



**UNIVERSITY**  
*of*  
**GLASGOW**

# **AN ADAPTIVE APPROACH FOR IMAGE ORGANISATION AND RETRIEVAL**

by  
**Jana Urban**

A thesis submitted to the  
Faculty of Information and Mathematical Sciences  
at the University of Glasgow  
for the degree of Doctor of Philosophy

January 2007

© Jana Urban 2007

---

## ABSTRACT

---

Image retrieval is an intrinsically hard problem. Manual labelling is impractical for most purposes and automatically extracted content-based features do not describe what humans recognise and associate with an image, referred to as the semantic gap. The semantic gap complicates the query formulation process for the searcher. Moreover, image meaning is subjective and context-dependent. Finally, information needs are often vague and dynamic, since image searches are usually coupled with creative tasks. These problems render current image retrieval systems difficult to use. Unlike most previous work in the field, which has studied the retrieval system as a self-contained problem, the approach described in this dissertation takes a holistic view, in which information access is considered as part of a larger work process. By taking into account the design of both the interface and retrieval algorithms, all of these intrinsic issues of image retrieval are addressed together.

The starting point in creating a more user-friendly system was to redesign the interaction process between user and system. Based on analysing user studies—both from previous work in the literature and those described within this document—the organisation of information has been found to help structure the thought process of the searcher. Therefore, the proposed system, *EGO* (Effective Group Organisation), combines image management and search. This is achieved by incorporating a workspace in the interface, allowing the user to organise search results into groups on the workspace. A recommendation system, which suggests new images for existing groups, assists the user in this task. The grouping process is incremental and dynamic: through usage a semantic organisation emerges that reflects the user’s mental model and their work tasks. Hence, *EGO* aims to represent the context in which the images are used, in short a “retrieval in context” system.

This dissertation describes the iterative design and evaluation process of an adaptive approach for image organisation and retrieval from both the user’s and system’s perspective. Two major user studies, evaluating a total of four different interface approaches, were conducted to assess the user’s view. The studies helped to unveil the inherent problems mentioned above and helped to gather evidence of how they are addressed in the proposed approach. Finally, the performance of the retrieval model—based on a graph representation of visual, textual and semantic features and the theory of random walks for retrieval—was evaluated in simulated experiments.

---

## ACKNOWLEDGEMENTS

---

I could pursue this PhD thanks to an EPSRC scholarship that paid my fees and to the department of Computing Science for providing me with an enjoyable tutoring job to earn my living. Most importantly however, I would like to thank my parents for their financial and moral support that kept me afloat.

I am grateful to the Royal Society of Edinburgh for providing me the opportunity to experience research in the USA through the JM Lessells travel scholarship. I had an interesting time at Thomas Huang's group at the University of Illinois at Urbana-Champaign, IL and a very warm welcome from Nick Belkin's group at Rutgers University, New Brunswick, NJ with a lot of useful comments and discussions of my work.

Special thanks go to Mark Baillie, Gareth McSorley and Robert Villa for carefully reading earlier versions of this dissertation and their comments on grammar, spelling and contents, even though they sometimes made my life harder. I would like to apologise here for moaning about some of the harder corrections they suggested.

My sincere thanks are due to my supervisors Joemon Jose and Keith van Rijsbergen for their support and guidance. Thank-you also to my examiners, Alan Smeaton and Karen Renaud, for managing to read the whole dissertation so carefully and for a surprisingly enjoyable Viva. Thanks to Karen Renaud's thorough corrections, the number of misspellings was drastically reduced and the number of commas drastically increased.

A big 'thank you' to all the members and visitors of the IR group for creating a pleasant working atmosphere. I am also thankful to the many friends who have shared with me the moments of joy and encouraged me during the occasional bouts of frustration I have experienced during this work. Iraklis Klampanos deserves a special mention for sharing the same experiences and helping me along the way.

I would also like to thank all the people who agreed to participate in my user experiments. I appreciate the time and effort that it cost them and I'm very grateful for all their useful comments and insights into their working practices.

Finally, I thank my family, relatives, colleagues and friends for patience, support, friendship and love. All these people and many others contributed to making these four years unforgettable. Thanks very much!

---

## **DECLARATION**

---

I declare that this dissertation was composed by myself, and that the work contained herein is my own except where explicitly stated otherwise in the text. This work has not been submitted for any other degree or professional qualification except as specified.

*Jana Urban*

---

## TABLE OF CONTENTS

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Proposed Solution . . . . .	2
1.2.1	Thesis Statement . . . . .	2
1.2.2	The Interface . . . . .	3
1.2.3	Retrieval Model . . . . .	4
1.2.4	Summary of Advantages . . . . .	5
1.3	Outline . . . . .	5
1.4	Contributions . . . . .	8
<b>2</b>	<b>Background and Related Work</b>	<b>10</b>
2.1	Introduction . . . . .	10
2.2	Image Representation . . . . .	12
2.2.1	From Content-Based towards Concept-Based Features . . . . .	13
2.2.2	Textual Features . . . . .	14
2.2.3	Primitive Content-Based Features . . . . .	16
2.2.4	“Semantic” Features and Retrieval . . . . .	20
2.2.5	Summary . . . . .	27
2.3	Learning from Relevance Feedback . . . . .	27
2.3.1	Overview of Relevance Feedback . . . . .	28
2.3.2	Geometric Approaches a.k.a. Query Refinement . . . . .	30
2.3.3	Statistical Approaches . . . . .	35
2.3.4	Discussion . . . . .	40
2.3.5	Summary . . . . .	41
2.4	The Interface . . . . .	41
2.4.1	Interaction with Image Search Systems . . . . .	42
2.4.2	Existing Interactive CBIR Interfaces . . . . .	43
2.4.3	Summary . . . . .	50
2.5	Evaluation . . . . .	50
2.5.1	System Evaluation . . . . .	50
2.5.2	User Experiments . . . . .	52
2.6	Other Issues . . . . .	53
2.7	Summary . . . . .	53
<b>3</b>	<b>Experiences from a User Study</b>	<b>55</b>
3.1	Results from a User Study . . . . .	56
3.1.1	Motivations of the Ostensive Approach . . . . .	56
3.1.2	Ostensive Relevance . . . . .	57
3.1.3	The Systems . . . . .	58
3.1.4	Query Adaptation Techniques . . . . .	61

## TABLE OF CONTENTS

---

3.1.5	Experimental Methodology . . . . .	63
3.1.6	Results Analysis . . . . .	65
3.1.7	Results Summary . . . . .	74
3.2	Open Issues . . . . .	75
3.2.1	The Meaning of an Image . . . . .	75
3.2.2	The Query Formulation Problem . . . . .	76
3.2.3	Dynamic Nature of Information Needs . . . . .	79
3.3	Discussion . . . . .	80
3.4	Summary . . . . .	82
<b>4</b>	<b>EGO—Effective Group Organisation</b>	<b>83</b>
4.1	Background and Related Work . . . . .	84
4.1.1	Problem Solving by Organisation . . . . .	84
4.1.2	Workspaces for Search and Management . . . . .	86
4.1.3	Summary . . . . .	89
4.2	A Holistic View . . . . .	90
4.2.1	Who are the Users? . . . . .	90
4.2.2	Summary . . . . .	93
4.3	The Interface . . . . .	94
4.3.1	The Interface Components . . . . .	95
4.4	Unique Characteristics . . . . .	100
4.5	Summary and Conclusion . . . . .	101
<b>5</b>	<b>The Recommendation System</b>	<b>103</b>
5.1	Background and Related Work . . . . .	103
5.1.1	Image Representation . . . . .	104
5.1.2	Relevance Feedback by Learning a Transformed Feature Space . . . . .	105
5.1.3	Multi-Point Queries . . . . .	106
5.2	Group-Based Query Learning . . . . .	107
5.2.1	Evidence Combination for Multi-point Queries . . . . .	107
5.3	Experimental Setup . . . . .	109
5.3.1	Features . . . . .	110
5.3.2	The Techniques . . . . .	110
5.3.3	Performance Measures . . . . .	111
5.4	Results Analysis . . . . .	111
5.4.1	Testing the Parameters . . . . .	111
5.4.2	Performance with Relevance Feedback . . . . .	117
5.4.3	Discussion . . . . .	120
5.5	Conclusions and Future Work . . . . .	120
<b>6</b>	<b>User Evaluation of EGO</b>	<b>123</b>
6.1	Introduction . . . . .	123
6.2	Experiment 1 . . . . .	124
6.2.1	The Interfaces . . . . .	124
6.2.2	Experimental Methodology . . . . .	127
6.2.3	Results Analysis . . . . .	131
6.2.4	Discussion . . . . .	136
6.2.5	Summary . . . . .	137
6.3	Experiment 2 . . . . .	137
6.3.1	The Interfaces . . . . .	139
6.3.2	Experimental Methodology . . . . .	139
6.3.3	Results Analysis . . . . .	143

## TABLE OF CONTENTS

---

6.3.4	Discussion . . . . .	153
6.3.5	Summary . . . . .	154
6.4	Combined Results . . . . .	154
6.4.1	Task Analysis . . . . .	155
6.4.2	Organisation Analysis . . . . .	158
6.4.3	User Satisfaction with Systems . . . . .	159
6.5	Conclusion . . . . .	160
6.6	Benefits of <i>EGO</i> . . . . .	162
6.6.1	Benefits from the Users' Perspective . . . . .	162
6.6.2	Summary of Benefits . . . . .	163
6.6.3	Addressed Issues . . . . .	164
6.7	Summary . . . . .	165
<b>7</b>	<b>The Personalised Recommendation System</b>	<b>166</b>
7.1	Introduction . . . . .	166
7.2	Individualist Retrieval System . . . . .	168
7.2.1	Peer Feature . . . . .	168
7.2.2	Textual Feature . . . . .	171
7.2.3	Visual Feature . . . . .	171
7.2.4	Combination of Results . . . . .	171
7.3	The Image Context Graph . . . . .	171
7.3.1	Related Work . . . . .	172
7.3.2	Mathematical Background . . . . .	174
7.3.3	Constructing the ICG . . . . .	176
7.3.4	Maintaining the ICG . . . . .	177
7.3.5	Evaluating a Query . . . . .	179
7.3.6	Relevance Feedback . . . . .	180
7.4	Experimental Setup . . . . .	182
7.4.1	Dataset . . . . .	183
7.4.2	Features . . . . .	183
7.4.3	Tasks . . . . .	183
7.4.4	Performance Measures . . . . .	184
7.5	Results . . . . .	185
7.5.1	Initial Runs without Relevance Feedback . . . . .	185
7.5.2	Performance with Relevance Feedback . . . . .	187
7.5.3	Variations of Group Size . . . . .	189
7.5.4	Introducing Feature Weights . . . . .	191
7.5.5	Computational Comparison . . . . .	194
7.6	Future Work . . . . .	196
7.7	Summary and Conclusions . . . . .	196
<b>8</b>	<b>Conclusions and Future Work</b>	<b>198</b>
8.1	Summary and Contributions . . . . .	198
8.1.1	The Interface . . . . .	199
8.1.2	The Recommendation System . . . . .	200
8.2	Analysis and Conclusions . . . . .	201
8.3	Future Work . . . . .	203
8.3.1	Study of Long-term Effects from the User's Perspective . . . . .	203
8.3.2	<i>EGO</i> in a Collaborative Context . . . . .	203
8.3.3	Browsing the Group-Space . . . . .	204
8.3.4	Applications to other Domains . . . . .	204

## TABLE OF CONTENTS

---

8.3.5	Minor Ramifications . . . . .	205
8.3.6	Additional Ideas . . . . .	205
<b>Bibliography</b>		<b>207</b>
<b>Appendices</b>		<b>218</b>
<b>A Quantitative Evaluation of the Ostensive Model</b>		<b>218</b>
A.1	Introduction . . . . .	218
A.2	The Simulation Setup . . . . .	219
A.3	Results Analysis . . . . .	219
A.4	Limitations of the Study . . . . .	220
A.5	Conclusions . . . . .	221
<b>B Architecture and Implementation of EGO</b>		<b>222</b>
<b>C Image Features</b>		<b>224</b>
C.1	Overview of Implemented Features . . . . .	224
C.2	Colour Features . . . . .	224
C.3	Texture Features . . . . .	226
C.4	Shape Features . . . . .	229
<b>D Experimental Documents</b>		<b>230</b>
D.1	Experiment 1 . . . . .	230
D.1.1	Tasks . . . . .	230
D.1.2	Information Sheet and Consent Form . . . . .	232
D.1.3	Questionnaires . . . . .	233
D.2	Experiment 2 . . . . .	242
D.2.1	Tasks . . . . .	242
D.2.2	Questionnaires . . . . .	245
<b>E Additional Results for the ICG Evaluation</b>		<b>253</b>
E.1	Parameters of the ICG . . . . .	253
E.2	Runs without RF . . . . .	259
E.3	Runs with RF . . . . .	259
E.4	Variations of Group Size . . . . .	259
E.5	Adaptive Weights . . . . .	271
E.5.1	Normalisation of Weights . . . . .	271
E.5.2	Performance of Adaptive Weights compared to Baselines . . . . .	271



---

## LIST OF FIGURES

---

2.1	Query Point Movement in a 2-D space . . . . .	31
2.2	Isosurfaces of distance functions (adapted from (Ishikawa et al. 1998)) . . . . .	33
2.3	SVM decision boundary . . . . .	38
2.4	The <i>QBIC</i> querying component . . . . .	44
2.5	The <i>MARS</i> interface . . . . .	45
2.6	The <i>ImageGrouper</i> interface . . . . .	46
2.7	The <i>CIRCUS</i> interface . . . . .	47
2.8	Zooming and panning in <i>CIRCUS</i> . . . . .	49
3.1	The ostensive path . . . . .	58
3.2	The interfaces. . . . .	60
3.3	Example Latin squares . . . . .	64
3.4	Work task scenario and task description . . . . .	66
3.5	Topics . . . . .	66
3.6	Semantic differential means per part (value range 1–7, lower = better) . . . . .	68
4.1	Annotated <i>EGO</i> interface . . . . .	94
4.2	Image Viewer window . . . . .	95
4.3	The interface where the results panel is enlarged and quick view is shown . . . . .	96
4.4	Step-by-step creation of a group on the workspace . . . . .	97
4.5	The interface where all but the workspace panel are collapsed . . . . .	98
4.6	The workspace (left) and its bird-eye-view (right) . . . . .	98
4.7	Recommendations on the workspace . . . . .	99
4.8	Selecting the “boats” group from the list in the recommended groups panel . . . . .	99
5.1	P(10) for various group sizes (average over all categories) . . . . .	112
5.2	Images from the roses category and the aviation category . . . . .	112
5.3	P(10) for various group sizes (average over homo- and heterogeneous categories) . . . . .	113
5.4	Number of clusters and cluster size vs group size . . . . .	114
5.5	P(10) for various group sizes comparing weighted vs non-weighted variants (average over all categories) . . . . .	115
5.6	P(10) for various group sizes comparing cutoff 100 and 10 (average over all categories) . . . . .	116
5.7	P(10) for various list cutoff values (group size 25, average over all categories) . . . . .	117
5.8	Number of images found per RF iteration (average over all categories) . . . . .	118
5.9	Number of images found per RF iteration (average over homo- and heterogeneous categories) . . . . .	119
6.1	Annotated WS interface used in Experiment 1 . . . . .	126
6.2	Annotated CS interface used in Experiment 1 . . . . .	126
6.3	Task description for Experiment 1 . . . . .	129

## LIST OF FIGURES

---

6.6	Relevance feedback in CS . . . . .	139
6.4	Annotated WS interface used in Experiment 2 . . . . .	140
6.5	Annotated CS interface used in Experiment 2 . . . . .	140
6.7	Task descriptions for Experiment 2 . . . . .	142
7.1	An example image-context graph . . . . .	173
7.2	The adjacency matrix for the ICG . . . . .	176
7.3	P(NR) for ICG and baselines . . . . .	188
7.4	P(10) over RF iterations . . . . .	190
7.5	P(100) over RF iterations . . . . .	190
7.6	R(100) over RF iterations . . . . .	190
7.7	P(100) of weighted ICG over RF iterations . . . . .	193
7.8	R(100) of weighted ICG over RF iterations . . . . .	193
A.1	Nr. of relevant images and nr. of iterations vs candidate size . . . . .	220
A.2	Example of fisheye display for candidate size of 15 . . . . .	221
E.1	P(10) for various values of $\alpha$ ( $k=3$ ) . . . . .	254
E.2	P(10) for various values of $k$ ( $\alpha=0.6$ ) . . . . .	254

---

## LIST OF TABLES

---

3.3	Semantic differentials . . . . .	67
3.4	Means and significance test results (value range 1–7, lower = better) . . . . .	68
3.5	Likert scale means for each statement . . . . .	69
3.6	Split of answers on changing ideas (number of responses per statement) . . . . .	69
5.1	Notations . . . . .	104
5.2	Average P(10) for weighted and non-weighted variants . . . . .	114
5.3	Number of images found per RF iteration . . . . .	118
5.4	Average number of images found after RF convergence . . . . .	119
5.5	Number (percentage) of null queries . . . . .	120
6.2	Number of relevant images found and corresponding levels of recall per category search topic . . . . .	131
6.3	Semantic differential results for the Task, Search Process and Images parts . . . . .	132
6.4	Semantic differential results for the System and Interaction parts . . . . .	132
6.5	Likert-scale results for the System part . . . . .	132
6.6	Comparison of system rankings . . . . .	133
6.7	Organisation and information need development results . . . . .	136
6.9	User effort indicators per task . . . . .	144
6.10	User effort indicators per task and system . . . . .	144
6.11	User perception of task performance per task (performance: higher = better, problems: lower = more problematic) . . . . .	146
6.12	User perception of task performance per task and system (performance: higher = better, problems: lower = more problematic) . . . . .	146
6.13	Semantic differential results for the Task, Search Process and Images parts . . . . .	147
6.14	Results for the system and interaction differentials and Likert-scales in the System part . . . . .	147
6.15	Interface effectiveness . . . . .	147
6.16	Relevance assessment with CS vs. grouping with WS . . . . .	147
6.17	Comparison of system rankings . . . . .	149
6.18	Organisation and information need development results . . . . .	151
6.19	Task listing . . . . .	155
6.20	Number of user samples per task . . . . .	156
6.21	Semantic differentials about task perception per task . . . . .	156
6.22	Semantic differentials about task perception per system for Experiment 1 . . . . .	156
6.23	Semantic differentials about task perception per system for Experiment 2 . . . . .	156
6.24	Organisation and information need development for all tasks with WS . . . . .	159
6.25	Semantic differential results for the System part (Experiments 1+2) . . . . .	160
6.26	Comparison of system rankings . . . . .	160
7.1	Tasks and their properties . . . . .	184

## LIST OF TABLES

---

7.2	Description of methods and their variations . . . . .	185
7.3	Comparison between baselines and ICG with and without peer information . . .	188
7.4	P(10) for individual tasks for baselines and ICG . . . . .	188
7.5	R(100) for individual tasks for baselines and ICG . . . . .	188
7.6	Average group size after 20 RF iterations . . . . .	189
7.7	P(10) for <i>IND</i> for various group sizes . . . . .	191
7.8	P(10) for <i>ICG<sub>p</sub></i> for various group sizes . . . . .	191
7.9	Average iteration number and time to solve the Random Walk (k=25) . . . . .	194
A.1	Average results for nr. of relevant images retrieved (R), nr. of iterations (I) and nr. of relevant per iteration (R/I) . . . . .	220
E.1	P(10) for $k = 25$ . . . . .	254
E.2	P(10) for $k = 3$ . . . . .	255
E.3	P(20) for $k = 3$ . . . . .	255
E.4	P(50) for $k = 3$ . . . . .	255
E.5	P(NR) for $k = 3$ . . . . .	255
E.6	R(10) for $k = 3$ . . . . .	256
E.7	R(50) for $k = 3$ . . . . .	256
E.8	R(100) for $k = 3$ . . . . .	256
E.9	R(P05) for $k = 3$ . . . . .	256
E.10	P(10) for $\alpha = 0.6$ . . . . .	257
E.11	P(20) for $\alpha = 0.6$ . . . . .	257
E.12	P(50) for $\alpha = 0.6$ . . . . .	257
E.13	P(NR) for $\alpha = 0.6$ . . . . .	257
E.14	R(10) for $\alpha = 0.6$ . . . . .	258
E.15	R(50) for $\alpha = 0.6$ . . . . .	258
E.16	R(100) for $\alpha = 0.6$ . . . . .	258
E.17	R(P05) for $\alpha = 0.6$ . . . . .	258
E.18	P(20) for baselines and ICG . . . . .	260
E.19	P(50) for baselines and ICG . . . . .	260
E.20	P(NR) for baselines and ICG . . . . .	260
E.21	R(10) for baselines and ICG . . . . .	261
E.22	R(50) for baselines and ICG . . . . .	261
E.23	R(P05) for baselines and ICG . . . . .	261
E.24	P(10) after the first RF iteration . . . . .	262
E.25	P(100) after the first RF iteration . . . . .	262
E.26	R(100) after the first RF iteration . . . . .	262
E.27	P(10) after the fifth RF iteration . . . . .	263
E.28	P(100) after the fifth RF iteration . . . . .	263
E.29	R(100) after the fifth RF iteration . . . . .	263
E.30	P(10) after the tenth RF iteration . . . . .	264
E.31	P(100) after the tenth RF iteration . . . . .	264
E.32	R(100) after the tenth RF iteration . . . . .	264
E.33	Average P(10) over 20 RF iterations . . . . .	265
E.34	Average P(100) over 20 RF iterations . . . . .	265
E.35	Average R(100) over 20 RF iterations . . . . .	265
E.36	P(20) for <i>IND</i> for various group sizes . . . . .	266
E.37	P(20) for <i>ICG<sub>p</sub></i> for various group sizes . . . . .	266
E.38	P(50) for <i>IND</i> for various group sizes . . . . .	267
E.39	P(50) for <i>ICG<sub>p</sub></i> for various group sizes . . . . .	267

## LIST OF TABLES

---

E.40	R(10) for <i>IND</i> for various group sizes . . . . .	268
E.41	R(10) for <i>ICG<sub>p</sub></i> for various group sizes . . . . .	268
E.42	R(50) for <i>IND</i> for various group sizes . . . . .	269
E.43	R(50) for <i>ICG<sub>p</sub></i> for various group sizes . . . . .	269
E.44	R(100) for <i>IND</i> for various group sizes . . . . .	270
E.45	R(100) for <i>ICG<sub>p</sub></i> for various group sizes . . . . .	270
E.46	Adaptive weights from individual indices using min-max normalisation . . . . .	272
E.47	Adaptive weights from individual indices using Gaussian normalisation . . . . .	272
E.48	P(10) for <i>IND</i> and <i>ICG</i> with adaptive weights under various normalisation techniques . . . . .	273
E.49	P(20) for <i>IND</i> and <i>ICG</i> with adaptive weights under various normalisation techniques . . . . .	273
E.50	P(50) for <i>IND</i> and <i>ICG</i> with adaptive weights under various normalisation techniques . . . . .	273
E.51	R(10) for <i>IND</i> and <i>ICG</i> with adaptive weights under various normalisation techniques . . . . .	274
E.52	R(50) for <i>IND</i> and <i>ICG</i> with adaptive weights under various normalisation techniques . . . . .	274
E.53	R(100) for <i>IND</i> and <i>ICG</i> with adaptive weights under various normalisation techniques . . . . .	274
E.54	P(10) for <i>IND</i> and <i>ICG</i> with various feature weighting strategies . . . . .	275
E.55	P(20) for <i>IND</i> and <i>ICG</i> with various feature weighting strategies . . . . .	275
E.56	P(50) for <i>IND</i> and <i>ICG</i> with various feature weighting strategies . . . . .	275
E.57	R(10) for <i>IND</i> and <i>ICG</i> with various feature weighting strategies . . . . .	276
E.58	R(50) for <i>IND</i> and <i>ICG</i> with various feature weighting strategies . . . . .	276
E.59	R(100) for <i>IND</i> and <i>ICG</i> with various feature weighting strategies . . . . .	276

---

## INTRODUCTION

---

*“The whole is more important than the sum of its parts.”*

— Aristotle, *Metaphysics*

### 1.1 Motivation

Image retrieval is complicated by problems inherent to the domain: the uncertainty of image meaning, the query formulation problem and time-varying and diverse information needs. The uncertainty of image meaning can be attributed to user subjectivity, on the one hand, and the *semantic gap* (Smeulders et al. 2000) between the low-level feature representation and the high-level concepts the user has in mind when looking at an image, on the other. As of today, we cannot devise automatic techniques to capture the general content of an image. This has further implications on the query formulation process. In order to satisfy their information need, the searcher poses a series of questions to the system until it returns sufficient satisfactory results. However, the question has to be translated into a language the system can understand, which can be a cognitively demanding process (Belkin et al. 1982, ter Hofstede et al. 1996). The query formulation problem is magnified when the representation of the documents is far from the user’s expectations as is the case in Content-Based Image Retrieval (CBIR), which is the collective term for retrieval techniques based on automatically extracted image features. To make matters even worse, the underlying information need can be ill-defined, vague or time-varying. Attempting to assist the searcher in developing their needs under these circumstances additionally complicates the system.

In order to address the semantic gap, a lot of effort has gone into improving the underlying image representation and retrieval algorithms. Consequently, early image retrieval interfaces were *computer-centric*. The system and its algorithms were considered the most important parts, and the interface simply provided the user with query input facilities that matched the system’s representation (eg *QBIC*, Flickner et al. 1995). Similarly, most research in the field of Information Retrieval (IR) has traditionally focused on the performance of the retrieval system. However, it has recently

been acknowledged that information retrieval is an inherently *interactive* process (Ruthven 2000), in which the user plays the most important role. In an effort to broaden the horizon of future search systems, researchers have attributed increasingly more importance to the human-computer interaction aspect of IR. Based on studies of information-seeking behaviour (Bates 1989, Ingwersen 1992) and user emotions and psychologies (Picard 1997), new interfaces for search systems have come into focus. Belkin (2003) has recently pointed out some grand challenges for information system design. By asking the question “*What might be the next steps to take in system design to support information seeking?*”, he identifies two issues, namely:

1. To design a system that supports a variety of interactions; and
2. Personalising the support of information interaction.

In this line, the aim of my work is to design an adaptive image search system. The above issues act as the primary design goals for its development. The system should be flexible, adapting to the diversity of users by supporting a variety of interactions. In particular, interactions should not be limited to strict image retrieval, but the retrieval aspect should be placed in the wider context of the work environment. What is sought is a holistic framework for retrieval, organisation and annotation. Moreover, the system should have the ability to adapt to the user by learning from user interaction. This facilitates personalisation and is expected to improve system performance, since the system is able to learn from the knowledge and interaction of individual users. I believe that it is vital to consider the system as a whole, from both the user and system perspective, in order to create an adaptive approach towards image retrieval that addresses the intrinsic problems of CBIR.

## 1.2 Proposed Solution

### 1.2.1 Thesis Statement

*A “retrieval in context” framework will help overcome the intrinsic problems of Content-Based Image Retrieval, such as query formulation, the semantic gap and time-varying information needs, by providing an integrated environment for image search and management in order to create and capture the context in which the images are used. This integration is achieved by the addition of a workspace to interactively group retrieval results, which supports the user, specifically where the user’s task is creative, and leads to a more effective system and increased user satisfaction.*

A “retrieval in context” framework, *EGO* (Effective Group Organisation), is proposed, which provides an integrated environment for image search and management in order to create and capture the context in which the images are used. I aim to tackle the intrinsic problems of image retrieval through the development of an interface that encourages the grouping of search results and a recommendation system based on semantic features to pro-actively support searchers. Instead of posing direct queries to the system, the user is engaged in an interactive organisation process supported through an adaptive recommendation system. The two pillars of *EGO*, and the work described in this dissertation, are:

1. An intuitive interface that allows the user to interleave the organisation and search processes to create a more intuitive information seeking environment.
2. A recommendation system based on an effective retrieval algorithm that can learn from the users' organisation and adapt to their understanding of image meaning.

In the following sections, I describe these two building blocks of an information seeking environment: its interface and retrieval model.

### 1.2.2 The Interface

The interface is the mediator between the system and its user. A properly designed interface assists the user with meaningful and intuitive ways of communicating their information need, and displays results in ways that stimulate the user and enhance performance.

The goal of the system designer should be to design a system that affords an intuitive interaction strategy close to the user's mental model of solving the underlying work tasks (Norman 1988, Järvelin & Wilson 2003, Ingwersen 1996). Organisation has been shown to be a vital tool to assist the thought processes of the searcher (Malone 1983, Kirsch 1995, Nakakoji et al. 2000), and the activities of organising and searching are inseparable (Malone 1983, Rodden 1999, Bauer et al. 2004). Hence, we can create an analogy to traditional problem solving strategies by helping the user to organise the information they find.

The main component of the proposed interface is the workspace panel provided in *EGO*. The workspace serves as an organisation ground for the user to construct groupings of images. The combination of search and organisation process creates an interaction metaphor for traditional ways of information management. The organisation is expected to help the user to conceptualise their search tasks and therefore express their information needs more easily. Over time, as I will show you by analysing organisation patterns in a user study, a semantic organisation emerges that reflects the context of the images' usage. To assist the user organising the collection, *EGO* includes a recommendation system. The recommendation system observes the users' actions, which enables it to adapt to their information requirements and to make suggestions of potentially relevant images based on a selected group of images. Using the recommendation system frees the searcher from having to formulate queries themselves, therefore addressing the query formulation problem.

There have been proposals in the literature to include a workspace in the retrieval interface to allow organisation of information (Hendry & Harper 1997, Cousins et al. 1997, Nakazato, Manola & Huang 2003). Search plans can be represented on the workspace by visualising elements of search activity, such as queries issued, results obtained and search services consulted. Tasks can be solved incrementally, since the search process can be interleaved and through the visual cues the users can track their progress. Hence, a workspace system can be used to interactively define and discuss a user's problems with the system.

The problem with the previous approaches to workspace systems in text-based IR environments (Hendry & Harper 1997, Cousins et al. 1997) is that they have emphasised the interaction and organisation of heterogeneous *retrieval* systems. The retrieval components themselves are



not interactive; they do not enable the user to interact with the results incrementally to improve retrieval. Instead, I investigate an information workspace used to facilitate the organisation of *results* rather than retrieval services. The categorisation of results allows an even more direct interaction with the information that is being sought. In the CBIR domain, Nakazato, Manola & Huang (2003) have introduced a workspace in *ImageGroupier* to assist grouping positive and negative images to improve the search results relying on a relevance feedback classifier. However, the emphasis in *ImageGroupier* does not lie in result organisation but simply in supporting an incremental and opportunistic search strategy.

To summarise, the workspace in *EGO* provides an organisation ground for the user to interact with search results. Grouping images on the workspace serves two purposes: (1) they aim to facilitate task conceptualisation since organising information is expected to support thought processes; and (2) they serve as query representatives and therefore aim to alleviate the query formulation problem. The organisation is persistent, and over time *EGO* can be used as both an information browser and archive—personalised to the user.

Moreover, varying types of information need can be supported in *EGO*. Short-term needs can be satisfied quickly by locating previously created groups that best match a users need. If there are no matching groups, the user can still resort to a traditional query facility. This is an important point, since the grouping facility is designed as an augmentation to traditional query facilities in current image retrieval systems rather than as their replacement. Furthermore, groups can be created and populated over time, reflecting long-term, time-varying needs.

### 1.2.3 Retrieval Model

Having discussed the problems from the user’s point of view, I feel obliged to also look at the system’s side, since they are naturally intertwined. Image retrieval system are primarily hampered by the semantic gap (Smeulders et al. 2000), and can thus be improved by: an underlying feature representation that is close to the intended semantics of the user; and a retrieval algorithm that can adapt to its users.

Retrieval effectiveness can be improved when the images are represented by a “semantic” feature—a feature that represents the intended semantics of the user. Ideally, such a feature is directly obtained from the user, because it should take into account an individual’s interpretation and context. However, people often feel overwhelmed when asked to provide semantic annotations explicitly, eg by labelling their image collection. An alternative is a semantic feature that can be mined implicitly by observing the user’s interaction with the collection.

With *EGO* I have aimed to create an environment for mining such a semantic feature. The groupings people create while searching are based on semantic concepts that take into account a user’s context, including work task and individual preferences. A group therefore suggests the existence of a semantic connection between those images contained within the group. These relationships form the basis of the proposed semantic feature.

The recommendations are based on a learning algorithm combining evidence from the visual features of the images in a group, their textual annotations and semantic information gained from the overall groupings of images. I propose a graph-model to represent these three feature modalities, in which querying (or recommending) involves finding those images that are best connected

and most easily reachable from a set of query images or terms. The semantic information provides the context of usage collected over time by a number of users. This enables it to make more personalised recommendations based on previous usage and user preferences.

#### 1.2.4 Summary of Advantages

The proposed approach provides a holistic, conceptual view of image search and management. In a nutshell, the advantages of my solution, and *EGO* as the proof-of-concept implementation, are the following:

- The interactive grouping is a flexible means to communicate both short- and long-term, specific and multifaceted information needs.
- The query formulation problem is reduced significantly by supporting an interaction metaphor for traditional ways of information management.
- The semantic gap is narrowed by the abstraction to high-level semantic groupings, reflecting an individual's task-specific mental model of the data.
- A personalised view of the collection can be provided, so that the users can go back to groupings previously created or explore trails of other users.
- The grouping information is the basis for a contextual feature learnt from the user that allows long-term learning in the system.

The remainder of this dissertation is concerned with investigating the validity of my thesis and, in particular, the advantages listed above. This is achieved by evaluating *EGO*'s success in alleviating the intrinsic problems of CBIR.

### 1.3 Outline

This dissertation begins by surveying background and related work in Chapter 2. This chapter introduces the motivation behind CBIR and its major challenges: why do we need CBIR in the first place, what can be achieved at the moment, what do the users expect, how can we bridge the gap between the users' expectations of CBIR systems and the available techniques today. This chapter also details the available techniques and how they are integrated into search interfaces. I survey related approaches under three general headlines: feature representation, relevance feedback learning and the interface, which encompass three of the most important components in an adaptive CBIR system. In addition, the evaluation of both retrieval techniques as well as the interface is covered in this chapter.

The discussion of related work points to problems which are still largely unsolved by the techniques surveyed in Chapter 2. Chapter 3 summarises the recurrent problems of CBIR, which include the uncertainty of image meaning, the query formulation problem and time-varying information needs. In order to investigate the extent of these problems, I performed a user evaluation of an adaptive image retrieval system based on the Ostensive Browser (Campbell & van Rijsbergen 1996). In the user study I gathered both qualitative data on user opinion (via questionnaires and

interviews) and quantitative data on search behaviour (via interaction logs and observations). In Chapter 3, I report the results and observations from this study, which helped me to identify these issues and formulate a new approach.

Having uncovered the main problems of CBIR, I have formulated an adaptive approach for image organisation and retrieval, in order to address these issues in a system that better fulfills the user's expectations. The proposed approach is implemented in the *EGO* system introduced in Chapter 4. Its design is motivated by considering the cognitive aspects of information seeking, studying existing user interfaces and observing people's search behaviour while using image search interfaces. The motivation is elaborated in Chapter 4, including previous work on workspace systems (Hendry & Harper 1997, Cousins et al. 1997, Nakazato, Manola & Huang 2003) and management of personal photographs (Rodden & Wood 2003, Grant et al. 2003, Bauer et al. 2004).

This has led to the idea of a "holistic view", which encompasses the integration of perspectives on two different levels. First, I consider a holistic view of system design that integrates both the user and system perspective. Second, I propose a holistic view of the image search process that integrates the retrieval and management process. The former takes the user as a starting point to analyse how the interaction process can be improved in an image retrieval system. Then, in an iterative design process, the interface and the underlying retrieval algorithm are developed hand-in-hand. The second integration level resulted from the goal to improve the interaction process in current image retrieval systems. I believe that combining the retrieval and management process can support a more intuitive interaction between user and system, and that it can create and capture the context in which the images are used. The combination is facilitated by the workspace in *EGO*, which encourages its users to group related images while searching.

Of course, to build an all encompassing system for professional designers, one should also integrate other tools they use for their jobs, such as image editing, page layout (such as Adobe's InDesign) and Web publishing software. However, the focus of my research is to improve the retrieval process, which, as these two activities are inseparable, should be tightly integrated with organising the results. An integration of other tools can then aid the designer's overall work flow further, and I consider my work as a first, but important, step towards a "holistic" image suite. My "holistic view" as elaborated above, even with its current limitations, should ensure that the intrinsic problems of CBIR are addressed from the user's perspective as well as the system's perspective. In Chapter 4 these ideas are elaborated, and *EGO*'s interface and how the user interacts with the system are detailed.

From the system's perspective, I propose a recommendation system that supports the interactive organisation of images by suggesting new images for selected groups on the workspace. Chapter 5 describes the initial implementation of the recommendation system. This covers the basic retrieval and learning algorithms based on visual features only. The choice of these algorithms was based on a comparative, simulated evaluation of various possibilities. In simulated experiments the algorithmic issues are isolated from interface design issues. Searcher behaviour was simulated in a relevance feedback scenario that determines which and how many images are selected for feedback in each iteration. In particular, a single query representation (where all images in a group are represented by a single representation) and multi-point query representations

(where a group is represented by multiple representations obtained by clustering visually similar images) were compared.

Having implemented both the interface and the basic retrieval model, a user evaluation of *EGO* followed, which is the subject of Chapter 6. The objective of this evaluation was to confirm some of the major claims made earlier on how an interactive organisation process assisted by the recommendation system provides a more effective information seeking environment. In this evaluation I aimed to study the usefulness of the interactive organisation process in *EGO*, based on an analysis of task-dependent search strategies and organisation patterns on the workspace. In particular, I investigated its influence on the query formulation problem under varying types of information need. In Chapter 6 the results are discussed and related back to the claims made throughout the dissertation of how the system would improve the intrinsic problems of CBIR systems.

The evaluation also helped to uncover problems with the content-based retrieval algorithm due to its lack of mapping to a higher-level semantic representation of the images. Some of these problems were also apparent in the evaluation of the recommendation system in Chapter 5, in which the difficulties of using content-based features only came to light. These issues have led me to investigate improvements to the recommendation system. Therefore, as a final step to formulate a “retrieval in context” approach, I propose a “contextualised” recommendation system. Chapter 7 introduces this improved recommendation system that facilitates long-term learning of a semantic feature by adding contextual information. Contextual information is gained from the groupings created by the users. This results in an additional feature representation, in which semantic relationships are encoded based on co-occurrences of images in the same group. I propose a graph-model to integrate the semantic feature with visual and textual features.

The proposed Image-Context Graph (ICG), in which images and their features are represented as several layers of nodes and feature similarities are represented as links between nodes. By representing all three feature modalities in a single graph, the problem of results fusion (Iyengar et al. 2005, Tong et al. 2005) is circumvented and interdependence between features can be modelled. The semantic relations defined by the user’s groupings are encoded by direct links between image nodes. These links are fostered over a long period of interaction with the system, which results in addition and reinforcement of links (positive feedback) and also deletion (negative feedback). Therefore the more links between two images exist (represented by link weights), the higher the probability that the images are semantically related. Retrieval in this model is implemented using the theory of Random Walks (Lovasz 1993). Similar to PageRank in the Web domain (Page et al. 1998, Brin & Page 1998), the computation of a Random Walk on the ICG results in an estimation of the relevance of an image based on the link structure in the graph.

Systematic simulated experiments were run in order to show that: the semantic feature provides an improvement over content only; and the graph-model is more effective than the typical individualist approach of combining the retrieval results from the various feature modalities. These results are also discussed in Chapter 7.

Finally, Chapter 8 summarises and reiterates the major findings. It also identifies avenues for future work, which include amongst others a study of the long-term effects of using *EGO* and its application to other media, such as music or videos, to create a multimedia management frame-

work.

## 1.4 Contributions

To solve the problems mentioned in the motivation above, this dissertation introduces a “retrieval in context” framework that provides pro-active support for its users to iteratively define their semantic needs. By placing the interactions between system and user at the centre, the system can better recognise and learn the user’s information needs and the user can concentrate on solving their tasks.

My research builds on previous and ongoing work on content-based representation and retrieval, retrieval interfaces and evaluation methodologies. My approach differs from previous work in this area by tightly integrating these three aspects to build a holistic system. It is motivated by reviewing and evaluating previous systems (Chapters 2 and 3) that resulted in the interaction and interface design of *EGO* (Chapter 4). User-centred evaluations have not only shown the effectiveness of the approach, but have also highlighted task-dependent search strategies and organisation patterns (Chapter 6). The work on the underlying retrieval system (Chapter 5) has culminated in the proposal of a novel retrieval model that integrates contextual information learnt from user interaction with content-based and textual features (Chapter 7). Taking all these components into account, my thesis is a formulation of an “adaptive approach for image organisation and retrieval”. In summary, the contributions of this work are the following (in chronological order):

- Critique of existing CBIR system—focusing on interface support—supported by results from a user evaluation of the Ostensive Browser.
- Proposed and evaluated a new adaptive query learning scheme for visual and textual features in the Ostensive Browser.
- Formulation of a holistic approach towards image retrieval based on cognitive ideas, previous user studies and interviewing design professionals.
- Comparative evaluation of visual recommendation algorithms focussing on evidence combination of multi-point queries.
- User-based effectiveness evaluation
  - Comparison of the effectiveness of the proposed interface to traditional relevance feedback interface;
  - Analysis of the extent of the query formulation problem in image retrieval interfaces;
  - Analysis of task-dependent search strategies;
  - Analysis of organisation patterns on the workspace.
- Contextual recommendation system
  - Proposal of a semantic feature learnt from user interaction, enabling long-term learning in the system;

- High-level feature integration provided by a graph-model—the Image-Context-Graph (ICG);
- Showed benefits of the proposed model in simulated experiments.

---

### BACKGROUND AND RELATED WORK

---

This thesis is an investigation of a “retrieval in context” system for image retrieval. This chapter provides the background for the research described in this dissertation and creates a context within which the work is situated. Automatic image retrieval techniques are based on visual features extracted from the images, and are known as Content-Based Image Retrieval (CBIR). This chapter will survey relevant work in CBIR: how images are represented, including content-based as well as meta-data based features; learning in CBIR; interfaces for image retrieval; and evaluation of CBIR systems.

#### 2.1 Introduction

Although historically speaking CBIR is still a young discipline, the literature on it today is vast (Datta et al. 2005). CBIR is considered to lie at the crossroads of many research areas. While it was mainly driven by image processing and computer vision in the early stages, artificial intelligence and human computer interaction have influenced its more recent advances.

This shift of interest has been triggered by the inability to find an acceptable solution to the image understanding problem, which is at the core of successful semantic retrieval. Even after decades of research in computer vision for image retrieval, object recognition in generic heterogeneous image collections remains a seemingly insurmountable challenge (Smeulders et al. 2000, Datta et al. 2005). After an initial optimism of purely content-based retrieval systems replacing the labour-intensive and expensive manual indexing procedures preceding systems had been relying on, the existence of the *semantic gap* (Smeulders et al. 2000) between the low-level features and the user (as mentioned previously in Chapter 1) finally had to be admitted. This gap has indeed been the reason for much of the disappointment in CBIR research. It is considered as probably *the* most challenging problem in CBIR by the research community. At the same time, the need to provide semantic-level interaction between users and content has been proven to be of vital importance. In each of the few existing user studies (*eg*, Garber & Grunes 1992, Markkula & Sormunen 2000, Armitage & Enser 1996, Cunningham et al. 2004) it has become apparent that the ability to query images based on semantic concepts is necessary for acceptability and practical applicability of image retrieval systems. Today, this need has moved towards the centre of current research

directions.

However, that does not mean that research in CBIR has come to a halt. On the contrary, researchers in the field are pushing the boundaries and are exploring new dimensions. The problems of fully automatic image understanding that computer vision is still trying to solve, have proven to be less critical for image retrieval purposes. The reason for this is that retrieval systems can exploit the knowledge of the user. Since recognising this fact, more and more inspiration has been taken from research in artificial intelligence and human computer interaction. Artificial intelligence research has driven the advance in machine learning, ie the problem of devising computer programs that automatically improve with experience. In the case of image retrieval applications this problem can be formulated as: Can we teach the computer to infer semantics from the low-level feature representation? Most of the proposed CBIR systems today encompass some sort of learning, in which the experience is drawn from the user's interaction with the system. Consequently, this has also led to borrowing ideas from the human computer interaction research community. It has become apparent that providing an intuitive and interactive environment, in which the system assists the user while browsing or searching, can improve the system's overall effectiveness in many ways and also compensate for its shortcomings due to the semantic gap.

There are a number of detailed reviews of the development and state-of-the-art of CBIR techniques and systems. Even in one of the earlier reviews, Aigrain et al. (1996) have pointed to the challenges of content processing techniques and the need for a combination of content-based features and metadata, such as keywords. They also acknowledge the important role the user interface plays for image retrieval systems. A more recent and also more comprehensive study by Smeulders et al. (2000) pays tribute to the historical development, as well as providing a detailed description of methods and a discussion of existing problems and future research directions. Datta et al. (2005) continue the survey of progress in the field from where Smeulders et al. left off. This follow-up survey covers new trends, such as automatic image annotation and task-related requirements from industry. The interested reader can also refer to Rui et al. (1999), who also compare existing systems, in addition to covering techniques and open issues. Addressing both research and commercial interests in CBIR, Eakins & Graham's JISC report (1999) presents an extensive and significant survey of image retrieval and additionally provides a large list of possible application areas and available software. The two books on Visual Information Retrieval by Del Bimbo (1999) and Lew (2001) provide an in-depth analysis of issues concerning multimedia retrieval. Finally, available systems are listed in (Eakins & Graham 1999, Rui et al. 1999, Müller 2002).

Although the importance of integrating features at all levels has long been argued, a large part of the earlier research has been devoted to a single aspect only, most importantly the extraction of meaningful and discriminating features and suitable similarity measures. Combinational approaches already advertised by all of the above mentioned reviews, alongside many other papers identifying open issues in multimedia retrieval (*eg*, Lu 1999, Eakins 2002, Enser 2000), have recently received major attention. As a result, approaches to unifying keywords and visual content have gained momentum (*eg*, Jeon et al. 2003, Su & Zhang 2002). A selection of such techniques facilitating image retrieval based on semantic concepts is discussed in Section 2.2.4.



The architecture of a CBIR system can be divided into a number of components. Vasconcelos & Kunt (2001) have identified two fundamental components of a retrieval system:

1. *representation*, and
2. *learning*.

Image representation is primarily derived from work in computer vision. The features used for image representation together with suitable similarity measures are vital for retrieval effectiveness and efficiency and depend largely on the collection domain. Learning is a common way to bridge the gap between the system and the user. One can distinguish between short- and long-term approaches. Most learning techniques are initiated by relevance feedback gained from the user during an interactive retrieval session. Other sources for learning stem from pattern recognition and classification methods. In this case, learning is achieved by a large-scale off-line training phase, usually based on a set of labelled images (supervised learning).

This review is centred around those two concepts. We will outline the issues concerning the image representation in Section 2.2, where we will provide a brief overview of the most basic image features and introduce some of the more recent techniques for obtaining “semantic features” along with combinational approaches of low-level features and textual annotation. Semantic features are often obtained through supervised training phases of partially labelled or categorised collections. Section 2.3 details recent learning methods for CBIR that involve a user in the loop. Such methods are summarised as relevance feedback techniques. Although this section emphasises their use in relevance feedback systems, the statistical learning algorithms discussed in Section 2.3.3 are also popular in other areas, such as the learning of semantic concepts.

As argued earlier, another very important component of an image retrieval system is the user interface and, in particular, visualisation techniques that communicate the meaning of similarity and invite rich user involvement. This topic is covered in Section 2.4. Finally, the importance of information retrieval system evaluation cannot be ignored. Section 2.5 discusses the issues of both automatic evaluation of retrieval techniques and interactive evaluation of retrieval interfaces. Throughout this review we will highlight open issues in CBIR related to the semantic representation of images and suitable interaction strategies. Finding a solution to these problems has acted as the primary motivation to formulate the holistic approach described in this dissertation.

## 2.2 Image Representation

Before the retrieval system is able to handle images, a suitable representation format needs to be found. The image itself is meaningless until distilled into “features” of some sort. For effective access, it is desirable to index the image by its most significant contents. Ideally, this would include objects the image contains, their layout and relationships among them. An image therefore needs to be transformed in a more compact representation that reflects significant *features* in the image. A *feature vector* (a line-up of different features) thus provides a compressed “view” of the image which emphasises certain attributes of the image. Along with the image representation, rules for comparing images have to be defined. These rules—referred to as *similarity measures*—are dependent on the feature space and usually each feature has its own measures. In summary,

the features together with their similarity measures are crucial for the system's *efficiency* and *effectiveness*.

In comparison, preprocessing in text retrieval systems involves splitting the original documents into tokens serving as units for indexing and subsequent matching between documents. Tokenising textual documents into words and phrases has proven to work reasonably well for retrieval purposes, since words carry some level of semantic meaning. In the visual domain, on the contrary, this is far from easy, since images cannot be readily decomposed into such semantic units. The content of an image can be described as the pixel distribution of certain colours, or the existence and direction of edges present in the image, for example. Such transformations from the space of image pixels to a feature space with better properties for retrieval and recognition—even though easy to automatically extract—lack a semantic interpretation.

As a result, images are often represented by both textual features, such as keywords obtained from annotation and visual features. Visual features are at the core of content-based retrieval and since a huge amount of work has gone into feature extraction, a large variety of visual features have been proposed. Some of those are general features, such as colour, texture and shape. Others have been developed for a specific recognition task or special domain, eg face recognition (Pentland et al. 1994) and trademarks (Eakins 2001). However, it has to be borne in mind that due to many difficulties, including perception subjectivity, one universally good feature set does not exist.

The remainder of this chapter will first discuss the development of features used for image retrieval. The features involved in each of the three “evolutionary” steps will be covered: from textual, to generic low-level (and hence most often used) and finally “semantic” or concept-based features. The semantic approach has become increasingly popular in another effort to bridge the semantic gap. Semantic features are usually obtained through (semi-)automatic image annotation or classification, of which a selection of example techniques will be provided in Section 2.2.4.

### 2.2.1 From Content-Based towards Concept-Based Features

In the earliest image database systems, images were retrieved on the basis of manual annotation and other metadata available for an image, such as date and photographer (Tamura & Yokoya 1984, Grosky et al. 1994). At least since the 90s it has been argued that such systems are too expensive to create and maintain and are not suitable for image search (Eakins & Graham 1999). The reason for this being that manual annotations are very subjective (eg, Rui et al. 1998, Eakins & Graham 1999) and can hardly cover every aspect the image might be searched on.

Thus, alternatives to the traditional approach were being sought. The new paradigm—*content-based image retrieval*—marks a significant departure from the early image databases. Despite its difficulties in bridging the semantic gap, content-based image features offer some attractive opportunities and some advantages over manual techniques. Firstly, they are independent of human intervention, which becomes increasingly necessary with the rapidly growing size of image corpora. Most importantly, however, by analysing the visual content directly, visual similarity of images can be exploited offering an entirely different view of image search. Today, there are numerous examples in which CBIR techniques are successfully deployed, including fingerprint and face matching, detection of unauthorised use of images on the Web, filtering out of pornographic images and automatic segmentation of video into shots and keyframe selection for storyboarding

(Eakins & Graham 1999). A discussion on the application of video-content analysis and retrieval techniques can be found in (Dimitrova et al. 2002).

Nevertheless, most professional and large-scale image archives are still indexed and searched on the basis of textual descriptions<sup>2-1</sup>. Some of the reasons for this have to do with the available techniques, which fail to take sufficient account of perceptual similarity, and the fact that visual query formulation is a difficult process (see Section 3.2.2). Another contributing factor is, to some extent, due to the missing dialogue between industry and research (Enser 2000, Dimitrova et al. 2002), where the research community has not made a big enough effort to convince others of CBIR's added values. Still, too little is known about the actual usage of image collections and a rigorous and comparative evaluation of CBIR systems remains a major obstacle. From the introductory discussion it can be seen that the most serious impediment, however, is the failure of content-based techniques to allow semantic or concept-based retrieval.

The next step in image retrieval research is now to combine the advantages of the two preceding paradigms. To enable querying images for semantic content while still maintaining predominantly automatic indexing facilities, people have started arguing for hybrid approaches to combine content-based and concept-based (usually textual) features (Enser 2000, Zhou & Huang 2002). This is not as straightforward as it might seem. Although many people have attempted to combine these two features before, only recently has there been a push towards more rigorous and well-founded ideas. Techniques to achieve this are mostly based on machine learning or pattern recognition techniques, which either involve semi-automatic annotation or image classification. Automatic annotation is achieved by label propagation, in which a partially annotated image collection is used to propagate their labels to other unlabelled images in the collection on the basis of visual similarity (eg, Jeon et al. 2003). Image classification is achieved by training a classifier on a set of training images to perform the classification task. This has been successfully employed for image retrieval by Oliva & Torralba (2001), who order natural landscape images on semantic axes, such as natural versus artificial. Online learning from user interaction is another possibility to improve the system's ability to discriminate semantic concepts in the images (Su & Zhang 2002, Zhou & Huang 2002). All of these approaches and other issues concerning semantic retrieval will be detailed in Section 2.2.4.

To lay the grounds for further discussion of CBIR techniques, an overview of the fundamentals of image features is essential and will follow.

### 2.2.2 Textual Features

The current indexing practices of large professional image collections rely on assigning metadata to each image. This metadata, in the form of textual descriptors, is then used as retrieval keys at search time with the help of traditional IR techniques (van Rijsbergen 1979). One can distinguish between indices that capture the formal description of the image and subject indexing. The former covers formal attributes of the image such as who, when and where, and is comparable to a bibliographical description of a textual document. This approach calls for a standardised set of formal attributes. Subject indexing depends largely on the make-up and purpose of the collection itself.

<sup>2-1</sup>eg, Getty Images ([www.gettyimages.com](http://www.gettyimages.com)) or Corbis (<http://www.corbis.com/>)

Many image libraries use their own indexing scheme geared towards the nature of the collection and the needs of their users.

Subject indexing is usually achieved by either describing the content of the image directly by assigning keywords from a specially designed thesaurus of words, or by classifying the images according to classification codes. Keywords are probably the most widely used approach in image libraries. *Getty Images*<sup>2-2</sup>—the company that markets the largest stock collection of imagery in the world—have developed a comprehensive thesaurus for indexing their photographs. It comprises more than 10,000 concepts, allowing users to pose queries at a range of levels, from very abstract to quite specific.

The alternative is to develop a strict classification scheme. Sometimes classification codes—alphanumeric representations of concepts in the library—are developed in favour of keywords because they are to a larger degree language independent and less prone to indexer subjectivity. Subjectivity arises because of inconsistencies in choices of keywords for indexing an image, which is a serious downside of existing manual indexing practices for image collections. Classification codes are usually employed to create a hierarchical structure. They work in a similar way to phone numbers, which include country codes and area codes to drill down a particular connection. One example is *ICONCLASS*<sup>2-3</sup> designed for the classification of works of art.

Keyword indexing schemes still succeed over content-based techniques because of their expressive power. They can capture the content of an image at various levels of complexity—one can list the objects depicted in the scene (eg house and tree), the layout of the objects (eg a tree in front of a house), the mood the image conveys (eg happiness) and even metadata that cannot be directly inferred from the image content itself, such as who took the picture at what time and where.

However, manual indexing processes have major drawbacks often quoted in the literature. Firstly, there is time and thus cost of indexing a collection manually. Secondly, the choice of keywords is very subjective and shown to be often inconsistent between different indexers. Furthermore, there is often a huge discrepancy between the keywords chosen by the indexer, who is usually a specialist in the field of library science, and those expected by the users. One of the few interesting, as well as entertaining, ways to collect image labels has been introduced by von Ahn & Dabbish (2004). They have developed an interactive online game to label images on the Web. A different approach has been pursued by Davis et al. (2004), who have developed an application for camera mobile phones that interactively collects spatial, temporal and social contextual metadata that can be shared amongst a user community.

User studies (Garber & Grunes 1992, Markkula & Sormunen 2000, Armitage & Enser 1996) have shown that manual indices are often inadequate and far from perfect. Eakins & Graham conclude that “*there is very little firm evidence that current text-based techniques for image retrieval are adequate for their task.*” (Eakins & Graham 1999, p. 22) This suggests that, although they are still used in favour of content-based techniques, there is a definitive need for alternative ideas. The most promising direction at the moment appears to be a hybrid approach between the two (Enser 2000). For more examples of classification and indexing schemes, software for image data

---

<sup>2-2</sup>[www.gettyimages.com](http://www.gettyimages.com)

<sup>2-3</sup><http://www.iconclass.nl/>

management, current indexing practice and research into indexing effectiveness refer to (Eakins & Graham 1999).

### 2.2.3 Primitive Content-Based Features

Content-based features are obtained by mathematical analysis of the pixel values of the image. They capture data patterns and statistics of the image using image processing and pattern analysis algorithms. The main requirements for feature extraction are (Lu 1999):

1. *Completeness/Expressiveness*: Features should be a rich enough representation of the image contents to reproduce the essential information.
2. *Compactness*: The storage of the features should be compact to facilitate efficient access.
3. *Tractability*: The distance between features should be efficient to compute.

For each feature a suitable similarity measure is defined that is used for determining similarity scores. During the retrieval process, images are presented to the user based on the similarity scores computed between the features of images in the database and the query features. Usually it is the case that each image is represented by a set of features, each feature type having its own similarity measure. Hence, to obtain a single similarity score, a means to combine the scores needs to be incorporated. In most cases this is achieved by a weighted sum of the normalised similarity scores for each feature type.

The three most prominent features—colour, texture and shape—are described below. In addition, the interested reader can refer to Datta et al. (2005), who have provided a review of which features have proven successful for CBIR.

#### Colour

Colour is *the* most fascinating attribute of an image. It has been studied by scientists, psychologists, philosophers and artists alike. It is used as a feature for image retrieval in order to retrieve and rank images on the basis of similar colour composition.

Intricate topics concerning the use of colour, which have to be born in mind when choosing a suitable colour descriptor for retrieval, are its variability with camera orientation and illumination, and human perception of colour that should act as the model for perceptual similarity measures. In addition, colour distribution gives no indication of the spatial layout of objects in the image.

Colour can be represented in different colour spaces. The choice of colour space for retrieval depends on the domain of use. The raw images are usually stored in RGB (Red, Green, Blue). However, RGB is not well suited for similarity retrieval. It is quite sensitive to illumination conditions and does not follow human perception of colour differences. This is a crucial criterion for a “good” colour space, which aims at mathematically modelling colour differences similar to how humans perceive and manipulate colour. Colour spaces approximating human perception, which are most often used for retrieval, are HSV (Hue, Value, Saturation) and CIE’s  $L^*a^*b$  colour space. Whereas  $L^*a^*b$  is specifically designed to be substantially perceptually uniform, its computation is a nonlinear conversion from RGB. On the other hand, HSV is easier to compute and furthermore has the advantage of invariance under the orientation of the object with respect to illumination and

camera direction. Overviews of various colour spaces can be found in (Gevers 2001, chapter in *Principles of Visual Information Retrieval*) and in any computer vision book (eg, Forsyth & Ponce 2003).

The most widespread descriptor is the *colour histogram* which encodes the proportion of each colour in the image. Apart from the choice of colour space, histograms are sensitive to quantisation effects, such as the number of bins (bars) and position of bin boundaries. By themselves, they also do not include any spatial information. Swain & Ballard (1991), who have introduced colour histograms, have proposed *histogram intersection* for matching purposes.

Other representations include colour moments and dominant colours. *Colour moments* have been proposed by Stricker & Orengo (1995) as a more compact representation and to overcome the quantisation effects of histograms. Colour moments are statistical descriptors that characterise the probability distribution defined by the distribution of colour in an image, such as the mean and variance of the distribution. Most often, the first three low-order moments (mean, variance, distribution skewness) for each channel in an image are calculated and used for retrieval. Colour moments are usually compared using a weighted Euclidean distance.

*Dominant colours* are obtained by clustering the colours in the entire image or a selected region of the image into a small number of representative colours. The descriptor contains for each dominant colour the representative colours, their percentages, spatial coherency of the dominant colours (to differentiate between large blobs versus colours that are spread all over the image) and colour variances. The objective of this descriptor is to provide a compact and intuitive representation of salient colours in a given region of interest. Their effectiveness depends on a suitable clustering algorithm, efficient similarity measures and indexing schemes, which can be looked up in (Manjunath et al. 2001). A similar approach is proposed by Smith & Chang (1996) in the form of *colour sets* as an approximation of the colour histogram, in which insignificant colour information is ignored while prominent colour regions are emphasised. The spatially localised colour sets are also an improvement over the global histogram, as it provides regional colour information.

The histogram is an efficient and the prevalent representation of feature distributions. However, it is inflexible, since the bin quantisation levels have to be decided beforehand, and hence it is difficult to achieve a good balance between expressiveness and efficiency. Alternatively, *signatures* in which the number and size of the bins (or clusters) is defined for each image individually have been proposed for representing feature distributions. Signatures have the advantage of adapting the number of clusters to the complexity of the images, so that simple images have short signatures whereas complex images have longer signatures.

Additionally, the similarity measures can be improved upon. The traditional *bin-by-bin* histogram measures (including histogram intersection) only compare the contents of the corresponding histogram bins (ie for histograms  $H = \{h_i\}_{i=1..n}$  and  $K = \{k_i\}_{i=1..n}$ : compare  $h_i$  to  $k_i \forall i$ , but not  $h_i$  to  $k_j$  for  $i \neq j$ ). This makes the measure very sensitive to the chosen bin boundaries. An improvement on effectiveness (but not efficiency) is to use *cross-bin* histogram measures, which also compare non-corresponding bins. Rubner proposed the *Earth Mover's Distance* as an effective similarity measure for histograms and signatures (Rubner 1999). He also provides a comprehensive comparison on feature representations and alternative similarity measures.

The interested reader can find numerous references on colour features in the literature. For ex-

ample, more information on the usage of colour for retrieval can be read in (Del Bimbo 1999, chapter 2). Manjunath et al. (2001) describe the colour descriptors that are proposed for the MPEG-7 standard<sup>2-4</sup>. They also cover texture features. Different distance measures for colour and texture features are summarised and evaluated in (Puzicha et al. 1999). From this extensive comparative study Puzicha et al. conclude that there is no single measure that exhibits best overall performance, but that the task at hand determines the performance.

### Texture

Colour alone is not discriminative enough for most image retrieval applications. For example, a part of sky cannot readily be distinguished from a lake based on colour similarity only. This is where texture can help. It is a phenomenon that is easy for a human to recognise but hard to define. Visual texture can be identified by variations of intensity and colour which form certain patterns. This makes texture analysis more complicated than the one of colour: a single pixel has no texture. For the computation of texture properties it is consequently necessary to take into account correlations of pixels in a certain neighbourhood. A lot of research has gone into the definition and extraction of texture properties.

There are some issues that need to be considered when dealing with textures:

- Texture is dependent on the scale at which the image is viewed. At a large scale, pebbles on a beach, for instance, create an effect interpreted as texture. Yet when focusing on a single stone at a finer scale, it will be seen as an object rather than a texture, until, while zooming in even more, the pattern, or texture, of the stone surface will become apparent.
- Natural images usually do not expose a homogeneous texture. They can be decomposed into regions within which the texture is constant. Texture segmentation is, however, an intricate task, which involves determining region boundaries and finding a suitable texture representation.

There are numerous approaches for texture features in the literature. A good introduction to texture features for content-based retrieval and a taxonomy of texture models can be found in (Sebe & Lew 2001).

Two distinct models that are established in CBIR are Picard & Liu (1994)'s *Wold decomposition* and the *Gabor filter decomposition* approach refined by Manjunath & Ma (1996). Picard & Liu (1994) have attempted to define a model in accordance with human perception of texture. It is based on the assumption that an image is a homogeneous 2D discrete random field. The Wold representation then decomposes an image into three mutually orthogonal components, which roughly correspond to *periodicity*, *directionality* and *randomness*. Those three components have been related to perceptual similarity dimensions in psychophysical findings. In addition, they offer some semantic referent. Since they agree with linguistic descriptions of texture, they have the advantage of allowing manual specification of the desired image properties in retrieval applications. In the *Photobook* system (Pentland et al. 1994) this model has been applied for retrieval of texture-swatch and keyframe databases.

<sup>2-4</sup><http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

Gabor filters, on the other hand, are believed to correspond to the way human vision works. A bank of Gabor filters can be considered as a collection of orientation and scale tunable bar filters (or edge and line detectors), which is analogous to the functioning of the visual cortex. The texture feature representation developed by Manjunath & Ma (1996), is based on a Gabor filter dictionary designed for image retrieval and browsing. In their *NeTra* system (Ma & Manjunath 1999), texture has been modelled using the mean and standard deviation of the filtered outputs, which is applied to search through large collections of aerial photographs. This texture feature characterises homogeneous image regions quantitatively, which is suitable for accurate search and retrieval given some query images. Lacking the possibility of a verbal description that the Wold model provides, Manjunath et al. (2000) have further extended their texture descriptor by a “perceptual browsing component”. Similar to the Wold attributes, this component characterises the perceptual attributes *directionality*, *regularity* and *coarseness* computed from the filtered images. This results in a very compact representation, which is more suitable for coarse classification of textures and browsing type applications. Both the similarity retrieval and the texture browsing descriptor have been adopted in the MPEG-7 standard (Manjunath et al. 2001).

Texture features are hardly ever used on their own. For retrieval and browsing of heterogeneous images, they are used in combination with a suitable segmentation algorithm to detect homogeneous texture regions (eg, Ma & Manjunath 1999). The segmentation is usually achieved by combining texture, colour and shape information. In addition, the obtained regions are represented by multiple features in both of the systems discussed above (Pentland et al. 1994, Ma & Manjunath 1999).

### Shape

Moving closer to the recognition of objects, shape is the third of the most prominent basic features. In contrast to colour and texture, which represent global intensity attributes of the image (unless used in combination with some segmentation technique), shape encodes inherently local geometric information.

The shape of an object within a 2-D image is defined as the contour traced by its boundaries. Formalising shape *similarity*, however, is a more delicate matter. In much the same way as colour and texture similarity, the ultimate goal is to match human perception of shape similarity.

The process of obtaining a shape feature vector is achieved in two steps. First, the image has to be segmented by detecting lines or similar in order to extract the shapes from a given gray-scale image. These shapes (in the form of binary images) are fed into shape analysis algorithms to arrive at a characterisation. Shape matching between the resulting shape vectors is used in the retrieval applications to determine the similarity between any two images.

The criteria that shape matching techniques must fulfill are—besides the ability to match human similarity perception—invariance to translation, scale and rotation, and robustness to noise. The modelling of shape similarity is seriously hampered by occlusion in the image and differences in view angle.

Since shape analysis plays a crucial role in object recognition, it has received extensive attention in the computer vision literature. Consequently, there exist numerous techniques. The interested reader is referred to a comprehensive survey of shape analysis techniques by Loncaric



(1998). Shorter introductions to shape analysis for image retrieval, including pointers to interesting approaches, can be found in any of the reviews of CBIR (eg, Rui et al. 1999, Smeulders et al. 2000, Eakins & Graham 1999). In the field of image retrieval, shape analysis has been studied extensively for trademark retrieval (Eakins 2001, Jain & Vailaya 1998).

### Combination of Features

Features have been proposed that, by themselves, already capture more than one aspect of image attributes. In theory, this is already the case for most texture features, since they capture changes in intensity values that can also indicate the existence of edges, which is the basis for shape matching. *Visual appearance* features are an extension of this idea. They are often used in attempts to recognise objects in images. The reasoning behind this is that in order to characterise the visual appearance of an object, which depends on an interplay of factors such as its shape, albedo<sup>2-5</sup>, surface texture and view point, a syntactic representation is more suitable for object recognition. So rather than extracting separate features for texture, shape, colour etc., only to later synthesise them again for similarity matching, the appearance feature approach circumvents having to separate the different factors constituting an object's appearance. Pentland et al. (1994) consider their Eigenface approach to face matching as an example of an appearance feature. Ravela et al. characterise visual appearance by the 'shape of the intensity surface' and propose features computed from Gaussian derivative filters for region matching (Ravela & Manmatha 1997) and global similarity retrieval (Ravela & Luo 2000).

Most commonly, however, a combination of (primitive) features is used in most visual retrieval systems (eg, Flickner et al. 1995, Ma & Manjunath 1999, Pentland et al. 1994). The most prevalent approach is to compute a single score as the weighted sum of the similarity scores of each feature. While this is a convenient means of computation, it is based on the assumption that the features are independent of each other (ie forming an orthogonal basis of the vector space spanned by the features as its dimensions). However, the fact that primitive attributes are inherently intertwined, provides a reason to question the independence assumption. Instead of a linear combination, Ma & Manjunath (1999) for instance, suggest a different treatment for uniting features. In their *NeTra* system an implicit ordering of features is assumed to prune the search. The search space is narrowed down using the first feature, followed by a re-ranking of the obtained set of images and final selection according to the whole set of features.

Since the number of proposed feature extraction algorithms is growing, a possible alternative to combining all available features is an intelligent feature *selection* approach. Depending on the application domain and the current query some features work better than others. Datta et al. (2005) review proposed feature selection algorithms in the literature.

#### 2.2.4 "Semantic" Features and Retrieval

Ever since the deficiencies of primitive content-based features were realised, interest has turned to "*semantic features*" and "*semantic retrieval*". Semantic features are now the ultimate goal in order to facilitate effective retrieval of visual data, but what are they? Smeulders et al. state the

<sup>2-5</sup>a measure of a surface or body's reflectivity of light

following:

“*Semantic features aim at encoding interpretations of the image which may be relevant to the application.*” (Smeulders et al. 2000, p. 1361)

There are two important points to note in this assertion. Firstly, semantics are about *interpretation*, and secondly the interpretation is to a large degree domain or *context dependent*. An image by itself usually has no intrinsic meaning. The meaning is bestowed upon the image by a human observer regarding the context of both the observer and the image, which will be discussed further in Section 3.2.1.

The goal of the semantic approach is to replace the low-level feature space with a higher-level semantic space that is closer to the abstract concepts the user has in mind when looking for an image. Since the endeavour of obtaining semantic features directly from the visual attributes was unfruitful<sup>2-6</sup>, mining for semantic concepts from a knowledge-base has been the focus of research to this end. Most of the existing attempts towards semantic features can be broadly categorised in two classes: *annotation-based* and *user-based*. This distinction arises from the nature of the knowledge base used: the first method relies on an (at least partially) annotated image corpus from which semantic concepts can be learnt and propagated to other images, whereas the latter learns semantic concepts from the user directly. While there are a number of general concepts that can be universally agreed upon, eg an ‘indoor’ vs. ‘outdoor’ classification, there are more subtle meanings that are subject to the observer’s interpretation, eg ‘a romantic scene’. The major difference in the two approaches hence lies in the interpretation context considered for deciphering the image’s meaning. It should become obvious that the annotation-based approach can only succeed in taking very general concepts into consideration, as opposed to user-based approaches that are tailored to the user’s expectations and interpretations.

It has to be noted that the majority of semantic feature extraction/learning approaches either aim at *image classification* or *image annotation*. Yet image classification can be seen as just a special case of annotation, in which the different classes the images are assigned to can be attached to the images as labels. Therefore, this distinction will not be made here explicitly, in favour of a more high-level categorisation depending on the knowledge base. Examples of both annotation-based and user-based techniques follow.

### **Annotation-based Semantic Mining**

Annotation-based approaches are flourishing and are the most prevalent today. The reason for this is that general concepts can be determined more easily without the user in the loop. They are more or less the same for each user, so that it is sufficient to use a knowledge base, such as a classification scheme or partially annotated images, that has been compiled manually by one person or a small group of people. It results in a kind of “batch mining” of general concepts, which—since dependent on some manual indexing—suffers from the same drawbacks (subjectivity, completeness, etc.).

<sup>2-6</sup>Since semantics should be seen as context-dependent interpretation (Jain 2003), an entirely automatic approach is deemed to fail in any case.

(Semi-)automatic annotation is achieved by large scale training on a set of predefined semantic concepts. Learning one concept often requires between 100s and 1000s of training examples, before it can be generalised to other images. This is why most techniques only report of being able to learn at most ten different concepts.

**Approaches** Various techniques for semantic label extraction have been proposed in the literature. As mentioned previously, some techniques focus on image categorisation (Oliva & Torralba 2001, Bradshaw 2000) whereas others on annotation (Chang et al. 2003, Jeon et al. 2003). Approaches also vary in the spatial extent of annotation. One can distinguish between global labelling (Oliva & Torralba 2001, Chang et al. 2003) and local region labelling (Bradshaw 2000, Lim 1999).

Oliva & Torralba propose a scene categorisation approach with the aim of labelling natural landscape images based on their global content and structural layout. Their studies have revealed that for rapid scene recognition, local object information might be spontaneously ignored in favour of a semantic scene representation built on a low resolution spatial configuration. Hence, the classification is achieved by characterising the *Spatial Envelope* describing the spatial structure or ‘shape’ of the scene. The properties of the Spatial Envelope are high-level descriptions of a scene. They include:

- Naturalness vs man-made
- Open vs. closed
- Roughness
- Expansion (for urban scenes)
- Ruggedness (for natural scenes)

The properties are interpreted as dimensions depicting a meaningful characterisation of the shape of the scene. This results in the determination of real-valued ‘semantic axes’ for each property along which the images are ordered. The Spatial Envelope is characterised by low-level perceptual features obtained from the global power spectrum of an image that has been filtered with a set of Gabor filters at various scales and orientations (see Section 2.2.3 on Texture Models). The properties of the Spatial Envelope are determined by a learning procedure, using 500 images for each classifier. The classifier (named “*Discriminant Spectral Template*”) computes an ordering of the images along the semantic dimensions, and is thus an improvement over most common image classification approaches that perform exclusive binary classification only.

Bradshaw proposes a probabilistic approach to a similar categorisation (Bradshaw 2000). The proposed method generates localised labels for man-made or natural objects and global labels for indoor or outdoor scenes. The resulting labels are represented as probabilities to capture the uncertainty associated with the automatic labelling process. The estimation of probabilities is based on colour and texture features of fixed-size image blocks. Training requires only a few 100 images, which is still prohibitive for on-line learning from user feedback.

Both Chang et al. (2003) and Jeon et al. (2003) propose an automatic probabilistic annotation approach obtained from a set of keywords assigned to each of the training images. Chang et al.’s *Content-Based Soft Annotation* (CBSA) based on Bayes point machines performs annotation using global primitive features (colour, texture). A training set of annotated images is used

to propagate their labels to all other images in the collection. As a result each image is associated with a global label vector (containing 116 keywords), in which each keyword is associated with a confidence factor depicting the likelihood of a label describing the image correctly, eg  $\{(landscape, 0.5), (cloud, 0.7), \dots, (tiger, 0.9)\}$ . When a text-based search is issued, images are retrieved and ranked based on the combined confidence factor in the matching labels.

Generative language models have been used by Jeon et al. (2003) for the task of associating words and image segments. Before annotation, the images are segmented into blobs generated from image features using clustering. A probabilistic *Cross-Media Relevance Model* (CMRM) then estimates the joint distribution of blobs and words. Instead of supporting a one-to-one mapping between words and blobs, the relation between words and blobs is retained globally for the image. They propose three sub-models to represent the images: (a) probabilistic annotation-based CMRM in which each image is represented by a vector of probabilities for each label similar to Chang et al.'s CBSA; (b) fixed annotation-based CMRM, in which only a small number (3–5) of labels without their probabilities is retained for annotation; and (c) direct-retrieval CMRM which does the opposite translation of query words into blobs. Having a vocabulary of both words and blobs allows flexibility for the retrieval model to compute the similarity between images and for query formulation supporting both text-based as well as query-by-example queries. This model is a good example of a well-founded unifying approach of textual and visual features. However, both Carneiro & Vasconcelos (2005) and Yavlinsky et al. (2005) have shown that a much simpler model that estimates the visual feature distribution associated with each word rather than jointly modelling the distribution of image segments and words performs better and is computationally more efficient.

Moving away from textual representations, Lim developed the idea of *Visual Keywords* (Lim 1999). Visual Keywords are visual prototypes extracted and learnt from the visual content that has been annotated with relevant semantic concepts. The author's approach characterises both types of visual objects (eg 'building', 'sky') as well as spatial configurations. An image is indexed as a spatial distribution of Visual Keywords. The detection of keywords is based on the existence of a vocabulary of visual templates, which are represented by a feature vector (eg colour, texture), to which the features of visual tokens are compared. The vocabulary is constructed by training on regions of sample images. Their spatial configuration is encoded by aggregating Visual Keyword occurrences according to spatial configuration templates, with the possibility of supporting different spatial configuration maps in order to exploit domain-knowledge. Users can specify queries by "Visual Constraints", ie selecting words from the visual vocabulary and placing them onto a canvas for the spatial layout (Lim 2000). Mulhem & Lim (2002) have integrated the Visual Keyword approach with a conceptual graph representation in order to encode and make inferences on relationships between image elements. The complete aggregation process results in a very compact representation of the images, however, it is not clear how it is possible to determine afterwards the Visual Keywords that resulted in the image being retrieved. This prevents the incorporation of relevance feedback techniques to improve the annotations.

The above examples constitute only a small number of recent approaches towards semantic image retrieval. Numerous other techniques have been introduced in the literature. Worth mentioning are, for example, Duygulu et al.'s translation approach (2002), the use of latent semantic

analysis (Zhao & Grosky 2000) to detect the latent correlation between low-level features and high-level concepts, a hierarchical classification approach that exploits ontological relationships between words (Srikanth et al. 2005), a graph-theoretic annotation approach (Pan et al. 2004) (discussed in Section 7.3.1), Hidden Markov Models (Ghoshal et al. 2005) and Wang et al.’s *SIMPLICITY* system (2001). Further, Datta et al. (2005) devote a section to recent work on annotation and concept detection in their survey.

Most approaches are tested on their annotation accuracy or retrieval performance in partially annotated photographic collections. A unique application is presented by Rath et al. (2004), who have built a system to retrieve historic handwritten manuscripts, in which automatic labelling is formulated as a translation problem between word images and words.

**Discussion** One of the main problems with automatic annotation is the necessity of agreeing on a vocabulary. Duygulu et al. (2002) provided an initial point of reference using a subset of the Corel dataset (COREL n.d.), containing 5000 images and 371 words, allowing performance comparisons. More recently, there have been efforts in developing a Large-Scale Concept Ontology for Multimedia (LSCOM) (Naphade et al. 2005) and a comprehensive classification challenge (Snoek et al. 2006) both using the TrecVid data (TrecVid 2005) (see Section 2.5.1).

Annotation-based semantic retrieval is based on the assumption that the user can easily relate to the semantics extracted by the system and subsequently create a mapping from the semantic space for querying. Even though this mapping is more intuitive than a mapping from low-level features, the semantic space still often lacks the richness of concepts the user has in mind. An example might make this point clearer. Bradshaw’s technique extracts four semantic classes: man-made/natural, inside/outside (Bradshaw 2000). The author claims that in order to retrieve “pictures from the holiday in Wales” the user maps this query to the semantics “outside-ness” and “natural-ness”. Not only will these broad categories most likely perform rather poorly for this quite specific query (depending on the *context* of the image collection), but other types of query cannot be mapped into the chosen semantics at all. For instance, it is very unlikely that these two categories reflect queries like “Find me pictures of our dog”. If the dog happens to have a distinctly textured and coloured skin, content-based primitive features might have performed much better in this case. This example highlights the deficiency of using only a very small set of concepts for image representation, since this might actually limit the types of query one can pose to the system rather than enhance its capabilities.

### User-based Semantic Mining

User-based approaches, on the other hand, attempt to extract and learn the semantic concepts the user has in mind. So, in addition to addressing the issues of how to effectively describe and extract semantic information of an image, a user-based approach is concerned with the question of how to learn and improve the semantic space from user interaction and feedback.

Relevance feedback refers to the techniques in which a user is given the opportunity to mark search results as either relevant or irrelevant (possibly by degrees) according to the current request. The degree of (ir)relevance of the chosen examples is fed back into the system, where it is used to improve its current knowledge state. Consequently the system will respond with a new set

of updated results (see Section 2.3). Under the assumption that all images marked relevant in a query session share a common semantic concept, relevance feedback results can be used to infer a semantic space.

A popular way to assimilate the user in the process of mining for semantic concepts is achieved in a probabilistic annotation-based setting. In such a model, an image is represented as a vector of labels including confidence factors (similar to the purely annotation-based approaches by Jeon et al. (2003) and Chang et al. (2003)). In the course of a search session, the user gives feedback on relevant images. Accordingly, the confidence factors of labels are continuously updated to reflect the user's feedback.

**Approaches** Su & Zhang's framework for relevance feedback in CBIR is an example of such a probabilistic approach (2002). The framework comprises a *semantic network* linking images to annotations. In this network, each keyword is linked to a number of images with associated weight links (reflecting the confidence of the annotation). During user feedback, keyword annotations are propagated to other images by means of a probabilistic learning process. Through the propagation process, the keywords that represent the actual semantic content of each image will receive a large weight, and ultimately the keywords with a majority of user agreement will emerge as the dominant representation of the content. Also, as more keyword queries are issued to the system, it is able to expand its vocabulary. The proposed framework additionally caters for cross-modality query expansion. Apart from the semantic network, the images are indexed by their visual attributes. The query expansion is supported, in that the system extends a keyword-based query into feature-based queries and vice versa.

Using the positive feedback, collected during an interactive search session, is also promoted as a way of obtaining semantically related images by Zhou & Huang (2002). The key to their approach again comprises a semantic network. Unlike in Su & Zhang's framework, their semantic network is in essence a term similarity matrix, which models keyword relationships. The idea is to generate a thesaurus incrementally that captures data-dependent and user-specific term associations. The knowledge about semantic groups of keywords is learnt from the user: If a group of images is selected relevant during a retrieval session, the similarities among all the keywords assigned to the current images are strengthened. In addition to the keyword annotations and the thesaurus, images are represented by low-level content descriptors, which are jointly used for retrieval and for "traditional" relevance feedback learning to estimate optimal parameters of the feature space (eg the relative importance of features) in order to further improve retrieval results (cf Section 2.3).

He et al. (2003) infer a semantic space from user interaction and image content. Their approach is different from the previous two in that textual annotations are not strictly necessary to represent the semantic concepts. Instead, they propose to construct *hidden semantic features* in the absence of information on specific textual attributes. The semantic space is defined as a  $m \times n$  matrix that contains  $m$  (the number of images) rows for each image in the collection, each represented by  $n$  hidden semantic features. This matrix is incrementally constructed by appending the relevance judgements after each query session. Thus, if images 1, 2 and 5 are selected as relevant in the first session, the first column of the semantic space matrix will be represented by a vector

containing the value 1 at positions 1,2,5 and the value 0 at all other positions. The resulting matrix is further reduced to a lower dimensional space using Singular Value Decomposition to emphasise the correlation between queries and to reduce noise from spurious relevance judgements (an idea similar to latent semantic indexing for text retrieval). This reduced matrix is optimised to retain the salient semantic concepts reflecting the largest user consensus. Again, the semantic space is accumulated over multiple retrieval sessions facilitating long-term learning, while a short-term learning method is included in the framework as well, and is used to refine the retrieval results in a single query session. Similarly, Lin et al. (2005) propose to learn a reduced dimensionality space by exploiting both image relationships based on low-level visual similarity and on user feedback. The relationships are represented in three different graphs: (1) similarity relations; (2) positive relations as indicated by the user; and (3) negative relations. The dimensionality reduction process is formulated as an Eigenvector problem. More graph-based techniques to represent user feedback are discussed in the related work section (Section 7.3.1) of Chapter 7, where we introduce our approach towards learning and representing a semantic feature.

Finally, Truran et al. (2005) consider learning the relationship between query terms and selected images in an image search engine on the Web. They propose to create a new feature space consisting of the union of query terms issued to the system and sort selected images along these dimensions. The approach is used for resolving query term ambiguity by clustering the images along the feature dimension given by the query term. The word-sense clusters identified are therefore based on the “co-active intelligence” of the search engine’s users. However, the approach does not attempt to relate the emergent text senses with visual features.

### Discussion

As pointed out earlier, the difference between the two semantic mining concepts lies in the source of knowledge from which the semantic concepts are drawn. Off-line annotation-based methods aim at learning general semantic concepts that can at best incorporate contextual information on the image domain. In contrast, user-based methods infer semantic concepts that are domain- as well as user-specific. The major disadvantage of methods that exclude the user is that classes are predefined. The training necessary for image categorisation demands the existence of a clear class structure with a predefined number of concepts. This is, as such, unsuitable for CBIR, since concepts representing the semantics of images are not well defined. Semantic concepts arise by definition from subjective *interpretation*, and hence, they are dependent on contextual factors regarding the user, the particular query, the collection domain and numerous other hidden factors influencing the retrieval context.

Most approaches, independent of whether they are annotation- or user-based, are grounded on the assumption that images with similar semantics share some similar features. Yet this assumption may not always hold. An approach that does not depend on any prior similarity measure is introduced by Yin et al. (2003). They define the similarity between two images only on the basis of co-occurrence of positive labels from relevance feedback logs. The similarity values are used to create semantic clusters of images. The resulting clusters approximate the semantic concepts the user’s actions reflect while querying the images. Since a user’s actions are often inconsistent (see Section 3.2.3), the quality of clusters obtained implicitly from such unpredictable actions alone

is likely to be inferior to approaches that encompass both subjective information from the user as well as objective information from visual features or other sources.

### 2.2.5 Summary

In summary, the unifying approach is, without a doubt, the most promising direction for the future. Firstly, low-level features should be employed in combination with conceptual features. Low-level features, on their own, lack the semantic capabilities required by most users, while semantic concepts are just too great a challenge to obtain independently from low-level content. When combined, visual features are useful for propagating semantic labels from (manually) labelled images (label should be understood generically, arising for instance from keyword annotations, relevance judgements, etc.) to others based on visual similarity. Secondly, user-assisted labelling techniques can help to improve and refine the semantics learnt from purely visual-based categorisation. In addition, a proper learning framework plays a crucial role in the personalisation of retrieval systems.

Nevertheless, semantic feature representation still remains an open issue, despite recent advances (cf Section 3.2.1). We believe the mining of a semantic feature should be based on the context provided by the user. Therefore, it should go hand-in-hand with the interface design. The goal of this thesis is to design an interface that “makes sense” to both the user and the system.

The techniques introduced in this section highlight the importance of learning methods in CBIR. Learning has indeed been the dominating factor to narrow the semantic gap arising from the low-level feature representation in the last few years. The following section will cover learning techniques used in image retrieval systems in a more principled manner.

## 2.3 Learning from Relevance Feedback

The semantic gap marks the greatest barrier for the advancement of current CBIR systems. However, it is not only due to the shortcomings regarding this gap that it is almost impossible for a state-of-the-art retrieval system to provide a satisfactory answer to a user’s request in the first iteration. Rather, the reasons for the failure of one-shot queries are manifold. Firstly, it is impossible to capture the semantic concepts depicted in the image by the low-level features available at present (see Section 2.2.4 and 3.2.1). Secondly, the user cannot easily grasp the low-level representation of the images, with the result that the translation of the user’s information need into a formal request poses a major obstacle for the user (see Section 3.2.2). And finally, making matters even worse, the need is likely to change over time (see Section 3.2.3). The consequence of the dynamic nature of information needs is that the system can only guess on the real need from the initial query and more importantly from the user’s interaction with the system.

Therefore, image retrieval has to be an inherently dynamic process in which the system learns from the user and the user learns from the system. In such an environment, the search process is initiated by a user-supplied query, returning a small number of documents to the user. Thereafter, the retrieval process consists of the following stages (note the alternation between the system and the user in the process):



1. the system makes suggestions in the form of a set of (ranked) images
2. the user provides feedback on the relevance (or irrelevance) of images in the set
3. the system updates previous suggestions

This process is iterated until a satisfactory answer—in the user’s eyes—to the current information need is found. During this interaction, both the user should learn about the system and the system should learn about the user. So, both the user and system can become more efficient as time passes. The user’s learning process depends largely on a well-designed interface that communicates its internal processes and representations—the system image—well. This discussion will be resumed in Section 4.1. In this section, we are concerned with the system’s ability to improve with the help of the user’s feedback. If the system does not seem to make intelligent suggestions, the process will be tedious for the user and the system will be rejected.

The system’s improvement is achieved through some learning algorithm. One can distinguish between *short-term* and *long-term* learning. If the learning takes place *within* a retrieval session, it is referred to as short-term. Long-term learning *across* retrieval sessions, on the other hand, requires the system to possess “memory”. Long-term learning is predominantly employed to discover semantic concepts in the images. A few examples of such techniques have been introduced in the previous section (Section 2.2.4). The following discourse will therefore concentrate on the short-term learning approach. Since this approach is dependent on the information gained from the user’s relevance judgements, it is simply referred to as *relevance feedback*.

### 2.3.1 Overview of Relevance Feedback

The idea of incorporating relevance feedback first emerged in text retrieval systems (*eg*, Rocchio 1971, Salton & Buckley 1990) and has been studied since. In comparison to purely text IR systems, it is even more valuable in the image domain: a user can tell instantaneously whether an image is relevant with respect to their current context (information need, awareness of information need, etc.), while it takes substantially more time to read through a text document to estimate its relevance.

**Motivations** Relevance feedback is regarded as an invaluable tool to improve CBIR systems, for several reasons. Apart from providing a way to embrace the individuality of users, they are indispensable to overcome the *semantic gap* between low-level image features and high-level semantic concepts. The user’s judgement of relevance is naturally based on their current context, their preferences and also their way of judging the semantic content of the images. The low-level image features are used as a quick way to ‘estimate’ the relevance values of the images. By prompting the user for relevance feedback, this initial estimation can be improved to steer the results in the direction the user has in mind. Rather than trying to find better techniques and more enhanced image features in order to improve the performance of what has been referred to as *computer-centric* sys-

tems (Rui et al. 1998), it is more satisfactory to the user to exploit the interface to refine high level queries to representations based on low level features. This way, the subjectivity of human perception and the user's current context are automatically taken into account. Consequently, it does not come as a surprise that various techniques to make use of relevance feedback in CBIR have been suggested in the literature. A comprehensive study of existing relevance feedback techniques in image retrieval can be found in (Zhou & Huang 2003).

**Overview of Approaches** Relevance feedback is engaged in finding optimised ways of updating the parameters of the retrieval algorithm. Traditionally, this has been achieved through query refinement approaches. These approaches underlie a geometric interpretation of the feature and query space. In most CBIR systems, the images are represented by their feature vectors in the vector space model (Salton & McGill 1983). The degree of dissimilarity between images can thus be interpreted as the Euclidean distance of the respective feature vectors. So, query refinement approaches strive to find the "ideal" query point that minimises the distance to the positive examples provided by the user. Prominent techniques for the geometric approach include:

1. *Query shifting*: moving the query vector closer to an area in the feature space that contains relevant documents.
2. *Feature re-weighting*: update the weights of the features to reflect the different relative contribution of the components, such as colour, texture and shape.

The geometric approach will be introduced in greater detail in Section 2.3.2.

From these initial heuristic approaches, borrowed directly from the text retrieval domain, relevance feedback research has moved towards optimised learning techniques that treat relevance feedback as a machine learning problem. This is motivated by the fact that machine learning is concerned with the problem of devising computer programs that automatically improve with experience. In this respect, relevance feedback can be considered a machine-learning task, which aims at *improving* the retrieval performance on the basis of the *experience* provided by the user through examples (Section 2.3.3).

### Characteristics

Different methods have been adopted on the basis of often diverging assumptions. One major variance is *what* actually is fed back to the system. Often, binary feedback for positive and negative examples is used (eg, Tieu & Viola 2000), some additionally associate a 'degree of (ir)relevance' (eg, Porkaew et al. 1999) and others interpret the feedback only as a 'comparative judgement' (eg, Cox et al. 2000). Depending on the assumptions taken in this respect, the resulting systems can be distinguished further: While positive feedback has been used for feature selection (eg, Peng et al. 1999) or feature relevance weighting (eg, Porkaew et al. 1999, Ishikawa et al. 1998), using both positive and negative feedback gives rise to treating the retrieval process as a classification or learning problem. Many systems now strike the latter path, transferring methods previously employed mainly in the field of artificial intelligence (eg, Tong & Chang 2001, Wood et al. 1998, Tieu & Viola 2000). However, they are hindered by one major obstacle, namely the *small sample issue* (Zhou & Huang 2003). The user feedback in each iteration only gives a tiny number of

training samples relative to the high dimension of the feature space and the possible number of classes for general multimedia data. This issue is further discussed in Section 2.3.3.

A further characteristic of existing systems is *how* they gain information about the user's judgement of relevance. One can distinguish between two distinct approaches: *explicit* and *implicit* relevance feedback. Explicit relevance feedback, which is assumed in most current systems (eg, Cox et al. 2000, Porkaew et al. 1999, Tong & Chang 2001), asks the user to explicitly state whether a returned document is relevant or not. Therefore, the user interface has to provide for facilities to input this judgement by the user, such as in *MARS* (Porkaew et al. 1999) described in Section 2.4.2. This additional task is often considered a burden to the user, since it is difficult for most users to assess the degree of relevance of one document in terms of a numeric value, which presumes considerable knowledge of the retrieval environment. Although it might be much easier to determine whether an image is actually relevant to the user compared to formulating a good query, it still often requires considerable cognitive effort from the user to communicate this relevance assessment to the system (Ruthven 2005). For this reason, a less-distracting possibility to gain relevance feedback is implicitly from the users, simply by observing their interaction with the system.

Another assumption underlying nearly all current relevance feedback techniques is that a user's information need is static and there is no provision for updating user judgements. Especially those techniques that attempt to classify or separate the document space into relevant and non-relevant, explicitly rely on the assumption of having constant relevance values. However, this is a rather simplifying view of the real-world. Not only are the user's *actions* time-dependent—resulting in giving inconsistent feedback, but even more importantly, the user's *goals* are also time-dependent and might change either gradually or quite abruptly. The trigger for such changes is most often a result of having come across something interesting that they have not even considered at the beginning of the search. For this reason Campbell & van Rijsbergen have proposed the *Ostensive Model*, which captures “*the intentionality of an information need that is assumed to be developing during the searching session*” (Campbell 2000a, p. 88). The model will be detailed in Section 3.1.2.

In the following some representative techniques will be surveyed. The survey will roughly follow the “evolution” of relevance feedback techniques in CBIR over the last decade.

### 2.3.2 Geometric Approaches a.k.a. Query Refinement

To relieve the user from the query formulation problem, a method that is able to guess or learn the *ideal* query reflecting the user's desires purely from a set of examples is believed to be very advantageous. Most query learning techniques are based on a geometric interpretation of the feature space.

In geometric approaches, images represented by a feature vector are interpreted as points in the (high dimensional) feature space. The distance between images is measured by the Euclidean distance (or variants of it) of their vector representations. This interpretation imposes an assumption of feature independence, which in regards to the highly correlated nature of some features is rather artificial.

Algorithms for CBIR that rely on query refinement as a way of incorporating relevance feed-

back have attracted a lot of interest (eg, Ishikawa et al. 1998, Porkaew et al. 1999, Rui & Huang 2000). They are based on early work in the text retrieval domain (Rocchio 1971), and have been adapted to CBIR. *MARS* is one of the earliest and probably most influential systems deploying relevance feedback. In the system developed by Rui et al. (Rui et al. 1997, Porkaew et al. 1999) algorithms were implemented to put into practice the mainly heuristic take-on of relevance feedback theory based on the geometric approach.

There are two major variants of these early relevance feedback techniques. Each of them aims at learning different components of the retrieval space:

1. The optimal query is adapted directly by query modification/shifting; or
2. A feature space transformation is learnt, involving an adaptation of the dimensions of the feature by feature re-weighting or selection.

I will describe each of these variants in turn, although most systems implement a combination of them (Porkaew et al. 1999, Ishikawa et al. 1998, Rui & Huang 2000).

### Query Shifting

The prevalent technique of adapting an initial query is *query shifting*. It aims at moving the query toward the region of the feature space containing the set of relevant documents and away from the region of the set of non-relevant documents (see Figure 2.1). There are two underlying assumptions in this technique. One is that relevant images are clustered in feature space, and secondly the user has an *ideal* query in mind. The system's task is consequently to find this ideal query locating the region containing the relevant images. The validity of these assumptions is questionable (see the discussion in Section 2.3.4), but for the moment these doubts shall be left aside.

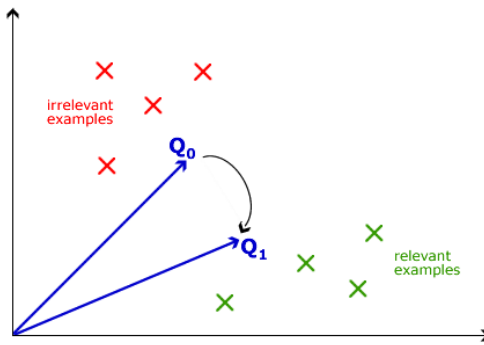


Figure 2.1: Query Point Movement in a 2-D space

The best known technique developed for text retrieval systems based on the vector space model is Rocchio's formula:

$$\vec{q}_1 = \alpha \vec{q}_0 + \beta \left( \frac{1}{n_R} \sum_{i=1}^{n_R} \frac{\vec{r}_i}{|\vec{r}_i|} \right) - \gamma \left( \frac{1}{n_S} \sum_{i=1}^{n_S} \frac{\vec{s}_i}{|\vec{s}_i|} \right) \quad (2.1)$$

where  $\vec{q}_0$  is the vector for the initial query,  $\vec{r}_i = [r_{i1}, \dots, r_{in}]^T$  the vector for *relevant* document  $i$  (and  $n$  is the total dimension of the feature space),  $\vec{s}_i$  the vector for *non-relevant* document  $i$ ,  $n_R$

the number of relevant documents and  $n_S$  the number of non-relevant documents. The new query  $\vec{q}_1$  is obtained by a linear combination of the ‘mean’ vectors of relevant and non-relevant, so that  $\vec{q}_1$  is close to the mean of relevant documents and far away from the non-relevant mean. The three parameters  $\alpha$ ,  $\beta$  and  $\gamma$  are usually chosen by experiment and make the formula subject to heuristic considerations.

Query point movement roughly based on Rocchio’s formula has been adopted in a few image retrieval systems (Porkaew et al. 1999, Ishikawa et al. 1998, Rui & Huang 2000). The query point movement (QPM) strategy as implemented in *MARS* (Porkaew et al. 1999) is a simplification of Rocchio’s formula in that only positive feedback is taken into account. After each round of feedback, a new query point is computed as the centroid of all images marked as relevant. Let  $R = [\vec{r}_1, \dots, \vec{r}_{n_R}]$  denote the  $n_R \times n$  matrix obtained by stacking all positive examples into a matrix and  $\vec{w} = [w_1, \dots, w_{n_R}]$  the relevance weight vector associated with the positive examples. The new query vector  $\vec{q}$  is obtained by:

$$\vec{q} = [q_1, \dots, q_n] = \frac{R^T \vec{w}}{\sum_{i=1}^{n_R} w_i} \quad (2.2)$$

$$q_j = \frac{\sum_{i=1}^{n_R} w_i r_{ij}}{\sum_{i=1}^{n_R} w_i} \quad (\text{for } j = 1, \dots, n) \quad (2.3)$$

As an improvement to QPM, the same authors have proposed a *query expansion* (QEX) strategy, as well. The expansion is achieved by clustering the positive points and adding a number of cluster representatives to the query. As opposed to QPM that only allows one point in the feature space as the query, the expansion technique results in multi-point queries. Evaluations have shown that QEX performs better than QPM (Porkaew et al. 1999). One of the explanation given by the authors is that identifying local clusters in the relevant set corresponds more closely to the nature of the ‘‘typical’’ information need. QPM combines all positive points into one centroid, treating all relevant images equally. Yet, according to the authors, relevant images often tend to be scattered in the feature space because of the semantic gap. This can be captured to a somewhat better degree with a query expansion strategy.

Although this technique could be shown to lead to significant improvements of retrieval effectiveness, it is often criticised as being heuristic without proper mathematical justifications. The next two approaches, on the other hand, define learning from relevance feedback as an *optimisation problem*. *MindReader* takes on this approach (Ishikawa et al. 1998). The authors examined the problem of query refinement from a more systematic point of view. They formulate the problem as a minimisation problem of the total distance of all positive examples from the query point. With respect to this, the derived ideal query point is proven to be the weighted centroid of all positive samples (as computed by Equation 2.2).

Rui & Huang (2000) claim that the *MindReader* approach, ‘‘even though elegant in theory’’, is based on an over-simplistic flat data model similar to most other CBIR systems. The problem lies in the combination of different feature representations. Since most low-level features can be represented as a real-valued vector, it is common practice to simply stack all the features’ elements into one overall feature vector. However, in order to discriminate between the features, Rui &

Huang propose a hierarchical feature model. They define a framework, in which, again, a solution for minimising the total distance between all the relevant images and the “ideal” query is sought. Both *MARS* and *MindReader* can be made to fit into their general framework. The ideal query point could again be shown to be the weighted average of the training samples. Rui & Huang’s model regarding feature representation and learning is described in detail in Section 5.1.1 and Section 5.1.2, respectively. It is the underlying retrieval model chosen for the initial recommendation system in *EGO* based on visual features only.

### Feature Re-Weighting and Feature Selection

Often query refinement methods are used in combination with *feature re-weighting*, which is based on a weighted similarity measure where relevance feedback is used to update the weights associated with each feature in order to model the user’s need (Porkaew et al. 1999, Ishikawa et al. 1998, Rui & Huang 2000, Minka & Picard 1996, Santini & Jain 2000).

Feature re-weighting is a simple form of feature space transformation. The approaches introduced in Section 2.2.4 aim at a mapping of the original visual feature space into a semantic space, that better captures the high-level concepts. The re-weighting of the feature axes can be seen as a special case of this.

**Preliminaries** Feature weighting is in fact a way of changing the parameters of the similarity function, so that it reflects “close-ness” in the feature space more accurately. Assuming the Euclidean distance to determine similarity between feature vectors, the set of  $k$  nearest points to the query vector is determined. If the features are equally weighted, the nearest neighbours are within a circle centred at the query point (see Figure 2.2(a)). When introducing different weights of the dimensions, the distance is computed by a weighted Euclidean. The shape of the isosurface of the query deforms with the weights. Thus, the circle transforms into an ellipsoidal shape, which stretches along the feature axes to adapt to the distribution of relevant features (see Figure 2.2(b)). In addition to reweighting the feature space, the general Euclidean distance can additionally model mapping into a new feature space resulting in a rotation of the ellipsoid (see Figure 2.2(c)). In the new feature space, correlations between features can be captured.

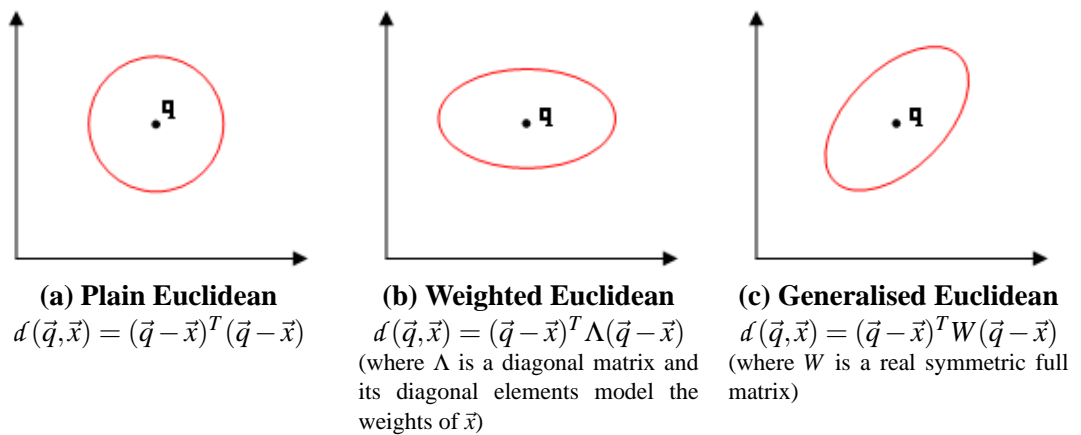


Figure 2.2: Isosurfaces of distance functions (adapted from (Ishikawa et al. 1998))

**Approaches** In Rui et al.'s *MARS* system (Rui et al. 1997, 1998) feature re-weighting is achieved according to a *standard deviation method*. The variance of each feature over the set of positive and negative examples is indicative for the relative importance of a particular feature. The intuition behind this idea is that, if the standard deviation of the positive examples according to a particular feature is high, this feature is not useful for discriminating relevant from irrelevant images. Thus this feature will receive a low weight. If a feature's values are consistent among all the relevant images, on the other hand, their standard deviation can be expected to be small. Features with small standard deviation are thus assigned a larger weight. The new feature weight is hence selected as the inverse of the standard deviation, and used for computing similarities according to the weighted Euclidean distance.

Ishikawa et al.'s *MindReader* (1998) offers an improvement over the previous method. Their proposed algorithm is independent of heuristic parameters (such as  $\alpha$ ,  $\beta$  and  $\gamma$  in Rocchio's formula) and they take into consideration the correlation between features. This is achieved by considering the generalised Euclidean distance (see Figure 2.2(c)). Rather than calculating a weight vector, the weights are represented as a full *matrix*, which can capture correlation between different feature dimensions. This matrix is determined from the covariance matrix of  $R$  (the matrix of all positive examples, see above).

Rui & Huang (2000) present an optimal solution for the query vector, the feature transformation and the similarity function. The authors reject using a "flat" data model simply stacking up all features (see above). Instead they base the computations on a hierarchical data model, in which each *individual* feature's similarity is modelled as the generalised Euclidean distance. An overall similarity is computed as linear combinations of the individual similarities. The weight matrix for each feature is obtained in the same way as in *MindReader*, however, one matrix for each feature is computed instead of an overall one. Additionally, a weight vector to combine each similarity score is derived as the optimal solution for minimising the total distances for each feature.

Despite being designed for on-line learning, the optimal learning approach is in effect achieved by batch learning, ie it requires all samples to be given simultaneously before it can learn, and there is no easy way to incrementally incorporate a new example without recomputing the weights. Further, its optimality criterion is only satisfied if the user gives sufficient feedback (more than the maximum of the dimensionality of each feature).

*FourEyes* (Minka & Picard 1996) incorporates a learning mechanism over various stages. For each feature separately, a hierarchical grouping of image regions or images is pre-computed. Each grouping receives a weight, which will be updated on the basis of feedback provided by the user. The chosen examples lead to clusters being updated and selected. Additionally, compound groupings can be created so that they include all positive examples and none of the negative examples.

Lastly, the approach taken in *El Niño* (Santini & Jain 2000) differs substantially from the previous techniques. Instead of presenting the images in a linear list or grid, a *configuration* of images is displayed. In this configuration the distance between any two images reflects their similarity as currently calculated by the retrieval system. The way of obtaining relevance feedback is also treated in a different manner. Their interaction model communicates *context feedback*, in which the user provides feedback by moving the images in the configuration. Thus, images the user considers similar will be moved closer to each other, while irrelevant (or belonging to a different

group) will be placed further away. The weights of the distance function will be calculated by minimising the error between the original distance of two images calculated by the system and the communicated distance given by the user. This “learning from the layout” of images marks a novel form of training.

### Discussion

The different approaches towards query refinement are complementary. Query shifting assumes that the original query can be centred in the region containing the set of relevant images. Once the centre has been found, the extent of the relevant region can be changed by updating the feature weights. Modifying these parameters of the similarity function in the direction of the most prominent features is aimed at retrieving a larger number of relevant images.

While it is a problematic but practical assumption that relevant images form one cluster in feature space, irrelevant images are at best clustered into several classes (Zhou & Huang 2003). Further, the class of irrelevant images is very heterogeneous and much larger in comparison to the relevant class. Thus, the small number of negative examples given by the user is unlikely to be truly representative for all the irrelevant classes. It has consequently been argued for an asymmetric treatment of positive and negative feedback (T.V. et al. 2002, Zhou & Huang 2003).

### 2.3.3 Statistical Approaches

Recognising the importance of relevance feedback in CBIR, the interest in learning techniques has created numerous alternative suggestions to the geometric interpretation. The advance of relevance feedback learning forms a momentous impact for CBIR systems. While the earlier techniques were formulated in the terms laid out by the treatment of relevance feedback in text retrieval applications, emerging new proposals are tailored specifically for the visual or multimedia domain. Freeing itself from its elder, and very different, brother, elaborate ideas based on well-founded theoretical frameworks have been introduced.

Consequently, the attempt of learning the ideal query point has been abandoned in favour of other techniques that are less dependent on the particular feature representation. As a result, relevance feedback has been formulated in probabilistic frameworks as belief propagation (Cox et al. 2000, Vasconcelos & Lippman 2000), or as a classification task (Wood et al. 1998, Tong & Chang 2001, Tieu & Viola 2000). The former belongs to the class of *generative methods*, while the latter is *discriminant*. The generative approach aims at generating beliefs of some learning hypothesis (eg estimating the probability of an image being the target of the search (Cox et al. 2000)). The goal of the classification approach, on the other hand, is to learn to discriminate the images in the database into a number of distinct classes. Although the distinction between generative versus discriminant approaches is made according to the nature of the underlying representation, techniques in both variants avoid formulating the problem in a particular feature space.

### Generative Methods

Generative methods are typically belief propagation algorithms. In a probabilistic setting learning involves estimating the probability of either an image belonging to the relevant or non-relevant



class, respectively (for *category search*), or an image being the target of the search (for *target search*). The probabilities are often propagated according to Bayes' rules (in the Bayesian settings). To optimise these probabilities, the algorithms attempt at minimising the classification error.

Cox et al. (1996) were one of the first to address learning from a different point of view than most other research teams. They pioneered treating it as a problem of predicting the user's actions. The predictive model has been defined in a probabilistic Bayesian framework and implemented in the *PicHunter* system (Cox et al. 1996, 2000). The model is formulated under the premise of a *target search*, ie the user looks for one specific image in the database. In each iteration the user selects those examples from the set of images displayed, that they consider being similar to the target image. Under this assumption, the learning algorithm involves estimating the likelihood of any database image being the target. These probabilities are conditioned on the history of relevance feedback actions collected over the entire retrieval session. In order to compute the target probabilities, a user model and image display strategy needs to be taken into account. The user model determines how the user's actions can be predicted in the face of the given history of actions and the currently displayed images. Predicting an action involves predicting human judgement of image similarity, which is estimated by calculating the visual similarity based on primitive features. The probability of an image  $I$  being the target is updated depending on the relative distance of image pairs formed by taking one selected and one displayed, but non-selected, image. The probability for  $I$  is increased or decreased depending on the similarity to the selected and the non-selected example in the pair. Instead of showing the "*most-probable*" images after each feedback iteration, *PicHunter* opts for a "*most-informative*" display updating scheme, in which images are chosen in order to minimise the expected number of future iterations, thus maximising the immediate information gain. In summary, the *PicHunter* framework treats relevance feedback as a problem of searching for decisions that are optimal with respect to the entire retrieval session. However, their scheme is computationally expensive, since the conditional probabilities for all images in the database need to be updated after each round of feedback.

Vasconcelos & Lippman (2000) provide another example of how to integrate relevance feedback as belief propagation. Retrieval and learning is considered as Bayesian inference, in which the goal is to minimise the probability of retrieval error. Beliefs are accumulated over the retrieval session, taking into consideration a *decay factor*. This factor can model changes in the information need over time by weighting the importance of the past. Lastly, Bayesian inference is also used by Meilhac & Nastar (1999) in category search to find an optimal separation between relevant and non-relevant images.

In fact, the probabilistic theory provides a number of advantages for multimedia retrieval, which have led to a growing number of proposed applications. These advantages include (de Freitas et al. 2002):

- incorporation of a priori knowledge or subjective preference
- cross-media modelling of multimedia data
- large number of possible application areas

De Freitas et al. (2002) propose Bayesian models for text, music and image documents, and show

their applicability for browsing, information retrieval, annotation and object recognition. The probabilistic annotation approach has recently received considerable publicity (eg, Jeon et al. 2003, Chang et al. 2003) to arrive at cross-media relevance models and ultimately capture semantic concepts through the combination of multiple media (see Section 2.2.4). Its advantages could be combined with the advantages of relevance feedback learning (Su & Zhang 2002).

Despite the advantages of providing interpretable models and principled ways to incorporate prior knowledge and data with missing values, alternative approaches compete with probabilistic frameworks. Discriminative methods are strong contestants in this competition. Their asset is typically claimed to be superior performance (Tong & Chang 2001).

### **Discriminant Methods**

Discriminant methods applied to relevance feedback strive to design the classifier that best separates the positive from the negative examples. This is accomplished by explicitly finding the boundaries in the feature space that best separate the two classes.

In general, discriminant methods are not necessarily binary classifiers. However, for relevance feedback applications, the number of classes is usually limited to two: *relevant* and *nonrelevant*. In image retrieval applications, the initial state is that no image in the database is assigned to any of these classes. The goal of the classifier is to give a label to each of them such that for any image the computed labels will agree with the user's labels.

**Issues** Before going into detail of the techniques used to achieve a classification, a few issues need to be pointed out. These concerns have to be taken into consideration when designing a classifier. Most importantly, the number of available examples to learn from is very small. The user cannot be expected to judge more than 10–20 images per iteration, and typically this number rather ranges between 1 and 5. In relation to the high dimensional feature space, this poses a great challenge to traditional classification algorithms that require between 100s and 1000s of examples to converge. This problem is referred to as the *small sample issue* (Zhou & Huang 2003). Consequently the results are unreliable on their own, requiring a lot of extra effort, eg an off-line training phase following the on-line search, as employed in (Wood et al. 1998), to arrive at meaningful results. This is often undesirable, since it militates against the real-time requirement of relevance feedback. The main advantage of relevance feedback, namely that it allows real-time learning from user interaction to improve the system's performance during one search session, is thus undermined.

The response-time is another critical aspect of the algorithm at hand. Since short-term learning from relevance feedback happens in an interactive setting, the updated result set should be delivered as quickly as possible to the user. The real-time requirement should always be kept in mind, and in some cases an approximation of the classification should be preferred over an optimal solution.

Discriminant methods share the same concern with query shifting techniques regarding the validity of the cluster hypothesis, namely that relevant and irrelevant images are clustered in feature space. Especially the asymmetry in feedback samples mentioned in the discussion of the previous section (Section 2.3.2) has led to proposals to treat relevance feedback as a multi-class problem

rather than a two-class one (Nakazato, Dagli & Huang 2003).

**Machine Learning Techniques** Conventional machine learning techniques are neural networks and Support Vector Machines (SVM). They have often been used for various classification tasks, such as face detection and character recognition. For relevance feedback learning in retrieval systems, the original techniques had to be adapted in order to make it possible to learn from the small number of training samples in comparison to the high dimensional feature space and the large number of potential image classes.

The adaptation can be achieved by making an “intelligent” decision of which images are chosen as training samples. Machine learning algorithms in different domains are usually applied on a fairly large number of randomly-chosen training samples. Applied to image retrieval, fast convergence using a small number of examples is crucial. Fast convergence can be supported by choosing the “*most-informative*” images for display to the user during learning. Most-informative images are those close to the decision boundary of the classifier that will result in the biggest impact for updating the boundary if they are labelled by the user.

This “*active learning*” component is suggested by Tong & Chang (2001) for use in combination with support vector machine (SVM) learning. An SVM in its simplest form is a binary classifier. The data is separated into a positive and negative class by determining a hyperplane dividing the representation space with a maximal margin between these two classes. Since it is not often the case that the two classes are linearly separable, SVMs allow one to project training data from the original representation space to a higher dimensional feature space. In the feature space, the training data is linearly separable by a hyperplane. The hyperplane determined in this way constitutes the decision boundary for the classification. The projected points lying on one side of the hyperplane receive a positive label and will be considered relevant for the current query, and the rest a negative one. An example of a decision boundary computed by a SVM is depicted in Figure 2.3.

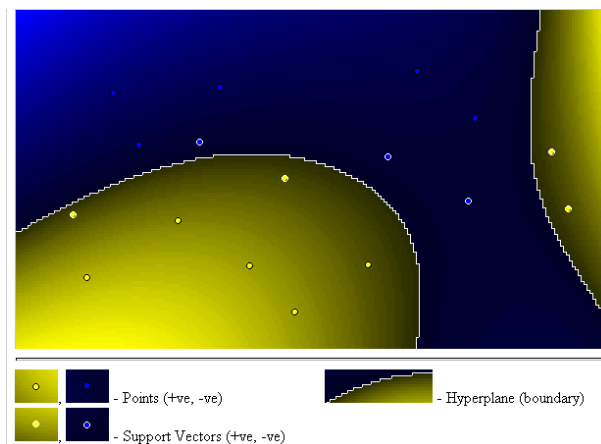


Figure 2.3: SVM decision boundary

SVM Applet provided by the Royal Holloway University of London, available at <http://www.clrc.rhul.ac.uk/research/svmoverview.htm>

Tong & Chang (2001) have shown the value of active SVM learning for image retrieval appli-

cations in comparative evaluations to traditional query refinement methods. Their SVM approach achieved around 90% precision after 3-5 rounds of feedback, in comparison to a substantially lower performance of between 65-80% of query shifting. All results that were presented, assumed 20 labelled images per feedback round. In a real-world situation this number is still quite high. It is possibly too demanding or tedious for the user to label 20 images in each of five iterations.

Variants of neural networks (Multi-Layer Perceptrons for this matter) are also used for classification by finding a hyperplane that best separates the feature space. However, the approach adopted by Wood et al. (1998) employs Radial Basis Function (RBF) neural networks, which describe the shapes of data clusters. A two-stage learning algorithm is proposed. The on-line training results in an initial clustering of image regions according to feedback provided by the user. According to the feedback from the first round, the image data is clustered, and consequently image regions are retrieved based on the minimum distance to the cluster representative. Re-clustering is performed to iteratively refine the search results until the user is satisfied. The neural network classifier comes into play in an off-line training phase as a follow up to the on-line learning. The result is a confidence of class membership for each image region given the training examples obtained during the user's interaction with the system. In this way, a class library can be built up over time, greatly facilitating future searches. Since the off-line phase is not restricted by the real-time requirement, the classifier can be trained much more accurately. A drawback of neural networks, worth mentioning at this point, is that they are known to be prone to overfit the data. SVMs in contrast, are better able to generalise.

**Statistical Procedures** A discriminant treatment can be achieved through different mechanisms than just the traditional machine learning techniques. In general, statistical procedures like the boosting approach (Tieu & Viola 2000) described below can be employed with the identical goal of discriminating between relevant and non-relevant classes.

Tieu & Viola (2000) base their approach on the observation that there are usually only a small number of features in a given set of example images that can successfully discriminate these images from the rest of the collection. If, however, like in most CBIR systems, only a small selection of features (like simple histograms or moments for colour, texture and shape) are implemented, it is unlikely that the one feature that is truly discriminative for any given example set is found. To remedy this shortcoming, their technique uses over 45,000 "highly selective features". On this extremely large feature set a *boosting* technique is employed to determine a classifier that greedily selects a small number of features (on average between 20-50 features) for which the positive examples are most distinct from the negative examples. However, in their approach random sampling of negative training samples was used to overcome the *small sample issue*, thereby taking the risk of treating relevant images as negative training samples.

The images that are presented to the user after each learning stage are: (a) a set of images that could be classified as positive; (b) a small set of randomly selected negative images; and (c) a set of images classified as negative, but which are close to the decision boundary. By displaying images from these three sets the system can update the decision boundary in three ways: (a) the user discards images in the first set, thus indicating false positives; (b) false negatives can be identified from the second set; and (c) the decision boundary can be refined by updating the relevance

judgements of images from the third set. This choice constitutes a reasonable compensation of the discrepancy between the *greedy* and *cooperative* model of selecting new images for display. The former always chooses the most-positive examples to satisfy the user, whereas the latter makes the selection in order to achieve the highest information gain by being labelled by the user. The images in the cooperative case are likely to be close to the current decision boundary, thus by receiving new labels the decision boundary can be adapted quickly (cf “active learning” in the SVM setting above).

### 2.3.4 Discussion

Both the geometric and the statistic methods have developed from initial heuristic procedures towards well-founded and optimised frameworks (Zhou & Huang 2003). Each of the introduced methods has benefits and drawbacks. Performance criteria to consider include the number of examples needed for convergence, the ability of progressive learning of new example, computational complexity and classification accuracy. Usually there is a trade-off to be made between conflicting criteria. So, while the initial query refinement approaches exhibit limited accuracy, their computational complexity is much lower than the later optimised approaches, for example.

Nevertheless, it is difficult, if not impossible, to make direct comparisons of the performance of different approaches because of the diversity of assumptions inherent in the different approaches, eg :

- “*What is the user looking for?*”

Most techniques assume either target or category search. There is little support for open-ended browsing.

- “*What to feedback?*”

Approaches range from positive feedback only, binary feedback of positive and negative examples, degrees of (ir)relevance, or comparative relevance.

- “*What to learn and how?*”

Some methods attempt to learn a new query and/or a (linear) transformation in the feature space, others treat it as a learning or classification problem.

The comparison of different techniques is further complicated by the lack of suitable testbeds and standardised performance criteria in CBIR. This problem is discussed in Section 2.5.

Yet there are some common issues with many relevance techniques. The premise inherent in the geometric approach, as well as discriminant methods, is that images that are relevant to one request form a cluster in the feature space. Furthermore, the group of relevant images must also be sufficiently distant to irrelevant images. Thus, the feature space has to be geometrically divisible into relevant and irrelevant parts. There is a crucial flaw in this hypothesis. Images, whose points are close to each other in feature space, are so because of their *visual* similarity. This is because the features used for representation can capture little more than low-level visual contents. The user providing the feedback, however, is more likely to judge the relevance of images on the basis of semantic concepts or meaning. So, images that are close in feature space are not necessarily considered relevant together. This reflects yet another instance where the semantic gap has detrimental effects on image retrieval.

The semantic gap has been acknowledged and also the fact that current techniques can hardly do any better than capturing the low-level visual similarity of images. Nevertheless, there is another vital argument against the hypothesis that relevant images can be clustered in feature space. It lies in the nature of the user's information need. The intention of the geometric approach is to find the *ideal* query point reflecting the user's information need. Hence, it explicitly assumes that such an ideal query exists. At the same time, discriminant methods presume that the relevant and irrelevant labels of images are static. However, the user's information need is known to be time-variant. A user might start off not knowing exactly what to look for. In the course of a search session, while being exposed to suggestions by the system, the need might change—often several times. So, while the system is updating its parameters restricting the search space to only a small area, the possibility of exploring other relevant images that do not fall within the region of visually similar images is taken away from the user. In fact, the majority of relevance feedback techniques, including geometric and statistical approaches, fail to address dynamic information needs.

### 2.3.5 Summary

As we have seen so far, learning techniques can be used to: compute a semantic representation of the images; and improve retrieval results interactively. Learning therefore addresses the semantic gap and query formulation problem. Yet these issues are far from solved. Often, the relevance feedback expected from the user is too restrictive: an image is expected to be either relevant or not. As we will see in Chapter 4, the approach in this thesis proposes a more open interpretation to relevance in the form of groupings. In this scenario, a user can decide on the nature of the groups (reflecting relevance classes or semantic concepts) and populate these groups. The user can concentrate on an organisation process rather than query formulation. The dynamic nature of information needs, also a factor ignored by the majority of relevance feedback approaches, is better addressed by the groupings too. The creation of a new group or switching between existing groups is assumed as an information need change, without the user having to make this fact explicit. In any event, in order to receive relevance feedback in one form or another, a suitable interface needs to be provided to the user. This is the topic of the following section.

## 2.4 The Interface

The interface is the mediator between the user and the computer. From the perspective of the user, it is the entry point to the system. A properly designed interface assists the user with meaningful and intuitive ways of communicating their information need to the system and displays results in ways that stimulate the user and enhance performance.

Early image retrieval systems were “*computer-centric*”. The system and its algorithms were considered the most important parts, and the user's role was simply to deliver the queries to the system (eg *QBIC*, Flickner et al. 1995). However, it has recently been acknowledged that information retrieval is an inherently *interactive* process (Ruthven 2000). In addition the previous section helped to highlight the important user feedback to improve image retrieval algorithms.

### 2.4.1 Interaction with Image Search Systems

The incentive to interact with an information retrieval system arises from a knowledge gap, which the user is determined to fill (Belkin et al. 1982). As a consequence, the system is used in order to seek for information. Information seeking is a very broad term and it can manifest itself in various ways.

The information seeking behaviour is greatly influenced by the nature of the information need. One user could for example search for “happy images” that remind her or him of holidays in the sun. Others may be looking for inspirations of images to illustrate a Web page. Yet again others want to find exactly one image that they had seen before. These examples illustrate just a few instances of distinct types of information needs of a user requiring the service of an image retrieval system.

The goal the user has in mind when interacting with a retrieval system is determined by their information need. However, the goal is rarely precise and might not even be known at the beginning of the search. The fulfillment of the information need is typically a very time consuming activity. Its success depends largely on the interaction strategy with the system.

Hence, the interaction strategy must take into account a variety of types of search tasks and other situational factors, such as the user’s knowledge about the domain and system or their personal preferences. For a successful retrieval system it is thus not only necessary to provide adequate document representations and matching functions, but more importantly to support the user in the process. It has been argued that the success of retrieval systems depends mainly on the user’s perception or mental model of the system (eg, Ruthven 2000). Thus the design of the system must be aimed at creating an environment that allows a better understanding of what the system does and how decisions are made. The two aspects of *feedback* on the system’s side over what the system is doing and *control* on the user’s side over their intended actions, should ultimately drive the interface design.

#### Interaction Metaphors

The diversity of search types can be supported by three basic interaction metaphors: *search*, *browsing* and *navigation*. The suitability of each is determined by the nature of the information need. Precise information needs are fulfilled by a set of documents possessing the desired characteristics, which can be located and accessed by direct searching methods. If the information need, on the other hand, is vague, browsing can allow for serendipitous discovery by providing a structural view of the collection helping the user to explore the database. Navigation is accessing relevant information within a logical unit based on a spatial metaphor. The considered unit can be an entire collection or a single document.

Since the type of information need is not fixed for a system, it needs the ability to adapt to changing requirements. Consequently, a retrieval system should combine searching, browsing and navigation, and create multipurpose interactive spaces.

### Visualisation

The representation of information has traditionally been confined to those suitable for retrieval. Thus, in image retrieval systems the interface was concentrated on query components facilitating the ability to specify the image features used for retrieval. However, in order to support the way information is used and managed, the *interaction* needs to be bestowed a representation. Ruthven (2000) suggests, among others, to represent term usage information in the text retrieval domain. Domain-independent interaction representations include displaying how the system's view of the search is changing over time and displaying relationships between documents.

In image retrieval systems, the major innovation to this end has been to replace the traditional linear result display, ranked by similarity to the query, with two- or three-dimensional maps of the returned images. These multidimensional displays aim at revealing relationships between images by visualising mutual similarities between any two images. The axes either represent feature dimensions directly, such as colour or textures, or are a result of dimension reduction methods, such as Principal Component Analysis (PCA), mapping the cardinality of the feature space down to the two or three most discriminative dimensions.

The goal of these visualisation techniques is to show the images in their surroundings or context. By depicting relationships between images in a global view, the user can form a more accurate mental model of the database and support navigation within it. A user study conducted by Rodden et al. (2001) has pointed to the benefits of a display organised by similarity for image browsing.

This visualisation technique has been used in *El Niño* (Santini & Jain 2000) to communicate computed distances between images by the system and actual perceived distances by the user in a 2D space (see Section 2.3.2). Rubner (1999), Chen et al. (2000) and Pečenović et al. (2000) are also among those people that have argued for a more meaningful display, which has consequently been incorporated in their systems. Because of its comprehensive approach towards both browsing as well as retrieval, Pečenović et al.'s *CIRCUS* system has been chosen as a representative of browsing systems and will be discussed in more detail below.

#### 2.4.2 Existing Interactive CBIR Interfaces

This section serves as an outline of the development of CBIR systems on the basis of their interface design. It will start with the early computer-centric systems, followed by a recount of the "typical" relevance feedback system, and finally selective examples of modern directions in interface design are presented. The first two systems were chosen because of their renowned status in the field: *QBIC* is inarguably the most influential CBIR system<sup>2-7</sup>, while the authors of *MARS* are pioneers of the relevance feedback approach for images. The three modern interfaces—*ImageGrouper*, *CIRCUS* and *AETOS*—were chosen because they each tackle the image retrieval problem from a novel and interesting angle: *ImageGrouper* adds a workspace to make relevance feedback more transparent, *CIRCUS* provides very neat overviews for browsing based on 2D projections and *AETOS* uses a novel graph-based representation for interactive nearest-neighbour browsing.

<sup>2-7</sup>At the time of writing, the article (Flickner et al. 1995) has a citation count of almost 2000 in [scholar.google.com](http://scholar.google.com).



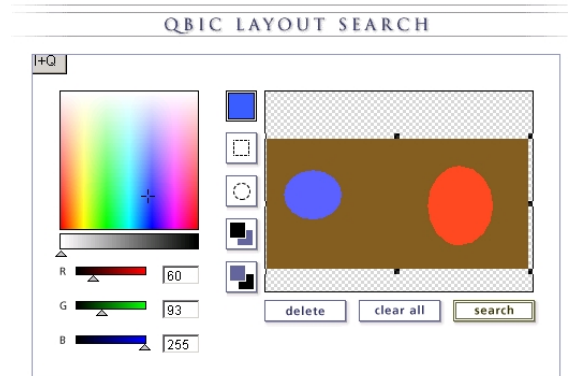


Figure 2.4: The *QBIC* querying component

### *QBIC*

The *QBIC* system developed by IBM is one of the earliest image retrieval systems with CBIR query facilities (Flickner et al. 1995). It will be used in the following as a representative for a number of other interfaces that have followed its example.

*QBIC* supports the retrieval of images based on a number of primitive image features, including colour, texture and shape. The query component is the most important aspect of the interface. In Figure 2.4 the query interface for “query-by-spatial-layout” is displayed<sup>2-8</sup>. It allows the user to specify the rough shape and colour of objects, which can be placed on the query canvas according to the spatial layout the objects in the retrieved image should convey.

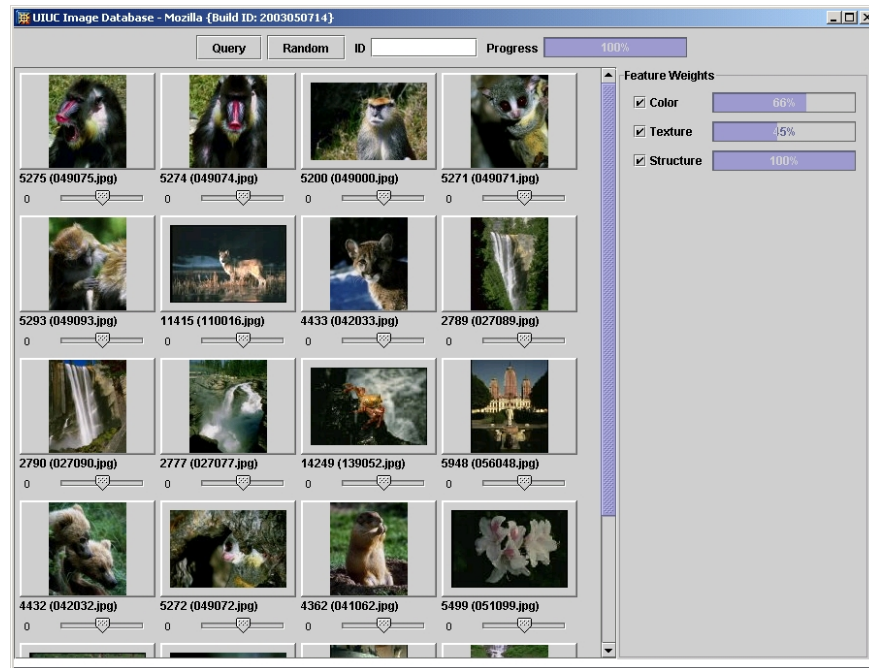
The query specified in this manner is automatically translated into the primitive features used for indexing the database images. After issuing the query to the system, the resulting images are displayed in a grid sorted by decreasing similarity scores to the query features.

The interface is hardly interactive in the modern sense. The only interaction taking place is the initial input of the query, to which the system reacts with a result set of images. If the user is not satisfied with the results, the only available option is to go back to the query and refine it manually.

For such a system to be successful, it is vital that the user can express their information need in the form of primitive attributes. It is a demanding task, in itself, for the user to formulate their information need in terms of the low-level image representation. Also, simply providing a query composition interface widget does not allow any support in refining the initial query or explanations by the system about which features are most expressive for a particular query.

In summary, this interface requires intuitive and meaningful query composition facilities, and relies on the user’s ability to map from the high-level concepts they have in mind when querying the low-level visual attributes the system understands and uses for retrieval. It hardly assists the user in their task and does not learn from user interaction.

<sup>2-8</sup>Taken from the Hermitage Web site <http://www.hermitagemuseum.org/>, which uses the *QBIC* engine for searching archives of world-famous art.

Figure 2.5: The *MARS* interface

### *MARS*

To alleviate the query formulation problem, recent systems have emphasised an interactive result refinement strategy made possible through relevance feedback. *MARS* (Porkaew et al. 1999) is used here as an example to illustrate the typical interface for relevance feedback (see Figure 2.5)<sup>2-9</sup>.

To initiate the search, these systems usually implement the “*Query-by-example*” paradigm. There, one user-supplied image, from which the query features are extracted, is used to bootstrap the search. After the first iteration, the user is asked to specify the relevance of images in the result set. In *MARS*, this feedback can be given by changing the value of a slider of any image indicating the degree of relevance when pushed to one side, or irrelevance when pushed to the other. The system responds with a new result set, which is improved based on the experience gained from the relevance feedback (see Section 2.3). This process is repeated until the user is satisfied with the results.

Hence, a two way interaction takes place between the system and the user, in which the user responds to the resulting set of images returned by the system, and the system responds to the relevance feedback given by the user.

The requirements for the interface are minimal in this case. Apart from letting the user choose an initial query image (or alternatively start with a random set of images), the user must be able to associate some relevance values with the images in the result set.

Nevertheless, the system does not provide sufficient information to assist the user in making vital decisions. For instance, the system does not give any indication of how many images to select for feedback, which images to select, what kind of effect feedback on a selected image has on the new results, etc. As a result, the user is forced to make decisions without having enough

<sup>2-9</sup>Online demo available at <http://www.ifp.uiuc.edu/~nakazato/CBIR/>

knowledge about the effects of their actions. Since the actions are usually irreversible this can have detrimental effects on the perceived performance of the system.

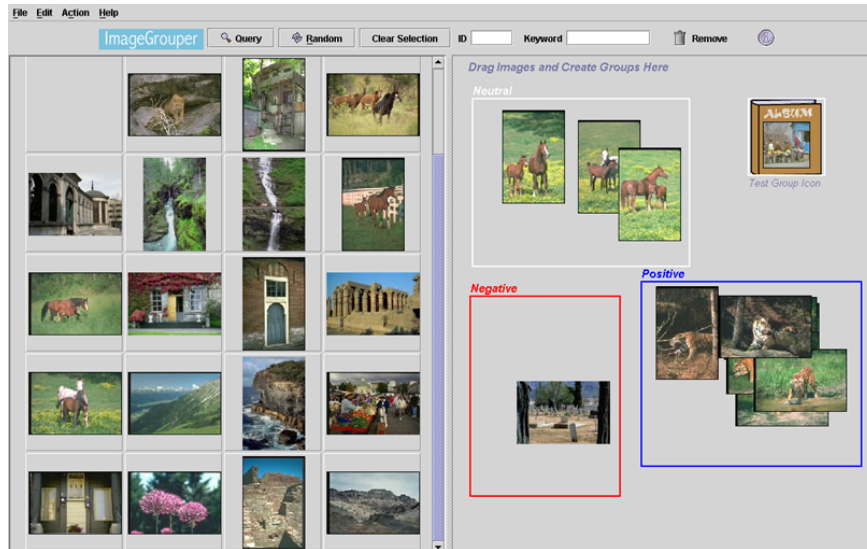


Figure 2.6: The *ImageGrouper* interface

### *ImageGrouper*

In order to address the problem of transparency in the traditional relevance feedback interface, the *ImageGrouper* system was developed by Nakazato et al. (2002). The major emphasis lies in group-based search, and this system combines the tasks of searching, annotating and organising digital images in groups.

Image retrieval in this interface (see Figure 2.6)<sup>2-10</sup> follows the *trial-and-error* approach as opposed to the traditional *incremental* search of most CBIR systems that incorporate relevance feedback. It is supported by separating the feedback display, in the form of a workspace, from the results display. The workspace serves as the organisation and storage area. Images can easily be dragged from the results panel onto the workspace, and consequently be organised into groups. Groups are created by drawing a rectangle around a cluster of images. For relevance feedback, the groups can be classified as positive, negative or neutral groups. The introduction of a separate workspace ensures that all images used for relevance feedback, and their organisation, are always visible. By dragging images around the workspace, ie in and out of groups, and selecting different groups as negative or positive examples, a *trial-and-error* search is easily supported. This relies on lightweight operations of creating groups (draw rectangle), assigning images to groups (drag'n'drop) and labelling the groups (simple popup menu). The organisation into groups is further enhanced by allowing subgroups inside a group and overlapping groups.

In addition to the image retrieval and organisation tasks, the interface supports a straightforward annotation operation. This annotation is naturally integrated in the search process. It is achieved by allowing the user to assign a number of (user-defined) keywords to a group of images, and thus follows the overall group-based paradigm. If groups overlap, images in the intersection

<sup>2-10</sup>Online demo available at <http://www.ifp.uiuc.edu/~nakazato/grouper/>

are annotated by the union of the keywords of each group. In summary, this interface integrates the three concepts:

1. Query-by-Groups
2. Annotation-by-Groups
3. Organisation-by-Groups

The *trial-and-error* approach ensures that actions are reversible, which is necessary due to the inferior capabilities of current CBIR technology in matching human similarity judgements. Nonetheless, *ImageGroupier* fails to deal with varying types of information need. The system learns to improve its retrieval results in order to satisfy the current information need. Although groups can be saved for later use, the contextual information they convey is not used to adapt the system in the long run.

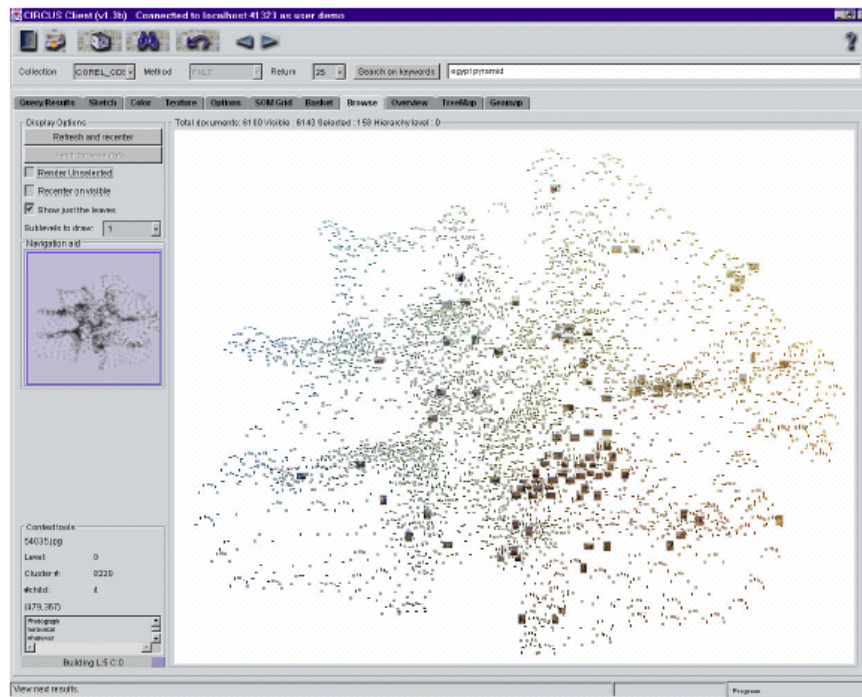


Figure 2.7: The *CIRCUS* interface

### *CIRCUS*

As opposed to learning from relevance feedback to free the user from having to specify an exact query, the *CIRCUS* system (Figure 2.7)<sup>2-11</sup> attempts to support the user by seamless combinations of query and browse-based views (Pečenović et al. 2000). Querying is catered for by a rich combination of possible query paradigms. The user can choose between querying by:

- *Properties*: metadata such as file name, file format, image dimensions, creation dates, etc.;

<sup>2-11</sup>An overview of the system including a manual is available at <http://lcavwww.epfl.ch/software/CIRCUS/>. However, the link to the actual demo has been disabled recently.

- *Example*: in the form of a positive image set for images similar to the target, and a negative image set;
- *Colour*: specify proportions of colours in the target image
- *Sketch*: create a collage query sketching objects in the target image;
- *Texture*: specify generic properties of textures (randomness, directionality, etc.) or choose templates from texture thesaurus;
- *Annotation*: specify keywords describing the contents of the target image;

or indeed any composite of the above. The idea of providing this large selection of tools is to assist the user in the construction of queries.

These tools for direct searching are combined with dynamic and interactive visualisations of the data to support browsing and navigation. The browsing mode permits an overview of the entire collection in a structured fashion. By panning and zooming the user can move to interesting regions and view the images in greater detail. An example pan-and-zoom sequence is depicted in Figure 2.8. Any of the images located in this way can be used as query-by-example images to initiate a search. The search results will be highlighted in the browsing mode, thus providing the user with the context of the returned images and a clue for navigating to the desired areas in the collection.

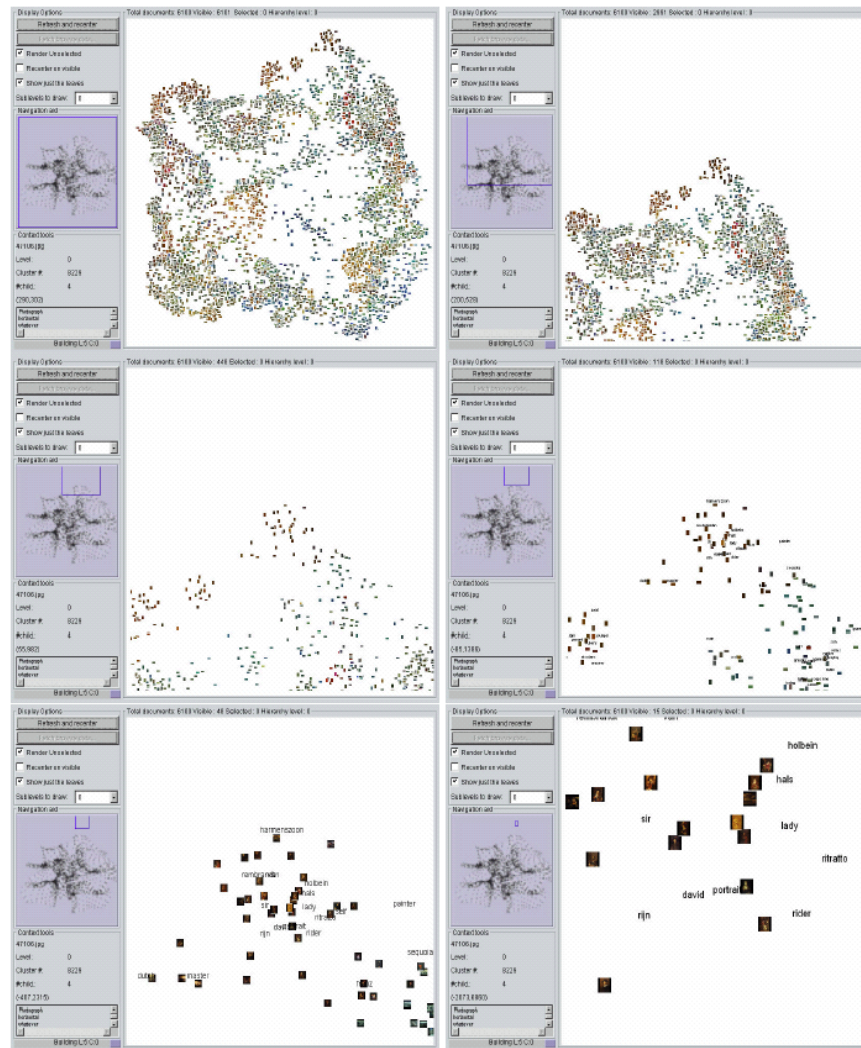
The display is constructed by creating a hierarchical clustering of the collection, projected into a 2D browsable space. Clusters are displayed by suitable cluster representatives, and a hierarchical view is generated by varying the size of images at various levels. The displaying strategy follows the overview and detail paradigm, which advocates the simultaneous display of detailed information while maintaining an overview to place the details into context.

The integration of browsing and searching mode is achieved by coupling the browsing display with the query results and allowing direct query specifications in the browsing mode. This immersive approach results in effective interaction sequences, in which the user can immediately perceive implications of their actions.

A completely automatic organisation of images computed without human feedback can never be perfect. It is not clear, however, if and how the organisation in *CIRCUS* can reflect changes in the similarity metric due to relevance feedback information. This could be achieved by clusters that change dynamically. So, for instance, if the user conducts a category search, the user would benefit from updating the pre-computed display so that the images belonging to a certain category identified by the user will be moved closer together in the display. In this way, the system learns from the user and the user sees the results of the interactions.

### **AETOS**

Heesch & Rüger (2004a) have developed a novel interaction technique referred to as “ $NN_k$  networks”. In this network each image is represented as a node in a graph and connections between them are based on the nearest neighbour relationship. An edge is created if a feature weighting combination exists that would yield one image as nearest neighbour to another. Based on this technique the authors have developed interfaces for retrieval and browsing of image collections

Figure 2.8: Zooming and panning in *CIRCUS*

(Heesch & Ruger 2004b). In the case of browsing, the user is presented with a number of hub images in the network as a starting point, that is those images with the highest number of links referring to other images (out-degree). Once one image is selected, the system displays its pre-computed nearest neighbour images. The user can select one of these images, which will result in the system displaying the nearest neighbour of the newly selected image and so on. The  $NN_k$  idea can also be used for retrieval. From an initial query image, the system first returns the set of precomputed nearest neighbours as in the browsing scenario. The user can then select relevant images, which causes the system to retrieve new images with the particular weight set associated with the selected images. The returned images are arranged in an Archimedean spiral that places the most similar images at the centre while the images further away are scaled down progressively. A demo of the *AETOS* system integrating both retrieval and browsing techniques is available at <http://mmis.doc.ic.ac.uk/demos/aetos.html>.

### 2.4.3 Summary

Interfaces that allow users to manage, browse and query large image collections have come a long way. The semantic problem, that is the problem of automatically bestowing meaning to an image based on their visual content, has led to the realisation that a tighter integration between user and system is necessary. The interactive systems that have recently been developed is testimony of this trend. As a result, browsing has taken a prominent role as in *CIRCUS* and *AETOS*. Directed access via querying is mostly integrated in order to fulfill more specific information needs more quickly.

Both *CIRCUS* and *AETOS* let the user browse a static organisation of the collection. The approach of this thesis allows the user to specify their own organisation, which is built up over time. A recommendation system assists the user in this process. Moreover, the system is endowed with the ability to learn semantic concepts from the user's organisation.

## 2.5 Evaluation

Evaluation of both retrieval algorithms and search interfaces is crucially important in order to find out which techniques or designs work and which do not. There are system-centric as well as user-centric evaluation models, both with their advantages and disadvantages. In this section, both methodologies are introduced.

### 2.5.1 System Evaluation

Evaluation has a long tradition in IR systems, largely due to the success of the Text REtrieval Conference (TREC) (TREC n.d.) initiative. TREC creates test collections and provides queries (topics) together with their relevance assessments. The traditional performance measures are *precision* and *recall* (van Rijsbergen 1979) defined as:

$$Precision = \frac{\# \text{ relevant images retrieved}}{\# \text{ retrieved images}} \quad (2.4)$$

$$Recall = \frac{\# \text{ relevant images retrieved}}{\# \text{ relevant images in the database}} \quad (2.5)$$

The availability of such testbeds is the prerequisite for systematic evaluation of retrieval systems and comparison between systems. The advantage of this evaluation model is that it provides a controlled environment in which it is easy to isolate the algorithmic issues from unwanted external factors, such as interface design or searcher behaviour. In addition, searcher interaction can be simulated in order to evaluate relevance feedback algorithms. In the most prevalent models of searcher interaction, the searcher is assumed to assess a selection of, or all of, the top  $k$  ranked documents, where  $k$  is usually small. White (2004) has proposed a much more complex simulation-based evaluation methodology to assess the performance of implicit feedback models.

### Testbeds for Image Retrieval

The most crucial obstacle for comparing the performance of image retrieval systems is the lack of a suitable testbed, including a test collection with ground-truth and standardised performance criteria. The Corel collection (COREL n.d.) has become the de-facto standard, because it has been categorised by domain experts, which is often used as ground-truth information. However, it is copyright protected and therefore not freely available to everyone. Hence, for a long time, each research team has been left to choose collections and performance criteria that suit their needs (Müller et al. 2002).

Only recently have people started making efforts towards creating a standardised testbed for image retrieval. The original initiative was proposed in the Benchathlon forum (Benchathlon n.d.). Markkula et al. (2001) have created their own testbed for journalists illustrating newspaper articles. Yet none of the purely image-based collections has had any impact on the image retrieval research community so far. Instead, a major push has come from the adoption of multimedia retrieval in TREC-like evaluation formats. On the one hand, the CLEF Cross Language Image Retrieval Track (ImageCLEF n.d.) was established in 2003 with the aim of evaluating image retrieval from multilingual document collections. Its focus is text-based retrieval techniques and automatic image annotation. The main content-based evaluation campaign, however, is TRECVID which is concerned with video data. It started as a video track in the TREC evaluation forum (TREC n.d.) and moved into its own forum in 2003 (TrecVid 2003). In 2005, the number of participants has reached a respectable 41 with even more participants expected in 2006.

In addition, as part of the EU MUSCLE Network of Excellence<sup>2-12</sup>, we can expect the CLIC testbed to be made publicly available soon (Moëllic et al. 2005). This testbed contains one million images, finally making it comparable in size to TREC text collections. The kernel of the testbed (ca. 15,000 images) is manually categorised into 16 major classes and subclasses. The remaining images are generated from the kernel through visual transformations. This suggests that the emphasis of this testbed lies on evaluating image analysis techniques. Its usefulness to evaluate image retrieval systems from a practical standpoint—“*does it do what people need?*” (Forsyth 2001, p. 242)—remains to be seen.

The lack of standardised testbeds and evaluation measures is a mirror of the lack of understanding of user needs—*what do people need?* Unlike text IR, practical CBIR applications are very rare. A major criticism of current evaluation practices has come from within the computer vision community itself. Forsyth states:

*“There is a substantial body of research on computer methods for managing collections of images and videos. There is little evidence that this research has had important impact on [...] any community yet. [...] In my opinion, there is little to be gained in measuring current solutions with reference collections, because these solutions differ so widely from user needs that the exercise becomes empty. The user studies literature is not well enough read by the image retrieval community. As a result, we tend to study somewhat artificial problems. A study of the user needs literature suggests that we will need to solve deep problems to produce useful solutions to image retrieval problems, but that there may be a need for a number of technologies that can be built in practice. I believe we should concentrate on these issues,*

<sup>2-12</sup><http://www.muscle-noe.org/>



*rather than on measuring the performance of current systems.”* (Forsyth 2001, p. 240)

In spite of all the new efforts towards creating bigger and more realistic testbeds in the interim (eg TRECVID), the overall situation has not changed substantially and Forsyth’s statement is still valid. Next, we will give a brief introduction to the evaluation practices in the information seeking community.

### 2.5.2 User Experiments

Evaluation of interactive systems is an even more difficult problem. The information seeking community, in particular, has been arguing for a long time that traditional IR evaluation techniques based on precision-recall measures are not suitable for evaluating adaptive systems (Ingwersen 1992, Borlund & Ingwersen 1997, Borlund 2003b, Jose et al. 1998, Dunlop 2000). Two of the most important reasons are the subjectivity of relevance judgements on the one hand, and the importance of usability for a system’s overall effectiveness, on the other. Usability can only be measured with the user in the loop, and will give valuable insights into what the users actually do rather than what we expect them to do. In addition, precision and recall can measure the effectiveness of the underlying algorithm relying on relevance judgements.

The concept of relevance is considered to be the common factor between the information seeking and information retrieval community (Ruthven 2005). However, relevance depends on a number of factors, such as topic, task and context, and further is subjective, multidimensional and dynamic (Borlund 2003a, Ruthven 2005). Moreover, it has been observed that interpretation of image content is particularly subjective and dependent on an individual’s experiences and view of life (eg, Squire & Pun 1998, Santini et al. 2001).

Borlund & Ingwersen (1997) argue that the actual information need should be used as the basis of judging relevance and hence performance. They propose *simulated work task situations*—“*a short ‘cover story’ that describes a situation that leads to an individual requiring to use an IR system*” (Borlund 2003b)—in order to trigger a simulated information need based on the user’s interpretation of the situation. These scenarios allow the user to develop a realistic information need, and hence searching behaviour, while providing control over the experiment. This method also takes into account that information needs are subject to change in the course of a search session. Different search systems and interfaces can thus be compared by experimental participants on the basis of situational relevance.

In this dissertation, we employ both simulated experiments to evaluate the algorithmic issues and user experiments to evaluate the system as a whole. The main user experiments are described in Chapter 6, where we have strived to create realistic scenarios by making use of simulated work task situations, providing realistic search tasks and inviting design professionals to participate. It has been important to us to understand and combine the cognitive and algorithmic issues in the design process, which is the prerequisite to “*strong research*” according to Ruthven: “*research that is motivated by an understanding of what cognitive processes require support during information seeking, and an understanding of how this support might be provided by an IR system*” (Ruthven 2005).

## 2.6 Other Issues

There are some other important components of CBIR systems and issues concerning their development that have not been considered in this review. In order to obtain an operable system, the CBIR architecture is dependent on efficient storage and access mechanisms. A discussion of system architecture in general and storage management in particular and further references can be found in (Böhm et al. 2001, Smeulders et al. 2000, Müller 2002, Rui et al. 1999, Lu 1999).

Human perception is a very important issue for feature extraction and similarity measures. Some features have failed to yield good performance, simply because they do not correspond to human perception. Ultimately, human perception—although admittedly hard to match—is the only reasonable model each CBIR system has to strive to imitate. A nice introduction to human physiology and human perception is provided in (Müller 2002), with interesting examples to set someone thinking about these issues.

Images are just one part of the story. Other media, such as video and sound, are gaining importance just as quickly. Moreover, applications are usually not confined to one single media. A survey of multimedia retrieval, in particular covering issues concerning video, can be found in (Aigrain et al. 1996). Dimitrova et al. (2002) additionally address concerns about possible application areas of multimedia retrieval. Finally the panelists of the MIR 2005 Panel on “Multimedia Information Retrieval: What is it, and why isn’t anyone using it?” discuss challenges, future directions and potential killer applications (Jaimes et al. 2005).

And still, there are uncountable other issues, such as network communication and Quality of Service, communication standards such as the MPEG-7<sup>2-13</sup>, data compression, etc, covered elsewhere (Eakins & Graham 1999, Lu 1999, Müller 2002). This list could be continued infinitely, which reflects the growth and expansion of the field.

## 2.7 Summary

This chapter provided an overview of current issues in CBIR research. Firstly, the representation of images for retrieval was covered, which can be seen as the backbone of every CBIR system. The development of image representations was traced and the main features used were explained. Secondly, learning techniques were discussed in the light of relevance feedback approaches to improve CBIR systems with experience gained from the user. Thirdly, user interface issues and visualisation techniques were compared. Without the development of such techniques, CBIR systems would most likely stay within the laboratory, which is the home to the majority of older research systems but from which emerging new systems are trying to break out. Last but not least, evaluation of retrieval techniques and interfaces is important to understand a system’s strengths and weaknesses.

In the following chapter we will report our observations from a user study of an adaptive image browser. This evaluation has helped to study typical user behaviour, their expectations of image retrieval systems and the problems they encounter during the interaction. We will then summarise and discuss these main unsolved issues in CBIR—the uncertainty of image meaning,

<sup>2-13</sup><http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

the query formulation problem and time-varying information needs. This study has led us to the conclusion that these deficiencies can only be tackled when taking all the major components of an information seeking environment into account: the image representation, the retrieval algorithm, the interface, the user and their work tasks. In this dissertation we formulate a novel 'holistic' approach, including interface, semantic image representation and retrieval algorithm, which is discussed in the remaining chapters.

---

### EXPERIENCES FROM A USER STUDY AND OPEN ISSUES

---

The previous chapter provided an overview of CBIR research, in particular tracing the development towards more intelligent systems. In this discourse some problems hindering current CBIR research have emerged. The most persistent questions, which we have identified in our research, are:

- “*What is the meaning of an image?*”  
If an image’s meaning or “aboutness” could be deciphered more easily, its contents would reveal itself for indexing and consequent retrieval.
- “*How can the user be assisted in communicating their information need?*”  
The query formulation problem has emanated as an information retrieval problem in general. The internal representation of documents is optimised for indexing efficiency and retrieval performance, but is, more often than not, rather alien to the user. Hence, there is the issue of teaching the user how to speak the language of the system, or, even better, teaching the system how to speak the language of its users.
- “*How can the time-varying nature of information needs be supported by the system?*”  
The initial idea of an image the user has in mind before starting a search session often deviates from the final results they will eventually be satisfied with. Whatever the reasons for this change, it shows that it is hard to derive an ideal query based on the initial query and consequent relevance feedback. Rather, we should attempt to trace the actions over the iterations in order to detect changes in the information need.

In order to find an even moderately satisfactory solution to any of these questions, it has become apparent that the *user* plays a very—if not *the* most—important role. After the initial euphoria of entirely automatic CBIR systems replacing the labour-intensive manual systems, it has been widely recognised that the user is a vital component in the chain. Without the user’s knowledge of the world and their superior visual system, CBIR system capabilities are limited. Moreover, user satisfaction greatly depends on subjective judgements of image contents and relevance. It is impossible to automatically accommodate the huge diversity of users. Yet the needs of individuals can be accommodated by learning their preferences.

In this chapter, we will present results of a user study, which was performed in order to investigate image searching and browsing from the user's point of view. The study compared an adaptive query learning approach based on the Ostensive Model (Campbell & van Rijsbergen 1996) to a traditional text-based interface. We will start by introducing the systems used in the study and our experimental methodology, before describing a detailed discussion of our findings. This study helped us identify the extent of the open issues mentioned above. Finally, we will generalise these open issues further, consider how they have been commonly addressed and why we think even more has to be done to overcome these problems. In summary, this chapter highlights some of the open problems in CBIR and provides motivation for new interface ideas, which will be introduced in the next chapter.

### 3.1 Results from a User Study

In order to investigate the problems a user faces in a typical CBIR system, we performed a user study (Urban et al. 2005, 2003). In this study, we compared a traditional manual query system to an ostensive browsing system. Our main goals were to determine the extent of the query formulation problem and the nature of information needs. Our hypothesis was that the design of a CBIR system interface has a significant impact on these two problems, and thus ultimately on its usability.

#### 3.1.1 Motivations of the Ostensive Approach

There are some issues that have been ignored in the large majority of proposed relevance feedback techniques. To start with, almost all learning techniques lack the ability to adjust the degree of relevance over time, with the notable exception of the probabilistic approach in (Vasconcelos & Lippman 2000). Often, it is not the case that the user's need is static or that there is an ideal query that fits the need. Therefore, it is a strong assumption to make that the document space can be divided in advance into relevant and non-relevant documents, and that after a number of iterations the system is able to approximate this division reasonably well.

In addition, existing approaches are reluctant to learn from *implicit* feedback. The user is always required to explicitly judge the relevance of the returned images. Even though the accuracy of explicit feedback in general is superior, a lot could yet be learnt from simply observing the user's actions (White 2004). This approach is less intrusive for the user, and can provide a different view on relevance. Sometimes, a user's real actions can tell a different story than the conscious interpretations given by the user.

Finally, browsing is typically not supported. The relevance feedback approaches usually assume category search or target search for simplicity of their algorithms. However, the user will greatly benefit from an environment in which both retrieval and browsing are combined. The possible nature of the tasks a user might want to perform is extremely diverse, and the user should not be restricted by the functionality of the system.

The *Ostensive Model*, introduced next, addresses all these issues. It derives a new interpretation of relevance in terms of information gained from implicit actions by the user varying over time.

### 3.1.2 Ostensive Relevance

The Ostensive Model (OM) of developing information needs was initially proposed by Campbell & van Rijsbergen (1996). It combines the two complementary approaches to information seeking: query-based and browse-based. It supports a query-less interface, in which the user's indication of interest in an object—by pointing at it—is interpreted as evidence for it being relevant to their current information need. Therefore it allows direct searching without the need to formally describe the information need. The query evolves automatically from a path of documents selected in the course of one search session.

By accepting that the user's need is dynamically changing during a search session, the OM adds a temporal dimension to the notion of relevance. A recently selected object is regarded more indicative of the current information need than a previously selected one. So, in this sense, the degree to which a document is considered relevant is continuously updated to reflect the changing context. Campbell's definition of Ostensive Relevance summarises the main points:

*“The **Ostensive Relevance** of an information object is the degree to which evidence from the object is representative/indicative of the current information need.”* (Campbell 2000b, p. 88)

The interaction with an Ostensive Browser follows an intuitive scheme. The user starts with one example document as the query, and as a result is presented with a new set of candidate documents (top ranking documents according to the similarity measure used). As a next step, the user—through selecting one of the returned documents—updates the query, which now consists of the original document and the selected document of the set of returned candidates. After a couple of iterations, the query is based on a path of documents. Since the whole path is visible to the users, they can jump back to a previous object along the path if they get the feeling that they are stuck or moving in the wrong direction. From there a new path can be explored, starting from the original object (the root) and the newly selected object. The resulting paths form a tree-like structure, originating from one root and branching at various objects (see Figure 3.1).

Similar to the Path Model described by Chalmers et al. (1998) for activity-centred information access, emphasis is set on the user's activity and the context, rather than the predefined internal representation of the data. A path represents the user's motion through information, and, taken as a whole, this is used to build up a representation of the instantaneous information need.

The weight of how much each document along the path contributes to the next query can be chosen with different objectives in mind. The weighting schemes are referred to as *ostensive profiles*, and reflect how relevance (or uncertainty) changes with age (age being interpreted as the order of selection or the position along the path). With the previously elaborated considerations in mind, the most plausible profile supports increasing uncertainty with age. The further back in time one document has been selected during the retrieval process, the more uncertainty is associated with it in actually reflecting the user's information need, or in other words, the less relevant it is considered for the query. This profile is also favoured by the original definition of Ostensive Relevance. For a comparative evaluation of different profiles and their interpretations please refer to (Campbell 2000a). The OM thus captures the developing information need of the user during a search process, and incorporates the uncertainty, which necessarily exists due to the imprecise

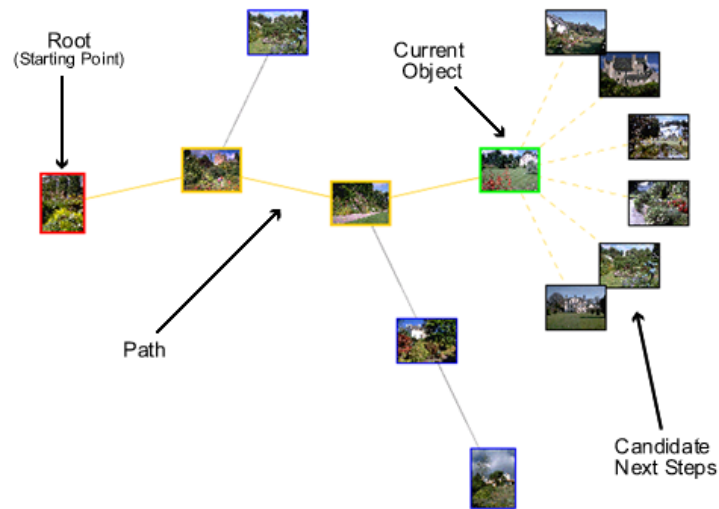


Figure 3.1: The ostensive path

nature of one's own information need and the difficulties of expressing it.

In its original conception, the OM was integrated with the Binary Probabilistic Model (BPM) of IR to create an operational retrieval model (Campbell 2000*b*). This was possible, since the images that were used in the implementation of the OM were represented by a set of index terms. However, if one takes into account content-based features to index images, the interpretation of the BPM becomes rather difficult. In the BPM, relevance scores are based on estimating or calculating the probabilities that, if the document is relevant (or non-relevant, respectively), a particular feature will be observed. In other words, the probability is assessed depending on whether some chosen feature is either present or absent. This interpretation was developed in the text retrieval domain, where a document can be represented by a set of index terms only. CBIR systems rely on more complex indexing features, in which it is hard to tell whether a particular feature can be observed. It is questionable whether or not content-based image features can be treated in a binary fashion, eg is it sensible to say the image contains the index term “green” if the colour histogram contains non-zero values for the bins referring to green? What makes matters even more complicated is the fact that most CBIR systems rely on multiple representations of image content. It becomes apparent that the interpretation of the binary probabilistic model in terms of content-based image features is rather inappropriate. For this reason, we introduce the use of adaptive queries within an operational retrieval system based on the OM.

### 3.1.3 The Systems

To test our ideas about adaptive query learning strategies, three prototype systems have been implemented and evaluated. In this section, we will describe these systems.

### Features & Similarities

The systems use two distinct features: *text annotations* and *visual features*. The text feature is extracted from the keyword annotations of the images, and the visual feature is based on colour histograms representing an image's global colour distribution represented in the HSV colour space.

An image is represented by two multi-dimensional feature vectors, which is a term vector (text feature) and a histogram bin vector (colour feature), respectively. The term vector is weighted by the  $\text{tf} \times \text{idf}$  (term frequency, inverse document frequency) weighting scheme (van Rijsbergen 1979). The similarity between the query,  $Q$ , and a candidate image,  $I$ , is calculated as the combined score of the two similarity values for each feature using the Dempster-Shafer combination (see Section 3.1.4). In the case of text similarity, the *cosine measure* (Salton & McGill 1983) is used:

$$\text{sim}(Q, I) = \frac{T_Q \cdot T_I}{|T_Q| |T_I|} = \frac{\sum_{i=1}^{l_T} T_Q[i] T_I[i]}{\sqrt{\sum_{i=1}^{l_T} T_Q[i]^2} \sqrt{\sum_{i=1}^{l_T} T_I[i]^2}}$$

where  $T_I$  and  $T_Q$  are the image and query term vectors, respectively,  $l_T$  the term vector's dimension (the number of terms in the index),  $T_I[i]$  the  $i$ -th entry in the vector ( $T_I[i] = \text{tf}_i \times \text{idf}_i$ , that is the number of times term  $i$  occurs in  $I$  multiplied by term  $i$ 's inverse document frequency measuring the number of times term  $i$  occurs in the whole collection) and  $|T_I|$  the image's term vector length (similar for  $T_Q$ ). Visual similarity is determined by *histogram intersection* (Swain & Ballard 1991):

$$\text{sim}(Q, I) = \frac{\sum_{i=1}^{l_H} \min(H_Q[i], H_I[i])}{\min(|H_Q|, |H_I|)}$$

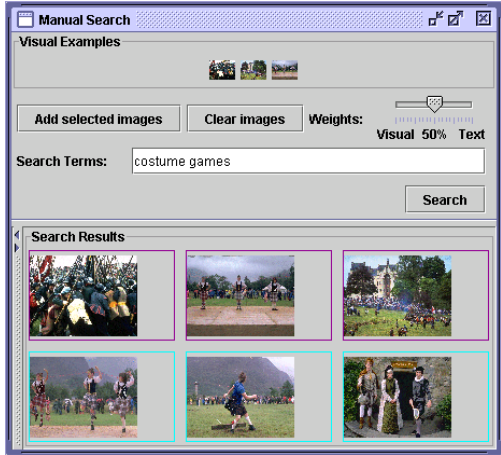
where  $H$  stands for a document's colour histogram vector, and  $l_H$  for the histogram vector dimension (256 in this case). Both similarity measures are widely used in combination with the chosen feature representation.

### The Interfaces

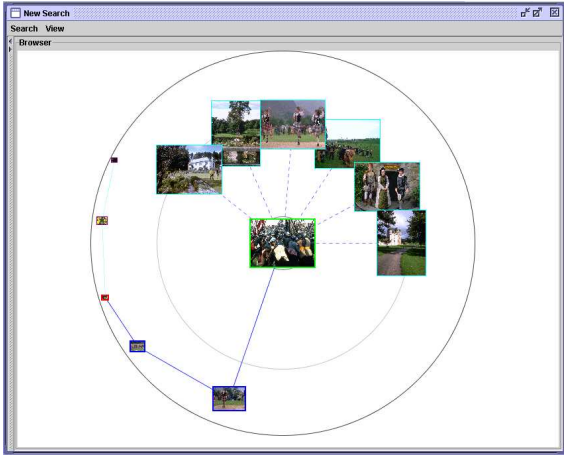
**The Ostensive Browsers** Two versions of the ostensive browsing approach have been implemented: one with a pure ostensive browsing scheme (Figure 3.2(b)) and the other allowing explicit feedback within ostensive browsing (Figure 3.2(c)). In both systems the user starts with an image in the browse panel (in Figure 3.2(c)-2). The initial image is obtained in a pre-keyword search from which the user is given the opportunity to choose an image to explore further. When an image is selected, the system returns a set of most similar images as candidate images. We chose to present six images as new candidates as a compromise between variety in candidates and space requirements<sup>3-1</sup>. Of those candidates, the user clicks on the most appropriate one. At this stage, the system computes a new set of similar images based on an adapted query and presents it to the user. As can be seen in Figures 3.2(b) & (c), this process creates a path of images, which is represented in the interface. At any point the user can go back to previously selected images on the path and also branch off, by selecting a different candidate. The complete search session can continue to iterate between keyword search followed by browsing sessions, as long as the user is

<sup>3-1</sup>As demonstrated in Appendix A, Figure A.2, a larger number of candidates could be achieved by incorporating a fish-eye distortion on the candidates.

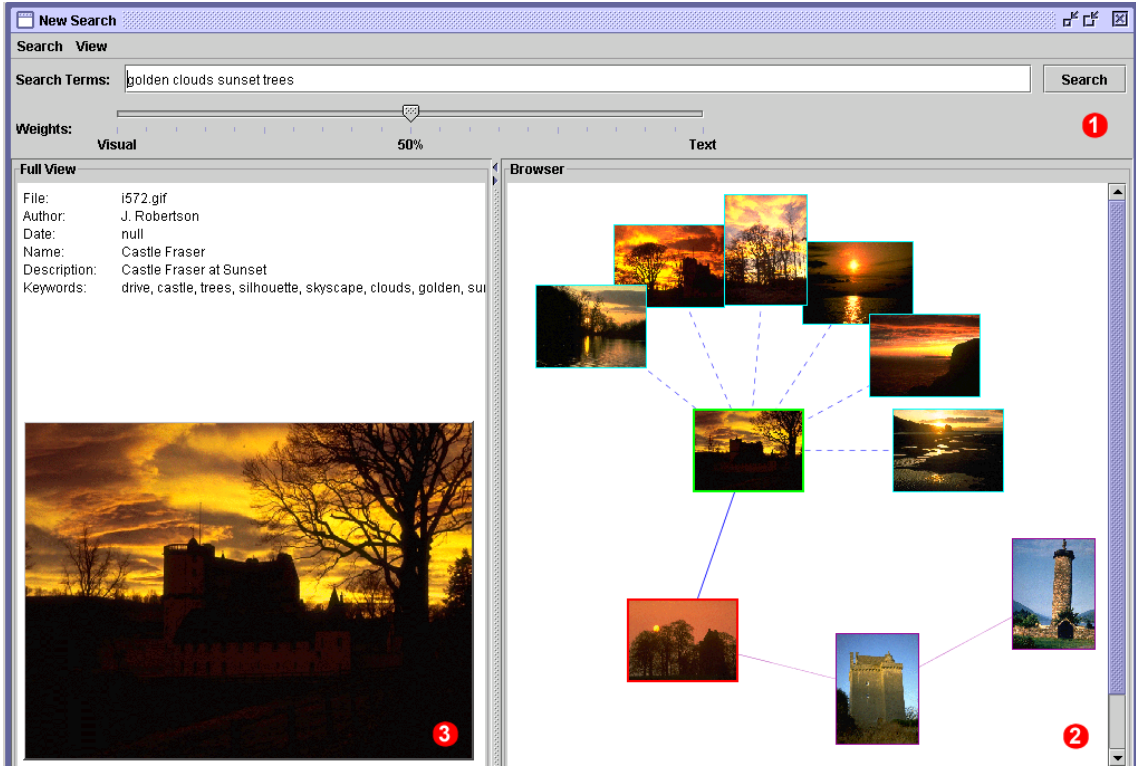




(a) MQS



(b) POB



(c) COB

Figure 3.2: The interfaces.

not satisfied with the retrieved images. Since the screen space is very limited the different paths are often overlapped resulting in a large degree of clutter, an alternative fish-eye view is provided (see Figure 3.2(b)). The user can switch between these views during the search.

To view the details of an image, there is the possibility of viewing a single selected image in full-size in a separate panel (in Figure 3.2(c)-3). It also contains some meta-data about the document, such as the photographer, title, date and description. In between the full-size view and the thumbnails, a quick view is shown as a popup when the user hovers the mouse over a thumbnail in the browse panel.

Both browsers (Figures 3.2(b-c)) attempt to adapt the query based on the user's implicit feedback, which will be described in Section 3.1.4. We provided two slightly different versions of the *Ostensive Browser* to allow for different levels of control. The *Pure Ostensive Browser* (POB) (Figure 3.2(b)) does not allow for any control of feature terms or weighting between the features. The system automatically adapts the query and also the feature weights. The learning of the feature weights is achieved in a similar fashion to (Porkaew et al. 1999), and they will be used as trust values in Dempster-Shafer's evidence combination (see Section 3.1.4) to combine the similarity scores.

In addition, the interface for the *Controlled Ostensive Browser* (COB) provides options for selecting the features and their associated weights (in Figure 3.2(c)-1). It displays the search terms the system used to obtain the currently shown candidates. The automatically selected terms (the strategy of the selection is described in Section 3.1.4) can be changed by the user and thus the current candidates are exchanged for the ones resulting from the updated query. Another aspect of control is the adjustment of the feature weights. The user can control the weights between the two features by means of a slider.

*How to start the search?* The problem with the ostensive search is the question of how to initiate the search, ie how to choose the first image that starts the path. As mentioned earlier, the current solution is to ask the user to formulate a keyword query, which returns a set of relevant images based on the textual feature. One of the returned images can then be chosen as the starting point. However, this is a rather ad-hoc approach, which again requires the user to formulate a query. One possible solution could be to pre-cluster the whole collection and let the user browse through these clusters to choose a starting point.

**The Manual Query System** As baseline system, we used the *Manual Query System* (MQS) (Figure 3.2(a)) resembling a 'traditional' image retrieval system, which returns a set of relevant images in response to a user-given query. A query can be formulated by a set of keywords and/or one or more images as 'visual examples' (QBE). The user can also set the weighting between the two features. If not satisfied with the results returned by the system, the user has to alter their query and so forth.

### 3.1.4 Query Adaptation Techniques

In the course of a search session in the *Ostensive Browser*, a user creates and moves along a path of images. During each iteration, the path changes and the query needs to be adapted accordingly. The selected documents are treated as evidence of the user's information need, with a changing

degree of uncertainty associated with each document: the older the evidence, the more uncertain we are that it is still indicative of the *current* information need. The degree of uncertainty is represented by an *ostensive relevance profile* (Campbell 2000a), used to weight the contribution of each path document. A separate query is constructed for each feature modality, text and visual, as a weighted combination of the documents' features.

**Text Query:** A new text query vector is created by updating the term weights with the ostensive relevance weights resulting from the ostensive profile. The query vector then consists of the union of the set of terms that appear in any of the documents in the path. A term's original weight is multiplied by the sum of ostensive relevance values for all documents in which the term appears:

$$w_t = \text{idf}_t \times \sum_{\substack{i=1 \\ t \in D_i}}^{l_p} (\text{ORel}_i \times \text{tf}_t(D_i)) \quad (3.1)$$

where  $w_t$  is the resulting weight of term  $t$  in the query vector,  $\text{idf}_t$  the term's idf value,  $l_p$  the length of the path,  $D_i$  the document at position  $i$  in the path (starting at the most recently selected object),  $\text{tf}_t(D_i)$  the term frequency of term  $t$  in document  $D_i$  and  $\text{ORel}_i$  the ostensive relevance weight at position  $i$ . The ostensive relevance weights are computed by the relevance profile function,  $\text{ORel}_i = \frac{1}{2^i}$ , and normalised to sum to 1:  $\sum_{i=1}^{l_p} \text{ORel}_i = 1$ .

Hence, the query terms are weighted with respect to the relevance profile and their corresponding idf values. A new query vector is computed based on the four highest ranking terms.

**Histogram Query:** There are different techniques for combining the query histogram from the individual histograms. A straight-forward approach in accordance with other query-point movement techniques (eg, Porkaew et al. 1999) is a linear combination of the constituent histograms and the ostensive relevance weights:

$$H_Q = \sum_{i=1}^{l_p} (\text{ORel}_i \times H_{D_i}) \quad (3.2)$$

The resulting query histogram  $H_Q$  is comprised of the bins computed as the weighted sum of the path documents' bins. It can be interpreted as the weighted 'centroid' of the corresponding histograms.

### Final Evidence

Two queries representing each feature are issued to the system, returning two result lists with different scores based on the respective similarity measure for each feature. For this reason, a means to combine the results to obtain one single ranked list of documents needs to be found. The *Dempster-Shafer Theory of Evidence Combination* provides a framework for this combination. The Dempster-Shafer mechanism has been widely used in the context of IR to combine information from multiple sources (Jose & Harper 1997, Jose 1998). The advantage of Dempster-Shafer's combination rule is that it integrates degrees of uncertainties or trust values for different sources.

For two features Dempster-Shafer's formula is given by:

$$m(\{d_i\}) = m_1(\{d_i\}) \times m_2(\{d_i\}) + m_1(\Theta) \times m_2(\{d_i\}) + m_1(\{d_i\}) \times m_2(\Theta) \quad (3.3)$$

and

$$m(\Theta) = m_1(\Theta) \times m_2(\Theta) \quad (3.4)$$

where  $m_k(\{d_i\})$  (for  $k = 1, 2$ ) can be interpreted as the probability that document  $d_i$  is relevant with respect to source  $k$ . The two sources in our case correspond to the similarity values computed from the text and colour feature respectively.  $\Theta$  denotes the global set of documents and  $m_k(\Theta)$  represents the uncertainty in those sources of evidence (also referred to as un-trust coefficients):

$$m_k(\Theta) = 1 - \text{trust}_k \quad (3.5)$$

where:

$$\text{trust}_k = \frac{\sum_{i=1}^p m_k(\{d_i\})}{\sum_{i=1}^p m_1(\{d_i\}) + \sum_{i=1}^p m_2(\{d_i\})} \quad (3.6)$$

$\text{trust}_k$  corresponds to the trust in a source of evidence  $k$ . This reflects the contribution of a given source in selecting that particular image. In our definition, it reflects the importance of each feature. The pieces of evidence, on which the trust in a particular source is based, are the calculated similarity values for the two features  $m_1(\{d_i\})$  and  $m_2(\{d_i\})$ . The resulting  $m(\{d_i\})$  is thus the combined belief for document  $d_i$ . Formulae 3.3 & 3.4 are a simplified version of Dempster-Shafer theory for IR purposes. Furthermore, they can easily be extended to accommodate more than two sources.

### 3.1.5 Experimental Methodology

As discussed in Section 2.5, evaluation in image retrieval systems is a difficult task. Traditional techniques based on precision-recall measures evaluating the retrieval effectiveness have often been criticised for treating the system as an independent entity (Ingwersen 1992, Jose et al. 1998, McDonald et al. 2001). The opponents of the traditional system-based evaluation have recognised the user's vital role in the design and evaluation of CBIR systems. Since image retrieval is an inherently interactive activity, a user-centred evaluation, in which 'real' people use the system in a 'real-world' setting, can provide invaluable insights into the system's overall performance that precision-recall measures can never anticipate. Important performance indicators ignored in traditional evaluations include user interface issues, task completion time and user satisfaction.

For this reason, we designed our evaluation to follow the guidelines of the evaluative framework for interactive, multimedia retrieval systems proposed by Jose et al. (1998). The main points in our evaluation following these guidelines are:

- Design professionals were asked to participate in the study in order to test the systems with real potential users.
- Context-situated tasks were created to place the participants in a 'real-life' usage scenario.

a)	<table border="1"><tr><td>A</td><td>B</td></tr><tr><td>B</td><td>A</td></tr></table>	A	B	B	A
A	B				
B	A				

b)	<table border="1"><tr><td>A</td><td>B</td><td>C</td></tr><tr><td>C</td><td>A</td><td>B</td></tr><tr><td>B</td><td>C</td><td>A</td></tr></table>	A	B	C	C	A	B	B	C	A
A	B	C								
C	A	B								
B	C	A								

c)	<table border="1"><tr><td>A</td><td>B</td><td>C</td><td>D</td></tr><tr><td>D</td><td>A</td><td>B</td><td>C</td></tr><tr><td>C</td><td>D</td><td>A</td><td>B</td></tr><tr><td>B</td><td>C</td><td>D</td><td>A</td></tr></table>	A	B	C	D	D	A	B	C	C	D	A	B	B	C	D	A
A	B	C	D														
D	A	B	C														
C	D	A	B														
B	C	D	A														

Figure 3.3: Example Latin squares

- A variety of qualitative measures indicative of user satisfaction (concerning the system, the tasks, the interface, etc.) was collected and analysed.
- Quantitative measures on task-completion time and images retrieved confirmed the qualitative measures of user satisfaction.

In our evaluative study, we adopted a randomised within-subjects design (Maxwell & Delaney 1990) in which 18 participants used three systems on three tasks. A within-subjects design is an experiment in which the same set of dependent variables is measured repeatedly on the same participant under different “treatments” (levels of independent variables). In our case, the treatments are system type and task type. The dependent variables are the responses from the questionnaires and other data collected from usage logs. The advantage of a within-subjects design is that effects due to the disposition of participants are minimised. This is beneficial because the variability in measurements is more likely due to differences among conditions than to behavioural differences between participants. There is one major weakness of this type of design: the learning effect, as participants’ behaviour in one condition will affect their behaviour in another.

To counterbalance the effect of learning, the order of the systems and tasks was rotated according to a Latin-square design (Maxwell & Delaney 1990). A Latin square is an  $n \times n$  table filled with  $n$  different conditions in such a way that each condition occurs exactly once in each row and exactly once in each column. Figure 3.3 a) shows an example Latin square for two conditions. In this case, participants are randomly assigned to groups of equal size: Group 1 is given condition A followed by condition B, while Group 2 is given condition B followed by condition A. Figures 3.3 b) and c) show the Latin squares if three or four conditions are tested.

The independent variable was system type; three sets of values of a variety of dependent variables indicative of acceptability or user satisfaction were to be determined through the administration of questionnaires. The searches were performed on a collection containing 800 photographs, created from the photographic archive of the National Trust for Scotland (Jose 1998).

### Tasks

In order to place our participants in a realistic work task scenario, we used simulated work task situation (cf Section 2.5.2). This scenario allows users to evolve their information needs in just the same dynamic manner as such needs might develop during a ‘real’ retrieval session (as part of their normal work tasks). Before starting the evaluation, the participants were presented with the work task scenario provided in Figure 3.4 stating they were responsible for designing leaflets for the Scottish Tourist Board. A draft design of the leaflet was also provided. For each system, they were given a different topic for the work task, each involving at least two searches (see Figure 3.5). The topics were chosen to be of very similar nature, in order to minimise bias in the performance across the systems.

## Systems

The Ostensive Browsers (Section 3.1.3) were evaluated against the ‘traditional’ image retrieval system MQS (Section 3.1.3), which supports manual query facilities. The Ostensive Browsers vary in the amount of control options granted to the user. The *Pure Ostensive Browser* (POB) relies only on automatic query adaptation as described in Section 3.1.4, whereas the *Controlled Ostensive Browser* (COB) additionally provides options for selecting the features and their associated weights.

## Hypothesis

Our experimental hypothesis is that the ostensive approach (reflected in both POB and COB) is generally more acceptable or satisfying to the user. It can be further distinguished in two sub-hypotheses:

1. Query adaptation coupled with an ostensive interface provides a better environment for CBIR; and
2. Providing an explicit control on the ostensive system results in better satisfaction on task completion.

## Participants

In order to obtain data as close to real-life usage as possible, we sought design professionals as participants. Our sample user population consisted of 18 post-graduate design students. We met each participant separately and followed the procedure outlined below:

- an introductory orientation session
- a pre-search questionnaire
- for each of the three systems in turn:
  - a training session on the system
  - a hand-out of written instructions for the task
  - a search session in which the user interacted with the system in pursuit of the task
  - a post-search questionnaire
- a final questionnaire

We did not impose a time limit on the individual search sessions. The complete experiment took between 1.5h and 2h, depending on the time a participant spent on searching.

### 3.1.6 Results Analysis

#### Pre-search Questionnaire

Through this questionnaire, information about the participants’ experience with computers and familiarity with using photographs was obtained. The participants were students at a post-graduate level in a design-related field (graphic design, photography or architecture). Their ages ranged

*Imagine you are a designer with responsibility for the design of leaflets on various subjects for the Scottish Tourist Board [...]. These leaflets [...] consisting of a body of text interspersed with up to 4–5 images selected on the basis of appropriateness to the use to which the leaflets are put.*

*Your task is to make a selection, from a large collection of images, of those that in your opinion would most effectively support the given topic. In order to perform this task, you have the opportunity to make use of a computerised image retrieval system, the operation of which will be demonstrated to you.*

Figure 3.4: Work task scenario and task description

**Task A:** *In this leaflet, we prefer **spring and autumn** photographs to depict the scenic splendour of Scottish countryside.*

**Task B:** *[...] **autumn and winter** photographs to depict the scenic splendour of Scottish countryside.*

**Task C:** *[...] photographs to depict the beauty of Scottish **seaside and coastal views**.*

Figure 3.5: Topics

between 23 and 30 years. The ratio between male and female participants was approximately 2:1. The responses revealed that all of the participants employed images extensively for their work, and that they were often required to retrieve images from large collections.

In summary, results from this questionnaire indicated that our participants could be assumed to have a good understanding of the design task we were to set them, but a more limited knowledge or experience of the search process. We could also safely assume that they had no prior knowledge of the experimental systems. The participants' responses thus confirmed that they were from the expected user population for the design task using an automated image retrieval system.

### Post-search Questionnaire

After completing a search session on one of the systems given a particular topic, the users were asked to complete a questionnaire about their search experience.

**Semantic Differentials** Each respondent was asked to describe various aspects of their experience of using each system, by scoring each system on the same set of 28 7-point semantic differentials. The differentials focused on five different aspects (see Table 3.3):

- three of these differentials focused on the *task* set (Part 1);
- six focused on the *search process* that the respondent had just carried out (Part 2);
- five focused on the set of images *retrieved* (Part 3);
- three focused on the user's perception of the *interaction* with the system (Part 4); and
- eleven focused on the *system* itself (Part 5).

The result was a set of 1512 scores on a scale of 1 to 7: 18 respondents scoring each of three systems on 28 differentials. On the questionnaire form, the arrangement of positive and negative descriptors was randomised.

Part I: Was the <i>task</i> ...?
(clear↔unclear), (simple↔complex), (familiar↔unfamiliar)
Part II: Was the <i>search process</i> ...?
(relaxing↔stressful), (interesting↔boring), (restful↔tiring), (easy↔difficult), (simple↔complex), (pleasant↔unpleasant)
Part III: Was the <i>retrieved set</i> ...?
(relevant↔irrelevant), (important↔unimportant), (useful↔useless), (appropriate↔inappropriate), (complete↔incomplete)
Part IV: Did you <i>feel</i> ...?
(in control↔lost), (comfortable↔uncomfortable), (confident↔unconfident)
Part V: Was the <i>system</i> ...?
(efficient↔inefficient), (satisfying↔frustrating), (reliable↔unreliable), (flexible↔rigid), (useful↔useless), (easy↔difficult), (novel↔standard), (fast↔slow), (simple↔complex), (stimulating↔dull), (effective↔ineffective)

Table 3.3: Semantic differentials

In our within-subject design, the sets of 18 scores on each differential for the three systems were compared. Our experimental hypothesis was that, in any individual case, the set of scores for both COB and POB was drawn from a population of lower (better) scores than that for MQS, and that COB scores were slightly lower than POB scores. Given the ordinal scale of the data, we had to use rank-based statistics. Since the data were not normally distributed, we calculated values of the non-parametric form of analysis of variance—the Friedman test (Maxwell & Delaney 1990). The null hypothesis in this case is: there is no difference in median ranks between groups on the criterion variable.

Overall, the Ostensive Browsers outperformed MQS, and usually COB’s scores were lower (better) than the scores for its pure counterpart. The means of all differentials for each part is depicted in Figure 3.6, which shows the trend that MQS scores are poorer than the scores for the other two systems, supporting our initial claim that query adaptation along with an ostensive interface provided a better environment for CBIR. The graph also shows quite clearly that POB’s scores are comparable with COB, except for the scores for Part 3. This part focused on the retrieved images, thus backing up our second sub-hypothesis, namely that providing an explicit control on the ostensive system resulted in better satisfaction with task completion.

For each differential, we tested the hypothesis that the scores for each system type were sampled from different populations. The results are collected in Table 3.4. The subset of differentials, which showed a significant level at  $p < 0.05$  (p-value after adjustment for ties) are: ‘restful’, ‘pleasant’; ‘comfortable’; ‘flexible’, ‘useful’, ‘novel’, ‘simple’, ‘stimulating’ and ‘effective’. Dunn’s multiple comparison post test (Maxwell & Delaney 1990) was performed to determine between which of the systems the difference occurred. For most differentials the significant difference occurred between MQS and COB. The most significant results are found when comparing the differentials for the system part (Part 5). Most notable is the variance in judging the system’s usefulness, and it should be pointed out that the advantage of the POB as being the simplest tool to use is reflected in the results, as well.



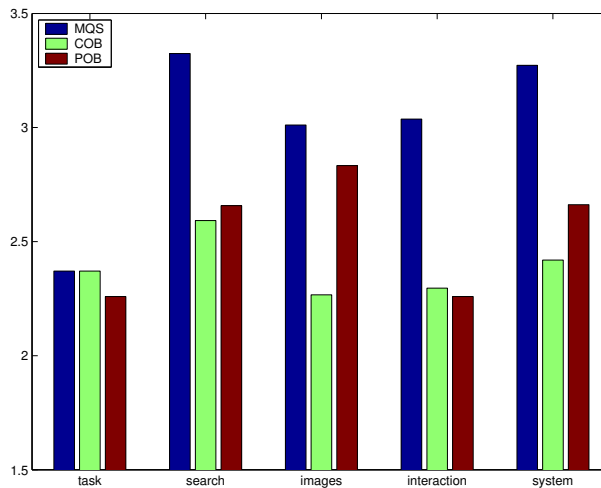


Figure 3.6: Semantic differential means per part (value range 1–7, lower = better)

Table 3.4: Means and significance test results (value range 1–7, lower = better)

Differential	MQS	POB	COB	p-value	Dunn's post test
<i>Part 2:</i>					
restful	3.9	3.1	2.8	0.008	MQS vs. COB < 0.05
pleasant	3.4	2.6	2.2	0.050	-
<i>Part 4:</i>					
comfortable	3.2	2.2	2.2	0.014	-
<i>Part 5:</i>					
flexible	3.7	3.4	2.4	0.007	MQS vs. COB < 0.05 POB vs. COB < 0.05
useful	3.4	2.6	1.9	0.001	MQS vs. COB < 0.01
novel	3.3	2.4	2.0	0.010	MQS vs. COB < 0.05
simple	2.9	2.2	2.9	0.030	-
stimulating	3.3	2.6	2.1	0.003	MQS vs. COB < 0.05
effective	3.2	2.4	2.1	0.007	MQS vs. COB < 0.05

There were no significant differences for Part 1 (concerning the tasks), neither across the systems nor across the tasks, which shows that the topics were well-balanced and should not have confounded the results significantly.

**Likert Scales** Each user was asked to indicate, by making a selection from a 7-point Likert scale, the degree to which they agreed or disagreed with each of seven statements about various aspects of the search process and their interaction with the system. There were four statements concerning the *user's information need*. They were phrased in such a way that responses would indicate the extent to which:

1. the user's initial information need was well-defined ("I had an idea of the kind of images that would satisfy my requirement before starting the search.");
2. the user was able to find images representative or coextensive with that need ("The retrieved images match my initial idea very closely.");
3. the user's information need changed in the course of the search ("I frequently changed my mind on the images that I was looking for.");

Stmt.	MQS	POB	COB
1	1.8	1.4	2.2
2	3.2	3.0	3.2
3	4.2	4.0	3.4
4	3.3	3.0	2.4
5	2.8	2.7	2.3
6	4.1	3.0	2.9
7	2.9	2.4	2.3

Table 3.5: Likert scale means for each statement

	MQS	COB
<b>Statement 2</b>		
images don't match initial idea	4	5
<b>Statement 3</b>		
changed mind on images	8	9
didn't change mind	7	4

Table 3.6: Split of answers on changing ideas (number of responses per statement)

- the change of his need was due to the facilities offered by the system (“Working through the image browser gave me alternate ideas.”).

The remaining statements captured the *user's satisfaction* with the search process and the system. Their responses would indicate the extent to which the user was satisfied with:

- the outcome of their search (“I am very happy with the images I chose.”);
- the level of *recall* attained (“I believe that I have seen all the possible images that satisfy my requirement.”);
- the overall outcome of their interaction with the system (“I believe I have succeeded in my performance of the design task.”).

Like before, each user was asked to respond to these statements three times (after each task they carried out on the different systems). The result was a set of 378 scores on a scale of 1 to 7 (with 1 representing the response “I agree completely” and 7 representing the response “I disagree completely”): 18 respondents scoring each of three systems with respect to each of the seven statements. The mean results are shown in Table 3.5.

Furthermore, since an evaluation based on the retrieved images *after* the search has been completed is hindered by subjective bias (Black et al. 2002), the participants were invited to draw sketches of the kind of images they had in mind before starting the search (if they had any). This ensured that there was a point of reference for them to judge whether the retrieved images matched their initial sketches.

*Information Need Development:* The scores for the respondents' reactions to the statements regarding their information need requires careful consideration. When they were asked about their initial idea of the images they were looking for, the responses showed that their initial need was reasonably well-defined (Stmt. 1). Users of COB were more inclined to change the initial need than for MQS and POB (Stmt. 3). However, the responses for the second statement whether the retrieved images matched their initial information need, were uniform across the systems (Stmt. 2). Still, when asked whether they thought the system gave them alternate ideas, COB scored significantly better (Stmt. 4). The significance of the difference is reflected in the values of the Friedman test statistics calculated in order to test the experimental hypothesis that the scores for COB are better (lower) than for MQS. The value of the Friedman statistic was found to be significant at a level of  $p < 0.05$  ( $p = 0.024$ ).

Analysing the comments about why they thought the images matched their initial idea (Stmt. 2) and why they changed their idea (Stmt. 3) sheds more light on the above results. We split the responses for these two statements into two categories: either their initial idea changed or did not change. For each category we considered only the responses where people stated they agreed (answers on the scale of 1–3) or disagreed (5–7). Table 3.6 shows the resulting split of answers.

A comparison of the responses for MQS and COB yields the following results (responses for POB are very similar to those for COB and are therefore omitted). For MQS, all of the 4 users, who believed that the retrieved images do not match their initial idea (Stmt. 2), indicated that was because they could not find the images they visualised: *“I could not find the right ones”* or *“the system gives you slightly unexpected results”*. The same reasons were also brought forward by 4 out of 8 users when asked about their opinion of why they changed their mind (Stmt. 3).

On the contrary, the comments for COB suggest that 4 out of 5 people deviated from their initial idea rather because they were offered a bigger choice and variety in the images: *“there were plenty of images to choose from”* or *“I found other cool images first”*. 7 out of a total of 9 who changed their mind in COB thought this was the case because they were offered a better selection of images: *“the idea of having related images displayed next to each other evokes reconsideration of choices or sparks off other ideas. It makes it easier to choose between images”* also showing the advantages of the presentation of the retrieved images. These comments highlight the reasons for changes in their information need in the course of the interaction with the system. This shows that: (1) a suitable interaction and presentation technique can assist a user’s developing needs (and thus assist the creative process); and (2) there is a necessity for system adaptation to reflect changing needs.

A similar comparison can be made for the users’ judgements of why they thought they did *not* change their minds. All 4 users who indicated that their information need remained constant on COB stated that they just had a clear idea of what kind of images they wanted: *“got more of the images I wanted”*. The reasons of why it remained constant on MQS are quite different. Only 2 out of 7 people in total who claimed they did not change their mind believed that they had a clear image: *“had ideas and stuck to them”*. 4 users however pointed out that the reason was the missing option of exploring the database: *“I saw less images—could not explore lines of images”* and *“more direct way of searching not leaving as many images to choose from”*.

To summarise, from the above analysis it emerges that, while most users had a mental model of candidate images, this model was changing during the search process. The system used had a major impact on the reasons for such changes. COB supported an explorative search causing their needs to evolve by offering a large selection of alternative choices. In MQS, however, many people at some point faced the problem that they were unable to retrieve any more images (usually when they exhausted keywords). They often had the feeling that the images they were looking for were not in the database, and they were puzzled and frustrated because they could not tell whether the images were indeed not there or whether they could not formulate a proper query. The majority of people who changed their mind on the initial images interpreted that in a negative way as a result of not being able to find the right ones. One person’s comment reflects this mood: *“My expectations have been adapted to the available images. This, however, is not how a designer wants to think, he doesn’t want limitations to influence decisions.”*

*User Satisfaction:* When analysing the scores of the statements concerned with the overall user satisfaction, no significant differences could be shown to conclude an overall improvement on satisfaction on task completion. Still, MQS's scores were always poorer, and the user comments presented below support the observation that they were generally more happy with the selection of images in the browsers. There are various other factors that can influence the satisfaction on task completion, too, for example the available images in the collection. After all, if a user is not really happy with the available images, none of the three system would be able to change this. Due to the relatively small sample size in our study, only a small number of such outliers can have an effect on the statistical significance of the results.

**Open Questions** In order to gain more insights into the users' preferences, the participants were asked to specify which features of the system they liked, which ones they disliked and what features they would like to have seen added. The responses obtained here were quite similar to the ones in the final questionnaire. To avoid repetition, they are presented together in the next section.

### Final Questionnaire

After having completed all three tasks, the participants were asked to rank the three systems in order of preference with respect to: (i) the one that *helped* more in the execution of their task; and (ii) the one they *liked* best. Both questions resulted in a very similar ranking of the systems. 10 out of the 18 participants ranked COB more highly than the other systems, and 12 placed both ostensive interfaces as their top two. The mean of the ranks were: MQS 2.5, POB 1.9 and COB 1.6. Again, in order to test the experimental hypothesis that the sets of 18 post-search scores for each system type were sampled from different populations, the Friedman statistic was calculated, which was found to be significant at a level of  $p = 0.017$  (for both Questions (i) and (ii)). Dunn's post test showed that a significant difference was between MQS and COB (with  $p \leq 0.05$ ), however not between MQS and POB. Our conclusion, therefore, was that people liked COB significantly better than MQS, and found it significantly more useful for the task we set them.

Respondents who ranked MQS highest appreciated the system's accuracy and being able to control the search, eg "*fastest of the 3 systems in finding specific images*". The features liked most were the combination of visual and textual features. However, some users found it hard to interpret the features and how to specify the correct combination. The complexity of formulating a query in MQS emerged in many comments: "*quite complex*", "*have to input too often*", "*confusing to control*". Some people also found MQS "*too restrictive*". Other participants, who used one of the other systems first, missed the ability to browse the images or return to previously retrieved images.

Those respondents who preferred either of the ostensive browsing approaches valued the fact that they were very intuitive and appreciated the "*visual representation of the search process*" ("*easily understandable 'line of choices'*", "*ability to compare images on screen + backtracking*"). They considered the search process a "*very image-based selection*". The difference between the two approaches seems to be the flexibility (COB), on the one hand, and the ease of use (POB), on the other. POB was generally referred to as: "*very intuitive, fast*"; "*pleasure to use*"; and "*relaxing*". Arguments for the POB approach included: "*it is easier to pick images rather*

*than to choose words*” and *“very fluid movement—just the images”*. POB’s drawbacks were concentrated on the missing ability to control the search: *“being stuck in a sequence, not being able to edit and control it”*. The additional control options, however, were also criticised by some users in COB. A few people disliked the system’s automatic selection or found it *“offered too much control, there’s too much to think about”*.

Apart from this, most responses about COB were entirely positive. It was still deemed *“easy to understand”* and *“very straight-forward”*. In addition, people liked its adaptability and versatility. They seemed to consider this system a more complete approach (*“most options, best display of information”*) and regarded the system *“very helpful”* and *“intelligent”* in making *“smart selections”*. The effectiveness of the system is reflected in a lot of responses: *“it is most efficient to use and get the desired results”*, *“search seemed more consistent”* and *“felt more extensive”*. Hardly anyone ever got stuck during the search process, and one of the features liked best included *“the fact that it kept going and going”*.

### Quantitative Results

In order to test the actual user performance in quantifiable, objective measures, a number of usage data was logged during the experiments. The kind of data logged included:

- time taken for the complete search session
- number of distinct images retrieved during the search session
- number of searches per session

Most interestingly, the time taken for the whole session was not significantly different between the systems. On average the completion times were 15min20sec for MQS, 15min30sec for COB and 13min54sec for POB. Comparing the individual times for each user it emerged that the completion time largely depended on the user: people tended to spend approximately the same amount of time for each system. A further factor is tiredness or boredom that might have affected the timing. The ordering of the systems had a slight impact on the time spent for searching: 16min29sec for the first system used, 14min43sec for the second and 14min31sec for the third. Again, the differences are not large enough in order to conclude that tiredness influenced the evaluation adversely.

In contrast, the number of distinct images retrieved in approximately the same time span was much higher in the browsing systems. On average the number of distinct images for MQS, COB and POB were 58.2, 82.9 and 83.0 respectively. The difference between MQS and COB could shown to be statistically significant ( $p = 0.012$ , value of Friedman statistic, adjusted for ties).

Finally, we have looked at the number and types of queries issued. In MQS, an average of 32 manual queries were issued. In POB and COB, the total number of ostensive queries (that is the number of times an image in the path was clicked on) was 33 and 32, respectively. In addition, there were on average 15 forks in the paths in POB, that is the number of times the users went back to previous images on the path to create a new branch from a different candidate. In COB, in contrast, there were only 2 forks. This was replaced by manually changing the system’s query, which took place 13 times on average (7 weights adjustments and 6 keyword adjustments).

Interestingly, the number of manual queries necessary to select a starting image in COB was reduced to 4, compared to 8 in POB.

This is an indication that the browsing systems, by reducing the number of times they have to formulate explicit queries, succeeds in the user seeing more images in the collection. We believe that the time the user has to spend on the query formulation and re-formulation in MQS is used in a more productive way in the browsers. In fact, POB (in which there is no query formulation process necessary at all except to find starting images) has the highest rate of image recall per minute (6.0 compared to 5.4 for COB and 3.8 for MQS).

### Observations

The observations of the participants using the system revealed further interesting facts. One—probably the most prevalent—issue to arise was the problems associated with the use of keywords. First of all, only few people used more than one search term at a time. Furthermore, they were often surprised at the results they obtained. Subjectivity in the choice of terms to describe an image was apparent throughout (“*summer? that’s not my definition of summer!*”). This was especially limiting in MQS, since the keyword search was the most used feature in this system. As a result, most people considered the task to be finished after they could not think of any more keywords to use. The only option for retrieving more images in MQS, when the user exhausted words, was to play around with the weighting between the two features. Many participants took this approach, but it was mostly a trial-and-error process used in order to see whether they could retrieve any different images.

Another problem, which became most apparent in MQS, was that people cannot easily relate to content-based image features. Even though they were told that the feature used was ‘colour only’, most people when selecting ‘query-by-example’ representations, had the idea set in their mind that they wanted ‘more images like this one’. They could not distinguish between ‘images that have the same colour’ and ‘images that are generally similar’ (in terms of semantic content, layout, colour, etc.). As a result, they often obtained unexpected results, since the returned images did not resemble—in their minds—the ‘query-by-example’ images.

A further interesting point to notice is that the ostensive browsing approaches seemed to be more successful in giving the users confidence and insight in the available images. The users got the impression that “*there are so many more images to choose from*”. On the other hand, when using MQS, people thought the image collection to be “*limiting*”. In addition, people often felt lost, because they could not tell whether the problem was that they were not able to formulate the query or whether the images were simply not in the collection. While using the browsing systems, this uncertainty did not arise due to the different approach to searching. The perception of the participants is reflected in numbers: the number of distinct images seen was indeed higher for the browsing approaches (see Section 3.1.6, Quantitative Results).

Most of those points mentioned are the subjective observations of the evaluators only, and cannot be considered to be representative or ‘statistically significant’. However, they do convey interesting aspects and are believed to help in the further design of image retrieval systems. Moreover, the findings are in accordance with those of other studies (Jose et al. 1998, Markkula & Sormunen 2000, McDonald et al. 2001).

### 3.1.7 Results Summary

In this section, we will summarise the results from this user study from a specific perspective, in which we compare the results concerning the three evaluated interfaces and draw conclusions on the experimental hypotheses.

The evaluation showed that people preferred the search process in the ostensive browsing scheme, felt more comfortable during the interaction and generally found the system more satisfactory to use compared to the manual interface (see Section 3.1.6). In a study concerning the nature of the information need, it emerged that the Ostensive Browser (OB) provides for an explorative search that reflects and supports dynamically changing needs (see Section 3.1.6, Likert Scales). The analysis of user comments supported the view that the user's underlying need changes while they explore the collection, although they mostly have a mental model before starting the search. The OB was more successful in both eliciting such changes and adapting the retrieval in response. This defends our proposition for an adaptive, interactive retrieval system. The two versions of the OB we provided revealed a tradeoff between simplicity (for the pure version) and flexibility (by providing additional control facilities). While most participants preferred the flexibility, they also appreciated the pure browser's simplicity.

To conclude, we believe the evaluation proved the success towards an effective and versatile approach in the form of the OB equipped with additional control facilities (COB). It provides a simple browsing-like interaction that allows for an explorative search and serendipitous discovery. The adaptive scheme emulates the development of the user's need during such explorative phases.

We have also learnt that explicit control in the OB is often necessary to steer the search in certain directions (for approximately every second selected image the candidates were changed manually in COB by selecting new query terms and/or changing the feature weights, see Section 3.1.6, Quantitative Results). Without this control, the users can still browse through the collection but are not as satisfied with the results they receive.

Another problem we have briefly mentioned in the beginning is the question of how to start a search in the OB, or the page zero problem. In this study, we have—rather crudely—solved the problem by allowing manual searches to retrieve starting images for the OB. This meant that users occasionally had to switch between two interfaces, which occurred on average 8 times in POB and 4 times in COB (see Section 3.1.6, Quantitative Results). This shows that the OB is not ideal for supporting abrupt changes in information need or multi-faceted needs. Whenever a new facet, which is unrelated to previous results, is explored a new OB window has to be started.

There were other issues with our experimental methodology. The image collection used is relatively small and might have had an impact on practice effects. Although the choice of topics for the tasks was such that the overlap of images suitable for each topic was minimised, further studies are needed with a much larger collection in order to generalise the results. Moreover, the study included only one *type* of task with an emphasis on design and creativity, which biased the comparison of the systems towards their ability to support exploratory searches. In fact, there were indications that the OB is not as good for very direct or targeted searches (see previous paragraph). The difficulty and necessity of judging an image retrieval interface's ability to support various types of (realistic) tasks has resurfaced and consequently been addressed in the main evaluation in Chapter 6.

In the meantime, however, the results discussed above highlight many important aspects of CBIR interface design. While the usability of a system depends largely on its interface, the performance of the underlying algorithms cannot be neglected for judging a system's overall effectiveness. The retrieval performance of the OM-based query learning scheme is better judged in comparison to other relevance feedback techniques in a more objective quantitative evaluation. A simulated evaluation we have conducted to this end showed that performance can be increased in the ostensive browsing scenario. The results are discussed in Appendix A.

This evaluation has helped us expose common issues in CBIR systems, including the extent of the query formulation problem and the dynamic nature of information needs. In addition, we found evidence on the subjectivity of image meaning. Since these problems are central to CBIR, we feel obliged to discuss each individual issue in greater detail in the following section. The chapter will then conclude with a summary of our observations regarding these problems. This summary will discuss evidence that surfaced in this study, providing a general perspective of the experimental results.

## 3.2 Open Issues

### 3.2.1 The Meaning of an Image

The *meaning* of an image is, without doubt, delicate to grasp. As a work of art—similar to poems—an image's meaning can hardly be pinpointed with universal consensus. Current CBIR technology has difficulties in extracting the major objects contained in an image, let alone its meaning.

One way to overcome the *semantic gap* between the system and the user, is to learn semantic concepts in order to move closer to decoding meaning. Semantics—the study of meaning—deals with the relationships between linguistic symbols and their meanings. Since there is no universal meaning, the semantic concepts depicted in, or otherwise emerging from, an image are individual to a user. The dependency of semantic concepts on individual *interpretation* and *context* has been widely acknowledged in the CBIR literature:

*“We argue that images don't have an intrinsic meaning, but that they are endowed with a meaning by placing them in the context of other images and by the user interaction.”* (Santini et al. 2001, p. 337)

*“[Semantics] emerges as the final result of the interactive exploration of the data.”* (Jain 2003)

*“[High-level] conceptual aspects are more closely related to users' preferences and subjectivity. Concepts may vary significantly in different circumstances.”* (Zhao & Grosky 2001, p. 15)

Approaches towards semantic concepts for image retrieval are mushrooming (*eg*, Oliva & Torralba 2001, Bradshaw 2000, Lim 1999, Chang et al. 2003, Jeon et al. 2003, Duygulu et al.



2002, Zhao & Grosky 2000, Srikanth et al. 2005, Pan et al. 2004, Wang et al. 2001, Su & Zhang 2002, Zhou & Huang 2003, He et al. 2003, as discussed in Section 2.2.4). Nevertheless, there is the need to define a general framework of how to extract, encode and consequently use semantic concepts for image retrieval (eg, Naphade et al. 2005, Snoek et al. 2006). Moreover, only few of the approaches interpret semantics as they emerge from user interaction (Su & Zhang 2002, Zhou & Huang 2003, He et al. 2003, Lin et al. 2005). Even worse, the context the semantics are interpreted in is usually confined to the retrieval environment.

In an experiment to assess the agreement between machine and human measures of image similarities, Squire & Pun (1998) have shown that there is a large discrepancy between the clusters chosen by human and machine. Agreement between pairs of humans was a lot higher, although the average agreement was only one third higher than expected by chance. The authors conclude that “*the appropriate measure will depend not only on the individual user, but also on the genre of images, and the task the user is performing*”. Therefore, successful approaches have to recognise the importance of context, which is not within the retrieval engine, but is determined by the tasks and work environment. To truly make an effort towards comprehending an image’s meaning, the image has to be placed in a wider context.

### 3.2.2 The Query Formulation Problem

The lack of a semantic representation of images in CBIR systems makes it more difficult for the user to pose adequate queries to the system. It is, however, not the only factor that influences the query formulation process. Query formulation has also troubled many information retrieval researchers (eg, ter Hofstede et al. 1996, White 2004, Ruthven 2005).

Every information seeking process is necessarily initiated by an information need on the user’s side. To be able to interact with the system, the user is asked to formulate this need into a query, which the system can process. However, the translation of an information need into a query is hampered by many problems.

#### The Problems

To start with, the user often does not know how the documents are represented. This is especially true for CBIR systems due to the low-level representation of content-based features. Current systems often require the user to sketch a query image or even express a query in terms of those features, which usually include options such as:

- Covariance of Lab colours;
- RGB Histogram, 256 bins;
- Gabor texture of luminance, etc.

System designers often think that the more options they provide, the better the user can define a query. This is only true if the user knows precisely what all the features actually mean (and even then it is hard to estimate whether two images have a similar “Gabor texture of luminance”, for example). None of those approaches are desirable for an average user.

A further problem is the underlying information need itself. The need is typically vague—“*I don't know what I'm looking for, but I'll know when I find it*” (ter Hofstede et al. 1996), which complicates its translation into a formal query language. Due to the uncertainty about what information is available or about the actual need itself, a search process usually starts with an *explorative phase*, in which the user tries the system and its options, and tests what kind of information is returned. During this process, the user's need can change quite dramatically due to the exposure to new information. This often leads the user reformulating the initial query to either make it more precise after having gained some knowledge about the collection make-up, or steer it in different directions after having seen other interesting documents, or a combination of both.

Image retrieval systems, in particular, should address the vagueness of the underlying information need (Garber & Grunes 1992). Image retrieval systems are often used in design-related environments and employed for highly creative tasks. Therefore, the search session should be treated more like an explorative process, which necessarily needs to capture the process of evolving information needs.

### Proposed Solutions

Assisting the user in the query formulation process is a crucial factor in retrieval systems. For image retrieval systems, the following solutions have been proposed and developed over the past decade.

**Query-by-Example** The difficulties with translating an information need into the low-level pictorial attributes has led to the development of an alternative form of query input. Instead of specifying image features directly, in the *Query-by-Example* (QBE) paradigm they are implicitly provided by one or more example images. It is significantly easier for the user to choose images that are in some way similar to the kind of images they are looking for (eg, Yang 2004).

There are two crucial obstacles associated with the QBE paradigm. First, if the user does not have a suitable image to hand (probably that is why there is the need to use the retrieval engine in the first place), it is impossible to start the search in this way. Second, the semantic gap comes into play again. Since the user is likely to look for images based on abstract concepts (eg, Enser 2000, Markkula & Sormunen 2000, Armitage & Enser 1996, Cunningham et al. 2004, Forsyth 2001), the QBE images will be likely to be chosen based on this level. The retrieval engine, on the other hand, can only extract the low-level features from the examples, and will thus base the search on this low-level visual similarity. As a result, the retrieved images often do not match the user's perception of similarity (Squire & Pun 1998).

**Query by Visual Keywords or other Semantic Concepts** Later approaches aim to alleviate the query formulation problem by creating a higher level representation of the images. Visual keywords, for instance, aim at developing a sort of visual dictionary to specify queries. With this dictionary, the user can compose the desired image from the “words” provided. The words in the dictionary can be represented in different ways. In one of the earlier versions, the dictionary consisted of various patterns, which are made available to the user as templates. The most popular example is the texture thesaurus developed by Ma & Manjunath (1998). A more recent approach

uses more sophisticated image templates, which include buildings, water, grass, faces, crowd, etc (Lim 2000). Since many of the current users still find it most natural to use keyword-based searches, attempts have been made to “translate” image templates into keywords (Jeon et al. 2003).

Since object recognition is still an unsolved problem in computer vision, advanced image segmentation techniques can serve as a suitable approximation to obtain visual templates. Visual templates are based on either fixed-sized block segmentation (Lim 2000) or region-based segmentation (variable-sized blobs) (Jeon et al. 2003). The advantage of blobs is that the segments are more coherent than fixed-sized blocks, but they largely depend on the quality of the segmentation algorithm. A retrieval system based on segmented images can furthermore exploit local structure and composition in the retrieval.

Visual templates or keywords are obtained by training the system on annotated image segments. The visual templates can also be subject to adaptation in an iterative search process. Since recognition can only be approximated, the uncertainty of the recognition process is typically modelled in a probabilistic framework. The probabilities that are associated with the segments in an image are then updated when new information is made available by the interaction with the user (eg, Su & Zhang 2002, Zhou & Huang 2002).

A large variety of additional techniques for semantic features has been proposed recently. A discussion on a selection of approaches regarding their benefits and drawbacks has been presented in Section 2.2.4. To summarise, the major problem that still needs to be overcome is the large-scale training required to learn semantic concepts. Furthermore, there is the question of how to learn and adapt semantic concepts from the user directly. There is a trade-off between user-centred semantic concepts and automatically derived semantic classes. The user-centred approach learns the semantic concepts as they are interpreted by the user, which are consequently most appropriate for them (see also Section 3.2.1). At the same time, it is difficult to obtain reasonable semantic classes from the small number of training samples provided by the user.

**Browsing and Navigation** People have suggested incorporating browsing or navigation facilities as a step to circumvent the query formulation stage altogether. In order to support browsing and navigation, the images are structurally organised, eg clustered by visual similarity or organised into classification schemes based on semantic concepts. Such an organisation is employed to provide the user with an overview of the collection and insights into the collection make-up. This makes it easier for the user to locate areas of interest in the collection (Rodden et al. 2001). This approach is adopted, for instance, in the *CIRCUS* system introduced in Section 2.4.2. A user study conducted by Yang (2004) has shown the merits of the browsing approach (based on self-organising maps) over the QBE search.

While browsing should be an important part of image retrieval systems, replacing the direct search facility altogether is not the best option either. For some tasks, it is necessary to locate images matching certain criteria not necessarily reflected in the clustering of the whole collections. Thus, re-organising the pre-computed clustering according to criteria specified by the user would be a solution to integrate both browsing and searching paradigms. However, this is easier said than done. The main problem is that any dynamic approach to clustering is limited by the computational costs involved. The system designer cannot expect the user to be patient enough to wait until the

collection is re-clustered every time a new search is issued, for example. A dynamic categorisation approach has been adopted in the *Haystack Browser* (Low 1999) for textual documents, albeit without the automatic clustering of documents. In this system, query results are automatically filed by creating virtual folders in the user's file system. This is an exemplary solution to building an organisation based on users' past searching behaviour. We will come back to the *Haystack Browser* in the following chapter (Section 4.1.2).

**Relevance Feedback** Discussed extensively in Section 2.3, the relevance feedback approach is another method to overcome the problems with query formulation. It is an automatic process of improving the initial query based on relevance judgements provided by the user. The process is aimed at relieving the user from having to reformulate the query in order to improve the retrieval results. Instead a learning process is initiated, which aims to match the automatically generated retrieval scores with the human judgements of relevance.

Relevance feedback is a very effective method to both improve retrieval performance and assist the user by engaging them into an interactive search session. The search becomes more intuitive to the user, since they are only requested to give judgements on the returned images of whether they match their information need rather than having to generate an effective query themselves. Despite its apparent advantages, relevance feedback does not cure all problems. From a computational perspective, it is still an ongoing research challenge to accurately learn the information need from the user based on a few relevance judgements (Zhou & Huang 2003). From the user's perspective, judging the relevancy of returned items can still constitute a high cognitive load, and it can distract the user from their main goal of finding information (Ruthven 2005). Moreover, the user bases their relevance judgements on subjective, high-level similarities. As a result they are likely to be confused about results returned by a retrieval system based on low-level features. In this case, the semantic gap is widened rather than narrowed. This problem has also surfaced in our user experiments involving a relevance feedback system described in Chapter 6.

### Summary

There are, of course, many different types of users and different types of searches (well-defined, ill-defined, searching for a known object, searching to get inspiration, etc.). Also, there is a tradeoff between system accuracy and ease of use. In the case of image retrieval systems, however, there is a greater need for intuitive and flexible interfaces. What becomes evident is that in order to reduce the burden on the users, the system needs to support them in their search for information. In recognition of the fact that information seeking is an inherently interactive activity, the system should provide for an intuitive and interactive interface.

### 3.2.3 Dynamic Nature of Information Needs

Relevance feedback as a solution to learn the internal query representation to match the user's information need, is the most promising direction in CBIR. However, the fact that the information need is time-varying is often ignored in favour of simpler algorithms.

In contrast to this tendency, there are strong reasons to assume that the information need is subject to change during a retrieval session. The information need is dependent on the user's

knowledge state, which is a summary of their previous experience. While interacting with the system, the user is exposed to new images which extends their knowledge. The need might change in light of new information. For example, if the initial need is vague, knowledge about the collection make-up and search environment will help the user to evolve their need into one that is more precise or well-understood. If, on the other hand, the need is already well formulated in the beginning, the interaction with the search engine might still cause them to consider different options. For example, various circumstances causing a change in information needs have come to light in a study of art directors' searching behaviour (Garber & Grunes 1992).

There are some approaches that model the dynamic nature of information needs. For instance, Pentland et al. observe:

*“people are nonlinear time-varying systems whose behavior depends on unknown internal states.”* (Pentland et al. 1993, p. 21) (also appeared in (Pentland et al. 1994))

Thus the one-model-fits-all approach is deliberately avoided in favour of offering a variety of models, or *“society of models”* (Minka & Picard 1996), in which a selection takes place on the basis of interaction with the user. Also Vasconcelos & Lippman's probabilistic approach (2000) (see Section 2.3.3) incorporates a weighting factor for the importance of the past. A more rigorous methodology to take into account time-variant information needs is the Ostensive Model (Campbell & van Rijsbergen 1996) detailed in Section 3.1.2.

Still, more work needs to be done to this end. Current relevance feedback techniques treat the relevance judgements gained over a number of iterations homogeneously, sometimes even collecting them all in a pool before starting the learning procedure. It would be more beneficial if the relevance judgements were *traced* rather than *collected*. In this way, new feedback can be compared to previous feedback to detect changes over iterations. This approach is adopted in the Ostensive Model, which we presented in the beginning of this chapter. Our goal was to study its strengths and weaknesses from the user perspective in order to gauge the impact of all the issues elaborated in this section.

### 3.3 Discussion

In this section, we will revisit our user study and discuss the results from a general perspective. These general conclusions include the lessons learnt regarding the extent of the three open issues in CBIR and the derived requirements for an “ideal” interface that addresses these issues.

Recall that our main goals were to determine the extent of the query formulation problem and the nature of information needs. In addition, we found evidence on the subjectivity of image meaning. Therefore, all of the three main problems of CBIR identified in the beginning of this chapter have surfaced in this study.

**Image Meaning** As mentioned in our observations in Section 3.1.6, there were apparent problems with using a controlled vocabulary. The choice of keywords was limiting and the obtained results were often surprising to the users. This provides evidence that semantics are user and task-dependent.

We also found evidence on the subjectivity of relevance even when task context is taken into consideration. For each of the three topics we set, we asked the user to select all suitable candidate images before making a final choice while searching the collection. The number of candidates chosen was approximately 7 for each topic and system. If we determine the number of relevant images for a topic by taking the union of selected candidates of all users, we obtain 76, 80 and 82 for the three topics, respectively. Counting the number of users that selected the candidates reveals that a huge majority has only been selected by one user, 67%, 71% and 65%, respectively, showing that there is not very much user consensus at all. The percentages of candidates chosen by three or more users are 6.6%, 12.5% and 13.4%, respectively. This corroborates other people's observations (eg, Squire & Pun 1998, Santini et al. 2001).

These two findings suggest that it is more viable to base a semantic feature on user opinion rather than generic concepts. Of course, generic concepts, especially in the form of keyword annotations, are still useful in addition to a user-based mining approach.

**Query Formulation Problem** In addition to the problem of choosing good query terms, the users had even more difficulty with interpreting low-level features. Both these issues show the reality of the query formulation problem in image retrieval interfaces, which seriously impedes the usability of a manual system limited to keyword and QBE search. The problems render it very difficult indeed for users to become familiarised with the collection, leaving them uncertain about the availability of images and their ability to retrieve them.

In the OB, the user is not required to explicitly formulate their need as a query, which is instead incrementally constructed by the system based on the user's choice of images. However, it became apparent that the users had reservations about letting the system guide the search on its own. They often needed to manually change the system's predicted query, which did not circumvent the query formulation problem entirely.

**Dynamic Needs** Last but not least, we have uncovered plenty of evidence that information needs are likely to change during a search sessions. We could identify different reasons for such changes: new ideas came to the user's mind on their own accord while searching; the user came across better images sparking alternative ideas; or the user could not find images relating to their own ideas forcing a change. Moreover, the ability to actively support developing needs was thought to be a major advantage of the OB.

While the OB is able to trace gradually changing needs, it is unsuccessful at detecting abrupt changes. It also fails at detecting multi-faceted needs and has no support for developing long-term needs.

We have investigated an approach that supports a way of adaptive *content-assisted browsing*, addressing many of these difficulties the user has to face in an image retrieval system. However, we believe that it is best suited to exploratory-type searches, while more complex information needs are more difficult to satisfy. The objective of this thesis is to formulate a 'holistic' approach supporting all kinds of information needs. Therefore, we have taken the lessons learnt from this study on board for the design of the system that is subject of the next chapter. Yet we have wound up looking for alternative ideas to support user's searching behaviour. Nevertheless, the

advantages of the OB should not be ignored, and ideally a browsing panel should be integrated in the system introduced next.

### 3.4 Summary

This chapter has highlighted the deficiencies of current CBIR approaches. In order to study how the user is affected by these deficiencies we described and evaluated an adaptive retrieval approach towards CBIR based on an innovative browsing scheme. This approach is based on the concept of ostension. The underlying idea is to mine and interpret the information from the user's interaction in order to understand the user's needs. The system's interpretation is used for suggesting new images to the user. Both text and colour features were employed and combined using the Dempster-Shafer theory of evidence combination. A user-centred, work-task oriented evaluation demonstrated the value of the adaptive technique by comparing it to a traditional CBIR interface. It also showed how the development of the information need during the information seeking process affects and is affected by the search system.

The ostensive browser addressed the query formulation problem and time-varying information needs. However, the users did not completely trust the system to correctly estimate their information need and return relevant candidates at each step. They needed additional control over their queries as was the case in the controlled ostensive browser. Moreover, it was hard to follow up on multiple search threads (multi-faceted information needs). Although the ostensive browser allows the creation of branching paths, each branch is still connected to the initial query image(s). If a user wants to follow up on *distinct* search threads, they have to use different browse windows to do so.

In addition, we have not addressed the issue of how to create a semantic representation for images in this approach. We believe that it is essential to treat the search process as only part of the whole work process. One step towards this is to eliminate the distinction between *search* and *organisation*. More can be learnt from the user when combining the search and organisation process and it should lead to easier and more natural interaction with the system. In the next chapter we will introduce a 'holistic approach' that will address all three deficiencies.

---

### EGO—EFFECTIVE GROUP ORGANISATION

---

The study of current approaches in CBIR in the previous chapters highlighted some intrinsic unsolved problems, namely the uncertainty of image meaning, the query formulation problem and the dynamic nature of information needs. Most of these problems can be alleviated by more intuitive interfaces supporting richer interaction strategies—as indicated by the results of the user study comparing a browse-based and a traditional manual querying interface in the previous chapter. The query formulation problem is mitigated in the browse-based approach, while gradually changing needs are catered for through the concept of ostension (cf Section 3.1.2). The Ostensive Browser’s main shortcoming, however, is that it does not support complex information needs, limiting its usefulness to exploratory-type searches.

In the main user study of *EGO* (which will be discussed in Chapter 6), we asked design-professionals about ideal tools that would support their work tasks most effectively. A detailed analysis is presented in this chapter in Section 4.2.1. The most important features of an ideal system identified this way are: supporting the workflow and capturing the work task; supporting opportunistic search strategies; creating a personal image library; and collaborative work. We believe these are attainable by placing more emphasis on the way information is used and managed while searching. In Section 4.1, we show that many researchers are of the opinion that organising information helps to structure the thought process of the searcher. Consequently, the interface should be endowed with better result management and personalisation techniques to make the organisation process an integral part of any search interface.

The analysis of user expectations has led us to develop a tool, *EGO*, that places emphasis on the long-term management and personalised access to an image collection. The long-term usage provides additional search clues, such as usage histories of images and groups, that, when combined with the low-level image features, increase the retrieval effectiveness. *EGO* provides the means to describe a long-term multifaceted information need. To achieve this, the user and the system collaboratively group potentially similar images. The process of grouping images stretches over multiple sessions, so that existing groups are changed and new ones are created whenever the user interacts with the collection. Instead of implicitly assuming gradual changes in the information need as in the Ostensive Browser, users explicitly define the facets of their information need. The system is automatically informed of significant changes or “context switches” when observing



that the user switches back and forth between groups. As will be verified in consequent user studies (see Chapter 6), the process of creating groups to represent multiple facets of an information need comes naturally to the user.

By placing the groups on a workspace, the user leaves trails of their actions behind for themselves, or others, to inspect and follow. The process is incremental and dynamic: a dynamic organisation emerges and changes over time. A semantic organisation emerges that reflects the user's mental model and their work tasks. These are the two most important influences on the organisation of personal media as identified by Kang & Shneiderman (2003): "*There is no unique or right model; rather the mental model is personal, has meaning for the individual who creates it, and is tied to a specific task.*" Hence, by helping the user to effectively manage and search their images, *EGO* aims to represent the context in which the images are used, in short, a personalised "*retrieval in context*" system. It captures both short- and long-term information needs, communicated by leaving behind trails of actions, and used by the system to adapt to the user's need. How this is achieved in practice will be described in the following sections. These ideas were also published in (Urban & Jose 2004a, 2006c). First, however, we will introduce the conceptual ideas illustrated by previous approaches in the literature.

## 4.1 Background and Related Work

Organisation, both spatial and categorical, is a vital tool to assist the thought process of the searcher, and, as such, should be supported by the system. Before presenting examples of how search can be facilitated by organisation, it is beneficial to introduce the cognitive aspects of information seeking.

### 4.1.1 Problem Solving by Organisation

#### Mental Models

Information seeking can be regarded as an instance of problem solving. According to cognitive scientists the natural way of thinking is to construct *mental models* of the premises (Johnson-Laird et al. 1998, Gentner & Stevens 1983). Wikipedia provides the following definition of mental models for the layman to understand:

*"A mental model is an explanation in someone's thought process for how something works in the real world."*<sup>4-1</sup>

Mental models are a vehicle for reasoning, explanation and illustration, and help us to provide a working strategy for the problem at hand. However, mental models can be contradictory, incomplete or varying in time. Most importantly, they are *individual* and therefore user, task and context-dependent.

Of specific interest to us is how mental models supply people with a means of understanding the functioning of interactive systems. The user develops a mental model of how they think the system works through interaction with the system. This model is used to reason about the system,

---

<sup>4-1</sup><http://www.wikipedia.org>

to anticipate system behaviour and to explain why the system reacts as it does to user actions. However, there is often a gap between the designer's conceptual model—devised as a tool for the understanding or teaching of systems—and the user's actual model (Norman 1988). The system itself has to bridge this gap, since it is usually the only means of communication between the designer and the user. Through the *system image* (Norman 1988)—the visible part of the system—the designer can help users form an accurate and useful mental model, which will allow them to interact with the system successfully.

### **Conceptual Models for Information Seeking**

Conceptual models for information seeking behaviour are discussed by Järvelin & Wilson (2003), including Ingwersen's model of the IR process. Similar to Norman's ideas for physical systems, Ingwersen (1996) postulates that a comprehensive model of information seeking behaviour must include: both the designer's and the user's model, as well as the IR system. Users have models of their work tasks or their information need. Yet there is often a gap between the user's model and the designer's model of what the system should do and how it should function. Hence, the user's model is constantly modified throughout the interaction with the system, so-called "cognitive transformations" occur, until the point where relevant documents can successfully be identified (ideally coinciding with the user's information need being satisfied). In order to allow the user to form an accurate model of the system, the designer's conceptual model and the necessary transformations need to be effectively communicated throughout the system interface. Further, Ingwersen insists that the information seeking tasks should always be placed within the wider work task in order to move closer to the user model. The goal of the system designer is to anticipate the work process to solve the task and the type of result sought. The more complex the task, the more difficult these are to predict in advance.

The cognitive point of view thus implies that the information system designer's goal should be to move the mental model, that is necessary for operating the system, closer to the mental model an individual user has of how to solve the task in the life-world. This can be achieved by supporting organisation as part of searching, as will be explained next.

### **Organisation to Assist Mental Model Construction**

The power of a good mental model lies in its sense-making ability—its ability to attribute meaning to things (Norman 1988). It has been observed in a host of studies that organisation helps in understanding a body of information or a set of representations (like a collection) (Malone 1983, Kirsch 1995, Nakakoji et al. 2000, Rodden 1999, Grant et al. 2003). There are two important facets to organisation: spatial arrangement and categorisation.

The importance of space for structuring information has only recently been investigated methodically (Kirsch 1995, Nakakoji et al. 2000). Both Kirsch and Nakakoji et al. agree that people make extensive use of spatial layout when constructing meaning. Kirsch states that "*how we manage the spatial arrangement of items around us, is not an afterthought; it is an integral part of the way we think, plan and behave*" (Kirsch 1995, p. 31). His basic assumption is that we often rely more on spatial structuring than logical planning for problem solving, because orienting in

space is necessary for our day-to-day activities and thus more natural to us. His investigations of the role of spatial layouts in interactive systems lead him to the conclusion that, firstly, spatial arrangements *simplify choice*. Secondly, the spatial arrangement of items can make it easier to notice properties or categories, to find or track relevant items, therefore *facilitating perception*.

Nakakoji et al. (2000) stress the importance of spatial organisation in design tasks, such as writing, programming or architectural sketching. They observe that the design process is not typically planned as a series of calculated steps towards a specific goal, but rather the designer is engaged in a cycle of actions—producing sketches, mockups, notes, etc.—and engaging in reflections on them. The authors argue that positioning objects in a two-dimensional space supports a “reflection *in* and *on* action” process. Positioning objects allows designers to express their state of mind (reflection-in-action) and studying the spatial layout helps them understand the current state and design rationale behind the design (reflection-on-action). They conclude that computer-based tools facilitating spatial layouts are therefore closer to the designer’s cognitive process.

Categorisation, as well as spatial arrangement, is also important for managing collections of documents. For instance, Malone (1983) has studied how people organise paper documents in their office space. He suggests that the two most important functions of desk organisation are *finding* and *reminding*. Categorisation (or ‘filing’) is the most important means to support finding. However, sometimes people find it hard to decide on a classification scheme or do not have sufficient time to process all incoming documents. For this reason people have a tendency to generate piles, unsorted collections of documents usually accumulated on their desks. The spatial location is the most important cue for accessing information from piles. Location and spatial arrangements are even more important for reminding, eg a pile of things to do on the desk. Malone’s suggestion for the design of electronic office systems is to simplify the filing process by providing intelligent aids for categorising and retrieving information, not ignoring the power of location as reminding function.

In the information seeking domain, it has also been observed that the activities of organising (or managing) and exploring (browsing, searching etc.) are often inseparable (Bauer et al. 2004). Hence a retrieval system should also support the user in the organisation process. Introducing a workspace in the interface, where people can position and categorise documents, facilitates organisation. This approach has been adopted by several researchers and we introduce a selection of representative work in the following section.

#### 4.1.2 Workspaces for Search and Management

According to the principle of analogy (MacLean et al. 1991), the interface design process should be based around supporting familiar interaction metaphors in order to facilitate the user’s understanding of the system (user model). As discussed above, in an information seeking environment we can create an analogy to traditional problem solving strategies by allowing the user to organise the information they find. The user is then free to concentrate on working with the actual documents rather than wasting time and energy on formulating a good query just to find them.

### The Room Metaphor

Workspaces in graphical user interfaces were originally inspired by the room metaphor (Henderson & Card 1986)<sup>4-2</sup>. To solve a task requires the use of certain tools that are available in your workspace. Henderson & Card suggest the provision of a number of “rooms” to create various workspaces tailored for particular activities. Each “room” will provide a set of tools that are relevant to the room’s designated task. The tools in a room suggest activities that can be performed in them, and therefore switching to a room helps the user to establish the mental context for the task.

Inspired by the room metaphor, early work on workspaces in information retrieval systems has focused on providing tools to interact with many different services (Hendry & Harper 1997, Cousins et al. 1997). They were concerned with searching different resources. This was achieved by designing an interface that provides a consistent model for users to deal with a wide array of sources at the same time.

The Digital Library Integrated Task Environment (*DLITE*) system (Cousins et al. 1997) is based around the notion of “workcenters”. The tools provided in a workcenter are referred to as “components”, which include documents, collections, queries, services (eg search services, document summarisation) and representations of people (to implement access control etc.). The user solves their tasks by defining interactions between components. Queries are used to populate document collections (result sets) by means of search services. The documents within the collections can further be manipulated by other services. For example, a user can consult a document summarisation service to automatically create a summary for all the documents dragged to the service. Workcenters are created and maintained by domain experts who carefully select the collection of tools to tailor the workcenters to the needs of their users. Users then simply have to choose the workcenter appropriate for their task.

*SketchTrieve* (Hendry & Harper 1997, Hendry 1996) emphasises the spatial metaphor: the workspace gives the user control over the layout of search techniques, queries and results represented with a data-flow notation. Hendry & Harper (1997) argue that “*to seek for information, is to manage space*” (p. 1036). The workspace is considered a “canvas” on which queries and search results are added into arrangements that represent the user’s conceptual model of their activities. Queries are wired to search services in order to search a particular repository. The manipulation of search services, eg re-wiring queries to generate a new retrieval service to compare results from different repositories, allows incremental and opportunistic search strategies (similar to the cyclic design process discussed by Nakakoji et al. (2000)).

Both *DLITE* and *SketchTrieve* are useful because they provide access to heterogeneous services by means of a homogeneous interface. They also enable the searcher to capture the history of search activity for retrospective analysis, to plan future searches or to share with others. However, the retrieval components are not adaptive, since neither system stores historical information (apart from the static snapshot of the artifacts placed on the workspace) or allows interactive query refinement. For instance, they cannot help users discover commonly explored query sequences or suggest repositories to use for certain query types, which would help them avoid repeating work.

Information workspaces have also been used to facilitate the organisation of results rather than

---

<sup>4-2</sup>The room metaphor is also used to build interfaces that allow 3D navigation. However, here we focus on the *purpose* of a room, ie collecting tools in designated rooms or workspaces.

services. The categorisation of results—the topic of the remainder of this section—allows an even more direct interaction with the information that is being sought.

### Categorisation

We organise our data on a day-to-day basis into files and folders, creating a hierarchy of directories supported by the operating system. In order to support organisation of personal files, Low (1999) has developed a graphical interface on top of the *Haystack* retrieval system that resembles a user's file browser. The *Haystack Browser* provides the combination of a conventional file browser with an information retrieval system to allow dynamic categorisation of information. It creates a dynamic content-based hierarchy in response to a query. Query “folders” are created automatically in the folder hierarchy to hold documents that match the query corresponding to the folder. Manually created queries are also stored as new objects within the hierarchy and can therefore be returned as query results to future queries. Hence, the user and system interactively categorise the information space, including documents stored on the hard drive, emails, web pages visited, etc., by creating virtual query “folders”. The user has the option of refining a query by providing negative and positive feedback, and query results folders are automatically updated if new information becomes available. In addition, a user can efficiently navigate their information space by following links, or “ties”, between documents that represent relationships between two pieces of data (usually derived from meta-data). Recently, a more general *Haystack* client has been implemented (Karger et al. 2005). The focus in this client shifts even more to “orienteering” to replace the need for searching: starting at a familiar place and following an association chain via links to items of interest. The *Haystack* system relies heavily on the user providing meta-data to create these associations in order to connect pieces of personal information. Furthermore, the interface has no workspace, as such, in which the spatial layout of items contributes additional context. The focus of the *Haystack Browser* is on constructing meaning by categorisation and linking, rather than positioning, information objects.

### Organisation of Image Collections

Spatial organisation and categorisation is also the primary means to manage personal photographs (Rodden 1999, Rodden & Wood 2003, Grant et al. 2003, Bauer et al. 2004). Following on from a study of non-digital photographs (Rodden 1999), Rodden & Wood (2003) have investigated how people manage their digitised photographs. Additionally, Grant et al. (2003) and Bauer et al. (2004) have studied the use of digital tables for organising personal photographs. All studies have concluded that organisation—including “piling” and developing meaningful spatial structures—is the main activity while exploring image collections. Hence, a workspace would seem to be the key component of any tool that allows image sorting.

There is a plethora of commercial or Web-based digital photo management systems (eg ACD-See<sup>4-3</sup>, Adobe Photoshop Album<sup>4-4</sup>, Canon ZoomBrowser<sup>4-5</sup>, iPhoto<sup>4-6</sup>, Picasa<sup>4-7</sup>). They typically

<sup>4-3</sup><http://www.acdsystems.com/>

<sup>4-4</sup><http://www.adobe.com/>

<sup>4-5</sup><http://www.powershot.com/>

<sup>4-6</sup><http://www.apple.com/iphoto/>

<sup>4-7</sup><http://picasa.google.com/>

support thumbnail-based features for organising, labelling, viewing and editing digital images. Several research prototypes have developed innovative variations and extensions of these basic features, such as advanced layout mechanisms in *PhotoMesa* (Bederson 2001), improved support for labelling in *FotoFile* (Kuchinsky et al. 1999), *PhotoFinder* (Shneiderman & Kang 2000) and *MediaBrowser* (Drucker et al. 2004), audio annotation and content-based retrieval techniques in *Shoebox* (Rodden & Wood 2003), or automatic classification based on time/location (eg, O’Hare et al. 2005, Drucker et al. 2004) and events (eg, Girgensohn et al. 2003). The retrieval techniques in personal photographic collections are mostly text-based, hence the large interest in providing tools for labelling. Wenyin et al. (2003), for example, developed media agents that automatically collect textual annotations from documents that are related to multimedia data in order to relieve the user from this burden.

An interesting semantic organisational approach is introduced by Kang & Shneiderman (2003). In their prototype system they make use of *Semantic Regions*, which the user creates by simply drawing a rectangle and placing it somewhere on the workspace. Regions are associated with their semantics by assigning a combination of attributes of the media data, such as time, names of people or places to the regions. Semantic regions can reflect mental models by allowing a certain setup and layout of the regions, for example maps, calendars, organisation charts or critical paths. Images in a collection can easily be organised according to one or more mental models by dragging the images onto the regions containing the layout of the semantic regions. Each image is then automatically assigned to the regions that fit its semantics according to a “fling-and-flock” metaphor.

In summary, personal photo management systems rely on efficient browse- and search techniques for locating events and people, which can be facilitated by labelling and automatic classification techniques. Visual features, on the other hand, have not been regarded very useful in this domain, because they cannot detect personally meaningful relationships.

The *ImageGrouper* interface (Nakazato, Manola & Huang 2003), which was introduced in Section 2.4.2, is concerned with searching arbitrary image repositories rather than personal collections. It uses content-based retrieval techniques for searching. A workspace is introduced to detach the images used to formulate the query from the results. Recall that the query images can be grouped on the workspace to form positive and negative example groups used in the retrieval system. The emphasis, therefore, is not on result organisation but simply to support an incremental and opportunistic search strategy (cf *SketchTrieve* (Hendry & Harper 1997)). It is still mainly a *search* interface.

### 4.1.3 Summary

People create mental models to simplify reality and allow the mind to match an “adequate” solution to a given problem. The IR system designer should try to understand the user’s *actual* mental models underlying information seeking tasks, and create a system image that affords little adaptation of their actual mental models. Studies have shown that organisation and spatial arrangement of information support the thought processes of the user during information seeking activities.

An information workspace allows organisation of information. Search plans can be represented on the display by visualising elements of search activity, such as queries issued, results

obtained, search services consulted. Tasks can be solved incrementally, since the search process can be interleaved and through the visual cues the users can track their progress. The system can be used to define and discuss a user's problems, thus mediating the "conversation" between the user and the system. This process helps the searcher in task comprehension and search planning. Thus, search interfaces that are based on a workspace create "*environments for ongoing, reflective problem solving*" (Hendry 2006).

In summary, this section highlighted the benefits of supporting the user in organising information, which can be achieved by integrating a workspace into the search system. This approach was also adopted in the proposed system, *EGO*. In the following sections we will describe *EGO* in more detail and show how this approach addresses the deficiencies of traditional image retrieval systems as outlined above.

## 4.2 A Holistic View

The importance and ubiquity of multimedia data has a big influence on the working environment. It is no longer dominated by a single media, single purpose (if it ever has been?), style of work process. Users are at ease in a cross-media working environment, and therefore need a tool which is universally applicable.

Often searching for and performing a selection of images is embedded in other tasks (*e.g.*, Markkula & Sormunen 2000). Therefore, a solution to accommodate the needs of today's users must be flexible, support multiple tasks and contexts, and allow exchanges or even seamless integration with other applications used for the work tasks. In order to understand and support the user, the system has to be placed in a usage context.

### 4.2.1 Who are the Users?

We argue that the expected user profile for the proposed "retrieval in context" system is a "power user". Professional (or semi-professional) users, especially in design related areas, usually spend large amounts of time searching for or selecting images from a large collection. This results in a great opportunity for learning, adaptation and personalisation. In such environments, the search task cannot be seen as a separate entity but is at least equal in importance to understanding and capturing the workflow (*eg.* Garber & Grunes 1992, Markkula & Sormunen 2000). Only through this, can one place the search in a meaningful context and incorporate it in the overall work process.

Interviews and questionnaires during the evaluation of *EGO*, described in Chapter 6, helped to analyse typical usage patterns of regular image search system users. 24 people were interviewed, and then three types of users were identified: the hobby designer, the graphic designer and the photographer. For each "prototype user" we state the image collection they use, their typical tasks and workflow, as well as the problems they encounter.

#### The Hobby Designer

The hobby designer makes regular use of images in their profession or hobby which is not primarily related to imagery. Amongst the participants of our evaluation there was a diverse need for

image search, such as for jewellery design, Web publishing or book illustration. The image search process is primarily to seek inspiration. Most participants make regular use of an internet image search engine for this purpose. The most popular image search engine on the Web is arguably *Google Images*<sup>4-8</sup>.

**The Collection** The Web is the largest universally accessible image repository. The images found therein are of diverse quality, ranging from tiny icons to premium-quality photographs. Images on the Web are generally not annotated nor do internet search engines index visual features. Instead images can be searched by keywords based on terms extracted in the surrounding text of the image on the Web page. Due to the differing quality and topicality results are unpredictable. Yet, because of the sheer number of images being published on the Web, people do usually manage to find something of interest eventually.

**The Workflow** The search process is started by thinking about which keywords to use. This begins a trial-and-error search, in which a number of related keywords are tested to see if any relevant images are returned. The results in each stage are scrutinised carefully, exploring deep down the results list in order to find a suitable image. Once a nice image has been identified, the searcher downloads it onto his/her own computer from the Web page where the image is published.

**The Problems** The search process using an internet search engine is very tedious:

- Looking for images matching some vague idea is very difficult. Since internet search engines only support keyword search, the user is forced to think about related words that are vaguely linked to an idea, which is vague in the first place. People often feel they have to think around three corners.
- Looking for images of specific concepts is often very difficult, because search engines do not allow you to specify advanced search options. No refinements of search results are possible.
- It is almost impossible to find specific images, like finding an image which you have seen before.
- It is tedious to save search results, which requires the creation of folders on your own file system and the downloading of images one by one. Comparing new images to those already found is not supported, requiring frequent switching between your own folders and the Web browser.
- Results from previous searches are lost and it is tedious to repeat searches.

### **The Graphic Designer**

Most of a graphic designer's work involves creating, selecting or searching images. Just like the hobby designer, the professional designer mainly searches images in a foreign repository to collect

---

<sup>4-8</sup><http://images.google.com/>



ideas for a certain design. In addition, s/he has a large image base of his/her own. Surprisingly, s/he does not typically make use of specialised tools that help him/her manage or search his/her own images.

**The Collection** The graphic designer uses a myriad of image repositories, including the Web, stock image collections and his/her own images. Stock image collections, such as *Getty Images*<sup>4-9</sup> or *Corbis*<sup>4-10</sup>, are usually organised into a fixed number of concepts and subjects and are annotated by a set of controlled keywords.

**The Workflow** When working on a project, such as designing a brochure, the graphic designer uses the internet and stock image collections in the initial stages to get ideas for the design. Since all these search engines rely on textual descriptions, the designer collects terms relating to the key messages the brochure should convey. S/he then tries his/her best to find images that match his/her ideas. The main goal of the search process is to visualise ideas better. Once s/he has formed a clear idea of what images would be best suited for the brochure, s/he creates his/her own version of it, also to avoid paying royalties. This is obtained by selecting an image from his/her own image bank if s/he already has a similar image. Alternatively, s/he organises a photo shoot, where the ideas that have been collected and developed while searching and browsing the other collections will be used to take his/her own version.

### The Problems

- There is no searching facility in his/her own image bank. S/he only uses the standard file browser to categorise and keep track of the images.
- Sometimes the key ideas can be hard to put in words. Since there is no visual search facility, s/he can only resort to browsing the image collection.
- Long-term projects are difficult to keep track of. Often similar projects come round every six months to a year. S/he can reuse images from a similar project in the past only if s/he can remember such a project existed. Even if a similar project can be identified, most of the work collected during the inspirational phase, where ideas are collected and developed, is lost. The inspirational phase is difficult to retrace or reproduce, since the results of searches that lead to certain design ideas are hardly ever saved.
- Generally the designer works as a member of a team. Collaborative work is not very easy, because the images obtained during someone's explorative search phase are not typically shared. The ideas are often communicated verbally and outside their usual search or management system.

### The Photographer

The photographer is different from the designer in that s/he does not search foreign image repositories. S/he primarily needs a system to store and catalogue his/her own images.

<sup>4-9</sup>[www.gettyimages.com](http://www.gettyimages.com)

<sup>4-10</sup>[www.corbis.com](http://www.corbis.com)

**The Collection** The photographer uses almost exclusively his/her own images. They are kept organised into folders on the PC. A consistent naming scheme helps to keep track of the photographs and enables the use of the file browser's searching facility when looking for a particular image.

**The Workflow** The photographer does not typically use any specialised image management system either, but relies solely on the functionality of the operating system's file browser. The images are organised in a folder hierarchy, which makes him/her heavily dependent on an intuitive classification scheme in order to locate specific images at a later time. Most of the images are related to a certain project, and hence the categorisation scheme is often project-based. Thumbnail views are the most useful tool to browse and search the image collection. Sometimes the file browser's search tool is used to perform a search on file names to locate a specific image.

### The Problems

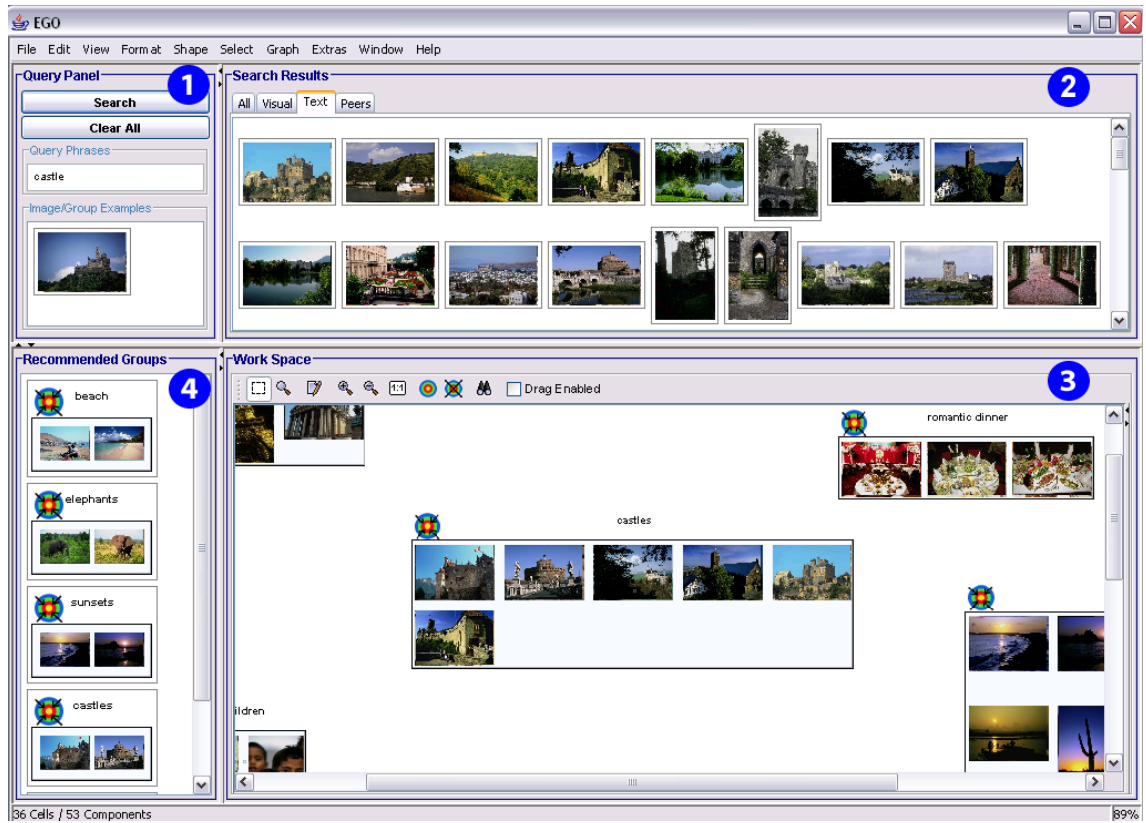
- Categorisation into folders poses problems, because the classification scheme is rigid. One classification scheme is usually not enough to facilitate every possible type of search. The photographer is forced to decide on a unique folder to put his/her photographs in, which can cause confusion at a later time: *“the reason that I can't find something is because I've put it in the wrong place”*.
- Every search essentially has to be translated into a search for a specific image. For example, if s/he wants to find images of buildings, s/he needs to remember the projects in which s/he took an image of a building. S/he can then browse to the specific project's folder to see if it contains a suitable image.
- The photographer is heavily reliant on his/her (episodic) memory for re-locating images. Sometimes photographs are not used anymore because s/he has simply forgotten about them.

We will return to these user groups in Chapter 6 and show how their problems are addressed in *EGO* and how they benefit from using *EGO* to solve their tasks.

### 4.2.2 Summary

For professional users, incorporating the context and process of the search into the system is a desirable goal. The context of a search is determined by the specific task (immediate context) and the work situation (general context). The main idea that drives the system design is to provide an environment for the day-to-day usage of the data, in which both search and organisation processes take place and are interleaved with each other.

To conclude, what is needed is a *“holistic view”* on personal image organisation and retrieval. The provision of a workspace in such a system integrates the search and organisation processes. The workspace also supports sharing, re-use and persistence. In the following sections we describe the interface of the proposed system. The description of the interface components also helps to discuss how the system would typically be used and how it adapts based on the interaction with the user.

Figure 4.1: Annotated *EGO* interface

### 4.3 The Interface

This chapter discusses the conceptual ideas behind *EGO*. Therefore, this section is limited to describing the interface of *EGO*. Some technical information about its implementation are available in Appendix B.

The interface is shown in Figure 4.1. In *EGO* the user will be involved in an organisation process, in which the user and the system interactively group images. As a starting point, the system provides a query panel (Figure 4.1, panel 1), in which Query-by-Keyword (QbK) and Query-by-Example (QbE) queries can be issued. The search results will be displayed in the panel beside it (panel 2). The user can then drag images from the results into groups on the workspace (panel 3). This constitutes the start of the interactive group creation. For the currently selected group (see Figure 4.7) the system provides recommendations of new images based on the images already contained in the group. The system's suggestions are displayed in the orange rectangle just below the selected group. The user can select recommended images to add (by dragging them into the group), and the system will update its previous suggestions. This process can iterate as long as the user is looking for more images to add to that group. We will now look more closely at the components making up the interface.

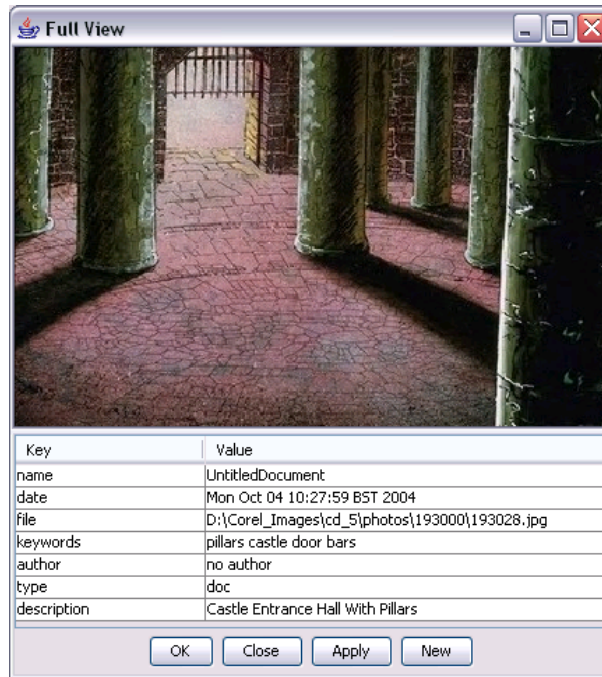


Figure 4.2: Image Viewer window

### 4.3.1 The Interface Components

The *EGO* interface comprises a query editor, results display area, workspace and image viewer. By providing these facilities, different types of requirements are catered for, enabling the user to both search and organise results effectively. In the following, the main components are discussed in more detail.

#### The Image Viewer

Double clicking on an image will open an image viewer window shown in Figure 4.2, which allows the user to look at an image at a larger scale. It also shows the properties of the image, including its title, description, keywords etc. If the user does not want to open a new window, the workspace can also be zoomed in to see more details in the image. A quick view is also shown if the mouse hovers over an image (see Figure 4.3)

#### The Search Panel

The upper half of the screen is devoted to the search facilities. It consists of the query and the results display panel (panels 1 and 2 in Figure 4.1). It should be noted that the size of all main components in the interface can be changed or even hidden on demand, since all panels are contained within split panes. In Figure 4.3 the divider between the query and workspace panels is moved down so that the results can be inspected more closely, for instance.

In the query panel (panel 1 in Figure 4.1), the user can trigger a search by choosing example query images and/or inputting some keywords. At the moment, both QbE and QbK are supported in *EGO*. The search results are displayed in the results panel beside the query construction widget (panel 2 in Figure 4.1). It allows for different views of the results based on the supported features.

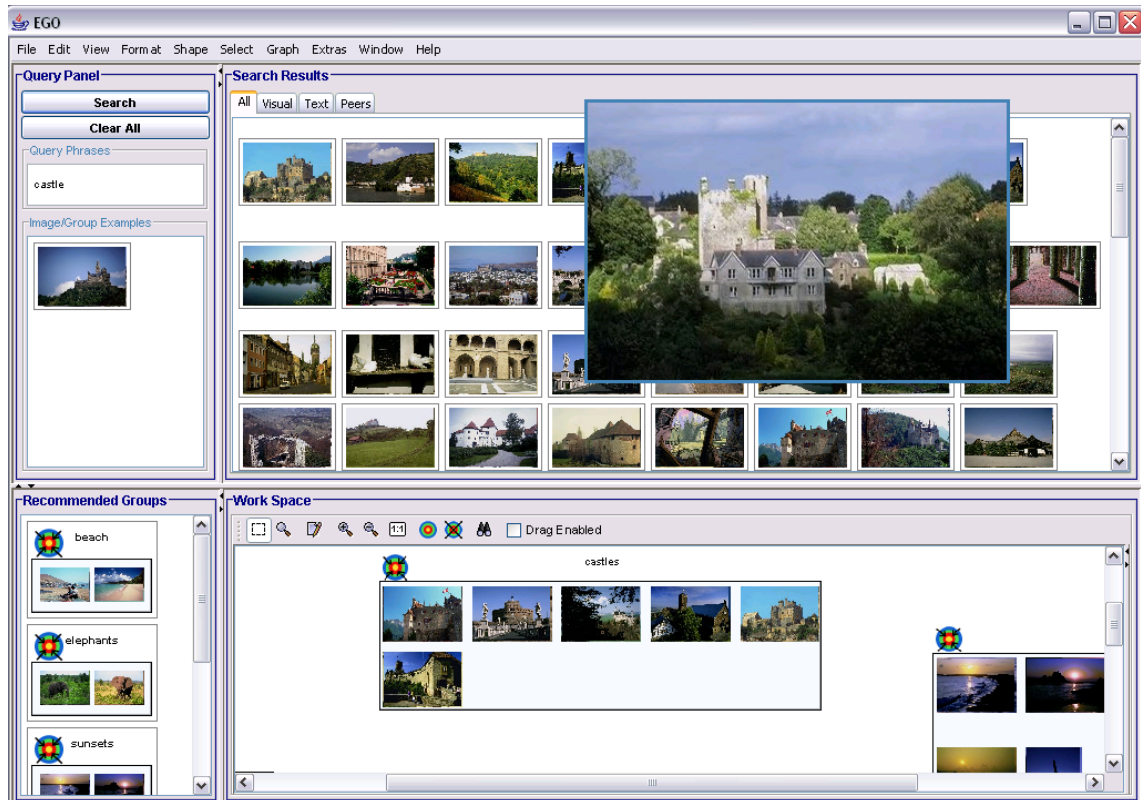


Figure 4.3: The interface where the results panel is enlarged and quick view is shown

The user can choose to view the overall results or results for only one feature category (visual, text or peers, respectively) by selecting the appropriate tab. The features are described in later chapters, in particular Chapter 7. Again, we have only implemented a linear result display, but other visualisation techniques, such as the ones mentioned in Section 2.4 or that of the Ostensive Browser discussed in Section 3.1, could be an additional enhancement of the system.

The search component provides the user with a basic query facility to search the database, which is useful for both the fulfillment of very specific information needs and serves as an entry point to the collection. From the search results the user can easily drag relevant images onto the workspace to start organising the collection.

### The Workspace

The main component of the interface is the workspace panel provided in *EGO* (panel 3 in Figure 4.1). The workspace serves as an organisation ground for the user to facilitate construction of image groups. Images can be dragged onto it from any of the other panels or imported from outside the system. The creation of groups is straight-forward as illustrated in Figure 4.4. A number of images can be selected by dragging the mouse over them (Figure 4.4(a)). When the “Group Bundle” icon from the toolbar is pressed, a new group is created that bundles the selected images together (Figure 4.4(b)). Figure 4.4(c) shows the newly created group, in which the images are automatically arranged in a grid layout to avoid clutter. Finally, the title can be edited for a group by selecting the group and pressing the “Edit” icon in the toolbar (Figure 4.4(d)). Traditional drag-and-drop techniques allow the user to drag other images into a group or to reposition the group on

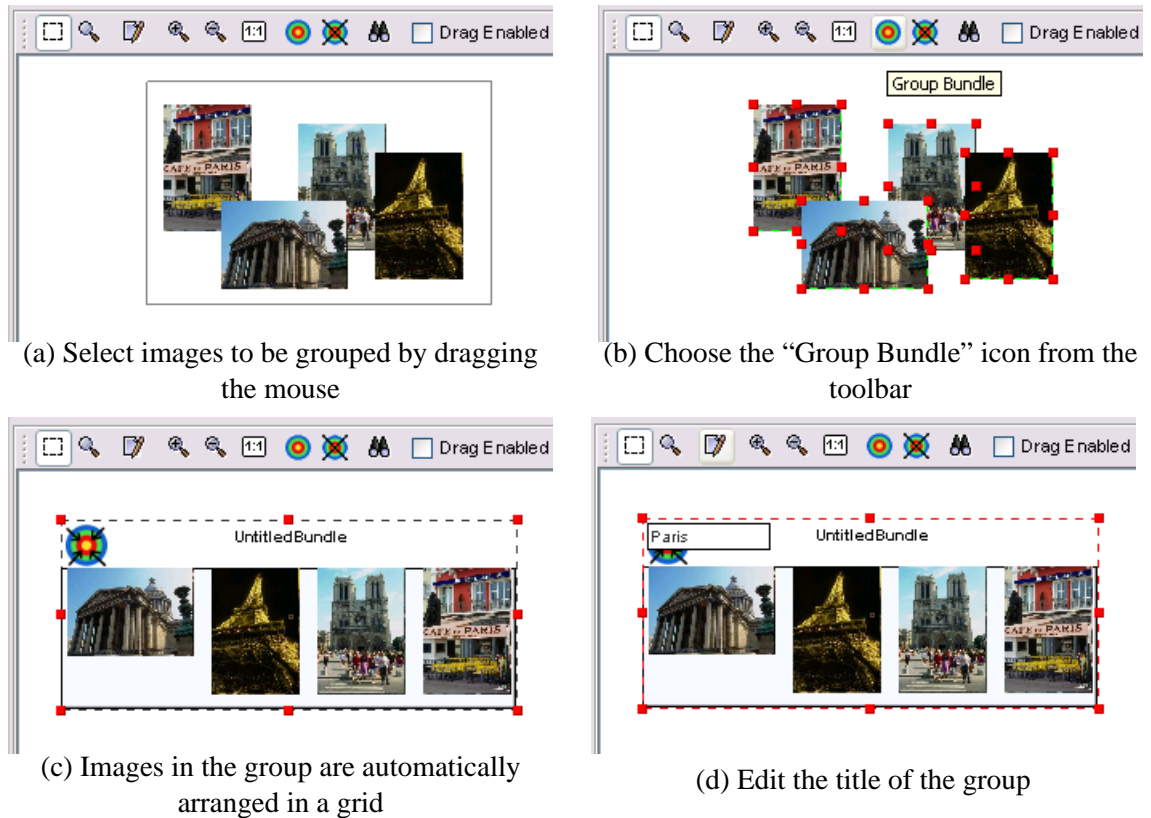


Figure 4.4: Step-by-step creation of a group on the workspace

the workspace. It should be noted that, unlike conventional file systems, an image can belong to multiple groups simultaneously. Finally, a user’s workspace including all its groupings, and their positions, can be saved and re-opened for later use.

The workspace is designed as a potentially infinitely large space to accommodate a large number of groups. Panning and zooming techniques are supported to assist navigation in a large information space. Additionally, a bird’s eye view of the workspace is available. It provides an overview, in which the whole workspace is visible, and a sense of location by marking the position of the current view with a red rectangle (Figure 4.6). Additionally, a fish-eye view may be beneficial to provide a view of the whole organisation and reduce clutter.

*EGO* includes a recommendation system that assists the user in the interactive grouping process. The recommendation system observes the user’s actions, which enables it to adapt to their information requirements and to make suggestions of potentially relevant images based on a selected group of images. An example is depicted in Figure 4.7, where the user has selected the “castles” group in the centre of the workspace. When selecting the “Search” icon (the binoculars) from the toolbar, the system will present suggestions of images. The system’s recommendations will appear as a popup below the currently selected group. The user can either accept some of the suggested images by dragging them into the current group, or simply ignore the recommendations. There are a few constraints in the recommendation system that arise from the *EGO* interface. First, an image that is already contained in the group should not be recommended again. Second, since organisation and interaction with the interface are the primary concern, the recommendations should be limited to a small number of images presented close to the location of the group on the

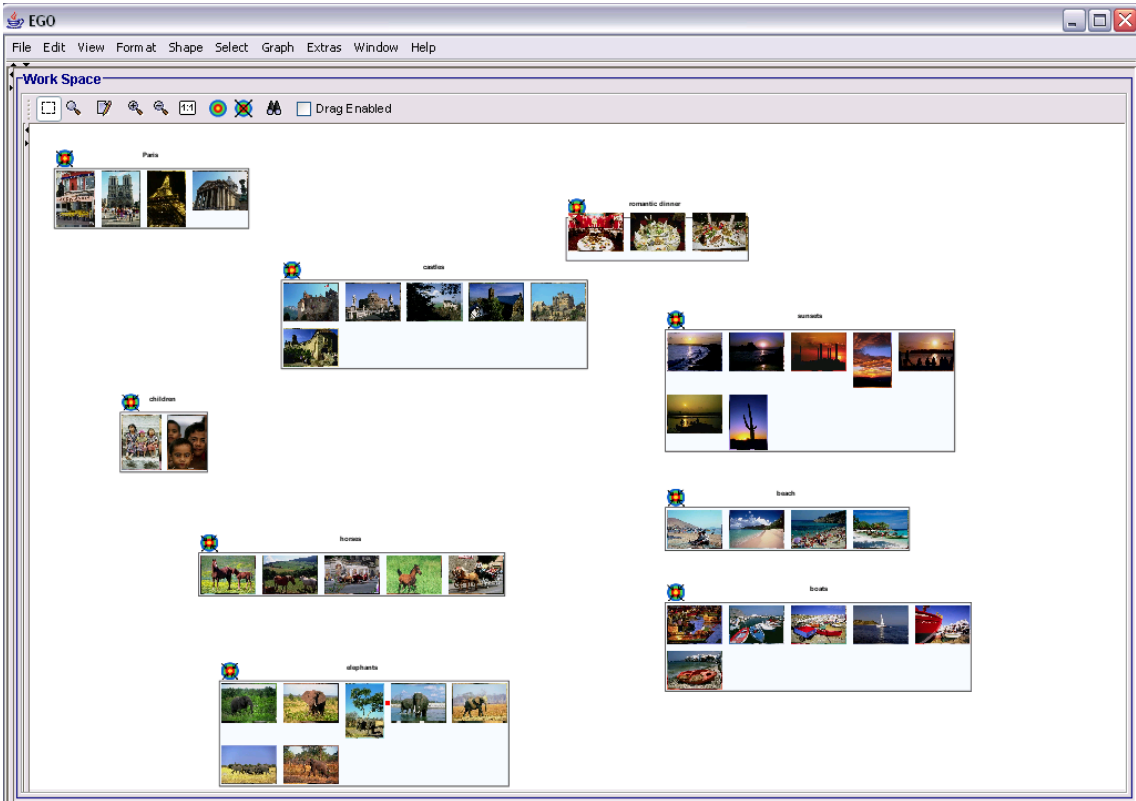


Figure 4.5: The interface where all but the workspace panel are collapsed

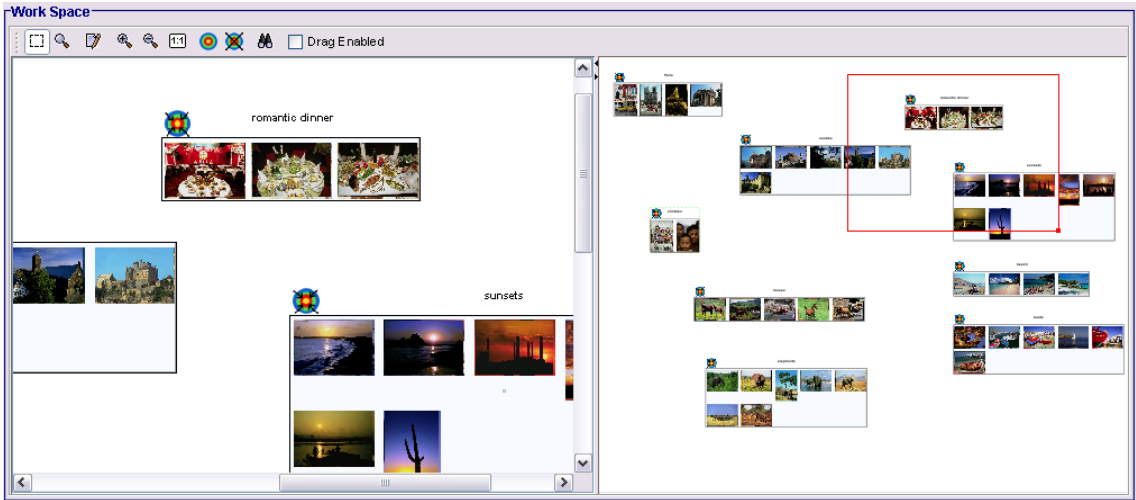


Figure 4.6: The workspace (left) and its bird-eye-view (right)

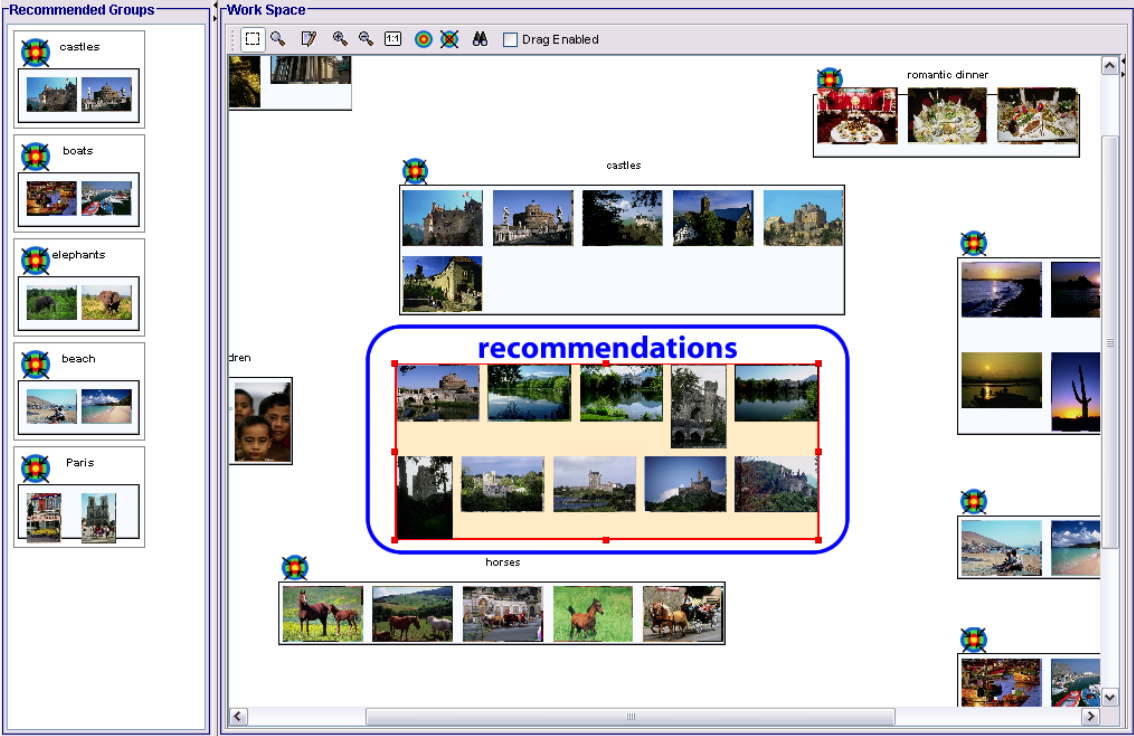


Figure 4.7: Recommendations on the workspace

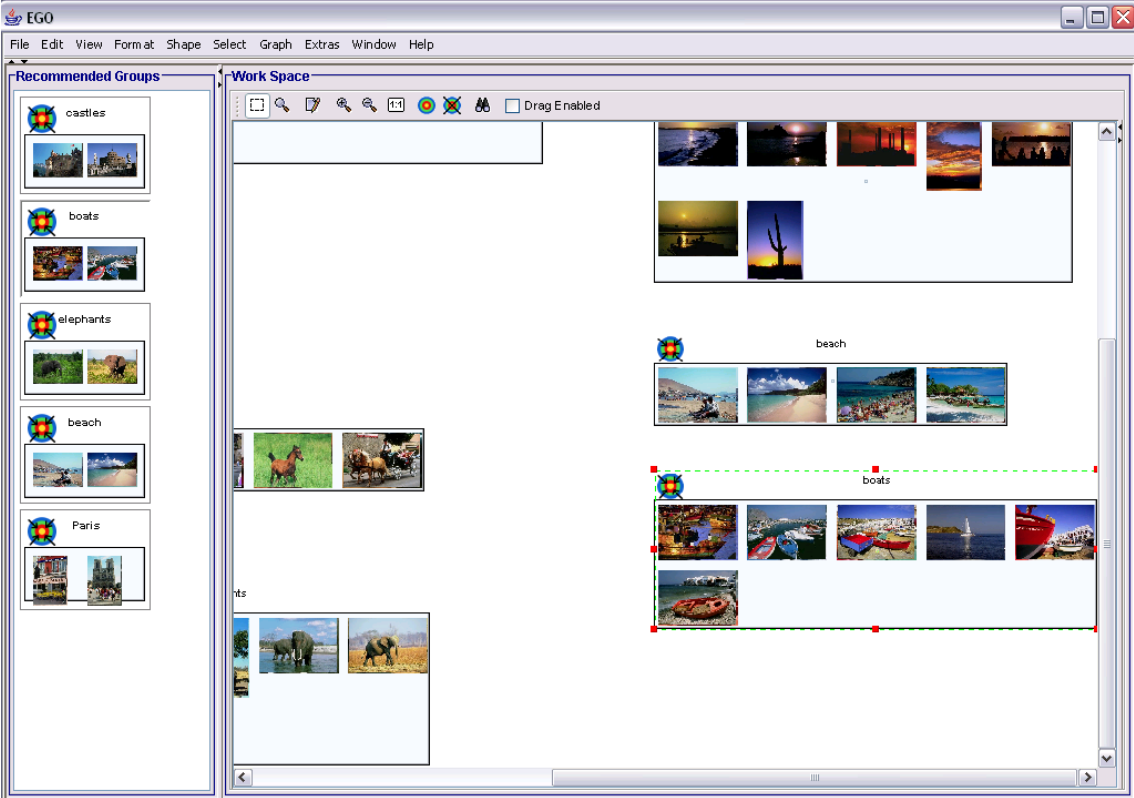


Figure 4.8: Selecting the “boats” group from the list in the recommended groups panel



workspace (see Figure 4.7). So as not to burden the user, the number of recommended images is based on the standard cognitive limits of  $7 \pm 2$  (Miller 1956). However, the results panel will also show a larger number of recommended images when recommendations are requested.

The system can adapt its recommendations based on learning the features that images in a group have in common and observing user actions and preferences over time. When new images are inserted in the group, the system updates its learning parameters in order to improve its future recommendations. Since the user ultimately decides on group memberships, the groups reflect the current semantics in the context of usage of the image collection. The recommendation system, which we will describe next in Chapter 5, is based on learning the similarities of images. The recommendation quality can further be improved by taking contextual and usage information into account to better capture the semantic information, as presented in Chapter 7.

### The Recommended Groups Panel

Finally, groups will also be retrieved as a whole by searches issued in the query panel or recommendations. They will be displayed beside the workspace (Figure 4.1, panel 4). Recommended groups are limited to the top five matching groups in the current implementation. The group is represented by the centroid of all its child images for matching purposes in the retrieval algorithm (see Section 5.1 for more details). The group results act as links to the actual groups on the workspace. When a result is selected, the corresponding group on the workspace will be highlighted and moved into view if necessary. An example is shown in Figure 4.8, where the “boats” group has been selected in the results (second from list) and the corresponding “boats” group is highlighted with a green<sup>4-11</sup> border on the workspace. The group search facility allows the user to search their own organisation. This has two benefits: first, short-term needs can be fulfilled more quickly if a matching group can be retrieved, provided that a such a group exists; second, long-term needs are better supported, because groups related to a certain task can easily be re-located and the user can continue with their work.

## 4.4 Unique Characteristics

Related workspace systems were discussed in Section 4.1.2. Here, we will take the opportunity to reiterate the main differences to these systems and highlight *EGO*'s unique characteristics.

The two previous approaches to workspace systems in text-based IR environments—*Sketch-Trieve* (Hendry & Harper 1997) and *DLITE* (Cousins et al. 1997)—have emphasised the interaction and organisation of heterogeneous *retrieval* systems. While these systems are adaptable to the user's own requirements they are not *adapting* based on the user's interaction with results. The retrieval components themselves are static. Instead we have proposed an information workspace to facilitate the organisation of *results* rather than retrieval services. The organisation of results allows an even more direct interaction with the information that is being sought.

Furthermore, *ImageGrouper* (Nakazato, Manola & Huang 2003) introduced a workspace in a general purpose image retrieval system. The workspace assists the user in grouping positive

<sup>4-11</sup>The choice of colours is customisable by the user.

and negative images to improve current search results relying on a relevance feedback classifier. It essentially serves to visualise the images selected for feedback, facilitating incremental changes of feedback examples. This is an improvement over traditional relevance feedback systems (eg *MARS* (Porkaew et al. 1999)), where previously selected examples are not visible to the user. The emphasis in *ImageGrouper*, however, does not lie in result organisation but simply in supporting an incremental and opportunistic search strategy (trial-and-error approach, cf Section 2.4.2). The system learns to improve retrieval results in order to satisfy the *current* information need, but does not adapt in the long-term. We also target general purpose image collections. Our approach enhances: image retrieval interfaces, such as *ImageGrouper*, by better supporting long-term organisation and management; and the traditional photo browser by better search support. Our goal is to assist designers in tasks that require searching and managing digital photographs.

In summary, the workspace in *EGO* provides an organisation ground for the user to interact with search results. Grouping images on the workspace serves two purposes: they facilitate task conceptualisation since organising information supports their thought processes; and they serve as query representatives and therefore alleviate the query formulation problem. The organisation is persistent, and over time *EGO* can be used as both an information browser and a personalised archive. In order to investigate these potential benefits, we performed a user evaluation of *EGO*. This study will be discussed in detail in Chapter 6, which concludes with a list of its benefits in Section 6.6.

## 4.5 Summary and Conclusion

Information retrieval has often been studied as a self-contained problem. Hence, there have been a lot of advances regarding feature representation, matching and learning from user interaction. Yet from the user's perspective, information retrieval is part of a larger process of information use or a *work task* (Ingwersen 1996). It follows that, rather than treating the retrieval system as a separate unit cut off from its environment, information access has to be considered as part of a larger work process.

Organisation of information has been found to act as a secondary notation in support of memory and information seeking. The act of grouping information is a natural means of managing information to support diverse, complex and often simultaneous tasks (Malone 1983, Rodden 1999, Grant et al. 2003). This metaphor allows the user to resort to traditional problem solving techniques, freeing them from the necessity of query formulation, which should ultimately create a natural and enhanced information seeking environment. This closes the gap between the user's actual mental model of the work process and their understanding of the system.

These considerations have motivated the design of *EGO*, a tool for personalisation and multimedia management, which has been the subject of this chapter. We have described how the user can interact with the system, and how the system adapts to the user's actions. To conclude, the design of *EGO* as a tool to create a task-specific organisation of images reflecting an individual's mental model, aims to overcome many of the problems of traditional CBIR systems.

The next chapters will deal with the implementation details of *EGO*: the way images are represented in the system and the matching and recommendation algorithms. Following this, in Chap-

ter 6 we report results from a user evaluation that was conducted in order to verify the claims made about *EGO*'s benefits.

---

## THE RECOMMENDATION SYSTEM BASED ON VISUAL FEATURES

---

This thesis is motivated by the open problems in image retrieval, which have been identified and discussed throughout this dissertation. The previous chapter provided a description of the *EGO* interface as a conceptual framework for organising images, and how the problems of traditional image retrieval systems are addressed in this interface. In this chapter, we are concerned with one particular aspect of the *EGO* system: how can we best retrieve relevant images based on a given group of images. To this end, we investigate relevance feedback techniques to achieve group-based learning from visual features. Specifically, we use a form of query expansion to learn a new query representation for a group similar to the technique employed by Porkaew et al. (1999). This involves determining multiple query points forming the representation of a group. To compute the overall results of a multi-point query, an evidence combination scheme is required that merges the individual lists returned from the query representatives. Evidence combination is an intricate topic (Lee 1997), and we have performed a quantitative comparison of three different fusion schemes for the task at hand. A summary of these results was published in (Urban & Jose 2004b).

In the following Sections 5.1 and 5.2 we present techniques that provide the underlying implementation of the proposed framework. Section 5.3 supplies the experimental details used to perform a simulated user-evaluation of the proposed fusion strategies for multi-point queries. The results and implications of this study are discussed in the remaining sections.

### 5.1 Background and Related Work

This section details the specific algorithms used to build the recommendation system in *EGO*. The relevant literature includes an introduction to the hierarchical image representation model, a description of the relevance feedback technique employed and an introduction to the multi-point query approach. All necessary notation and techniques used later are described. Table 5.1 lists some of the notations used in this chapter.

Table 5.1: Notations

Notation	Description
$I$	number of features
$K_i$	feature dimension of feature $i$ ( $1 \leq i \leq I$ )
$\vec{x}_i$	$i$ -th feature vector of database object $x$
$\vec{q}_i$	$i$ -th feature vector of query representation $q$
$g_i(q, x)$	$i$ -th feature distance between database object $x$ and query representation $q$
$d(q, x)$	overall feature distance between database object $x$ and query representation $q$
$W_i$	feature transformation matrix of feature $i$ (intra-feature weights)
$\vec{u}$	feature weight vector (inter-feature weights)
$N$	total number of items in collection
$M$	number of items in a specific group
$k$	number of recommendations
$c$	cutoff value (number of items in the individual lists, length of individual lists)
$L$	number of lists to combine

### 5.1.1 Image Representation

A hierarchical object model for image representation is proposed by Rui et al. (1998). In this model an image is represented by a set of feature vectors, one for each distinct feature implemented. The distance between an object  $x$  in the database and a given query representation  $q$  is computed in two steps: calculating the distances according to the individual features; and calculating the overall distance as a linear combination of the individual distances.

First, the individual feature distances,  $g_i$  (for  $i$  in  $1..I$ , where  $I$  is the number of features), are computed by the generalised Euclidean distance:

$$g_i(q, x) = (\vec{q}_i - \vec{x}_i)^T W_i (\vec{q}_i - \vec{x}_i) \quad (5.1)$$

where  $\vec{q}_i$  and  $\vec{x}_i$  are the  $i$ -th feature vectors of the query  $q$  and the database object  $x$  respectively, and  $W_i$  the *feature transformation matrix* used for weighting the feature components.  $W_i$  is a  $K_i \times K_i$  real symmetric full matrix, where  $K_i$  is the  $i$ -th feature dimension. The intra-component weights in  $W_i$  are estimated on a per query basis, based on the images obtained through relevance feedback, as detailed in the next section.

The second step is then to combine the individual distances to arrive at a single distance value  $d$ . This is achieved by a linear combination between  $\vec{g}(q, x) = [g_1(q, x), \dots, g_I(q, x)]^T$  and a feature weight vector  $\vec{u}$ :

$$d(q, x) = \vec{u}^T \vec{g}(q, x) \quad (5.2)$$

The overall feature weights,  $\vec{u}$ , are again estimated from relevance feedback (see below). We use the same notation as Rui & Huang (2000), since we employ their proposed method of determining the optimal feature weights given a number of training samples. This approach will be outlined in

the following sections.

### 5.1.2 Relevance Feedback by Learning a Transformed Feature Space

In interactive CBIR systems, relevance feedback is used to improve the system's matching function based on experience gained from the user's feedback. Section 2.3 reviews representative techniques in this area (Cox et al. 2000, Ishikawa et al. 1998, Meilhac & Nastar 1999, Minka & Picard 1996, Peng et al. 1999, Porkaew et al. 1999, Rui et al. 1998, Rui & Huang 2000, Santini & Jain 2000, Su & Zhang 2002, Tieu & Viola 2000, Tong & Chang 2001, Vasconcelos & Lippman 2000, Wood et al. 1998, Zhou & Huang 2003). After studying the various techniques, we selected one main approach suitable for the implementation of *EGO*'s recommendation system. This approach is based on the geometric interpretation discussed in Section 2.3.2 and includes both Query Shifting and Feature Re-Weighting. For feature re-weighting, the parameters of the matching function are continuously updated in order to adjust to the training samples. Many relevance feedback techniques achieve this by determining an optimal feature transformation matrix  $W_i$  used in the calculation of the feature distance (see Equation 5.1) (Rui & Huang 2000, Nakazato, Dagli & Huang 2003).

Rui & Huang (2000) present an optimised framework for calculating the transformation matrix, when only positive feedback is considered. Due to the hierarchical object model, it distinguishes between component and feature weights. Three steps of computation are required to determine the optimal feature weights regarding a number of training samples. First, we need to choose an optimal query vector  $\vec{q}$  to represent the training samples. Second, the intra-component weights,  $W_i$ , are computed for each feature  $i$ . Finally, we can find the inter-feature weights,  $\vec{u}$ , that best capture the feature inter-similarity between the training samples.

The *optimal query vector*  $\vec{q}_i$  (for the  $i$ -th feature) is calculated as the weighted centroid of the  $P$  positive examples specified by the user:

$$\vec{q}_i^T = \frac{\vec{\pi}^T X_i}{\sum_{p=1}^P \pi_p} \quad (5.3)$$

where  $\vec{q}_i$  is the query vector for feature  $i$ ,  $X_i = [\vec{x}_{i1} \dots \vec{x}_{iP}]$  the matrix, whose columns are the  $P$  positive example vectors according to the  $i$ -th feature, and  $\vec{\pi} = [\pi_1 \dots \pi_P]$  the degrees of relevance for each positive example. In our setting, we have constant relevance values ( $\forall i, j \in [1, P] : \pi_i = \pi_j$ ), since each image in the group is equally relevant.

The *optimal feature component weights* are given by the feature space transformation matrix,  $W_i$ , which can be calculated as:

$$W_i = \det(C_i)^{\frac{1}{k_i}} C_i^{-1} \quad (5.4)$$

where  $C_i = \sum_{p=1}^P \pi_p (\vec{x}_{ip} - \bar{x}_i)(\vec{x}_{ip} - \bar{x}_i)^T$  is the *weighted covariance matrix* of the  $P$  positive examples and  $\bar{x}_i$  is the mean vector of the positive examples.  $W_i$  takes the form of a full matrix, if  $P$  is larger than the dimensionality of the  $i$ -th feature, otherwise only the diagonal entries are considered.

Finally, the *optimal feature weights*,  $\vec{u} = [u_1, \dots, u_I]$ , are solved by:

$$u_i = \sum_{j=1}^I \sqrt{\frac{f_j}{f_i}} \quad (5.5)$$

where  $f_i = \sum_{p=1}^P \pi_p g_i(q, x_p)$ . The total distance of a database image to the average query vector,  $\vec{q}$ , is then computed by Equations (5.2) and (5.1), using the optimal feature weights,  $\vec{u}$ , and optimal feature component weights,  $W_i$ .

When negative feedback is available, it is possible to view the learning as a classification problem as is done in the *ImageGrouper* system (Nakazato, Dagli & Huang 2003) (cf Section 2.4.2). In this system group-based learning strategies based on discriminant analysis are studied. *ImageGrouper* allows for an interactive query formulation process by the creation of positive and negative groups by the user. The learning objective is formulated as an  $(x+y)$ -class problem, meaning that the system learns to discriminate between multiple positive and multiple negative classes. The learning system's goal is to return only images belonging to any of the positive classes and none of the negative classes. The  $(x+y)$ -class formulation leads to the calculation of a feature transformation matrix based on *Group-Biased Discriminant Analysis* (GBDA) (Nakazato, Dagli & Huang 2003). GBDA is an extension of Fisher's linear discriminant analysis (Duda et al. 2001). However, this method relies on a large number of feedback samples, both negative and positive, to work reliably. In particular, the learning algorithm implicitly assumes that the samples are consistent in the visual feature space. Yet we expect groups to reflect semantic concepts rather than visual coherence. Consequently, we have deemed the Group-Biased Discriminant Analysis not applicable for our purposes.

### 5.1.3 Multi-Point Queries

In the *MARS* system (Porkaew et al. 1999) (cf Section 2.4.2) a query expansion scheme is proposed for CBIR applications, in which a multi-point query is constructed from the cluster representatives obtained from clustering the relevant images. The choice of multiple query points rather than a single representation is motivated by the semantic gap between the user's high-level perception and the low-level feature representation. Due to this gap, the relevant images might not necessarily be close in the feature space but rather they may form multiple disjoint clusters. The overall similarity score to a multi-point query is defined as the weighted sum of the scores calculated from issuing each cluster representative as a query, where the weight of a query point is proportional to the cluster size. It is shown by Porkaew et al. that this query expansion scheme performs better than using a single average query vector of all feedback samples in combination with learnt feature weights.

Porkaew et al. have not studied the combination strategy of multi-point queries extensively. We will address this issue by comparing the simple linear combination with both score and rank-based combination strategies. These are based on ranked list aggregation methods used in the Web retrieval domain and the Dempster-Shafer Theory of Evidence Combination. The details of the various combination techniques are discussed in the following section.

## 5.2 Group-Based Query Learning

The proposed group-based learning scheme involves: (1) updating the system's matching parameters; (2) creating the multi-point query representation and computing a ranked list for each query point based on the learnt parameters; and (3) combining the individual result lists for the new recommendations.

The parameter adaptation is achieved by finding a feature space transformation matrix according to the scheme described in Section 5.1.2 and proposed by Rui & Huang (2000). The creation of multi-point queries for each group follows, whereby each query point represents one cluster of visually similar images in the group. The clusters are computed by an agglomerative hierarchical clustering algorithm, using Ward's minimum variance criterion (Theodoridis & Koutroumbas 1999). The ideal number of clusters is automatically estimated using the method proposed by Salvador & Chan (2003). The query points are the cluster centroids.

When issuing the multi-point query to the system, a separate result list will be returned for each cluster representative, which need to be combined. We outline three combination strategies below.

### 5.2.1 Evidence Combination for Multi-point Queries

Each of the combination strategies discussed below has their own terminology for describing the sources that produce the individual ranked lists. In the group-based recommendation scenario, the sources are the query points that form the multi-point query. These are determined as the cluster representatives that result from clustering all the images in a group. The following terms are thus used interchangeably to refer to the entity that produces one individual ranked list: query point, cluster representative, information source, voter, ranker.

#### Query Expansion

In *MARS*, the overall similarity score to a multi-point query is defined as the weighted sum of the scores with respect to each query representative, where the weight of a query point is proportional to the cluster size. The *Query Expansion scheme (QEX)* studied here uses a simple linear combination as in *MARS*.

#### Voting Approach

The combination problem for multi-point queries parallels with the ranked list aggregation problem in the Web retrieval domain (Fagin et al. 2003, Dwork et al. 2001). It is exemplified in meta-search engines, whose task is to synthesise single orderings of Web pages returned by the individual search engines into an optimal aggregated ranked list. Since the scores attributed to the results from different search engines are not easily comparable, the combination is rank rather than score-based. In this interpretation, the search engines can be seen as the "judges" or "voters" with their own preferences of candidates. The task of the meta-search engine is then to find a maximum "consensus" ranking.



Inspired by this idea, we also consider an aggregation method purely based on ranks. In the *voting approach* (VA) each query representative is treated as a voter producing its own individual ordering of candidates (images). The final combined list is computed based on the *median rank aggregation* method proposed by Fagin et al. (2003). This method is shown to be a reasonable heuristic for the rank aggregation problem based on combining partial lists (ie top  $c$  lists that do not contain all database objects). It assumes a number of independent voters that rank a collection based on the similarity to a query. The aggregation rule then sorts the database objects with respect to the median of the ranks they receive from the voters.

The proposed algorithm MEDRANK is efficient and database friendly. The idea can be sketched as follows. Assume each voter produces a ranked list. From each list, access one element at a time, until a candidate is encountered in the majority of the lists, place this candidate as the top ranked of the final list. The second candidate will be placed second top, and so on. Continue until top  $k$  candidates are found, or there are no more candidates. If less than  $k$  candidates can be found that appear in more than half the lists, simply append the remaining candidates (sorted according to their partial median rank).

### Dempster-Shafer Combination

The Dempster-Shafer (DS) Theory of Evidence Combination is a powerful framework for the combination of results from various information sources, and has been extensively studied for IR purposes (Jose & Harper 1997). We have already employed this technique in the Ostensive Browser approach introduced before (see Section 3.1.4). In this chapter we investigate two variants: a score and a rank-based combination.

There are three steps required for calculating the final scores of items in the merged results. First, each information source (query point) is assigned an un-trust coefficient,  $\beta_j$  ( $0 \geq \beta_j \geq 1$ ), which represents the uncertainty of the source of evidence. Initially, we use constant un-trust coefficients, ie  $\beta_j = 1/L$ , where  $L$  is the number of information sources (lists).

Second, we calculate the mass function for document  $d_i$  of information source  $j$ :

$$m_j(\{d_i\}) = S_{ij} \times (1 - \beta_j) \quad (5.6)$$

where  $S_{ij}$  is the initial score of  $d_i$  from information source  $j$ . We have determined the score in two ways in this evaluation: (1) the distance from the cluster representative is used directly (score-based); and (2) the score reflects the rank in the list from information source  $j$  (rank-based). This leads to the following two formulae. The score-based  $S_{ij}^s$  is calculated as:

$$S_{ij}^s = \frac{d(q_{c_j}, d_i)}{\sum_{i=1}^c d(q_{c_j}, d_i)} \quad (5.7)$$

where  $q_{c_j}$  is the query representation (cluster representative of the  $j$ -th cluster) that produced the ranking and  $d_i$  the  $i$ -th document's representation. The denominator acts as normalisation factor, so that the scores sum to 1, where  $c$  is the number of items in the individual lists. While the

rank-based  $S_{ij}^r$  is determined by:

$$S_{ij}^r = \frac{c - (r_{ij} - 1)}{\sum_{i=1}^c i} \quad (5.8)$$

where  $r_{ij}$  is the rank of  $d_i$  in the list produced by the information source (query point)  $j$ , and again normalised by the denominator.

Finally, the results from different information sources are combined by applying the Dempster-Shafer Theory of Evidence Combination as follows. For each two information sources 1 and 2, the new mass function of document  $d_i$  is given by:

$$\begin{aligned} m'(\{d_i\}) &= m_1(\{d_i\}) \otimes m_2(\{d_i\}) \\ &= m_1(\{d_i\}) \times m_2(\{d_i\}) + m_1(\{d_i\}) \times m_2(\{\Theta\}) + m_2(\{d_i\}) \times m_1(\{\Theta\}) \end{aligned} \quad (5.9)$$

where  $\Theta$  denotes the global set of documents and  $m_j(\{\Theta\})$  is the un-trust coefficient of information source  $j$  (initially set to  $\beta_j$ ). The new un-trust coefficient  $m'(\{\Theta\})$  of the combination is obtained by

$$m'(\{\Theta\}) = m_1(\{\Theta\}) \times m_2(\{\Theta\}) \quad (5.10)$$

Any new set of results, from a third information source, can be folded in by reusing Equations (5.9) and (5.10).

Having introduced the algorithms underlying the recommendation systems and detailed possible evidence combination strategies, we will now present the experimental setup followed by a detailed analysis of their performance under a variety of settings.

### 5.3 Experimental Setup

Experiments are conducted on the Corel image collection (COREL n.d.). We use a reasonable subset, photo CD 7 of the Corel collection, containing 23,796 images. Due to the collection's difficulty for CBIR systems (and also limitations in computational resources) it is common practice to select a reasonable-sized subset for evaluation (Müller et al. 2002). The total number of images used for this evaluation is similar to, if not above, the limits of current CBIR system evaluation, for instance a similar number is used by Kim & Chung (2003), whereas a significantly smaller subset is used by Nakazato, Dagli & Huang (2003).

Domain experts have organised the collection (photo CD 7) into 238 categories of ca. 100 images each, which reflect high-level semantic concepts. The results are based on 10 query categories selected for the evaluation ("aviation", "bob sledding", "flags", "minerals", "roses", "rock formations", "stamps", "tribal people", "volcano", "dolphins"). We use the category information as ground-truth, that is, images from the same category as the images in the query group are considered relevant. Note that many CBIR system evaluations are based on hand-selected categories rather than the predefined ones from Corel (eg, Rummukainen et al. 2003). The hand-selected categories contain visually more similar images from the whole collection, which makes it easier for the retrieval system to learn the shared concepts. We have chosen to adhere to the Corel categories for two major reasons: first, the ground-truth from Corel is easily available for everyone,

facilitating a comparison of our results to other approaches; second, the categories reflect realistic high-level semantic concepts that our *EGO* system should be able to deal with. Images in one group might not all be visually similar to each other (eg a group for “tribal people” might contain close-up shots of people and scenery shots of their habitat, see Figure 5.2(b)), and not every image that is visually similar to other images in the group belongs to it (eg a shot of a person wearing lots of make-up during carnival will not necessarily belong to the “tribal people” group). The choice of ground-truth categories complicates the retrieval system, but also leads to interesting and realistic results, which will be analysed in Section 5.4.

From the ground-truth, we have constructed queries by randomly selecting a number of starting images from a given category. The number of starting images is varied in the runs below. Since this evaluation is meant to test the recommendation algorithm of *EGO*, we mainly refer to a query as a “group”. Hence the group size is equivalent to the number of query images. The results are based on 50 queries (or 50 distinct groups of a given size) for each category, resulting in a total of 500 queries. All results are presented as the average over all categories (unless otherwise stated).

### 5.3.1 Features

We use the following six low-level colour, texture and shape features (feature dimension in brackets):

**Colour** Average RGB (3), Colour Moments (9) (Stricker & Orengo 1995)

**Texture** Co-occurrence (20), Autocorrelation (25) and Edge Frequency (25) (Sonka et al. 1998, Sharma et al. 2001)

**Shape** Invariant Moments (7) (Hu 1962)

These features are described in Appendix C. They were chosen to construct a rapid initial prototype implemented purely in Java because they were readily available in Java. The recommendation system, however, does not rely on this specific set of features. In fact, future improvements should incorporate the descriptors proposed for the MPEG-7 standard<sup>5-1</sup>, since those features have proven successful for a variety of retrieval tasks. This would also allow better comparison of retrieval techniques.

### 5.3.2 The Techniques

The evaluation presents a comparison of the three fusion strategies QEX, VA and DS for multi-point queries described in Section 5.2.1.  $DS_r$  and  $DS_s$  refer to the rank-based and score based combination, respectively. The fusion strategies are based on combining the top  $c$  results (list length or cutoff value) from each query point, and return the overall top  $k$  results from the combined list. Throughout the evaluation  $k$  is set to 10. All of these techniques are compared to a single *average query point* AVG (Rui & Huang 2000) as baseline.

<sup>5-1</sup><http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

### 5.3.3 Performance Measures

Recall from Section 2.5.1 that the traditional measures in Information Retrieval are *precision* and *recall* (van Rijsbergen 1979) defined as:

$$Precision = \frac{\# \text{ relevant images retrieved}}{\# \text{ retrieved images}} \quad (5.11)$$

$$Recall = \frac{\# \text{ relevant images retrieved}}{\# \text{ relevant images in the database}} \quad (5.12)$$

In our application, only the top  $k$  (where  $k \ll N$ ) images are ranked and returned to the user. Also, the query images will not reappear in this ranking, because they are already contained in the group. Thus, the traditional precision versus recall curve is not applicable for an evaluation in this scenario. We are primarily concerned with the quality of the recommendations, that is how many of the  $k$  returned images are relevant. The precision after the  $k$ -th image retrieved,  $P(k)$ , provides a good indication for this, since it measures its composition of relevant and non-relevant images. The recall value measures how many of the total available relevant images are returned. Over one iteration, the recall in the recommendations is not of primary importance, since the recommendations are limited to a very small number ( $k = 10$ ). The total number of relevant images found only becomes an important performance measure when running the recommendation system over a number of feedback iterations. Therefore, we present the  $P(k)$  performance for each run allowing evaluation of the settings of the combination methods without using feedback iterations.  $P(k)$  values are in the range  $[0, 1]$ , corresponding to 0-100% precision. The recall performance, in terms of the total number of relevant images found, is presented for the final run allowing evaluation of the performance over a number of feedback iterations.

## 5.4 Results Analysis

We have considered four axes of variation that can affect the performance of the mergers. First, we have tested the effect of the group size on the merging performance. Second, we have introduced a weighting mechanism for weighting the contributions of the individual lists and compared it to the non-weighted combination. Third, the effect of the list cutoff value,  $c$ , that determines the length of the individual lists is studied by varying it from  $k$  (number of recommendations wanted) to 1000. Finally, we report the results of a pseudo-relevance feedback run, where all of the three parameters above are fixed. It is meant to test the performance of the recommendation system “in action” using either of the studied fusion methods.

### 5.4.1 Testing the Parameters

#### Variations of Group Size

The objective of the first run in the experiment is to evaluate the effect the group size (number of query images) has on the recommendation system using either of the proposed merging strategies. The group size is varied from 5 to 50 in steps of 5. Note that the total number of relevant images for one group, that is the maximum group size, is 100. The cutoff value,  $c$ , is set to 100.

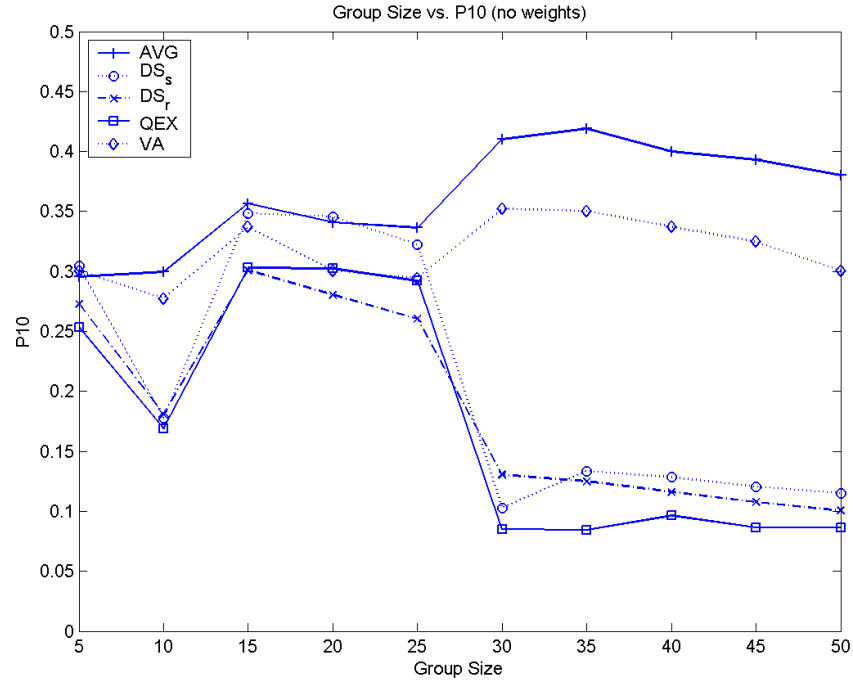


Figure 5.1: P(10) for various group sizes (average over all categories)

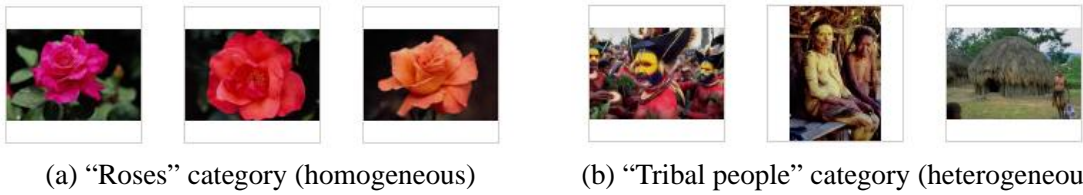


Figure 5.2: Images from the roses category and the aviation category

The graphs in Figure 5.1 reflect a “scissor trend”, where all methods start at approximately equal performance for a group size of 5, but then dramatically diverge from a group size of 25. AVG and VA tend to increase performance with growing group size, while all other multi-point query methods worsen considerably for larger groups.

Analysing the individual categories, we could identify two classes, namely *homogeneous* and *heterogeneous* categories. Homogeneous categories contain visually similar images and are well distinguishable from other categories (eg “roses”), while heterogeneous categories contain visually less similar images and/or are not easily distinguishable from other categories (eg “tribal people”). Our sample categories contained 5 of each. Figure 5.2 displays example images in the “roses” and “tribal people” category.

Figures 5.3(a) and (b) show the results for the homogeneous and heterogeneous categories, respectively. Here we can see that AVG is best suited for homogeneous categories, where one can assume an “ideal” query representation to describe it. In these categories it generally outperforms the multi-point query approaches, because it is successful in learning the “ideal” query representation with a growing number of training samples. In heterogeneous categories, which are not necessarily described best by a single representation, the multi-point queries succeed in a slight in-

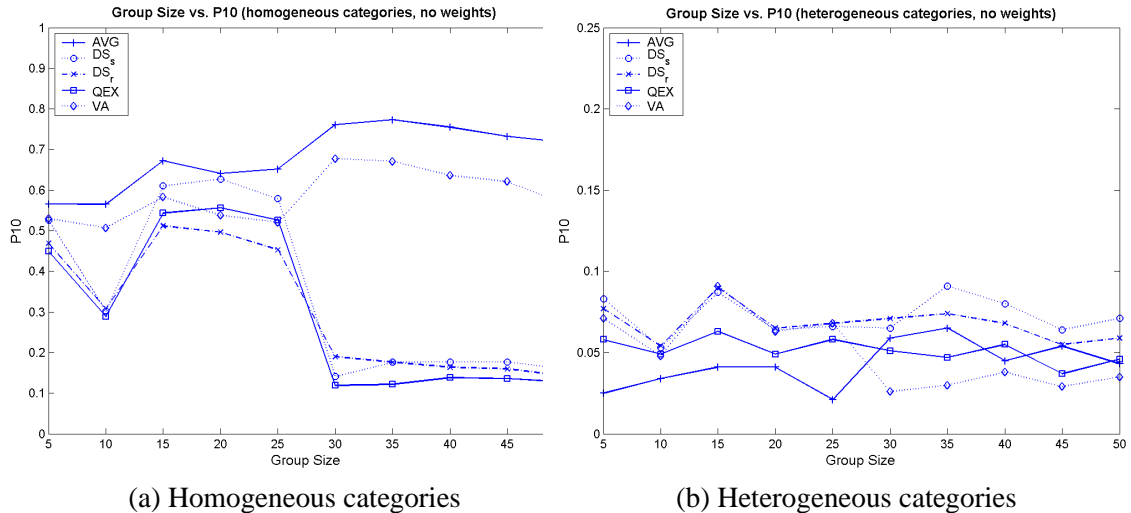


Figure 5.3: P(10) for various group sizes (average over homo- and heterogeneous categories)

crease in performance. Again, from a sufficiently large group size (25+), the single representation is smoothed enough so that it can keep up with the multi-point approaches.

The decline in performance for group sizes larger than 25 of the multi-point queries relying on adding the scores or ranks from the individual lists, QEX and DS, is probably due to the introduction of too much noise. The list aggregation in both these approaches in essence averages the scores (ranks) of the candidate images in all individual lists. Combining a larger number of lists tends to lead to a smoothing of the candidates' scores, unless the individual scores (or orderings) in the various lists are very similar. VA, on the other hand, is less sensitive to this kind of noise, because it simply counts the number of occurrences of the candidates rather than averaging their ranks or scores.

The large jump in performance at group size 25 has led us to an investigation into what influence the cluster algorithm might have on this. Figure 5.4 plots the number of clusters and the cluster size, respectively, versus the overall group size. The two graphs show a sudden jump at 25. The average number of clusters increases steeply, while the average cluster size for 30 query images even falls below that of 25. Again, this strengthens our hypothesis of increased noise introduction at the critical query size of 30. The reason for this sudden jump must lie in the method of determining the optimal number of clusters in the hierarchical clustering method (Salvador & Chan 2003). A thorough investigation into the influence of the cluster algorithm might yield more insights here.

### Introduction of Cluster/List Weights

In the next run we have introduced a weighting scheme for the multi-point queries similar to the one proposed in *MARS* (Porkaew et al. 1999). Each query point is associated with a weight proportional to the cluster size it represents, ie  $w_i = \frac{m_i}{M}$ , where  $m_i$  is the number of images in cluster  $i$  and  $M$  the total number of images in the group.

In VA, the weights influence the ranking in two ways. First, the lists are sorted in descending order of their weights, as this algorithm is sensitive to the sequence in which they are processed in

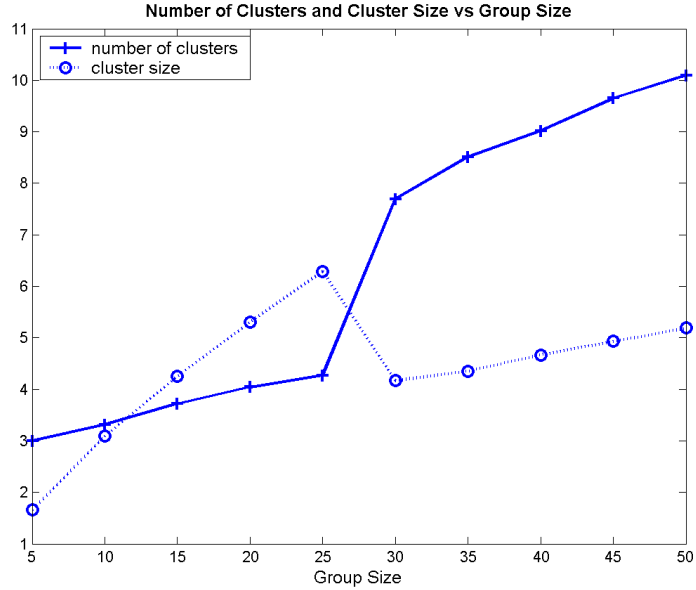


Figure 5.4: Number of clusters and cluster size vs group size

Table 5.2: Average P(10) for weighted and non-weighted variants

	QEX	DS <sub>s</sub>	DS <sub>r</sub>	VA	AVG
weighted	0.1588	0.1749	<b>0.2111</b>	<b>0.3448</b>	0.3631
non-weighted	<b>0.1759</b>	<b>0.2099</b>	0.1875	0.3175	

the merging process. Second, the scores each list gives its candidates will be weighted. Formally, to incorporate the query-point weights,  $w_i$ , each list,  $l_i$  (where  $1 \leq i \leq L$  and  $L$  the number of voters), is able to score its candidates by its weight. The overall score of a candidate  $x$ ,  $s(x)$ , is accumulated:  $s(x) = \sum_{i=1}^L w_i$ , where  $l \leq L$ . The majority criterion from above, which states that a candidate is carried forward to the final list if it is seen in more than half of the lists, is fulfilled if  $s(x) > 0.5$  (this candidate is seen in the weighted majority of lists)<sup>5-2</sup>.

In all the other methods, the inverse of weights are used, since the lists are sorted by distance or rank values directly, ie the smaller the weight the smaller the increase in distance values. Thus  $w'_j = \frac{1/w_j}{\sum_{i=1}^L w'_i}$ . In QEX, these weights are used to combine the weighted linear combination of distance scores from the individual lists, ie  $s(x) = \sum_{j=1}^L w'_j s_j(x)$ , where  $s(x)$  is  $x$ 's overall score and  $s_j(x)$  its score in the  $j$ -th list. In DS, the weights are used to derive the un-trust coefficient, ie  $\beta_j = (1 - w'_j)$ .

The graphs in Figure 5.5 contrast the weighted and non-weighted performance for each individual method. The two rank-based methods, VA and DS<sub>r</sub> perform slightly better if list weights are introduced in the merging process. The average P(10) values increase by ca. 2-3% points for both VA and DS<sub>r</sub>, as can be inferred from Table 5.2. However, the score-based methods QEX and DS<sub>s</sub> cannot benefit from the weighting. On the contrary, their performance drops by 2% and 3.5% points, respectively, when the individual list contributions are weighted to arrive at the final score.

<sup>5-2</sup>The majority criterion is a parameter in the algorithm and can be set to different values to adjust to the application domain.

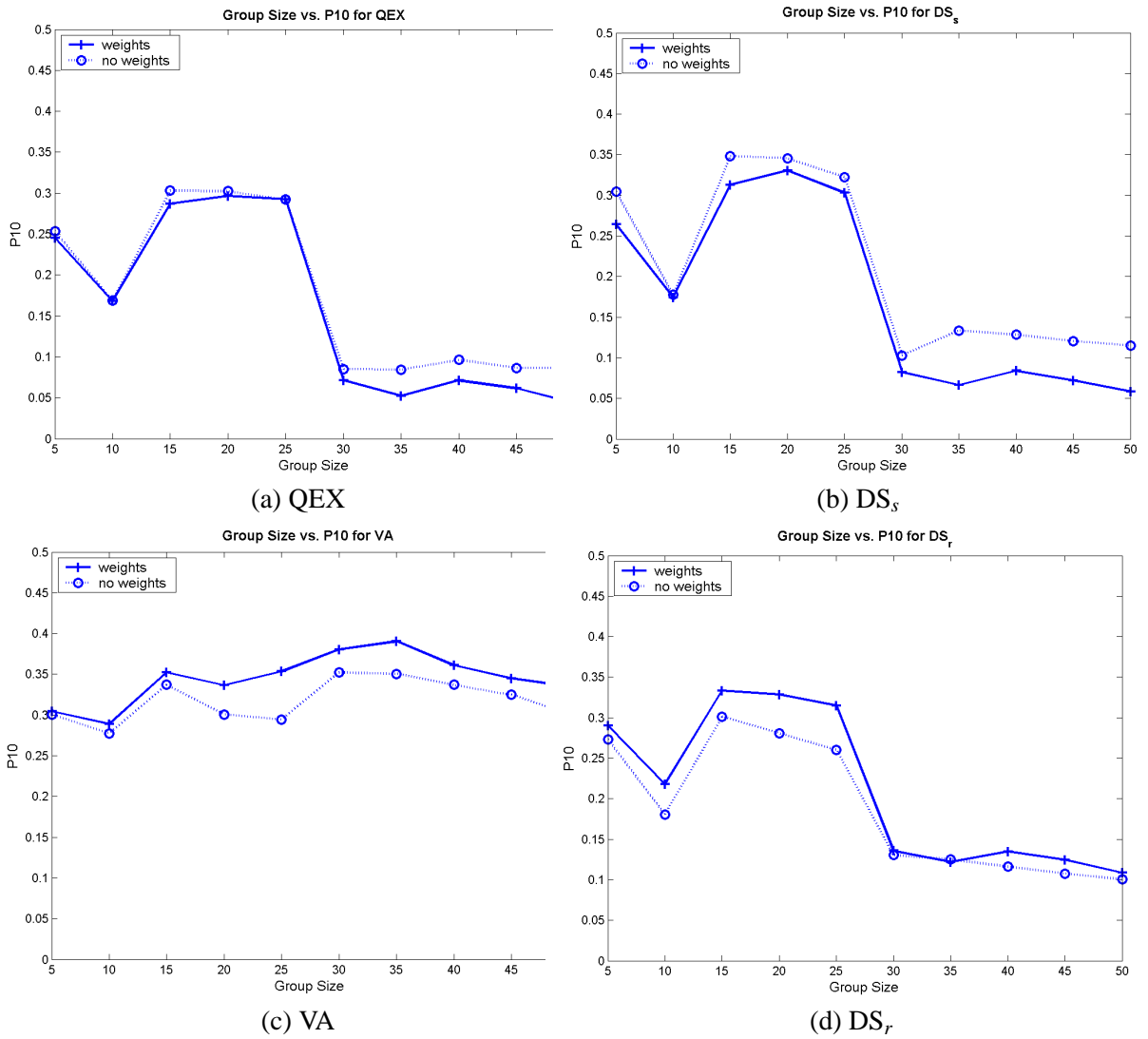


Figure 5.5: P(10) for various group sizes comparing weighted vs non-weighted variants (average over all categories)



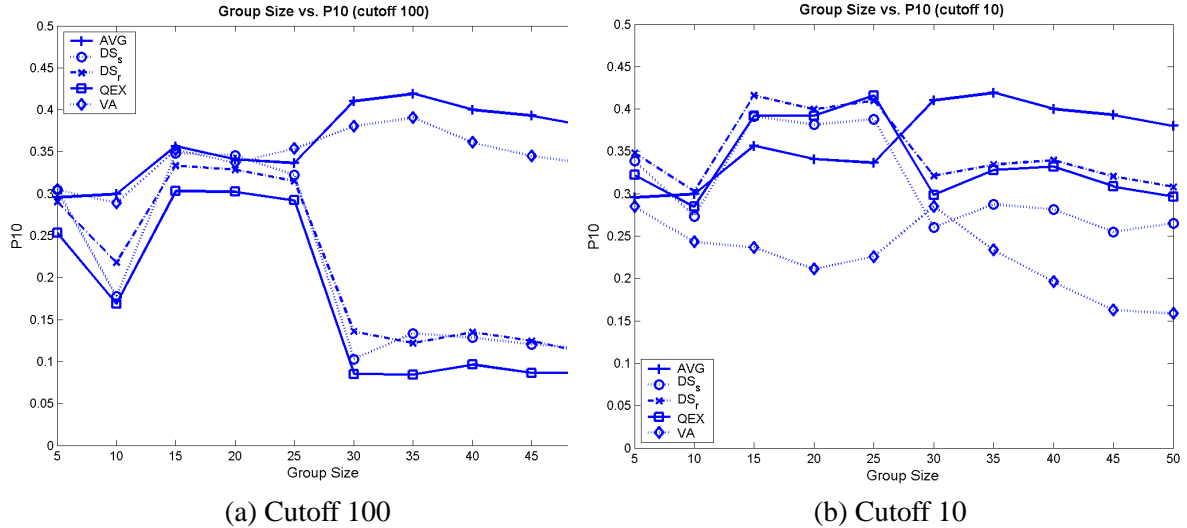


Figure 5.6: P(10) for various group sizes comparing cutoff 100 and 10 (average over all categories)

The weighting introduces noise adversely affecting the final ranking. This again shows that the raw scores cannot easily be compared.

### Variations of Cutoff Value

In the various fusion methods, we limit the length of the individual lists being merged,  $c$ , to  $k \leq c \leq N$  (where  $N$  is the total number of items in the collection and  $k$  the number of recommendations), for computational and retrieval performance reasons. The objective is to confirm the previous observations that some of the fusion methods tend to suffer from the introduction of noise when merging large lists, by varying the list length or cutoff value,  $c$ , in this run.

First, the effect of a lower cutoff value is tested against the various group sizes. Figure 5.6 contrasts the performance of the new cutoff value of 10 to the previous results. Figure 5.6(a) depicts the best performances of the previous run (the rank-based methods are weighted, whereas score-based methods do not use weights). In Figure 5.6(b) the same settings are applied, only now  $c$  is set to 10. It shows that only VA benefits from a larger cutoff value, while all other mergers perform better when short lists are combined. This confirms the claim from above that not only the number of lists to combine, but also their length adversely affects the performance of QEX and DS. The possible noise increases with a larger number of lists as well as a larger list size. A larger cutoff value does not have the same smoothing effect in VA, however. In fact, VA does not even need to look at all candidates in the lists, but stops at a certain depth as soon as the top  $k$  candidates are determined. A larger cutoff value is only beneficial for VA, if the individual lists disagree (the lists have to be processed further down to find the top  $k$  candidates that appear in the majority of lists), otherwise it does not harm the performance. On the other hand, a small cutoff value in VA yields a suboptimal performance, since the majority criterion might have to be compromised when not enough candidates appear in the majority of the lists. This is clearly visible in Figures 5.6(a)&(b), where VA's performance at  $c = 10$  drops substantially below that of  $c = 100$ .

The critical group size of 25 is again visible in the results. All multi-point approaches drop

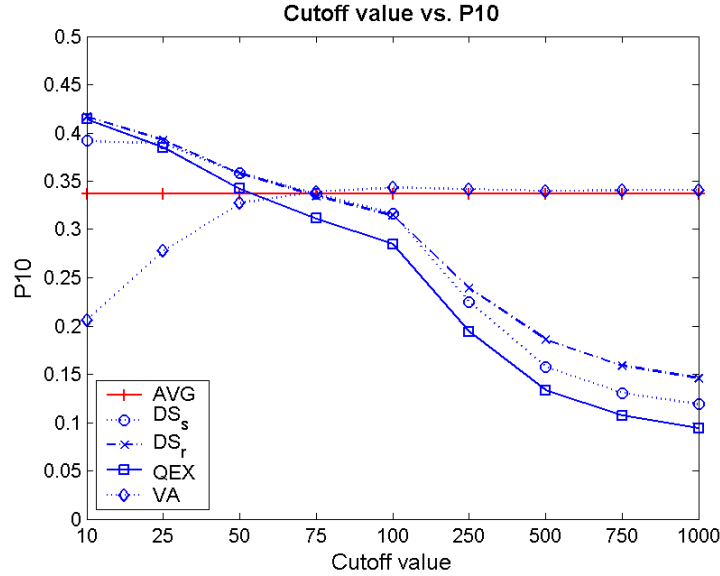


Figure 5.7:  $P(10)$  for various list cutoff values (group size 25, average over all categories)

after that (albeit not as dramatically as with  $c = 100$ ). With the lower cutoff value, the multi-point approaches QEX and DS now outperform the baseline up to this critical point.

To determine the exact influence of the cutoff value  $c$ , we have varied it from 10 to 1000. The group size is set to 25 in this run. The graph in Figure 5.7 plots the cutoff value versus the  $P(10)$  performance. As the graph shows, the performance of QEX and DS is best at  $c = 10$ , decreasing rapidly with a growing  $c$ . The curve for VA exhibits the opposite behaviour, increasing steadily up to a peak at  $c = 100$ , from which point onwards its performance cannot be increased any further.

### Summary

- The multi-point query approaches performed generally better than AVG in heterogenous categories.
- QEX and DS are very sensitive to the choice of clusters and perform significantly worse for larger group sizes (also due to the automatic choice of the optimal number of clusters used).
- List weighting improves performance of the rank-based mergers, VA and DS<sub>r</sub>, while it adversely affects the score-based mergers, QEX and DS<sub>s</sub>.
- The cutoff value has a significant effect on all mergers. QEX and DS perform best at a small cutoff value of 10 (outperforming the baseline by around 5% points). VA, on the other hand, reaches its peak at a cutoff of 100.

### 5.4.2 Performance with Relevance Feedback

Having established some critical settings of the fusion methods, we now proceed to comparing their performance in an interactive scenario. In this run, user interaction is simulated by starting with a group size of three, that is a group containing three randomly chosen images from a given category, and performing pseudo-relevance feedback from the recommendations (top 10 returned

Table 5.3: Number of images found per RF iteration

	1	2	3	4	5	6	7	8	9	10
<i>AVG</i>	5.76	7.93	9.64	11.00	12.15	13.15	13.86	14.30	14.57	14.70
<i>DS<sub>s</sub></i>	5.20	5.50	5.62	5.69	5.71	5.73	5.74	-	-	-
<i>DS<sub>r</sub></i>	5.34	5.69	5.89	5.99	6.04	6.07	6.07	6.07	6.08	6.08
<i>QEX</i>	4.91	5.67	6.13	6.42	6.63	6.76	6.85	6.90	6.91	6.92
<i>VA</i>	5.93	7.96	9.54	10.66	11.60	12.39	13.02	13.59	13.96	14.22

	11	12	13	14	15
<i>AVG</i>	14.78	14.84	14.87	14.88	14.89
<i>DS<sub>s</sub></i>	-	-	-	-	-
<i>DS<sub>r</sub></i>	6.09	-	-	-	-
<i>QEX</i>	6.92	-	-	-	-
<i>VA</i>	14.39	14.51	14.57	14.62	14.65

images). In each feedback iteration the simulated user adds all relevant images in the recommendations to the current group. A query run terminates, when no more relevant images can be found.

From the previous runs, we determined the optimal settings for each fusion method. The list cutoff value is set to 100 for VA and to 10 for all other fusion methods. Further, the rank-based methods (VA and  $DS_r$ ) incorporate a weighting of the query points, while in the score-based methods ( $QEX$  and  $DS_s$ ) lists are combined without weights.

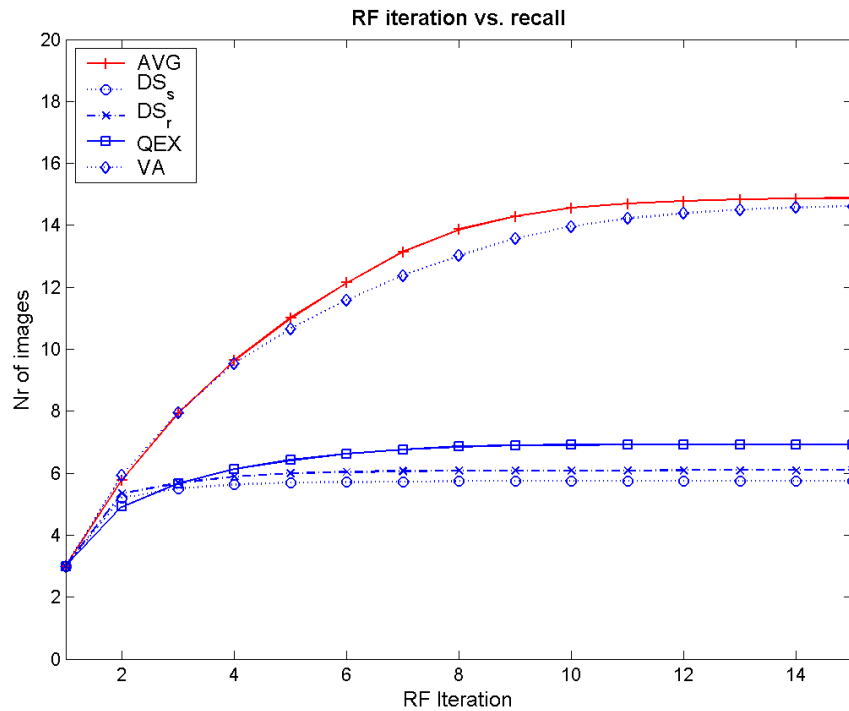


Figure 5.8: Number of images found per RF iteration (average over all categories)

Figure 5.8 shows the results for the simulated run just described. The graph depicts the average number of relevant images found in each iteration, based on 50 queries per category. Overall, AVG outperforms every multi-point query strategy. While VA's performance is almost as good

Table 5.4: Average number of images found after RF convergence

	QEX	DS <sub>s</sub>	DS <sub>r</sub>	VA	AVG
all categories	6.47	5.55	5.86	12.36	12.69
homogeneous	9.04	7.26	7.95	20.97	21.92
heterogeneous	3.90	3.84	3.76	3.76	3.46

as the baseline, all other strategies perform considerably worse. The exact numbers are listed in Table 5.3.

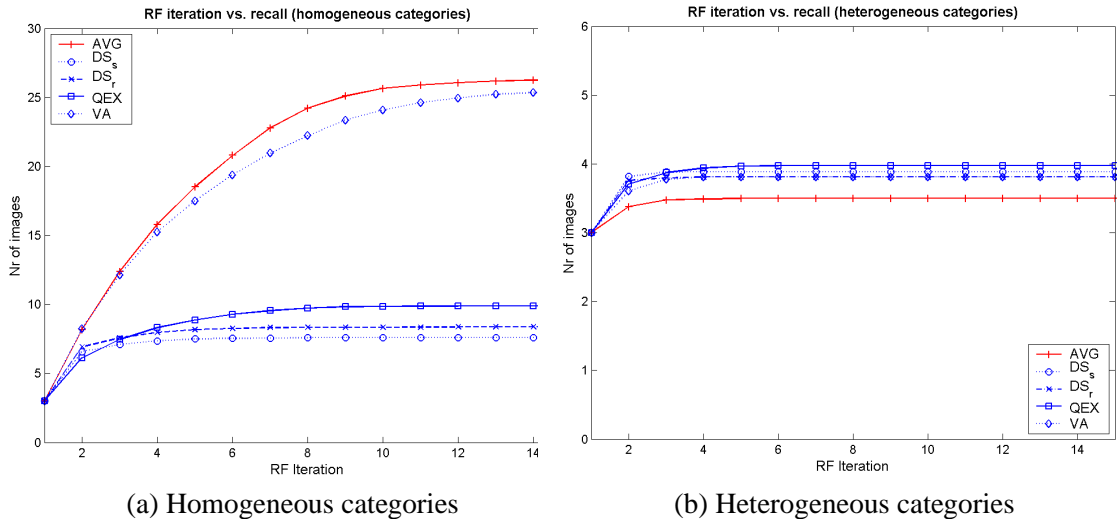


Figure 5.9: Number of images found per RF iteration (average over homo- and heterogeneous categories)

The results for the homogeneous categories are displayed in Figure 5.9(a), while Figure 5.9(b) depicts the heterogeneous categories. It shows that AVG performs very well on homogeneous categories, while it performs slightly worse than the multi-point queries on heterogeneous categories. However, VA manages to capture a group’s query representation well in both circumstances. These results are summarised in Table 5.4.

It should be noted that for each mechanism, there is a set of “null queries” for which no relevant results can be found during the first iteration. With no relevant images returned, there can be no relevance feedback. The incident rate of null queries averages at roughly 1/3 of queries, but varies from system to system. AVG is particularly poor with almost 40% of these queries, compared to 35%, 29%, 28% and 32% and for QEX, DS<sub>s</sub>, DS<sub>r</sub> and VA, respectively. The numbers of these null queries are listed in Table 5.5. This shows that the multi-point approaches tend to be better in finding relevant images when only three query images are available. These findings suggest that an adaptive recommendation system might be able to improve the overall performance, which initially uses multiple query points, but switches over to a single query representation once a sufficiently large group size has been reached. The adaptive recommendation system then benefits from improved performance, and at the same time computation costs are kept low (since the overhead of computing the multi-point queries is less for a small group size than for a large one).

Table 5.5: Number (percentage) of null queries

	QEX	DS <sub>s</sub>	DS <sub>r</sub>	VA	AVG
all categories	17.4 (34.8%)	14.7 (29.4%)	13.8 (27.6%)	15.8 (31.6%)	19.8 (39.6%)
homogeneous	6.8 (13.6%)	4.0 (8.0%)	2.8 (5.6%)	2.0 (4.0%)	5.0 (10.0%)
heterogeneous	28.0 (56.0%)	25.4 (50.8%)	24.8 (49.6%)	29.6 (59.2%)	34.6 (69.2%)

### 5.4.3 Discussion

The evaluation has confirmed that list combination is an intricate topic, and previous results of superior performance of multi-point queries over single point queries in general as reported by Porkaew et al. (1999) or Kim & Chung (2003) should be used cautiously. Factors, such as the cluster algorithm, the list cutoff value, the weighting of lists, can have a detrimental impact if not applied carefully.

Overall, multi-point queries can provide a benefit over a single group representative, but only if a suitable combination strategy is employed. A simple linear combination of the raw scores is sensitive to noise, especially when the number of lists becomes large and the lists are very different from each other. In this case, computing the average of scores acts like a smoothing operation. Kim & Chung (2003) have already observed that this form of query expansion creates a large contour covering all query points. In other words, averaging indicates that an image should match *all* query points. They argue that, if the query points are far apart from each other, the contours should be separated to allow a discriminative search. They have suggested the use of the minimum distance (disjunctive OR operation of distances) rather than the average as in QEX (AND operation) in the combination process, which might help to prevent this problem. On the other hand, VA has exhibited stable performance and is the only fusion method with comparable performance to the single-point query approach, AVG, under various settings.

In general, multi-point queries perform better than a single point query in heterogeneous groups, where the images will indeed form multiple distinct clusters. On the contrary, a single query point is sufficient to describe homogeneous groups. In addition, when the group size gets large, a group also benefits from a single query representation.

## 5.5 Conclusions and Future Work

In this chapter we introduced the underlying mechanisms for a recommendation system based on content-based image features. Underlying the recommendation system is a group-based learning technique that is achieved by: (1) adapting the feature weights to reflect common features in the group; (2) creating multi-point queries as group representatives; and (3) an evidence combination scheme to compute the final recommendations. We presented a quantitative evaluation of three possible algorithms for combination.

We identified a number of parameters that influence the multi-point queries. There are still a large number of additional parameters we have not yet studied. First, the results might have been influenced by the clustering method implemented. Hierarchical agglomerative clustering algorithms are known for their suboptimal performance when wrong decisions about cluster merging

are made early on in the lower hierarchies (Theodoridis & Koutroumbas 1999). Further, the automatic selection of the number of clusters to use is another factor influencing the performance of multi-point queries. We could consider better clustering algorithms, such as the adaptive scheme proposed by Kim & Chung (2003), which should further improve the results of multi-point queries. Ideally, the cluster algorithm should be able to distinguish visually homogeneous from heterogeneous groups, so that for the former only one cluster is returned while the latter is divided into multiple clusters. In this way, the recommendation algorithm for homogeneous groups would employ the efficient and more effective AVG method in these circumstances, while the query expansion approach would be employed for heterogeneous groups. For this reason, we suggest investigating the possibility of a metric for group homogeneity that can be used by the cluster algorithm.

Moreover, the feature weighting influences the multi-point queries. In the proposed multi-point recommendation algorithm, the overall weighting for all images in the group is computed and then used to weight the individual clusters. We made this decision because the weight computation is only reliable if there are enough samples, ie more samples than the feature dimensionality. Since the clusters can contain as little as one image, computing individual feature transformation matrices would be error prone. The analysis of the results, however, pointed to a deteriorating performance for all but the VA approach with increasing group size. This might be preventable if individual weights were employed for the larger clusters.

Relevance feedback algorithms relying solely on visual features tend to converge after a few iterations, after which no more new relevant images can generally be returned. (The average recall in the homogeneous categories was 21% for the best performing method in Section 5.4.2.) For this reason, the group-based recommendation system is not enough for a successful image retrieval and management tool. Hence, we need to study both improvements to the recommendation system, as well as alternative retrieval aids in the system acting as extensions to the recommendation system.

As a simple fix, one could consider alternative presentation techniques. No matter which fusion method employed, the aggregation of results can always miss relevant images. Instead of combining the lists of the multi-point query for the overall recommendations, one could retain the individual lists and present these as separate recommendations to the user. In this case, the user is presented with the different senses or facets of the group recognised by the recommendation system. This idea is similar to Truran et al.'s approach (2005) for query term sense disambiguation. Despite its potential benefits, this approach has not been implemented yet, in favour of improving the overall quality of the recommendations. Visual features alone have turned out to be insufficient to satisfy real searcher's expectations when using the recommendation systems (cf Section 6.2.4), and we deemed it more fruitful to find an alternative approach that can integrate both visual and textual queries. Furthermore, multiple recommendation windows would also clutter and complicate the interface.<sup>5-3</sup>

In order to improve recommendation quality, we will look at how contextual information can be incorporated as another source of evidence beside the content-based features. Contextual information will be gained from personal preferences by analysing all existing groups on the workspace

---

<sup>5-3</sup>A simple alternative to multiple recommendation windows was adopted in the interface once different feature modalities (such as visual features and text) were to be integrated. This was achieved by adding multiple tabs to the results panel: one showing the overall results and then one tab per modality to show the individual results (see Figure 4.1, panel 2).

(giving rise to co-occurrence counts of images). Since the voting approach is independent of scores, it is employable when combining information from different feature modalities. Based on this observation and the results presented in this chapter, we have chosen the voting approach as the initial solution for the improved recommendation system that includes visual, textual as well as contextual features. Before describing the unified framework including contextual information in Chapter 7, we report the results of a user study of *EGO* in the next chapter. With the basic recommendation system in place we felt it was necessary to investigate the system's effectiveness from the user's perspective. The observations during the user experiments have also led us to improve the recommendation system along the way, which finally resulted in the proposed method discussed in Chapter 7.

---

### USER EVALUATION OF EGO

---

In this chapter we present the user evaluation of EGO’s interface focusing on the support it offers the user to search for images and organise their results. Our experimental hypothesis is two-fold. First, we aim to collect evidence on whether the proposed system helps the user to conceptualise their search tasks, and therefore clarify their information needs. Second, we want to establish whether it helps to overcome the query formulation problem, since—if the user relies on the in-built recommendation system—there is no need to create a query in order to initiate a search. We measure EGO’s success in these two issues compared to a traditional relevance feedback system as a baseline.

The evaluation is based on “real”<sup>6-1</sup> users performing practical and relevant tasks, and captures a large amount of interaction data that can be used in follow-up evaluations. By employing different types of information seeking scenarios, the evaluation shows that the proposed approach succeeds in encouraging the user to conceptualise their tasks. The grouping, combined with the recommendation facility, helps overcome the query formulation problem experienced in the relevance feedback system. Overall, the workspace interface leads to increased user satisfaction. The proposed interface is stronger at supporting complex tasks requiring diversified searches. The relevance feedback approach, on the other hand, is good for selecting many images for a specific topic. This work was published in (Urban & Jose 2006*d*, 2005, 2006*e,b*).

#### 6.1 Introduction

After having argued that a system supporting an interactive organisation process leads to a more intuitive interaction paradigm in Chapter 4, a user experiment was designed to investigate the actual effectiveness of the workspace. This experiment was exploratory in nature. By observing and analysing the users’ organisation strategies we will answer the following questions: How was the workspace used? What influence did the task have on this? More importantly, however, we would like to determine the workspace’s role in helping the user both to conceptualise their search tasks and to overcome the query formulation problem.

---

<sup>6-1</sup>as opposed to simulated



However, image retrieval systems are particularly difficult to evaluate (cf Section 2.5.1). To date there still does not exist a common testbed despite several efforts (eg the Benchathlon network (Benchathlon n.d.) and more recently ImageCLEF (ImageCLEF n.d.)). What makes creating a testbed so challenging is the lack of objective measures for realistic image search tasks. People have employed category search (eg Nakazato, Dagli & Huang 2003) and target search tasks (eg Cox et al. 2000), where the set of relevant images can be determined beforehand and hence traditional precision and recall measures (van Rijsbergen 1979) can be used. However, image searching is an inherently creative activity. The target user population is expected to use our system for design-related work tasks. In these scenarios it is seldom the case that an image retrieval system is consulted to search for such a clearly defined set of images (Garber & Grunes 1992). On the contrary, the underlying information need is typically vague, and the result set is fuzzy.

For these reasons, we have adopted a user-centric, task-oriented experimental methodology. We have devised several design-oriented tasks and asked design-professionals to participate in order to create a realistic search experience. Each task description is accompanied by a scenario, which describes a simulated work task (Borlund & Ingwersen 1997) (cf Section 2.5.2). The simulated work task situation is aimed at emulating tasks from an individual's working life. This allows the users to develop their own interpretation of the task and use their own judgement for choosing relevant images. This way we can study how information needs evolve and what influence the interface has on their search and organisation strategy.

The experiment was carried out in two stages. Different tasks were chosen in each stage, and only the second stage incorporated a textual search facility and negative feedback. The results are analysed per stage with combined results provided at the end.

## 6.2 Experiment 1

In order to understand how people organise their workspace and what influence the task has on this, the first stage of the experiment was designed with two different tasks in mind: a category search task and a design task. *EGO* was evaluated against a system that has essentially the same relevance feedback mechanism, but without the organisation capabilities provided by the workspace. Analysing, in particular, the image organisation resulting from pursuing the various tasks, but also more generally the users' performance and satisfaction with the system, should highlight the difference made by the workspace.

### 6.2.1 The Interfaces

In this experiment, the *EGO* interface described in Chapter 4 is evaluated from the perspective of the participants. The underlying retrieval mechanism has been described in Chapter 5, and a traditional relevance feedback interface using the same retrieval mechanism serves as a baseline.

For the purpose of the evaluation, a slightly different version of *EGO*'s interface, as described in Chapter 4, was used. Chapter 4 described the final version of the system, which was a result of an iterative design process. The evaluated interface, referred to as "Workspace System" below, lacks two search components: the group search facility; and the Query-by-Keyword (QBK) search. Keyword annotations were not available initially, and hence the interface did not provide for QBK.

Only the feedback we received during Experiment 1 led us to include keywords for the later versions of the interface. The group search facility was not implemented in the first experimental version either. Furthermore, negative feedback was not available yet.

The recommendation system is based on only an average query representation as in (Rui & Huang 2000) instead of multi-point queries as described in Section 5.2. This is mainly due to computational complexity (so as not to stretch the users' patience), but also due to some anomalies we found during the earlier evaluation arising from the clustering algorithm used (cf Sections 5.4.1 and 5.5).

### **Workspace Interface—WS**

A workspace in the interface allows the user to organise their search results and provides both retrieval and management facilities. Images can be dragged onto the workspace from any of the other panels (or imported from outside the system) and organised into groups. The grouping of images can be accomplished in an interactive fashion with the help of a recommendation system. For a selected group, the system can recommend new images based on their similarity with the images already in the group. The user then has the option of accepting any of the recommended images by dragging them into an existing group.

Since our main objective in these experiments is to evaluate the usefulness of the workspace (and also to avoid biasing the participants by the naming of the experimental systems), this interface is referred to as the Workspace System (WS). The WS interface depicted in Figure 6.1 comprises the following components (the following numbers correspond to the panel numbers in the screenshot):

1. **Given Items Panel:** This panel contains a selection of images provided to the participants for illustration purposes and can be used to bootstrap the search. Three images per task were chosen by the evaluator that were believed to reflect the particular topic well. Please refer to Section 6.2.2 below for the description of the tasks.
2. **Query Panel:** This panel provides a basic Query-by-Example (QBE) facility to search the database by allowing the user to compose a search request by adding example images to this panel. At this point of the evaluation, textual annotations, and thus Query-by-Keyword (QBK), were not available for the collection yet. Clicking on the "Search" button in this panel issues a search, which causes the system to automatically construct a query from the examples provided and compute the most similar images in the database.
3. **Results Panel:** The search results from a query constructed in the QBE panel are displayed in this panel. Any of the returned images can be dragged onto the workspace to start organising the collection or into the QBE panel to change the current query.
4. **Workspace Panel:** The workspace holds all the images added to it by the user and serves as an organisation ground for the user to construct groupings of images. Groupings can be created by right-clicking anywhere on the workspace, which opens a context menu in which the option can be selected. They can also be created by using a button located in the toolbar on the top of the workspace. Traditional drag-and-drop techniques allow the user to drag

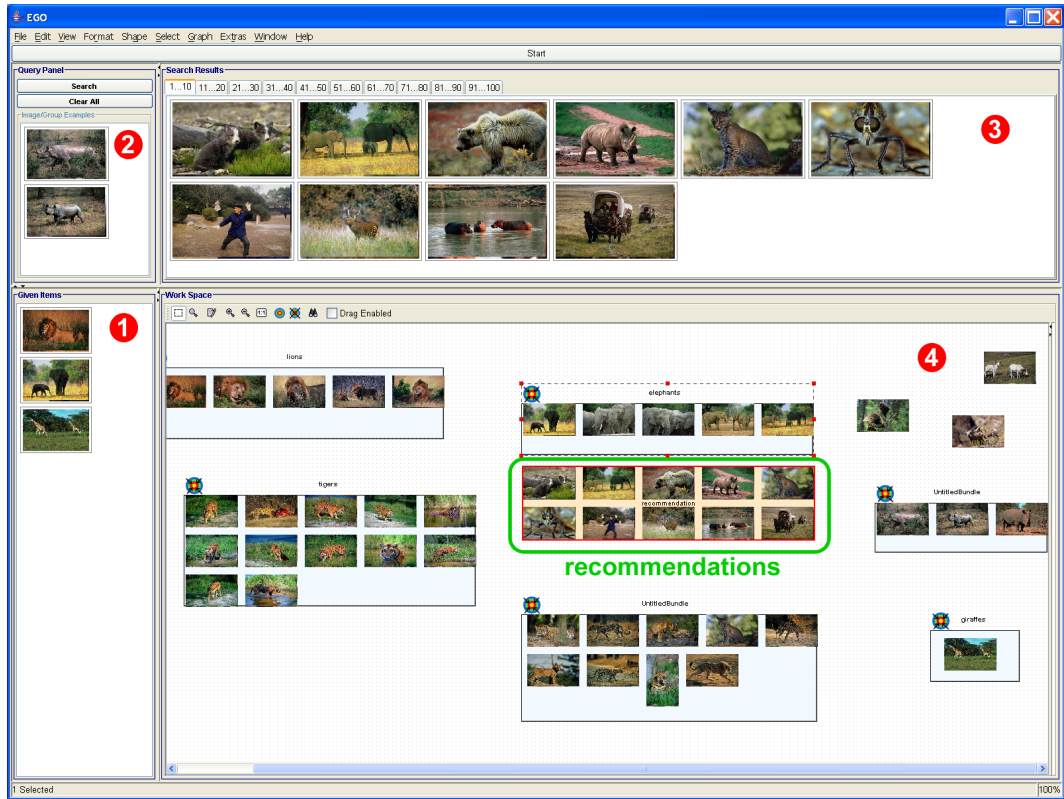


Figure 6.1: Annotated WS interface used in Experiment 1

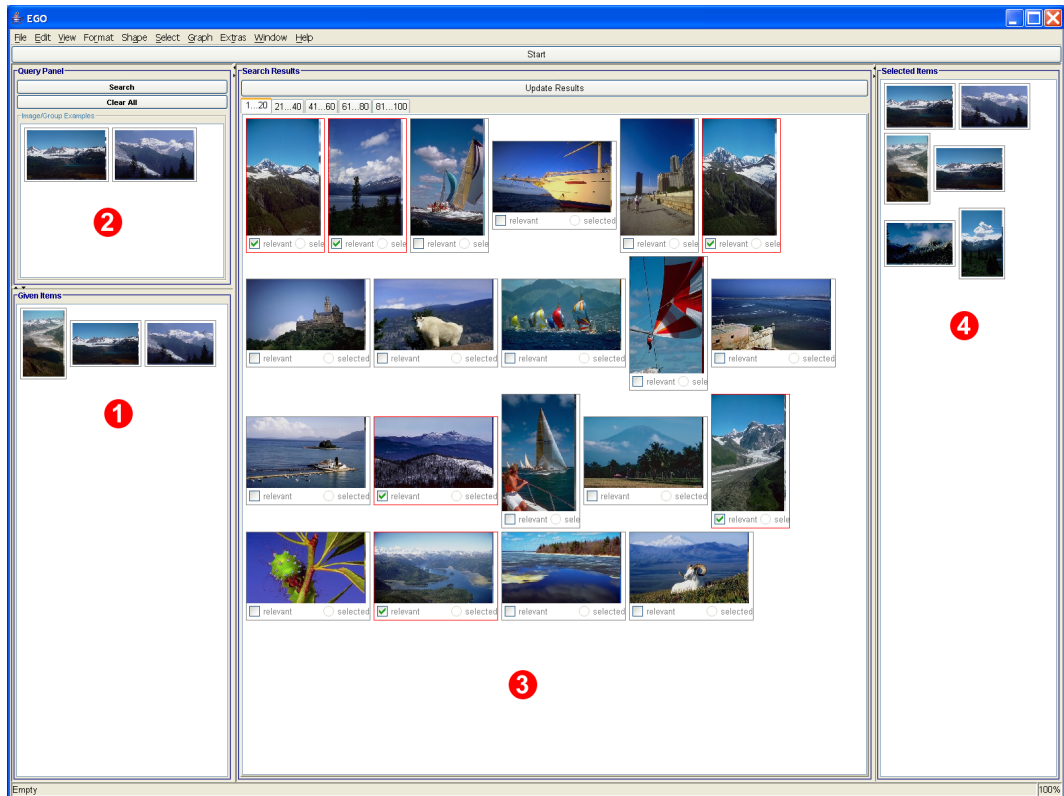


Figure 6.2: Annotated CS interface used in Experiment 1

images into (or out of) a group or reposition the group on the workspace. An image can belong to multiple groups simultaneously. Panning and zooming techniques are supported to assist navigation in a large information space. Also, the recommendations are displayed close to the selected group on the workspace (see centre of workspace in Figure 6.1). So as not to burden the user, the number of recommended images (set to 10 in this evaluation) is based on the standard cognitive limits of  $7 \pm 2$  (Miller 1956).

To reiterate, the query facilities available in the WS interface are: (1) manually constructed queries by providing one or more image examples (QBE); and (2) user-requested recommendations.

### Relevance Feedback Interface—CS

The baseline system is a traditional relevance feedback system, referred to as CS (for Checkbox System). As discussed in Section 2.3, relevance feedback (RF) is an automatic process of improving the initial query based on relevance judgements provided by the user (Rui et al. 1998). The process is aimed at relieving the user from having to reformulate the query in order to improve the retrieval results incrementally. The search becomes more intuitive to the user, since they are only requested to label the returned images as either relevant or not. Nevertheless, it is still an ongoing research challenge to accurately learn the information need from the user based on a few relevance judgements (Zhou & Huang 2003).

Figure 6.2 shows the CS interface with the following components (the following numbers correspond to the panel numbers in the screenshot):

1. Given Items Panel: as above. The same starting images as in WS were provided to the participants for each task.
2. Query Panel: as above.
3. Results Panel: As above, but instead of dragging a relevant image onto the workspace the user has the choice of labelling it by selecting a checkbox underneath the image. After relevant images have been marked the user can ask the system to update the current search results (based on the feedback provided) by clicking the “Update Results” button in this panel.
4. Selected Items Panel: All items selected relevant during the course of the search session are added to this panel. The user can manually delete images from this panel if they change their mind at a later change. This panel serves as an intermediate store of relevant images for the task.

Finally, CS supports two query facilities: (1) QBE as above; and (2) automatic query reformulation by the user feedback provided in the search results (RF).

### 6.2.2 Experimental Methodology

Based on frameworks for evaluating interactive systems (Jose et al. 1998, Borlund 2003b), we have designed the experiments to be as close to real-life usage as possible: we have chosen par-

ticipants with a design-related background and have set tasks that are practical and relevant. We employed a subset of the Corel collection (CD 1, CD 4, CD 5 and CD 6 of the Corel 1.6M dataset), containing 12,800 photographs in total (COREL n.d.). 12 participants used two systems in a randomised within-subjects design (Maxwell & Delanay 1990), and a Latin-square design (Maxwell & Delanay 1990) was used to rotate the ordering of systems and tasks to counterbalance the effect of learning (cf Section 3.1.5).

The independent variable was system type; two sets of values of a variety of dependent variables indicative of acceptability or user satisfaction were to be determined through the administration of questionnaires (provided in Appendix D.1). In addition, users' actions were logged and analysed.

### Participants

Our sample user population consisted of post-graduate design students and design professionals. Responses to an entry questionnaire indicated that our participants could be assumed to have a good understanding of the search and design task we were to set them, but a more limited knowledge or experience of the search process. We could also safely assume that they had no prior knowledge of the experimental systems.

There were 12 participants in total: 9 male and 3 female. The average age was 26 years. They had on average 5 years experience in a design-related field (graphic design, architecture or photography). Most people dealt with digital images at least once a day.

The participants were also asked about prior experience with image search engines, professional image search services, and image management systems for organising their own images in the entry questionnaire. All participants had used an internet image search engine before (mainly Google Images), whereas only 5 people had used a stock image collection (such as Getty Images, Corbis, Corel). Concerning the organisation of their images, 9 people did not use any management system but just organised their images into folders. The image management systems that were used by the remaining 3 users were ACDSsee, iPhoto/iView, Picasa and Extensis Photo Studio.

People thought that using folders was easier, more relaxing and satisfying than either Web or stock search engines. They also felt they were able to find images using their own organisation more often than using search engines. People expect a search engine to not only return relevant images matching their search criteria, but also a "*wide variety of images to choose from*". Therefore, it is also important to detect new images outwith their initial search criteria by looking at related images supported by tools to easily navigate, view, survey and compare large numbers of images at once. Many people want to search by criteria other than just textual descriptions, for instance file type, file size, aesthetics, quality and style. An attractive interface was also important to some people. People also stated they needed a straightforward way of cataloguing images in a "*more human-based interface and search process*", or "*a library of images grouped into my own categories*"<sup>6-2</sup>.

---

<sup>6-2</sup>Note that these answers were provided before the participants were introduced to the experimental systems.

## Tasks

We used a simulated work task situation as conducted by Jose et al. (1998). An abbreviated description of the work task scenario and tasks is provided in Figure 6.3. Please refer to Appendix D.1.1 for the full task description and the search topics.

**Task Scenario**  
*Imagine you are a designer with responsibility for the design of leaflets on various subjects for the Wildlife Conservation (WLC). The leaflets are intended to raise awareness among the general public for endangered species and the preservation of their habitats. These leaflets [...] consisting of a body of text interspersed with up to 4–5 images selected on the basis of their appropriateness to the use to which the leaflets are put.*

**Category Search Task (Tasks A and B):**  
*You will be given a leaflet topic from the list overleaf. Your task involves searching for as many images as you are able to find on the given topic, suitable for presentation in the leaflet. In order to perform this task, you have the opportunity to make use of an image retrieval system, the operation of which will be demonstrated to you. You have 10 minutes to attempt this task.*

**Design Task (Task C):**  
*This time, you're asked to select images for a leaflet for WLC presenting the organisation and a selection of their activities (some of WLC's activities are listed overleaf but feel free to consider other topics they might be involved in). Your task is to search for suitable images and then make a pre-selection of 3–5 images for the leaflet. You have 20 minutes to attempt this task.*

Figure 6.3: Task description for Experiment 1

**Category search task:** In the category search scenario users were asked to find as many images as possible from a given topic. The topics in Task A represent simple and concrete topics (“mountains”, “tigers”, “elephants”), while the topics in Task B comprised multiple facets (“animals in the snow”, “African wildlife”, “underwater world”).

**Design task:** This task resembles an open-ended design task, where the participants had to search for and make a choice of 3–5 images.

The first task was set on both systems, CS and WS, while the latter was performed with WS only after having completed the category searches. A maximum time was set for all tasks in order to limit the total time spent on the experiment. This was 10 minutes for the category search and 20 minutes for the design-task.

Initially, the design task was only planned as a complementary task to determine how the workspace was used and how images were organised. We felt the experiment would be too long if the design task was set on CS as well, since the total time for one session was already two hours (see below). In hindsight, this was not a good decision, since we were unable to compare the different effects of the systems for various tasks. This was one of the reasons to perform a second experiment with more tasks, as described in Section 6.3.

## Hypotheses

Since investigating the workspace's usefulness is a high-level goal, the experimental hypothesis has been broken up into the following more manageable sub-hypotheses:

1. The addition of a workspace leads to a more effective system and increased user satisfaction.
2. The workspace helps users to conceptualise their tasks:
  - Helping users to organise their ideas;
  - Helping users to detect and express different task aspects;
  - Helping users to follow up on various task aspects, thus diversifying their search.

Results are analysed according to these two points, which is expected to shed light on the use of the workspace and its usefulness.

## Procedure

We met each participant on a separate occasion and adhered to the following procedure:

- an introductory orientation session
- a pre-search questionnaire
- a hand-out of written instructions for the tasks and setting the scenario
- **Part 1:** category search task
  - for each system (CS and WS)
    - \* a training session with the system
    - \* a search session in which the user interacted with the system (max 10min)
    - \* a post-search questionnaire
  - a questionnaire comparing the two systems
- **Part 2:** design task
  - a search session with WS system (max 20min)
  - a post-search questionnaire

The total time for a session was two hours.

## Data Capture

**Questionnaires:** The questionnaires elicit people's opinion on the tasks performed, the images found during the search session, the usability of the systems and their satisfaction with their task performance. Please refer to Appendix D.1.3 for the documents. User opinion was captured on five-point semantic differentials, five-point Likert scales and open-ended questions. The results for the semantic differentials and Likert scales are in the range [1, 5], with 5 representing the best value. In the results analysis, statistically significant differences are provided where appropriate with  $p \leq 0.05$  using the non-parametric Wilcoxon matched pairs signed rank test (Lewis & Trail 1999).  $\overline{CS}$  and  $\overline{WS}$  denote the means for CS and WS, respectively, while  $\widetilde{CS}$  and  $\widetilde{WS}$  denote the medians.

**Usage logs:** The data logged included total session time, images selected during the search, types of queries issued and number of queries issued. These results are analysed and summarised to reflect the users' performance and effort required to complete the tasks.

Table 6.2: Number of relevant images found and corresponding levels of recall per category search topic

	Task A			Task B			AVG
	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	
Total #Relevant	549	114	103	220	865	402	375.5
#Rel AVG	56.5	14.0	15.25	44.0	38.75	36.75	34.2
#Rel CS	71.5	18.0	18.5	54.5	50.5	34.0	41.2
#Rel WS	41.5	10.0	12.0	33.5	27.0	29.0	25.5
Recall AVG	10.3%	12.3%	14.8%	20.0%	4.5%	7.8%	11.6%
Recall CS	13.0%	15.8%	18.0%	24.8%	5.8%	8.5%	14.3%
Recall WS	7.6%	8.8%	11.7%	15.2%	3.1%	7.2%	8.9%

### 6.2.3 Results Analysis

There are two objectives of this experiment: (1) to contrast the two systems in terms of their effectiveness and user satisfaction; and (2) to analyse how people make use of the workspace depending on the nature of the task.

#### System Comparison

The first objective of the experiment was to compare the two interfaces. It involved two category search tasks, one on each system. The analysis is based on data obtained through questionnaires and usage logs. The questionnaires present a subjective view indicative of the system’s acceptability and usability, while the log data provides a means of judging task performance objectively.

**Task Performance** Data in the usage logs sheds light on how people actually used the system. From this data we can obtain information on the number of relevant images found over the course of the search session. The category search tasks are the only tasks that have an associated set of relevant images<sup>6-3</sup>. Table 6.2 shows the number of relevant images for each of the topics and systems. The total number of relevant images varies greatly per task. The level of recall (number of relevant images found over number of total relevant images for the topic) attained depends therefore not only on the complexity of the task but also on the number of relevant images available in the system. The topics were chosen so that Task A represented simple and concrete topics (“mountains”, “tigers”, “elephants”), while Task B comprised multiple facets (“animals in the snow”, “African wildlife”, “underwater world”). Looking at the data in Table 6.2 it can be inferred that users generally performed better in CS independent of the nature of the task. Yet the questionnaire analysis below suggests that there was a stronger focus in WS to find appropriate images for the leaflet, facilitated by the superior tool for exploring the task and the image collection.

**User Satisfaction** After having completed a task, the participants were given a questionnaire about their search experience (the post-search questionnaire in Appendix D.1.3 ). Finally, they were asked to compare the two systems in the exit questionnaire. In this section, we analyse the users’ opinion of the systems as inferred from the answers provided in the questionnaires.

<sup>6-3</sup>The ground-truth was obtained by manually labelling relevant images in the collection for each topic.



Table 6.3: Semantic differential results for the Task, Search Process and Images parts

	Differential	$\overline{CS}$	$\widetilde{CS}$	$\overline{WS}$	$\widetilde{WS}$	p
Task	clear	4.8	5	4.8	5	-
	easy	4.5	5	4.3	5	-
	simple	4.8	5	4.5	5	-
	familiar	3.8	4	3.7	4	-
Search	relaxing	<b>4.6</b>	5	3.9	4	-
	interesting	3.6	4	<b>4.3</b>	4	0.02
	restful	3.8	4	3.7	4	-
Images	relevant	4.2	4	4.2	4	-
	appropriate	4.2	4	4.3	4	-
	complete	3.3	3	<b>4.1</b>	4	0.03

Table 6.4: Semantic differential results for the System and Interaction parts

	Differential	$\overline{CS}$	$\widetilde{CS}$	$\overline{WS}$	$\widetilde{WS}$	p
System	wonderful	3.7	4	4.1	4	-
	satisfying	3.9	4	4.1	4	-
	stimulating	3.2	3	<b>3.8</b>	4	0.01
	easy	<b>4.6</b>	5	4.1	4	0.03
	flexible	2.8	3	<b>3.9</b>	4	0.01
	novel	3.1	3	<b>4.2</b>	4	0.02
Inter	effective	4.3	4	4.3	4	-
	in control	4.3	4	4.2	4	-
	comfortable	4.4	5	4.6	5	-
	confident	4.3	4	4.4	5	-

Table 6.5: Likert-scale results for the System part

Statement	$\overline{CS}$	$\widetilde{CS}$	$\overline{WS}$	$\widetilde{WS}$	p
learn to use	<b>4.8</b>	5	4.1	4	0.03
use	4.5	5	4.0	4	-
explore col.	3.3	3	<b>4.3</b>	4	0.03
analyse task	3.1	5	<b>4.5</b>	5	0.02

1. *Post-Search Questionnaire*: In the post-search questionnaire, people were asked about the task they performed, the images received through the searches and the system itself.
  - **Task**: The first part of the post-search questionnaire covered the user's perception of task complexity. The tasks were rated according to the five-point semantic differentials: clear (vs. unclear), easy (vs. difficult), simple (vs. complex) and familiar (vs. unfamiliar). The results are shown in Table 6.3 (scores from 1 to 5, higher = better). There are no significant differences on any of the differentials. All scores are well above 3, showing that the users generally considered the tasks to be *clear*, *easy*, *simple* and *familiar*. However, the tasks were considered slightly more *easy* and *simple* in CS. Note that their perception depends on the users' overall search experience, since these responses are received in the post-search questionnaire.
  - **Search Process**: The users were asked to rate the search process according to the five-point semantic differentials: easy (vs. stressful), interesting (vs. boring) and restful (vs. tiring). The search process was considered slightly more *relaxing* and *easier* in CS, but significantly more *interesting* in WS. However, people tended to agree more with the statement that they had enough time to complete their task in CS:  $\overline{CS} = 4.6$ ,  $\widetilde{CS} = 5$  and  $\overline{WS} = 4.3$ ,  $\widetilde{WS} = 4$ .
  - **Images**: The retrieved images were rated on the semantic differentials: relevant (vs. irrelevant), appropriate (vs. inappropriate) and complete (vs. incomplete). They were considered equally *relevant* and *appropriate*, but significantly more *complete* in WS (see Table 6.3).

Table 6.6: Comparison of system rankings

System	(a) learn	(b) use	(c) effective	(d) liked best
CS	5 (42%)	5 (42%)	4 (33%)	3 (25%)
WS	3 (25%)	<b>6 (50%)</b>	<b>6 (50%)</b>	<b>8 (67%)</b>
no difference	4 (33%)	1 (8%)	2 (17%)	1 (8%)

More people agreed with the statement, that they discovered more aspects of the category than initially anticipated during the search with WS ( $\overline{CS} = 2.4$ ,  $\widetilde{CS} = 2$  and  $\overline{WS} = 4.4$ ,  $\widetilde{WS} = 5$ ;  $p = 0.02$ ). On the other hand, people tended to be equally satisfied with their search results in both systems ( $\overline{CS} = 3.6$ ,  $\widetilde{CS} = 4$  and  $\overline{WS} = 3.6$ ,  $\widetilde{WS} = 4$ ). There is no apparent correlation between actual task performance and perceived task performance. This shows that people had other performance criteria apart from finding as many images as possible. We suspect that the users of WS, by possessing better tools for exploring and analysing the retrieved images, concentrated more on selecting appropriate images (see below for more details, in particular Section 6.3.3).

- **System:** The users considered CS significantly more *easy* than WS, while they considered WS to be significantly more *stimulating*, *flexible* and *novel*. Table 6.4 shows the results for these differentials.

People found CS significantly easier to *learn to use*, while there was only a marginal difference between *ease of use*. By contrast, people thought WS helped them to explore the collection better, as well as analyse the task better. The results for the responses to these statements are provided in Table 6.5.

2. *Exit Questionnaire:* After having completed both category search tasks using both systems, the users were then asked to indicate the system that was: (a) easiest to learn to use; (b) easiest to use; (c) most effective; and (d) they liked best overall. Table 6.6 shows the users' preferences of systems for each of the statements. It shows that, while it is easier to learn to use CS, people did not have a problem using WS, and the majority of people preferred WS. In open-ended questions, the participants were invited to give their opinion on what they liked or disliked about each system. The advantages listed for CS were that it was fast, efficient and easy to use. Some user comments that reflected these issues were: "*is was very efficient in finding many images*", "*very simple and easy to understand*". Its disadvantages included that the users felt they did not have enough control over the search and that its interface was less intuitive, eg "*too regimented; not enough user control*", "*too abstract*", "*slightly confusing*". In WS, people liked the ability to plan their searches by organising the results into groups, and the overview they had of the results and searches that the organisation brought along, eg "*allowed you to plan/organise images, whilst finding them, saving time later*", "*the constant overview of all results*", "*it allowed flexibility [...] therefore I selected more, then dispensed with those that weren't useful*". In addition, the system's flexibility and variety of control options were noted as advantages. The disadvantages were mainly concerned with the poor quality of the recommendations and that the handling of groups was sometimes cumbersome. Both of these issues are not inherent in the interaction

paradigm of the proposed system itself, and can consequently be improved or even avoided in the future. The recommendation quality can be improved by a better choice of visual features and also by recommendations based on other people's groupings. The handling of the groups and images within groups is an implementation detail.

**Summary** In terms of effectiveness and usability, the following advantages and disadvantages of WS could be identified:

- CS is better for finding many images for the category search task. In spite of this, there was no difference in people's perception of task performance and system effectiveness. In fact, people found the selection of images received in WS to be more complete. We found evidence in the questionnaire responses and by observing people's behaviour that they were more concerned with finding good quality images in WS. This can be explained by the fact that the task description not only asked users to find as many images as possible, but also had the additional qualifying statement "*suitable for presentation in the leaflet*". This was an ambiguity in the description, because it was not clear for some users whether all images matching the concept were also suitable for the leaflet.
- WS is more difficult to learn to use. The learning period is extended, since the interface provides a more complex and flexible interaction strategy, initially increasing the cognitive load. Deciding on how many groups to create and which images to add to which group, for example, requires additional cognitive effort. This was reflected not only in the judgement of ease of the system compared to CS and the ranking in the final questionnaires, but also affected the user's perception of the ease and simplicity of tasks and led to a less relaxing search process.
- The longer learning period and increase in cognitive load is not perceived as negative. On the contrary, the search process is considered significantly more interesting, and the system itself is significantly more stimulating, flexible and novel. At the end of the experiment, the participants thought it was at least as easy to use and effective as CS, and the majority preferred WS. The learning process is generally two-way: the system learns about the user and the user about the system, both becoming more efficient over time.
- WS helps users to analyse the task better, discover more aspects of the task than initially anticipated and explore the collection more effectively. For this reason, WS seems to be better for exploratory searches with vague information needs or complex, multi-faceted tasks. This observation will be reevaluated in Experiment 2, when more types of tasks are compared on both systems.
- The feedback on the workspace was entirely positive. The participants claimed they particularly liked the ability to plan their searches and organise their results. In comparison, they considered they were lacking control over their searches in CS.

Yet the first sub-hypothesis could not be verified entirely: while WS scores better in terms of user satisfaction, CS is the more effective system for the category search tasks.

### Organisation Analysis

The second objective of the study is to judge the workspace's usefulness in helping the user to conceptualise their task. In order to find out how people make use of the groupings and organise their workspace, we have created two different task scenarios in the experiment: the category search scenario and the design task scenario. The former (set on both WS and CS) aims at maximising recall, while the latter aims at finding a selection of good quality images that work well together (only with WS). By analysing the number of groups created and the average number of images per group for the various tasks, we can determine how these numbers relate to task complexity.

The results are summarised in Table 6.7. For the focused category search (Task A) people only created around one group, whereas images for the complex category search (Task B) were organised into approximately four groups. In the design scenario, people created even more groups to organise their selection of images. These results show that the groupings are related to (available<sup>6-4</sup>) task aspects. Furthermore, in the design scenario the search is broader and more aspects are consequently followed up.

The organisation of images into groups seems to be more helpful in the design scenario than in the category search scenario. The average of the responses as to whether the organisation of images into groups helped them express different aspects of the task, is 4.4 and 3.9 for the design task and category search task, respectively. The difference is even more pronounced when comparing the different task groups for the category search tasks. The average response is 3.0 for the focused tasks and 4.8 for the more complex tasks. So, while the organisation is helpful in general, it is dependent on, and reflects, the nature of the task.

These observations could be supported by the questionnaire data that point to differences in *user perception* of their information need depending on the task nature. The responses suggest they had a clearer idea of the images that were relevant for the task in the category search scenario (average 4.4, on a scale from 1–5, higher = better), as compared to the design scenario (3.7). Hence, their need was better defined in the category search tasks. In a comparison between the two systems for the category search, we found that WS helped more to develop and broaden their need, although their initial idea did not vary much across the systems. This is reflected in the responses that the users detected significantly more aspects of the category than initially anticipated in WS ( $p = 0.02$ ), especially for the multi-faceted topics.

As an aside, we could identify two different types of behaviour concerning the organisation strategy in the design scenario. About half the people saved all candidates on the workspace organised into several groups (between 4 and 9) that reflect different aspects of the task, before making the final selection. The other group of users only added a few images to the workspace, and mostly all in the same group. Our observations were that the latter user group made their pre-selection of images suitable for the task while searching, rather than saving all suitable candidates to the workspace first. The average number of images saved on the workspace for the first selection strategy was 53 images in 6.5 groups. On the other hand, the other group of users saved only 14 images in 1.5 groups on average.

---

<sup>6-4</sup>The selection of images available in the collection obviously limits the task aspects that can be followed up on in the searches.

Table 6.7: Organisation and information need development results

	Task A			Task B			Cat AVG	Task C
	CS	WS	both	CS	WS	both		
# Groups	-	1.2	-	-	4.3	-	3.4	4.4
# Images/Group	-	18.8	-	-	11.9	-	15.4	7.5
# Selected Images	39.3	26.2	32.8	53.2	36.8	45.0	38.5	36.6
Initial idea	4.3	4.5	4.4	4.2	4.3	4.3	4.3	3.7
Detect more aspects	2.2	3.0	2.6	2.7	4.7	3.7	3.1	4.3
Satisfied with results	4.2	3.7	3.9	3.5	3.5	3.5	3.7	3.0
Organisation useful	-	3.0	-	-	4.8	-	3.9	4.4

**Summary** To summarise, we found a correlation between the number of groups created and the complexity of the task set. Furthermore, responses in the questionnaires showed that the management of search results was deemed more helpful in the design scenario, which is more flexible and open to interpretation than the category search scenario. In the category search scenario, the usefulness of the organisation also depended on the complexity of the task: the more facets the task comprised, the more useful the workspace was considered. The evident dependency between both the number of groups created and the users' perception of the workspace's usefulness, has led us to the conclusion that our approach does indeed help in conceptualising the task.

#### 6.2.4 Discussion

By analysing user behaviour in different task scenarios, we have been able to show that the grouping facility was used to reflect the various task facets, and therefore helped to conceptualise tasks. On the other hand, it is more difficult to draw a definite conclusion on the other hypothesis, namely that our approach leads to a more effective and usable interface.

The responses in the questionnaires suggest that the participants were more satisfied with their overall search experience with WS and that it was at least as effective. By contrast, the actual task performance does not reflect the users' perception. The number of relevant images found per task were generally higher in CS than in WS. Based on the analysis of the questionnaire data above, the reason for this is that the selection of relevant images is much faster than the dragging of images. Also, the users spent time creating groups of images and moving images between groups in the WS system. Underlying these activities is an additional cognitive effort. The users spent more time thinking about task aspects and the types of groups to create, as well as on the images which would be appropriate for the leaflet. Since we have set a maximum time limit, the number of images found was generally higher in CS, where the user was not "distracted" by managing their search results. On top of this, the task description was found to be ambiguous as mentioned before. We suspect that people had a slightly different objective in WS, which supported a more selective search strategy, rather than the quantity of images.

In addition, the failure of the recommendation system, based on visual features only, has most probably contributed to these results. Analysing the users' comments, we could identify that many people thought the recommendation system would potentially have been a useful feature, but it was not employed due to its inability to recommend relevant images. The main problem was that

only the top 10 recommendations were visible, whereas in CS the top 100 images were shown. The overall hypothesis underlying this work, namely that the recommendation system helps to overcome the query formulation problem, could not be verified directly. On the other hand, when analysing the way the users manually created queries, we observed an interesting pattern. They usually started off with a small number of example images (from the given items and some initial results). Once they had created a group on the workspace that contained several relevant images, they used the whole group in the QBE search to find similar images to the *group*. We assume that, had the recommendation system worked better, users would have used the recommendations instead of the QBE search. Since this was not the case, however, they had to resort to the manual facility of finding more similar images for the group.

One more issue with the workspace was that the handling of groups was sometimes cumbersome. This was caused by unexpected resizing behaviour when images were dragged around in groups. Instead of an image being deleted from a group once it is dragged beyond a group's boundaries, the bundle size adjusts automatically. Although just a minor coding issue, it led some people to avoid creating and using groups.

In conclusion, the difference in performance can be attributed to the additional effort—both physical (slower selection process) and cognitive—required in WS. While the users commented on the additional physical effort, they did not perceive the additional cognitive effort as negative. On the contrary, they thought the organisation to be supportive for solving their tasks as well as potentially beneficial for others to use in the future.

### 6.2.5 Summary

The first stage of the experiment helped to explore the benefits of the workspace and led to interesting conclusions. The participants generally preferred the proposed approach and there was evidence that it helped them to conceptualise their tasks. However, there was not enough evidence in order to study the effect of tasks on searching and organisation behaviour in detail. In particular the design task was only performed with WS, which made it impossible to compare the differences in searching experience the two interfaces might have caused. To be able to do this, we needed to investigate a larger variety of tasks. In addition, there were some issues with the evaluation set-up. Instead of continuing with the same set-up, we decided to remedy these problems and introduce a different set of tasks for the second stage of the experiment.

## 6.3 Experiment 2

Based on the results obtained from the first set of participants, the experimental set-up was scrutinised and consequently redesigned to take into account the lessons learnt. The following changes were made:

- The handling of groups was improved. The major criticism was that the display could be disturbed easily, because of unexpected behaviour when trying to move images in and out of groups. This has been addressed by automatically laying out the images in a group and disabling the manual resizing of groups.

- The recommendation system was not used to its full potential, due to its inability to recommend relevant images. This has been addressed in two ways. First, instead of just showing the top 10 recommendations on the workspace, the Results panel now also shows the complete results (limited to 100 images as in CS). Second, a textual search facility has been introduced, since the visual features seemed insufficient to solve more abstract tasks. Textual annotations for the Corel collection, obtained from (BerkleyCorel 2005), were incorporated and implemented according to the vector-space model (Salton & McGill 1983) (cf Section 7.2). The results of the visual and textual features are combined using the same voting approach as applied for combining the multi-point queries in the recommendation system (cf Section 5.2.1 and Section 7.2.4). The ability to query by keywords is expected to provide a more realistic search experience.
- The retrieval mechanism was further improved by allowing negative feedback, as people had complained about the inability to continue a search when the majority of returned images were irrelevant. Since incorporating negative feedback is a difficult endeavour (Zhou & Huang 2003), we have opted for a quick and safe approach: irrelevant images are added to a negative filter excluding them from being returned for the same search. While it was straightforward to implement this in CS where negative feedback can easily be provided explicitly, there was a choice of adopting either an implicit or explicit approach in WS. An explicit approach could be implemented for instance by a “waste bin”, eg a dedicated group on the workspace, into which irrelevant images can be dragged. However, we have chosen an implicit feedback strategy, whereby an image is automatically added to a negative filter for a group when it has been ignored (ie not dragged into this group) after having been returned three times amongst the top 10 recommendations. We expected the explicit strategy of dragging irrelevant images to a waste bin to be too cumbersome for the user. In future work, a comparison between implicit and explicit feedback mechanisms in the interface could be very enlightening.
- A new set of tasks has been introduced. We felt that more tasks were needed in order to draw definite conclusions on the workspace’s usefulness in helping to conceptualise tasks. In addition, after having questioned design professionals about their “usual” kind of work and search tasks it became apparent that they rarely have to perform an exhaustive search on a specific topic as is required in the category search task. Therefore, a greater emphasis was placed on creativity when devising the new set of tasks. The participants agreed that the chosen tasks were very similar to their own tasks during the exit interviews. To address the problem of ambiguity in the task description in Experiment 1, we have explicitly asked for a specific number of images for each task. This ensures that the users have a clear target in terms of the requested outcome of the task.
- Finally, no time limit was set on the tasks, addressing this particular problem in the previous set-up and supporting an even more creative search session. In creative search sessions, and in this experiment, effectiveness is more important than efficiency.

With this improved evaluation set-up, Experiment 2 should help clarify the validity of the experimental hypotheses.

### 6.3.1 The Interfaces

The interfaces were essentially the same as in Experiment 1 (see Section 6.2.1), with only minor adaptations described below.

#### Workspace Interface—WS

The WS interface of Experiment 2 depicted in Figure 6.4 comprises the following components:

1. Query Panel: The Query panel has an additional text box for a user to provide a set of keywords to use in a search (QBK). The QBE panel is the same as in the previous interface.
2. Results Panel: same as in the previous interface.
3. Workspace Panel: The only difference to the previous interface is that the complete recommendation results are displayed in the Results panel in addition to the top 10 box on the workspace itself.
4. Recommended Groups Panel: For each query or recommendation issued the existing groups are ranked in order of similarity to the current query/group and the five top matching groups are displayed in this panel. Each returned group contains a link to the original group on the workspace.

#### Relevance Feedback Interface—CS

Figure 6.5 shows the CS interface used in Experiment 2 with the following components:

1. Query Panel: as above.
2. Results Panel: As in previous system, but the checkbox for marking relevant images is replaced by three combo boxes to mark images as one of relevant, irrelevant or neutral (see Figure 6.6).
3. Selected Items Panel: as in previous system.

Since the keyword search provides an adequate solution to bootstrapping a search session, we no longer needed to provide a set of given items. Hence, the given items panel is no longer present in the interfaces.



Figure 6.6:  
Relevance feedback in CS

### 6.3.2 Experimental Methodology

The experimental methodology is similar to the previous experiment detailed in Section 6.2.2. The main differences are: a new set of tasks; a revised set of experimental hypotheses; and a slightly different procedure. The experimental documents can be found in Appendix D.2.



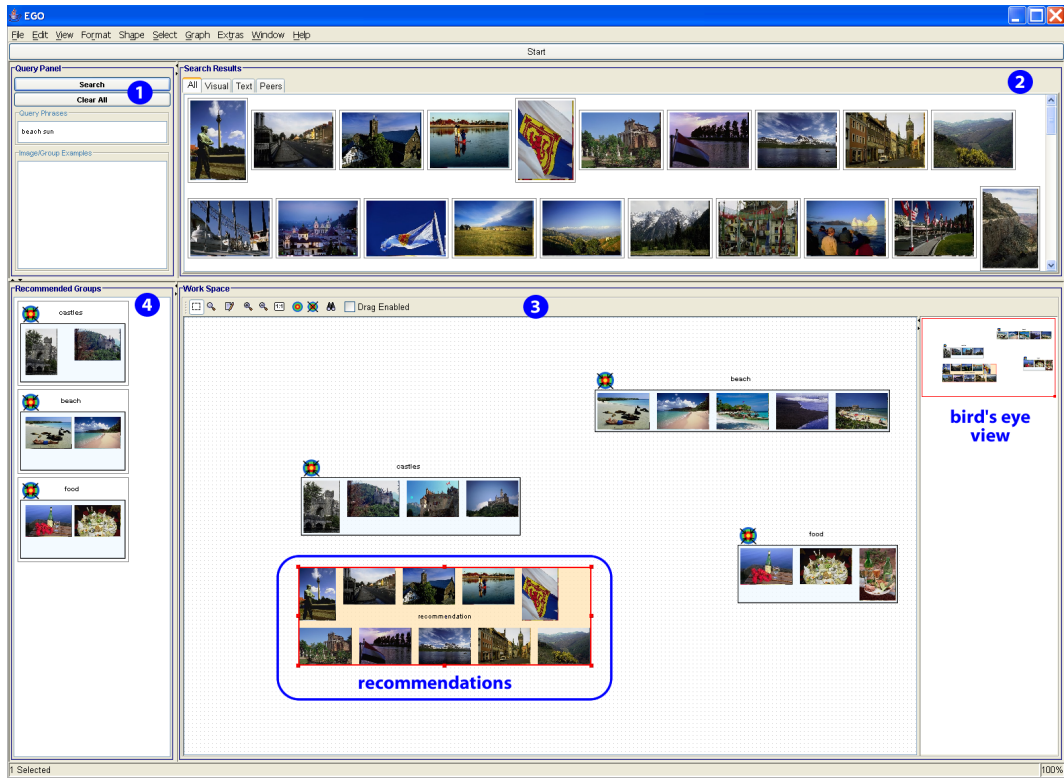


Figure 6.4: Annotated WS interface used in Experiment 2

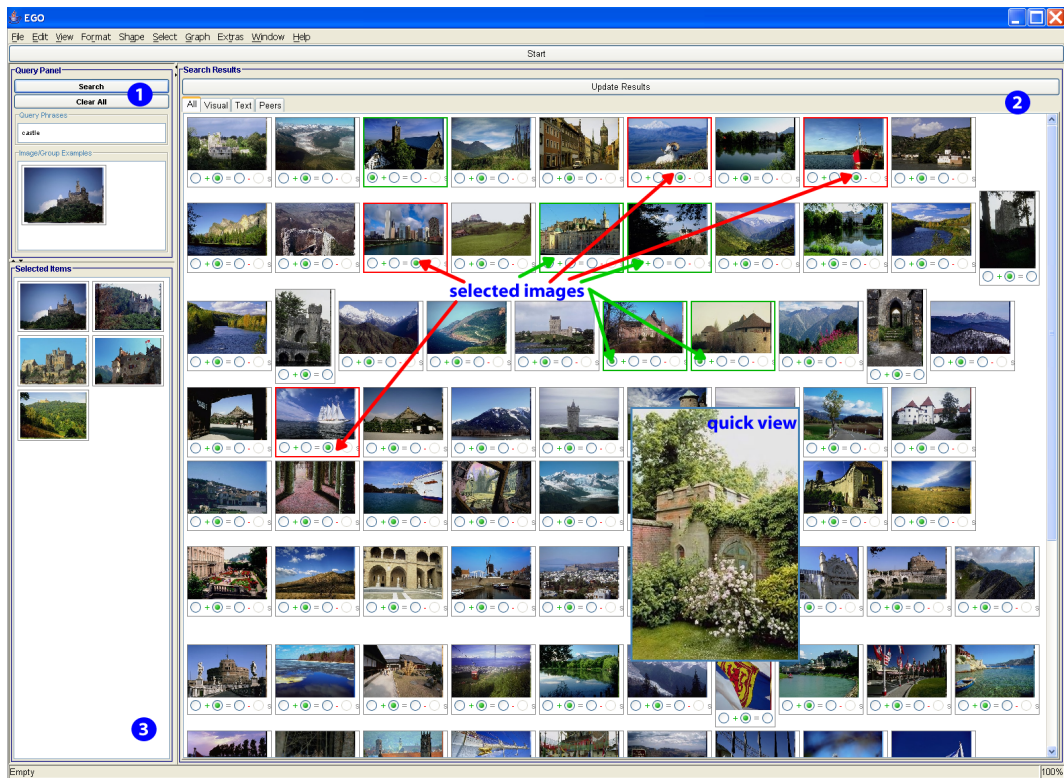


Figure 6.5: Annotated CS interface used in Experiment 2

## Participants

The user profile is similar to the one described in Section 6.2.2. Again, there were 12 participants: 7 male and 5 female. There was a wider variety of ages in the range of 20–50 years and the average age was slightly higher at 28. The participants came once again from a design-related field and were equally experienced in it. In Experiment 2, there were 6 people that had used the services of a stock image provider before. Concerning the management of images, there were 8 people who used only the operating system's folder structure to organise and manage their images. The responses to the usability of Web search engines, stock image collections and image management system showed the same trends as in Experiment 1.

## Tasks

**Theme search task:** In this task people were asked to find an image fitting into a specified theme. The theme was illustrated by three example images and the task involved searching for and selecting *one further image* complementing this set (see Figure 6.7).

**Illustration task:** The task was to illustrate a piece of text for publication on the Web or an advertising slogan with *three images*. There were four topics in total from which the participants had to choose two (one on each system). One example scenario and task description is provided in Figure 6.7.

**Abstract search task:** Here, people were asked to select *at least one image* representing a given abstract topic. The simulated work task situation prescribed selection of an image for a photo competition.

No time limits were set on these tasks, as it was learnt from Experiment 1 that this adversely affected people's performance.

## Hypothesis

As in Experiment 1, more evidence was to be collected for the following sub-hypotheses:

1. The proposed approach leads to an increased effectiveness and user satisfaction.
2. The workspace helps to conceptualise and diversify tasks.

This time round, an additional perspective was explicitly introduced:

3. The grouping and recommendations help to overcome the query formulation problem.

## Procedure

We met each participant on a separate occasion and adhered to the following procedure:

- an introductory orientation session
- a pre-search questionnaire
- for two types of tasks:
  - search session on system 1

**Theme search (Task D)**  
 Look at the three images provided below. They all share a common theme. Your task is to find and select a fourth image complementing the set.



Topic 1: “people in national costumes”



Topic 2: “seasons in the country”

**Example illustration task (Task E):**  
 Imagine you are the Web designer for an online travel agency called *PerfectHoliday*. In order to gain more customers, they have decided to hold a competition entitled “Win your dream holiday”. They have provided you with the details of the competition (see below) and have asked you to select some images to illustrate the text.  
 Your task is to find one main and two additional images that you would place on the webpage along with the competition details. The images should draw people’s attention and spark their imagination.

**Win your dream holiday!**  
 What if you could make your dream holiday become reality? Where would you go and what would you do? *PerfectHoliday* is giving you the chance to win that dream! We will be giving away £2000 to the lucky winner for the holiday of their dreams! What would you do with the money? Swim with the dolphins? Stay on a French castle or sail the Mediterranean on a luxurious sailboat? Do you imagine yourself white water rafting in the Alps? Or would a secluded beach with pearly white sands be for you? No matter what your dream holiday looks like, we will make your dreams come true.  
 To enter this competition, simply send us a description of the perfect holiday before midnight on [...] So don’t hesitate! Send your details to [...] and you could be packing your bags!

**Abstract search (Task F):**  
 Imagine you want to take part in a photo competition, where you could win £100 for a picture that depicts the following theme: *Dynamic [//Cute]*  
 In order to get ideas for the competition, you want to look for already existing photographs conveying the same theme. Your task is to select at least one image that represents the theme well.

Figure 6.7: Task descriptions for Experiment 2

- \* a training session on the first system if not used before
- \* a hand-out of written instructions for the first topic of this task
- \* a search session in which the user interacted with the system (ca 15min)
- \* a post-search questionnaire
- search session on system 2 (same as above with second topic of this task)
- repeat with second task
- an exit questionnaire/interview comparing the two systems

The whole session lasted approximately two hours. Tasks and systems were rotated according to a Latin-square design in order to compensate the learning bias (see Section 3.1.5).

### 6.3.3 Results Analysis

In the results analysis, the systems are first compared according to: their effectiveness; and user satisfaction. Finally, the users' organisations of images on the workspace are analysed and related back to the task that was performed and the nature of the users' underlying information needs.

The results for Likert-scales and semantic differentials are in the range [1,5], the higher the value the better. Statistically significant differences are provided where appropriate with  $p \leq 0.05$  using the non-parametric Wilcoxon matched pairs signed rank test (Lewis & Trail 1999).  $\overline{CS}$  and  $\overline{WS}$  denote the means for CS and WS respectively, while  $\widetilde{CS}$  and  $\widetilde{WS}$  denote the medians.

#### Effectiveness

The systems' effectiveness is investigated both objectively and subjectively: from the perspective of the required effort as determined from the usage logs and from the perspective of the participants.

**User Effort** Due to a lack of objective performance measures for the tasks in Experiment 2, we provide an analysis of the number of images selected per task and the amount of user effort required to select them. Indeed, the effort users have to invest in order to complete a task is another interesting characteristic and should not only be seen as placeholder for task performance. Indicators for task completion effort include: total search time; number of images selected during the search; and number of queries issued. People can issue either manual queries—constructed in textual form, by providing image examples or a combination of both—or relevance feedback queries. The latter correspond to relevance feedback iterations in CS or group recommendations in WS.

Table 6.9 breaks up the results of the user effort indicators into the various tasks. The time invested is on average 4–5min higher for Task E. The participants also issued more queries (both manual queries as well as feedback queries). At the same time, more images were selected during a search session for this task. This confirms previous observations that people were more intrigued in pursuing this task. Moreover, the relevance feedback facility, either in the form of explicit relevance feedback or group recommendations, was most used in Task E. On the other hand, Task D is a lot more focused. As a result, fewer images are selected, fewer queries are issued and hence less time is spent on completing it.

Table 6.9: User effort indicators per task

Task	D	E	F
time	10'58"	16'22"	11'56"
#images	11.0	18.3	15.7
#queries	10.7	20.3	16.4
manual	8.0	14.0	11.7
RF	2.7	6.4	4.8

Table 6.10: User effort indicators per task and system

	D <sub>CS</sub>	D <sub>WS</sub>	E <sub>CS</sub>	E <sub>WS</sub>	F <sub>CS</sub>	F <sub>WS</sub>	$\overline{CS}$	$\overline{WS}$
time	9'55"	12'02"	18'26"	<b>14'18"</b>	9'40"	14'31"	12'40"	13'35"
#images	9.6	12.3	17.9	<b>18.6</b>	13.6	17.8	13.7	16.2
#queries	11.9	9.8	21.5	19.1	15.9	17.1	16.4	15.3
manual	8.4	7.6	15.8	12.2	11.4	12.0	11.9	10.6
RF	3.4	2.1	5.8	6.9	4.5	5.1	4.6	4.7

The same data per system, as shown in Table 6.10, reveals that the search session lasted on average 1min longer with WS than with CS. Again, we attribute the longer time spent with the system to an increased interest on the user's side. As is shown below, people found this system better for analysing the task and exploring the collection. Supporting this observation is that tasks performed with WS resulted in a slightly higher number of selected images, while the number of manual queries issued was lower.

It is even more interesting to look at the differences between the tasks depending on which system was used. Task E stands out for being completed in less time with WS (with a difference of about 4min) but achieving a slightly larger selection of images in the end. Task D required slightly fewer, Task F slightly more queries to be issued with WS. Nonetheless, the number of images is higher for both these tasks with WS.

**User Perception of Task Performance** After each task the users were asked if they thought they had succeeded in their performance of the task and also to rate potential problems that might have affected their performance. Table 6.11 reflects the general perception of performance success for each task. The table also highlights the problems that affected the performance (rated on a score from 1–5, lower = more problematic). The biggest problem encountered was that people thought the images they were looking for were not contained in the collection, followed by the system not returning relevant images. People were slightly less satisfied with their performance for Task E. The dissatisfaction was mainly attributed to the problem that they could not find the images they were visualising (ie because the images were not in the collection or the system did not return relevant images). Also, time was more of an issue in this task<sup>6-5</sup>. These results are to be expected, since this is the most creative of the three tasks.

Performing a task with WS was perceived as more successful, as can be seen from Table 6.12. With WS, people's understanding of the task might have had a little impact. Also, time was more

<sup>6-5</sup>Since no time limits were imposed, people completed a task when they were reasonably happy with the images they found. As the answers suggest, however, they sometimes felt they would have found better/more images had they spent even more time.

of an issue with WS than CS<sup>6-6</sup>. On the other hand, people's performance was hindered more by an uncertainty of what action to take next with CS. Together with the user comments presented below this indicates that—though a simple concept in principle—providing relevance feedback brings uncertainty as to which images to select for feedback in order to achieve better results. Again, this demonstrates the semantic gap and query formulation problems inherent in image retrieval systems. This also corroborates similar results in textual information retrieval (Beaulieu & Jones 1998).

The dependencies between task and system are displayed in more detail in the same table. For Task D, people were more satisfied with their performance with WS. The problems affecting their performance with CS more than with WS include the fact that CS did not return enough relevant images and that they were less sure of their actions with CS. On the other hand, Task E is relatively balanced concerning problems encountered during the search process. Once more, the uncertainty of their next action had a larger impact on people's performance with CS for Task F. With WS, they felt the lack of relevant images in the collection was the biggest issue. Still, they thought they completed the task more successfully with WS.

We also observed that the selected images with WS were of better quality, suggesting that WS's efficacy is superior. In an additional study, we asked people to judge the relative quality of the result sets obtained in the experiment to quantify this observation. We randomly selected 16 pairs of result sets per task (48 in total), where one set was retrieved with WS and the other with CS. The participants were given a copy of the original task description and a pair of result sets obtained for this task. They were then asked to (a) select one overall image from the two sets, which—in their opinion—was most suitable for the task; (b) cross out any images they thought were not relevant for the task; and (c) state which set they preferred overall. 16 people, judging on average three different pairs each, took part in this study. None of these people had participated in the actual experiment evaluating the systems. Out of the 48 pairs, there were 36 preferences for the sets obtained with WS and only 12 for the CS sets. Only on three occasions, people picked the best image from the set they did not prefer, and on one occasion the best image was present in both sets. The number of instances in which the best image was selected from the WS sets was also 36, compared to 13 for CS. Moreover, people disagreed more with the images in the CS sets: 2.9 images were deleted from these sets on average compared to 1.6 from the WS sets. There was no apparent trend that people simply preferred the larger result set: 26 votes for the larger set, 22 for the smaller. The differences of preferred set ( $p < 0.001$ ), best image ( $p < 0.001$ ) and irrelevant images ( $p < 0.05$ ) between the two systems are all statistically significant as determined by the Wilcoxon signed ranks test. Hence, the general consensus is that the selection of images obtained in WS is better.

### User Satisfaction

In this section, we discuss the results to the responses concerning user satisfaction with the system in general and the interface features in particular.

---

<sup>6-6</sup>In Experiment 1, people also tended to agree more with the statement that they had enough time to complete their task in CS:  $\overline{CS} = 4.6$  and  $\overline{WS} = 4.3$

Table 6.11: User perception of task performance per task (performance: higher = better, problems: lower = more problematic)

Task	D	E	F
performance success	4.4	4.1	4.3
did not understand task	4.9	4.9	4.8
images not in collection	4.3	3.5	3.6
no relevant images returned	4.2	3.6	4.4
not enough time	4.8	4.3	4.8
unsure of next action	4.3	4.3	4.2

Table 6.12: User perception of task performance per task and system (performance: higher = better, problems: lower = more problematic)

	D <sub>CS</sub>	D <sub>WS</sub>	E <sub>CS</sub>	E <sub>WS</sub>	F <sub>CS</sub>	F <sub>WS</sub>	CS	WS	p
performance success	4.2	4.6	4.1	4.2	4.3	4.4	4.2	<b>4.4</b>	-
did not understand task	5.0	4.8	5.0	4.9	4.9	4.6	5.0	<b>4.8</b>	-
images not in collection	4.2	4.4	3.5	3.5	4.1	<b>3.4</b>	3.9	3.8	-
no relevant images returned	<b>4.0</b>	4.4	3.6	3.5	4.5	4.4	4.0	4.1	-
not enough time	4.8	4.9	4.4	<b>4.1</b>	4.9	4.6	4.7	<b>4.5</b>	-
unsure of next action	<b>4.1</b>	4.5	4.4	4.4	<b>4.0</b>	4.3	<b>4.2</b>	4.4	-

**Tasks, Search Process and Retrieved Images** The trend on the user’s perception of the tasks themselves is reversed in Experiment 2: the tasks were considered slightly more *clear*, *easy*, *simple* and *familiar* with WS. As in Experiment 1 there were no significant differences concerning the tasks. The search process was once again perceived to be significantly more *interesting* with WS and the set of images received through the searches were more *complete*. The results for this part are shown in Table 6.13.

**System and Interaction** There is a clear trend that the participants were more satisfied with WS. They regarded WS to be significantly more *flexible* and the scores for the remaining differentials—*wonderful*, *satisfying*, *stimulating*, *efficient* and *novel*—were higher for WS as well. CS, on the other hand, was only thought to be *easier*. Table 6.14 shows the results for these differentials.

A similar trend is apparent concerning the interaction with the system. People felt more *comfortable* and *confident* while using WS. However, WS was deemed slightly more difficult to *learn to use* but equally *easy to use*.

**Interface Support** In Experiment 2, people were asked how effective they found the interface and rated the interface’s features contributing to the effectiveness. Table 6.15 summarises these results. Overall, WS was regarded significantly more *effective*. The three top rated features in WS were that it helped to *organise images*, *explore the collection* and *analyse the task*. The ordering of features in CS was: *find relevant images*, *explore the collection* and *detect/express different task aspects*. It is worth noting that the highest ranked feature in CS, ie *find relevant images*, has the same score in WS and CS, but it is the “least” useful feature in WS. Hence, it is no surprise that all other features are rated significantly higher in WS.

Table 6.13: Semantic differential results for the Task, Search Process and Images parts

	Differential	$\overline{CS}$	$\widetilde{CS}$	$\overline{WS}$	$\widetilde{WS}$	p
Task	clear	4.6	5	4.7	5	-
	easy	3.8	4	3.9	4	-
	simple	3.6	4	3.8	4	-
	familiar	3.4	4	3.5	4	-
Search	relaxing	3.7	4	3.7	4	-
	interesting	3.6	3	<b>4.3</b>	4	0.009
	restful	3.7	4	3.5	3	-
Images	relevant	4.0	4	4.1	4	-
	appropriate	4.1	4	4.1	4	-
	complete	3.5	3	<b>3.8</b>	4	-

Table 6.14: Results for the system and interaction differentials and Likert-scales in the System part

		$\overline{CS}$	$\widetilde{CS}$	$\overline{WS}$	$\widetilde{WS}$	p
System diffs	wonderful	3.3	3	<b>4.1</b>	4	-
	satisfying	3.2	3	<b>4.0</b>	4	-
	stimulating	3.5	3	<b>4.3</b>	4	-
	easy	<b>4.0</b>	4	3.8	4	-
	flexible	2.9	3	<b>4.2</b>	4	0.004
	efficient	3.3	3	<b>3.9</b>	4	-
	novel	3.7	4	<b>4.4</b>	5	-
Inter	in control	3.6	4	3.6	4	-
	comfortable	3.7	4	<b>4.3</b>	5	-
	confident	3.1	3	<b>3.8</b>	4	-
Likert	learn to use	4.1	4	3.9	4	-
	use	3.9	4	3.9	4	-

Table 6.15: Interface effectiveness

Statement	$\overline{CS}$	$\widetilde{CS}$	$\overline{WS}$	$\widetilde{WS}$	p
effective	3.7	4	<b>4.4</b>	5	0.032
analyse task	2.8	3	<b>4.3</b>	5	0.001
explore collection	3.5	4	<b>4.6</b>	5	0.001
find relevant images	4.2	4	4.2	4	-
organise images	2.7	3	<b>4.7</b>	5	0.001
detect/express task aspects	3.0	3	<b>4.2</b>	4	0.003

Table 6.16: Relevance assessment with CS vs. grouping with WS

Differential	$\overline{CS}$	$\overline{WS}$	p
easy	3.8	<b>4.4</b>	-
effective	3.3	<b>4.3</b>	0.019
useful	3.7	<b>4.4</b>	0.017



Table 6.16 compares the adaptive querying mechanisms in both interfaces: the relevance feedback (RF) in CS and the grouping in WS. It turns out that the grouping was considered significantly more *effective* and *useful*. It is also interesting to note that the relevance assessment was even considered more *difficult* than the grouping. In CS, the users had to think about selecting both positive and negative feedback, while the negative feedback was taken care of implicitly in WS. In order to examine if the difference was mainly due to this slight imbalance, the responses in Experiment 1 were consulted again, where both systems only required affirmative actions by the users. Although RF in CS was considered somewhat *easier* in Experiment 1 than in Experiment 2, it still scored worse than WS ( $\overline{CS} = 4.2$ ,  $\widetilde{CS} = 4$  and  $\overline{WS} = 4.4$ ,  $\widetilde{WS} = 5$ ). The RF facility in CS without negative feedback was also deemed more *effective* than its counterpart in Experiment 2, still WS scored slightly better ( $\overline{CS} = 3.8$ ,  $\widetilde{CS} = 4$  and  $\overline{WS} = 4.0$ ,  $\widetilde{WS} = 4$ ). Finally, Experiment 1 also confirmed that the grouping was considered significantly more *useful* than the relevance assessment ( $\overline{CS} = 3.4$ ,  $\widetilde{CS} = 3$  and  $\overline{WS} = 4.4$ ,  $\widetilde{WS} = 4$ ;  $p = 0.02$ ). Please note that these statements are also affected by the task, so that the differences in scores between Experiments 1 and 2 in CS cannot only be attributed to the existence of a negative feedback facility. Nevertheless, these results once again highlight the problems with the relevance feedback approach.

In open-ended questions, the participants were asked to state the most and least useful tools of the interface. The most useful tools in CS were stated as, in order of frequency of responses: textual query (10 responses<sup>6-7</sup>); QBE search (9); and relevance feedback facility (7). The least useful tools were: result filters for various features (5); relevance feedback (4); and lack of storing facility/overview of selected images (4). Users who thought the relevance feedback was a useful tool stated that it helped them to improve and/or narrow down their search, eg “*Selecting images definitely improved results and was a great alternative to choosing specific wording to further the search*”, “*It just worked and I don’t know how or why!*”. The problems with relevance assessment were mainly that it returned unexpected results and that it was difficult to follow what the system was doing. The following responses highlight these problems: “*At times the images returned seemed irrelevant to the query and this led to a lack of confidence on my part*”, “*for no obvious reason, a set of images showed up with an irrelevant context; no idea how*”, “*It was hard to keep track of what was going on*”.

In WS, people unanimously liked the grouping facility on the workspace. The three most useful tools in WS included: the grouping of images (14); group recommendations (10); and textual queries (5); and the least useful tools were: QBE (4); top 10 window of recommendations (3); and text search (2). This shows that using groups and recommendations was considered more useful than the manual search facilities. In particular, the QBE facility was deemed superfluous in this system. There was a plethora of comments about the workspace demonstrating its advantages: “*The workspace is useful, easy to use, clear and logical*”, “*I found the workspace useful to pull together and compare images from the query results*”, “*grouping was useful to keep track of associated images*”, “*emphasis was on sorting rather than searching; workspace and groups were used to categorise images and explore those categories further*”, “*I found [the groupings] very effective for identifying relevant images*”, “*easy to track thoughts on searches*”. The grouping’s only disadvantage that became apparent was that it was difficult to remove images from existing

<sup>6-7</sup>This question was asked after each task, thus 24 responses are possible per system.

Table 6.17: Comparison of system rankings

System	(a) learn	(b) use	(c) effective	(d) liked best
CS	9 (75%)	5 (42%)	1 (8%)	1 (8%)
WS	2 (17%)	5 (42%)	5 (42%)	8 (67%)
no difference	1 (8%)	2 (17%)	6 (50%)	3 (25%)

groups, which again is an implementation issue, not an issue with the concept.

These results support our view that WS, with its grouping and recommendation facility, assists the user in the query formulation process, while removing the need to manually reformulate queries. The picture in CS is quite different: people were divided on the usefulness of the relevance assessments and some still relied heavily on the manual query facilities. On average, people selected 2.4, 3.2 and 3.8 images per relevance feedback iteration for Tasks D, E and F, respectively. Compared to that, the groups in WS contained 4.9, 4.6 and 4.4 images. So the manual selection process was less productive than collecting the images in groups. Moreover, the grouping process has the additional benefit of supporting a diversifying search by allowing the user to declare and pursue various task aspects simultaneously.

**System Rankings** After having completed all four search tasks having used both systems, the users were again asked to determine the system that was: (a) easiest to learn to use; (b) easiest to use; (c) most effective; and (d) they liked best overall. Table 6.17 shows which system the users preferred for each of the statements. The rankings reflect the earlier findings of this experiment, namely that the majority of people find CS easier to learn to use, but only one person thought it was more effective and preferred it over WS. Moreover, after having used each system twice, people did not think using WS was more difficult than CS.

**Open-ended Questions** Finally, the participants were invited to give their opinion on what they liked or disliked about each system. The responses reconfirmed most advantages and disadvantages already identified in the previous experiment. The advantages listed for CS were that it was easy to use, fast and efficient especially for specific searches: eg “*could quickly indicate images closest to or further from what you wanted*”, “*it was more straightforward to use when looking for a specific image*”, “*enabled you to focus into selections*”, “*the system itself zoomed into the correct group required*”, “*easy to drill down and find 1 or 2 images you were looking for at the start*”. Its disadvantages included that the users felt they did not have enough control over the search and that its interface and search process was less intuitive: eg “*less flexibility*”, “*I really didn’t feel [the checkboxes for marking relevant/irrelevant results] worked very well*”.

People appreciated that WS was an organising tool. The workspace enabled them to plan their tasks and pursue alternative search threads, without losing the overview of intermediate results and searches: eg “*great tool for organising and building a collection of images*”, “*being able to group images and hold onto them in a new window while the search moved on*”, “*ability to group and then follow alternative search threads*”, “*felt like you were narrowing down the search and you had the results right in front of you*”, “*made search a bit clearer*”, “*very useful if looking for a variety of different images on the same topic*”. Once more, the system was regarded as more flex-

ible and offering better control over the search process. In Experiment 1, the disadvantages were mainly concerned with the poor quality of the recommendations and that the handling of groups was sometimes cumbersome. Both of these issues are not inherent in the interaction paradigm of the proposed system itself, and were consequently improved for Experiment 2. The recommendation quality was improved by taking textual annotations into account. The handling of the groups and images within groups was changed so that the system now automatically arranges the layout of the images in a group. Consequently, none of these issues resurfaced in Experiment 2.

**Summary** The two systems were compared directly in terms of user satisfaction. Although CS was considered *easier* to learn, WS scored higher in every general aspect, such as *stimulating*, *flexible* and *novel*. WS was considered significantly more effective for the tasks, and was deemed more helpful in organising the results, exploring the collection, analysing the task, detecting and expressing different task aspects.

The grouping facility was considered more useful and effective than the relevance assessment. It also emerged that the relevance assessment was mainly useful for specific searches; however, for other tasks people often felt confused and unsure of which items to select to improve the results. In this case, people had to resort to manual search facilities. The grouping facility, on the other hand, was considered helpful in all tasks for categorising the images, organising their thoughts, exploring the collection and identifying relevant images by means of better comparison opportunities. The relevance feedback mechanism in the disguise of the group recommendations did not suffer the same confusion of why the system returned certain images as the relevance assessment in CS did. Finally, the majority of people preferred WS over CS (67%).

On a side note, the implicit negative feedback strategy in WS did not seem to leave people feeling out of control. Although negative items could not be reset for groups, that did not have an impact, since the negative feedback was not used for changing the retrieval parameters. In the long run—when the system is used over multiple sessions and by more than one user—this would probably become more of an issue. Then it would be more important to allow the user to explicitly influence negative feedback, since their requirements or ideas might change.

### Organisation Analysis and Information Need Development

Experiment 1 has pointed to differences between users' organisation behaviour depending on the nature of task they performed. Following on the investigation into this dependency, three more types of tasks were introduced in Experiment 2. Again the number of groups created reflects our expectation of task complexity. In the theme task, Task D in Table 6.18, participants were required to find *one* image complementing a provided set of three images sharing a common theme. On average 1.5 groups were created for this task. The illustration task, Task E, required to search for *three* images to illustrate a given piece of text. The number of resulting groups is 2.9. Analysing the groups more closely, it turned out that people followed up on approximately three different aspects and finally chose one image of each aspect (see the analysis of the groups created below). In Task F, the target was to find *one* image that best represents a given abstract topic. Since an abstract topic cannot easily be pinned down by one idea, we expected a broader search to be required. Indeed people followed up on multiple aspects at once, 2.6 on average.

Table 6.18: Organisation and information need development results

	Task D			Task E			Task F			AVG			p
	CS	WS	both	CS	WS	both	CS	WS	both	CS	WS	both	
# Groups	-	1.5	-	-	2.9	-	-	2.6	-	-	2.3	-	
# Images/Group	-	4.9	-	-	4.6	-	-	4.4	-	-	4.6	-	
# Selected Images	9.6	12.3	11.0	17.9	18.6	18.3	13.6	17.8	15.7	13.7	16.2	15.0	
Initial idea	3.9	4.6	4.3	4.1	4.5	4.3	4.0	3.4	3.7	4.0	4.2	4.1	-
Detect more aspects	2.9	3.6	3.3	3.0	3.9	3.4	2.8	4.3	3.6	2.9	3.9	3.5	0.046
Images match idea	3.1	3.9	3.5	3.4	3.4	3.4	3.4	3.6	3.6	3.3	3.6	3.5	-
Seen all images	2.5	3.3	2.9	3.0	2.5	2.8	2.8	3.0	2.8	2.8	2.9	2.9	-
Satisfied with results	3.3	4.5	4.0	3.4	4.1	3.9	3.9	4.3	4.3	3.5	4.3	4.1	0.040
Organisation useful	-	4.6	-	-	4.8	-	-	4.7	-	-	4.7	-	-

The users started off their search with a rather well-defined initial information need in Tasks D and E, whereas their initial idea for the abstract topic, Task F, was not as clear. Compared to the design task in Experiment 1, the illustration task (Task E) in Experiment 2 had a much more detailed specification, hence the differences in the user’s initial idea: 3.7 and 4.3, respectively. People also discovered more task aspects during the search in Task E and the least in Task D: 3.7 and 3.3, respectively. They thought that there were more images in the collection that would have satisfied their requirements for all three tasks. Although they had the vaguest initial idea for Task F, they were generally more satisfied with their end results: 4.3 compared to 4.0 and 3.9 for Tasks D and E, respectively. The perception of the organisation’s usefulness for solving the task was generally high (4.7 on average), slightly above average for Task E and slightly below for Task D.

**Comparison between the Systems** As could be seen in the performance analysis in Experiment 1, the category search tasks were more successful in CS. For the tasks in Experiment 2, on the other hand, the participants managed to find a larger selection of images with WS than with CS. While their initial idea was clearer with WS, especially for Tasks D and B, they also discovered significantly more task aspects during the search with WS: 3.9 compared to 2.9 with CS. The participants were significantly more satisfied with their results across all three tasks, and as we have seen earlier in Table 6.12 also perceived their overall task performance as more successful.

**Analysis of Users’ Groups** The numbers and statistics about the organisation discussed in the previous section do not reveal the nature of the groups created by the users. In this section, we provide the missing details.

**Task D:** The theme search task was the most focused task in Experiment 2. Only 1.5 groups were created on average. There were two different topics: *people in national costumes* and *seasons in the countryside*. These themes were open to interpretation, since both were specified visually through a set of three images each. For topic 1, the participants usually only created one group containing various images of “people”<sup>6-8</sup>. Sometimes more groups were created for “groups of people” and “one person”, or “women” and “groups of people”. In topic 2, the “autumn” image was missing in the set, consequently all people had a

<sup>6-8</sup>The group names were either given by the participants directly by labelling their groups or else extracted by the evaluator by looking at the shared themes of the images in a group.

group of “autumn” images, mainly displaying leafy, red forests. Other groups created were “colourful fields”, “close-up of plants”, “boats” and “country houses”. From the resulting groups it turned out that the set of images in topic 2 was interpreted mostly as “seasons”, but sometimes also just as “countryside” or “fields”.

**Task E:** Unlike the design task in Experiment 1, the illustration task had a much more detailed specification. There were four topics in total: the holiday competition (as in Figure 6.7), a description of a tropical marine ecology course, and two advertising slogans: “*PowerHouse - In tune with nature all around the world!*” for a company producing renewable energy and “*Flash - Unleash the animal inside!*” for a company manufacturing sports clothing and equipment. People could choose two of the four topics.

Topic 1 was represented by the following groups: “beach”, “mountains”, “people”, “castles”, “food/dinner”, “ice/snow”. The groups for topic 2 were: “fish”, “corals”, “divers”, “misc water creatures”, “plants”, “night sky with stars/moon”. Topic 3 was represented by “people”, “countryside/landscapes”, “waterfalls/water” and “roads”, while the groups for topic 4 were “animals” and “sports”.

Most people created around three groups for this task and finally chose one image of each group as their final selection. The groups created by almost all people had common themes, like the ones listed above, with only two exceptions where people organised their images by their illustration qualities, eg “with space for logo and text”, “high contrast”, “landscape” and “portray”.

**Task F:** Again there were two topics for the abstract search task: *dynamic* and *cute*. While the average number of groups for Task F was 2.6, there was a noticeable difference between the two topics: 3.5 groups for topic 1 and 1.8 for topic 2. Topic 2 was represented unanimously by “baby animals” and “children”. The groups for topic 1 were more varied: “animals” (sometimes split into “flying birds”, “tigers/leopards”, etc.), “sports”, “mountains”, “waterfalls”, “sunsets/landscapes”, “boats/water”. There were also more unassigned miscellaneous images for topic 1.

The differences in the topics was reflected in the systems, too. The total number of selected images with WS for topic 1 was 19.3 and only 10.5 in CS. Topic 2, on the other hand, resulted in the opposite relation: 8.0 for WS and 25.0 in CS. This shows that CS is good for selecting many images for a specific topic (there were a lot of “baby animal” images in the collection), whereas WS is much better at supporting a broader search.

**Summary** One of the main objectives of this study was to determine whether there was any correlation between task characteristics and the way people organise images on the workspace. The observed result is that the more open or complex a task is, the more groups were created on the workspace. For these types of tasks organisation was deemed most useful and recommendations were requested more often.

The groups the participants created for any given task often overlapped in the overall themes of the groups, but not necessarily the images themselves. This shows that groups are definitely task-

dependent and hence people would possibly benefit from using and working with other people's groups. To reiterate, the two main advantages of having the workspace are:

- It leads to a better task conceptualisation, because the tasks can be divided into sub-aspects and each individual aspect can be followed up (one after the other or simultaneously). This is especially useful for open and/or complex tasks.
- It allows a progressive search process, spread across multiple sessions and multiple users. As could be seen, users tended to agree on the task aspects, so grouping has a long-term, time-saving benefit.

### 6.3.4 Discussion

We can make the following observations based on in the results analysis of Experiment 2:

- WS's effectiveness was better for the set of tasks provided in Experiment 2. Performing a task was generally perceived more successful with WS compared to CS and the interface was perceived significantly more effective for completing the tasks. Users also had to expend less effort, especially issuing fewer manual queries, in order to find a larger selection of images in WS.
- People needed more time to complete the tasks with WS. Our conclusion in Experiment 1 on the same issue was that the increased physical and cognitive effort called for a prolonged search session. By contrast, this time we found more evidence that this was mainly due to the system's ability to support the user in exploring the tasks from different perspectives. The system helped to analyse the task, explore the collection and detect and express more aspects of the task. These are all indicators that people were able to diversify their search better and follow up on multiple trains of thought simultaneously.
- Learning to use WS is still perceived as more difficult. While this trend was reflected in responses to various aspects of the whole experience, including the perception of tasks and search process, in Experiment 1, only the responses to direct questions on ease of use of the system itself resulted in lower scores for WS in Experiment 2. There often is a trade-off between ease of learning, on the one hand, and power, expressiveness and flexibility, on the other hand (Gentner & Nielsen 1996). We believe trading the latter for the former is an advantage of WS over CS, since it better supports users in solving their tasks.
- The longer learning period and increased cognitive load is not perceived as a disadvantage of WS. Only one person preferred CS in Experiment 2.
- People had more trouble with the relevance feedback approach than with the grouping and recommendations. The grouping was not only considered easier, more effective and useful, but was praised unanimously in open-ended questions. On the other hand, the relevance feedback facility caused more confusion. It became apparent that providing relevance feedback brings uncertainty as to which images to select for feedback in order to improve the results. Consequently, people relied more on the manual query facilities in CS than WS.

Although both systems have the same underlying learning mechanism, it is more intuitive to the user to provide feedback in a structured form by creating groups on the workspace instead of indicating relevant and irrelevant images indiscriminately. Thus, one can conclude that the grouping process is better at overcoming the query formulation problem.

All in all, Experiment 2 essentially reinforced the findings of Experiment 1 regarding the strengths and weaknesses of WS. Furthermore, it was shown that the effectiveness for realistic tasks was actually better in WS. There was also more evidence that the grouping and recommendations caused less confusion and were more intuitive to the users than the relevance feedback approach.

### 6.3.5 Summary

Experiment 1 helped us to identify avenues for improvement in both of the systems as well as the evaluation set-up. After having implemented these improvements, we could reinforce some of the earlier findings and revalidate some claims that could not be proven earlier. It was essential that we introduced a broader set of tasks, for example, in order to analyse the effect the organisation has on task conceptualisation. Also, more detailed questions on the tools provided in the interfaces, that is the grouping/recommendations and relevance feedback facility, made it possible to identify their strengths and weaknesses. The results finally provided sufficient evidence to accept all three experimental hypotheses:

1. The proposed approach leads to an increased effectiveness and user satisfaction.
  - The perceived effectiveness was better as well as the effort required to complete a task was lower with WS.
  - In terms of user satisfaction, WS scored higher on most aspects covered in the questionnaires and received all but praise when user's opinion was requested directly.
2. The workspace helps to conceptualise and diversify tasks.
  - Users indicated that WS helped to analyse and explore their tasks better.
  - The resulting groupings reflected our expectations of task complexity and were generally very similar amongst different users.
3. The grouping and recommendation system helps to overcome the query formulation problem.
  - The users had more problems with the relevance feedback facility than the recommendations. In the recommendations they could see which images contributed to the query, while at the same time hiding the details of the retrieval mechanism.

## 6.4 Combined Results

In this section, we present a detailed analysis of the combined experimental results with an emphasis on a task-based comparison. It provides a discussion on users' perception of task characteristics and performance, and a more objective view of user effort when attempting the tasks, in order to

Table 6.19: Task listing

Task	Description	Objective
A	Simple or focused category search tasks.	Find as many images as possible for the specified topic
B	Complex or multifaceted category search tasks.	Find as many images as possible for the specified topic
C	Design task.	Choose 3-5 images to design a leaflet.
D	Theme search tasks.	Choose one image to complement a provided set of three images of a specific theme.
E	Illustration task	Choose three images to illustrate a provided piece of text or advertising slogan.
F	Abstract topic search tasks.	Choose one image of a specified abstract topic.

analyse the specifics of each task. Furthermore, a summary of the organisation analysis should help to clarify how people use the workspace for all tasks performed with WS. It also analyses the nature of the information need and compares how each of the systems supported the user in either fulfilling or evolving their needs. Finally, the systems are compared directly in terms of usability and user satisfaction and each system's advantages and disadvantages are identified.

#### 6.4.1 Task Analysis

We have created a variety of realistic tasks, ranging from category search, an image-based theme search, abstract topic search, illustration task and an open design task. The tasks were designed to vary in terms of complexity, degree of abstraction and creativity. The participants confirmed that they were familiar with these types of tasks and that they encountered similar tasks in their own work or hobby. The tasks are described in Sections 6.2.2 and 6.3.2, and are summarised in Table 6.19. The number of users per task is specified in Table 6.20.

#### User Perception of Task Characteristics

The participants were invited to rate the task they had just performed after completing a search session. Furthermore, the search process and the resulting images were rated. The overall results per task are shown in Table 6.21 (scores from 1 to 5, higher = better).

We noticed differences in the task perception depending on which system was used. This can be explained by the fact that participants were asked to rate the task after having completed it using one of the systems. Thus the scores reflect people's perception of the task taking into account their experience with the system. Table 6.22 shows the results for the category search tasks in Experiment 1 on a per-system basis. The design task was only performed with WS, and is hence excluded from the following per-system analysis. Table 6.23 shows the equivalent results for the tasks in Experiment 2.

**Task:** There are no significant differences on any of the differentials across the various tasks.

Overall, the tasks were considered *clear*, but slightly less *familiar* (all scores are well above



Table 6.20: Number of user samples per task

Samples	Task A	Task B	Task C	Task D	Task E	Task F
CS	6	6	0	8	8	8
WS	6	6	12	8	8	8
total	12	12	12	16	16	16

Table 6.21: Semantic differentials about task perception per task

	Differential	Task A	Task B	Task C	Task D	Task E	Task F
Task	clear	4.8	4.8	4.8	4.8	4.8	4.4
	easy	<b>4.4</b>	<b>4.3</b>	3.9	3.9	3.8	3.7
	simple	<b>4.7</b>	<b>4.7</b>	3.9	3.6	3.6	3.7
	familiar	3.6	3.9	3.8	3.5	3.4	3.4
Search	relaxing	4.0	<b>4.5</b>	3.9	3.6	3.8	3.8
	interesting	3.9	3.9	<b>4.1</b>	3.8	<b>4.3</b>	3.9
	restful	3.3	3.4	3.6	3.6	3.6	3.7
Images	relevant	<b>4.4</b>	3.9	4.3	4.2	4.0	4.1
	appropriate	<b>4.5</b>	4.0	3.9	4.1	4.0	4.2
	complete	3.5	3.6	3.3	3.8	3.6	3.8

Table 6.22: Semantic differentials about task perception per system for Experiment 1

	Differential	A <sub>CS</sub>	A <sub>WS</sub>	B <sub>CS</sub>	B <sub>WS</sub>	$\overline{CS}$	$\overline{WS}$	p
Task	clear	4.8	4.8	4.8	4.6	4.8	4.8	-
	easy	4.7	4.2	4.3	4.3	4.5	4.3	-
	simple	4.8	4.5	4.8	4.5	4.8	4.5	-
	familiar	3.8	3.3	3.8	4.0	3.8	3.7	-
Search	relaxing	4.5	3.5	4.7	4.3	<b>4.6</b>	3.9	-
	interesting	3.8	4.0	3.3	4.5	3.6	<b>4.3</b>	0.016
	restful	3.3	3.3	3.3	3.5	3.3	3.4	-
Images	relevant	<b>4.7</b>	4.2	3.7	<b>4.2</b>	4.2	4.2	-
	appropriate	<b>4.7</b>	4.3	3.7	<b>4.3</b>	4.2	4.3	-
	complete	3.3	3.7	2.7	<b>4.5</b>	3.3	<b>4.1</b>	0.027

Table 6.23: Semantic differentials about task perception per system for Experiment 2

	Differential	D <sub>CS</sub>	D <sub>WS</sub>	E <sub>CS</sub>	E <sub>WS</sub>	F <sub>CS</sub>	F <sub>WS</sub>	$\overline{CS}$	$\overline{WS}$	p
Task	clear	4.6	4.9	4.8	4.8	4.4	4.4	4.6	4.7	-
	easy	3.6	<b>4.3</b>	3.8	3.9	3.8	3.8	3.8	3.9	-
	simple	3.6	<b>4.0</b>	3.6	3.6	3.6	3.8	3.6	3.8	-
	familiar	3.5	3.5	3.3	3.6	3.5	3.3	3.4	3.5	-
Search	relaxing	3.6	3.5	3.9	3.8	3.6	3.9	3.7	3.7	-
	interesting	3.4	<b>4.1</b>	4.1	<b>4.4</b>	3.3	<b>4.5</b>	3.6	<b>4.3</b>	0.009
	restful	3.8	3.4	3.6	3.5	3.6	3.8	3.7	3.5	-
Images	relevant	4.1	4.3	4.0	4.0	4.0	3.8	4.0	4.1	-
	appropriate	4.3	4.0	4.0	4.0	4.1	4.3	4.1	4.1	-
	complete	3.5	4.0	3.5	3.6	3.6	3.9	3.5	<b>3.8</b>	-

3). Task F was considered the least *clear*, but that did not have a noticeable effect on the users' perception of *ease* and *complexity* of the task (compared to Tasks D and E). The category search tasks (A and B) were both *easier* and *simpler* than all other tasks. The decision making process was less crucial in these tasks, since the objective was to find as many images as possible from a (well-defined) given category.

In Experiment 1, the tasks were considered equally *clear* and *familiar* in both systems, but more *easy* and *simple* in CS. Task A performed with CS was the winner for *simplicity*. There was a large discrepancy between *familiarity* of Task A versus Task B with WS: The simple category tasks were the least *familiar*, while the complex categories were the most *familiar*.

In Experiment 2, there is a notable difference on the perception of Task D depending on which system was used (see Table 6.23). It was considered more *clear* and *easy* and less *complex* when using WS. For the other tasks the scores were relatively balanced.

**Search process:** The search process was considered most *relaxing* for the complex categories (Task B), while all tasks were considered similarly *restful*. The design-oriented tasks (Tasks C and E) were considered more *interesting*.

In Experiment 1, the search process was considered more *relaxing* in CS, but significantly more *interesting* in WS. Task A led to the most *stressful* search process with WS.

In Experiment 2, the search process was considered most *interesting* when performing the illustration task (Task E). Looking at the same responses per system, summarised in Table 6.23, people found the search process significantly more *interesting* with WS for all three tasks in Experiment 2 (as well as in Experiment 1).

**Images:** The images received through the searches in the simple category search task (Task A) were considered most *relevant* and *appropriate*.

In Experiment 1, the retrieved images were considered equally *relevant* and *appropriate*, but significantly more *complete* in WS. In the per-task comparison, images were considered more *relevant* and *appropriate* for the simple categories with CS, while the opposite was true for the complex categories. It is also interesting to note that these two differentials scored the same on average for both the simple and complex categories with WS.

In Experiment 2, it seemed to be more difficult to find the right images for Task E. Compared to the other tasks, the images returned from the searches were considered slightly below average for the *relevant*, *appropriate* and *complete* differentials (see Table 6.21). As in Experiment 1, the responses per system revealed that the returned images were more *complete* when working with WS than with CS.

## Summary

Through the analysis of task characteristics and the resulting performance, we hope to identify the types of tasks that each system is most appropriate for. First, we looked at the task characteristics from the users' perspective. The tasks were perceived equally clear with an exception of the abstract search task which was slightly less clear. The category search was considered the easiest

and simplest, followed by the design task, the theme search and the illustration task on a par, and finally the abstract search. The search process was considered more interesting, the more creativity was asked for in a task. However, people's expectation of the appropriateness of the retrieved images was also higher for the creative tasks. Thus, the more specific the task, the more people thought the system helped to retrieve the right (relevant and appropriate) images. It also emerged that the perception of task complexity sometimes varied depending on which system the task was performed on. Most of all, the search process was considered significantly more interesting with WS for all tasks.

Next, the task performance was examined in the results analysis sections for Experiments 1 and 2 (Sections 6.2.3 and 6.3.3, respectively). We briefly reiterate our observations on the users' perception of their success in performing a given task. People were least happy with their performance in the more creative tasks, mainly due to not having had enough time to complete the task. People were more satisfied with their performance with WS, although time was a bigger issue here. On the other hand, uncertainty about the next action affected their performance more with CS. Moreover, the actual task performance for the category search tasks was consistently better in CS. However, there was no correlation between actual and perceived task performance.

Finally, we analysed the amount of user effort required to solve a task (for Experiment 2). Most time was spent on the illustration task, reconfirming user's perception on task performance in this respect. However, more images were selected and more queries were issued during the course of this task. Adaptive queries in the form of relevance feedback iterations or group recommendations were considered especially valuable for this type of task. WS helped the user to select more images for all three Experiment 2 tasks.

To conclude, we could see differences in the perception of tasks and the actual effort required both depending on the nature of the task as well as the system being used. In summary, CS seems to be particularly good for quickly finding many images for a specific/narrow topic. The strengths of WS show particularly for more complex or creative tasks. Especially if the information need is vague in the beginning, the grouping facility in WS allows the user to explore the collection and to discover and express different task aspects. Therefore, users of WS are encouraged to diversify their search and the workspace makes it possible to make a more informed decision on the final images selected from a larger set of alternatives.

#### 6.4.2 Organisation Analysis

In this section, we summarise people's organisation and the nature of their information need for all tasks performed with WS. Table 6.24 shows the relevant data per task.

We have observed earlier that the number of groups created corresponds to the number of facets the users detected and followed up on. From this perspective, Tasks A and D were represented by a single facet (approximately), while the other tasks had about 3–4 facets. Tasks C–F had clear instructions on how many images had to be selected. These targets are closely reflected in the number of groups created, with the exception of Task F. The target for Task F was to select only one image, but was represented by 2.6 groups on average. Since the topic for Task F was abstract (especially compared to Tasks A and D), people explored several alternatives, which correspond to the number of groups they created. Task B is also interesting in this respect. Although the target

Table 6.24: Organisation and information need development for all tasks with WS

Task	A	B	C	D	E	F
# Groups	1.2	4.3	4.4	1.5	2.9	2.6
# Images/Group	18.8	11.9	7.5	4.9	4.6	4.4
# Selected Images	26.2	36.8	36.6	12.3	18.6	17.8
Initial Idea	4.5	4.3	3.7	4.6	4.5	3.4
Detected more aspects	3.0	4.7	4.3	3.6	3.9	4.3
Satisfied with results	3.7	3.5	3.0	4.5	4.1	4.3
Organisation useful	3.0	4.8	4.4	4.6	4.8	4.7

was the same as in Task A, namely to find as many images as possible, people created more groups for the topics in Task B, which were more complex than the topics in Task A. Hence, the number of facets is influenced by two factors: (1) the complexity of the task; and (2) the number of images that were required for the task.

The nature of the underlying information need is captured by asking how clear people’s initial idea was (before starting the search) and if they detected more aspects while searching. The responses for their initial idea are again an indicator of how focused the tasks were perceived. Task F and C have the lowest score of initial idea, and are indeed more open to interpretation than the other tasks. As mentioned before, Task F is the most abstract and Task C the most creative. Interestingly, there is a relationship between the scores of initial idea and task aspects: they are roughly inversely proportional. So, the less defined their initial idea, the more aspects they detect during the search and vice versa. Task B is the only exception: the information need is well-defined but people also detected more aspects. This is not too surprising, because people can think of many images for the categories, for example “African Wildlife”, from the top of their head—unlike the abstract topic of Task F. Since these topics comprise a large number of facets (at least 4.3 that were detected on average) people can still detect some more during the search that they had not thought of before.

The large difference in result satisfaction between Tasks A–C and Tasks D–F can possibly be explained by the improved retrieval system in Experiment 2. Still we can see that the creative tasks (Task C in Experiment 1 and Task E in Experiment 2) have the lowest scores compared to the other tasks in the same experiment. We believe that this is due to higher expectations for these tasks. People are instructed to create a composition of images rather than select images with a specified requirement. As seen above, time restrictions were an issue affecting their performance satisfaction, probably affecting their satisfaction with the results as well.

Finally, the organisation feature was regarded as very useful. The only exception was for finding a large number of images from a focused topic. In fact, CS was generally preferred for this task.

### 6.4.3 User Satisfaction with Systems

In both Experiments 1 and 2, the participants were asked to rate the system they had just used in the post-search questionnaires. These results are given per experiment above. Nonetheless, we provide the combined results for all 24 users of both experiments in this section, since a larger

Table 6.25: Semantic differential results for the System part (Experiments 1+2)

Differential	$\overline{CS}$	$\widetilde{CS}$	$\overline{WS}$	$\widetilde{WS}$	p		$\overline{CS}$	$\widetilde{CS}$	$\overline{WS}$	$\widetilde{WS}$	p
wonderful	3.4	4	<b>4.1</b>	4	-						
satisfying	3.5	4	<b>4.0</b>	4	-	in control	3.8	4	3.8	4	-
stimulating	3.4	3	<b>4.3</b>	4	0.007	comfortable	4.0	4	<b>4.4</b>	5	-
easy	<b>4.3</b>	4	3.8	4	-	confident	3.6	4	<b>4.0</b>	4	-
flexible	2.9	3	<b>4.1</b>	4	0.000	learn to use	<b>4.3</b>	4	4.1	4	-
efficient	3.6	4	<b>3.9</b>	4	-	use	<b>4.2</b>	4	3.9	4	-
novel	3.4	3	<b>4.3</b>	5	0.005						

Table 6.26: Comparison of system rankings

System	(a) learn	(b) use	(c) effective	(d) liked best
CS	14 (58%)	10 (42%)	5 (21%)	4 (17%)
WS	5 (21%)	11 (46%)	11 (46%)	16 (67%)
no difference	5 (21%)	3 (13%)	8 (33%)	4 (17%)

sample size leads to more reliable results.

### System and Interaction

The participants considered CS more *easy* than WS, while they considered WS to be significantly more *stimulating*, *flexible* and *novel*. The scores for the remaining differentials, *wonderful*, *satisfying* and *efficient*, were generally higher for WS as well. Table 6.25 shows the results for these differentials.

While using the system, people felt more *comfortable* and *confident*. However, WS was deemed more difficult to *learn to use* and to *use*.

### System Rankings

After having completed all search tasks having used both systems, the users were asked to determine the system that was: (a) easiest to learn to use; (b) easiest to use; (c) most effective; and (d) they liked best overall. Table 6.26 shows the users' preferences of systems for each of the statements. 67% liked WS best and the majority also thought it was more effective. CS was clearly easier to learn to use, whereas the ranking for using the systems was relatively balanced.

## 6.5 Conclusion

Although a workspace has been introduced in several information retrieval systems before, for instance the *ImageGrouper* (Nakazato, Manola & Huang 2003) and *SketchTrieve* (Hendry & Harper 1997), none of these systems have formally evaluated its effectiveness. With this experiment we aimed to fill this gap and answer the following questions: How was the workspace used? What influence did the task have on this? Does it help to conceptualise tasks? Does it help overcome the query formulation problem? These are the answers we found for our specialised domain of results organisation for image retrieval:

**How was the workspace used and what influence did the task have on this?** As determined in the organisation and information need analysis, the workspace was used to create different groupings that reflected different semantic facets of the task. These facets often overlapped amongst the users for the same task.

In addition, we found a correlation between task characteristics and organisation behaviour. The workspace is most useful for exploratory searches with vague information needs or complex, multi-faceted tasks. Possible explanations include that it helps to analyse the task better, discover more aspects of the task than initially anticipated and explore the collection better.

Even for focused tasks, the organisation was still deemed useful, because it helped to maintain a better overview and hence better comparison opportunities of the selected images. For these tasks, the focus shifts naturally to selecting images with good quality rather than the pure quantity of images. In the users' eye this was a more realistic goal of image searching tasks.

**Does it help to conceptualise tasks?** Grouping search results on the workspace incites the user to organise results for their search/work task. This enabled the users to break up their overall search task into a small set of individual search tasks. By doing so, it helped users to analyse and conceptualise their tasks, and similarly their underlying information need, better.

Hence, the grouping process has the benefit of allowing the user to explore the task. People can pursue a progressive search strategy by following multiple search threads simultaneously, while maintaining a constant overview of intermediate results and searches. The groups are equivalent to task aspects, and the search threads are equivalent to trains of thought. This shows that dynamic needs are supported by encouraging an incremental and progressive search strategy.

**Does it help to overcome the query formulation problem?** The interactive grouping mechanism is successful at hiding the internals of the retrieval system without the user feeling lost or confused by the system's responses. Since the groups are equivalent to task aspects, users find it easier to categorise images into these aspects and interpret the system's results accordingly. The fact that the underlying retrieval mechanisms in the relevance feedback system and the workspace system were the same, proves the achievement towards a more intuitive search process.

It is commonly observed in the HCI community (*eg*, Gentner & Nielsen 1996) and the information seeking community (*eg*, Ruthven 2005), that the stringent separation of the user interface from the underlying technical processes is a poor strategy, since people cannot help but try to understand what goes on behind the scenes. In *EGO* we achieve a more intuitive interaction process than in the over-simplified relevance feedback approach.

We can therefore confidently conclude that the combination of the management and retrieval process achieved through *EGO*'s provision of a workspace and recommendation system is crucial for a more satisfying overall search experience for the following reasons:

- It helps to conceptualise and explore the task better.
- It supports a variety of information needs. In particular, dynamic needs are encouraged by allowing the user to follow up on different search threads—simultaneously or consecutively.
- People are more satisfied with their performance when they are able to organise their results. The system is perceived more effective.
- It is a positive step towards improving the query formulation problem.
- The workspace will prove especially useful in the long-term, since the groups are stored permanently and can be pulled out as a whole when required again. Thus, the individual search threads—or trains of thought—can be stored and pursued over multiple sessions and even shared between users.

## 6.6 Benefits of EGO

Having just concluded with EGO's advantages relating to our specific experimental hypotheses, we would like to take the opportunity here to summarise the general benefits of the proposed approach.

### 6.6.1 Benefits from the Users' Perspective

Recall that we have characterised potential user groups and highlighted their problems when using existing systems in Section 4.2.1. Here, we finally describe how each of these user groups will benefit from using EGO. These observations were gathered through discussions with the users interviewed during the evaluation described above.

#### The Hobby Designer's View

- The workspace makes it easy to store intermediate results.
- The grouping of images into concepts allows side-by-side comparison between groups and images, and make the final selection from intermediate results easier.
- The fact that images can be grouped and stored on the workspace, allows the user to branch off and explore different ideas while searching.
- The system recommendations provide clues to generate new ideas, which may not have been thought of by the user before.
- Visual search, according to one participant, is an “*incredibly useful*” alternative if initial keyword searches fail.

### The Graphic Designer's View

- It makes it easier for the user to manage their own images and build up an image library.
- Visual search facilities can be very useful for some types of searches, for instance when the ideas are difficult to put into words or for fine-tuning a search.
- The workspace allows better side-by-side comparison and facilitates exploratory searches, which help for vague information needs or ideas generation.
- The whole design process is recorded, because the workspace makes it easy to store results from the inspirational phase along with the final results (or in separate related groups).
- Long-term projects are easier to keep track of, because all groups relating to the project can be retrieved straight away.
- Collaborative usage is supported if workspaces are shared. Groups can then be populated by any member of the team. Also since each member leaves trails of actions on the workspace, someone else can more easily pick up from where another user has ended.

### The Photographer's View

- It helps to build up an image library, in which the photographs can be organised into multiple categories allowing multiple views of the collection.
- It makes search more exploratory, because the recommendation system can make you aware of images you might have forgotten about.

## 6.6.2 Summary of Benefits

With *EGO* we have proposed a contextualised retrieval and management system, which has several benefits:

- **Supporting the workflow and capturing the work task**

By analysing the context of the search system it has become clear that supporting different work tasks and capturing the flow of the work is a vital factor for a successful retrieval system (cf Section 4.1). One facet of achieving these goals is to manage not only the images but also the searches in some meaningful way, since searching and organising activities are interleaved in the typical workflow. In *EGO*, searching and organising images are coupled, which means that search results are organised and the organisation can be searched more easily. The search process aims to be closer to an individual's mental model and the resulting organisation captures the context of the work tasks.

- **Supporting opportunistic search strategies**

Lots of images can be saved and grouped on the workspace while searching. The grouping makes it easier to interleave multiple search threads, which is often necessary to support an erratic search process during the inspirational phase where information needs are vague. Additionally, it offers better side-by-side comparison and makes the selection of images from several candidates easier.



- **Creating a personal image library**

The retrieval system is crucially dependent on the underlying organisation of the image collection. One of the largest collection of images is the Web. There is no underlying organisation on the Web, which makes searching a tedious process of wading through a huge number of images in the hope of finding a few relevant ones. On the other end of the scale we can find stock image collections, which are organised rigidly based on predefined ontologies imposed by the administrator. The occasional user might have difficulty in finding out the peculiarities of the collection organisation. On the other hand, frequent users might have trouble with the rigid organisation. Neither type of collection offers the user any customisation options or ways to incorporate the searching and acquiring of images into the overall work process. Thus, such a collection is not tailored (or tailorable) to a particular user. We propose to support users in building up their personal image library while searching for images. This process is incremental and dynamic: an organisation is built up and changes through use. The resulting organisation thus reflects the user's preferences and their work tasks.

- **Collaborative Work**

Professional users often work in teams. In a collaborative context people work together and are inspired by and learn from each other's activities. If workspaces are shared, the search process can be seen in a collaborative context, in which not only the end results, but also the thought processes, are shared.

### 6.6.3 Addressed Issues

Reflecting back on the issues mentioned as the primary motivations for the design process, we can now see how *EGO* addresses the questions raised previously:

- *“What is the meaning of an image?”*

We do not claim we have solved the problem of automatically determining an image's meaning. As argued before, successful approaches have to recognise the importance of the context, which is not within the retrieval engine, but is determined by the tasks and work environment.

In *EGO* the semantic gap is narrowed by the abstraction to high-level semantic groupings. The resulting organisation from the long-term interaction, reflects a user's personal and task-specific mental model of the data. From this organisation, it is easier for the system to infer the intended semantic meaning. Our solution to incorporating such a semantic feature is described in Chapter 7.

- *“How can the user be assisted in communicating their information need?”*

*EGO* does not require the user to think in terms of the system (ie how to formulate a query, how a search works, etc.) but engages the user in an interactive organisation process to iteratively define their semantic needs. This process is closer to everyday solutions of managing information, hence affording traditional problem solving techniques and natural ways of communicating their information need. This allows the user to concentrate on their tasks rather than the system.

- “How can the time-varying nature of information needs be supported by the system?”

The interactive grouping is a flexible means to communicate both short- and long-term, specific and multifaceted information needs.

*EGO* invites the user to create groups according to the multiple facets of their need. This allows them to pursue various search threads simultaneously. The system does not need to worry about detecting changes implicitly, because the user can switch between groups that reflect their current needs.

Moreover, the groups can be created and changed over multiple sessions, so that they dynamically capture aspects of the user’s long-term need. Organised on a workspace, they leave behind trails of actions used by the system to adapt to the user’s need and enable users to trace and reflect on their actions.

To conclude, the design of *EGO* as a tool to create a task-specific organisation of images reflecting an individual’s mental model, successfully addresses many of the problems of traditional CBIR systems.

## 6.7 Summary

In a user study involving 24 design students and professionals *EGO*’s usability was investigated with emphasis on the usefulness of the organisation facility added by the workspace in its interface. A detailed analysis of task characteristics, both from the user’s perspective as well as from logging data showing user effort and organisation strategies, was presented. Further, the grouping mechanism was compared to the relevance feedback mechanism provided in the baseline system. The results showed that, although slightly dependent on the type of task being performed, the benefits of being able to organise search results generally outweighed the easier to use and faster relevance feedback system.

The results confirmed that the interaction in *EGO* is more intuitive to the user and hence closer to the user’s mental model as discussed in Chapter 4. We found evidence of the issues that have motivated the design of *EGO* introduced in Chapter 3 (see also Section 4.5). In particular, we could show that *EGO* assists the user in communicating their information need and that it supports a variety of information needs. We encountered both specific and multifaceted information needs. *EGO* is especially useful for vague and dynamic information needs as it supports interleaved and erratic search processes. The long-term aspects still have to be investigated in a longitudinal study, following a particular user over a longer period, although our analysis of the user’s groupings and the participants’ responses pointed to *EGO*’s potential benefits for long-term and collaborative usage.

The participants found fault with the quality of results in Experiment 1, and to a lesser degree in Experiment 2. In the following chapter we will introduce a new retrieval model to incorporate a semantic feature learnt from the users’ groupings (based on the usage data from the evaluation just presented). The semantic feature encodes the personalised context of the images’ usage. This will address the final issue of “What is the meaning of an image?” by bringing the feature representation closer to the user’s interpretations of the images.

---

## THE PERSONALISED RECOMMENDATION SYSTEM

---

The recommendation system introduced in Chapter 5 is the backbone of the image management system in *EGO*. It supports a fast and simple interactive organisation process, allowing the user to create groups of images and thus build up a personal organisation of the collection. However, relying solely on content-based features for the recommendation system is not sufficient to detect the common semantic concepts of a group of images, as became apparent in Experiment 1 of the user evaluation described in the previous chapter. We argue that the additional information that can be gained from a user's previous organisation behaviour will help to discover and disambiguate common semantic links between images.

The problems addressed in this chapter are: how to capture and model personalised usage information to improve retrieval performance; and how to integrate this information with other feature modalities (visual and textual) to model interdependencies between features. The proposed approach models both feature similarities and semantic relations in a single graph. Retrieval in this model is implemented using the theory of Random Walks. In addition, ideas on how to implement short-term learning from relevance feedback are presented. Systematic experimental results validate the effectiveness of the proposed approach for an improved recommendation system that takes into account personalised, contextual information. Moreover, the model is generic and can be used for image retrieval purposes under different circumstances and could even be extended to other domains. A summary of the results was published in (Urban & Jose 2006a).

### 7.1 Introduction

The common thread in this work is to find a way to narrow the semantic gap so as to provide a more efficient and effective image retrieval system. The user evaluation in the previous chapter has shown that the proposed search environment helps to overcome the query formulation problem by letting the user split up their tasks into related facets. These facets materialise as groups of related images on the workspace. From the user's perspective, the interactive organisation process helps to close the semantic gap, since they can and do group images based on semantic concepts (cf Sections 6.2.3, 6.3.3 and 6.4.2). If that information is incorporated in the retrieval system as a "semantic feature" we can build a bridge between the user and the system.

Semantic image retrieval is a topic of growing research interest with the goal to replace the low-level feature space with a higher-level semantic space, which is closer to the abstract concepts the user has in mind when looking for an image. Section 2.2.4 introduced a selection of proposed techniques to achieve this. In this section we categorised the existing attempts towards semantic features into two classes: *annotation-based* (eg, Jeon et al. 2003, Pan et al. 2004) and *user-based* (eg, Su & Zhang 2002, Han et al. 2005, He et al. 2005) depending on the nature of the knowledge-base used to learn semantic concepts.

Our approach is an implementation of the latter because contextual information is mined from user interaction. We use the groupings created in the user experiments (cf Chapter 6) to infer a semantic feature. Our underlying assumption is that all objects (images) in one group share some semantic concept (user, usage and task-dependent), eg images of snowy mountains, images with high visual contrasts, images that could be used as a background on the front of a flyer. This assumption seems reasonable based on the analysis of the groupings created by the users in the evaluation presented in Chapter 6: people created groups of images reflecting task-related concepts. Instead of trying to infer and label these concepts, however, we simply record that there is a semantic relation between images in a group. We refer to these relationships as *peer information*. Appropriately recorded, the peer information can be used to implement long-term learning of semantic concepts in the system.

The advantage of this approach is that the semantic concepts are tailored to the user's expectations and interpretations. After all, semantics are about *interpretation*, and the interpretation is, to a large extent, *domain or context dependent*. The resulting concepts will tend to reflect the meaning that is bestowed upon an image by a human observer regarding the context of both the observer(s) and the image.

In addition to the peer information, low-level visual features and textual annotations are further sources of information for the retrieval (and recommendation) system. However, the combination of different feature modalities is a big challenge in multimedia retrieval (TrecVid 2005, Iyengar et al. 2005, Tong et al. 2005). Most state-of-the-art systems treat each feature individually and fuse the result lists to obtain the final results. However, the method of fusion is far from obvious and such systems fail to capture dependencies between the features. Even worse, such systems have difficulties in exceeding the performance of a text-only system in information retrieval tasks (TrecVid 2005).

In our simple recommendation system, we adopt a separatist approach, and use a rank-based voting approach to combine the three sources of evidence. In addition, we have developed a more advanced model. Instead of a late fusion of results, we propose to integrate the different modalities in a single graph and use the theory of Random Walks (Lovasz 1993) to calculate retrieval results. In this model, images, terms and visual features are represented as nodes in an *Image-Context Graph* (ICG). The links between nodes represent: image attributes (relations between images and their features); intra-feature relations (feature similarities); and semantic relations (peer information).

We describe a retrieval model, based on Random Walks, that can retrieve both top matching images, as well as terms, to a query (consisting of both image examples and terms). Random Walks have most famously been used for information retrieval purposes in Google's PageRank

algorithm, which estimates the authoritative quality of a Web page depending on how many pages link to it (Brin & Page 1998). In addition, we show how short-term relevance feedback learning can be integrated into our model by adapting the link weights in the ICG. The main contributions of this chapter are:

- We propose a group-based contextual feature (peer information) based on mining usage information while searching in a multimedia collection.
- We show how the peer information can be integrated with already existing low-level visual features and textual annotation in a graph model.
- We define various learning strategies in the graph model.
- Through systematic experimentation the effectiveness of the proposed approach is validated and learning strategies are investigated.

The remainder of this chapter is organised as follows. We start by introducing the individualist retrieval system in Section 7.2 followed by the graph-model in Section 7.3. This section also reviews related work and explains the mathematical background. We outline the experimental methodology in Section 7.4, followed by the experimental results in Section 7.5, where the individualist retrieval system is used as a baseline. Finally, we summarise the chapter and point to future work in Sections 7.7 and 7.6.

## 7.2 Individualist Retrieval System

In the individualist approach, the three features, peer, textual and visual, are indexed independently. When a query is issued, each index is consulted to retrieve images relevant to the query in a ranked order. Before returning the overall query results, the individual ranked lists are combined to produce a single result list. This section describes how the three features are represented and queried, and finally how the results are merged.

### 7.2.1 Peer Feature

To model the group context in its most simplistic form we count the number of co-occurrences between images. If two images belong to the same group, their co-occurrence score is incremented by one. This information can be represented in a square matrix  $M$ , whose rows and columns are the images in the collection. The entry  $m_{i,j}$  denotes the number of groups that contain both image  $i$  and image  $j$ . The diagonal of  $M$  is set to 1.  $M$  is symmetric, thus  $m_{i,j} = m_{j,i}$ , since if image  $i$  co-occurs with image  $j$ , then image  $j$  also co-occurs with image  $i$ . In other words, images  $i$  and  $j$  are considered peers. Each image is then represented by a vector (the corresponding row in  $M$ ). In order to compute the similarity between two images, we can use the theory of the vector space model developed in the text retrieval domain (Salton & McGill 1983).

In order to interpret this information in parallel with traditional textual IR, we can consider the images in the collection the vocabulary, or the terms, with which our documents (the same images) are annotated. Each image is then represented by a term-vector that encodes its peers. First of all, we assign each term  $j$  a weight in document  $i$ :

$$w_{i,j} = \text{tf}_{i,j} \times \text{idf}_j \quad (7.1)$$

where traditionally  $tf_{i,j}$  is the term frequency (how often term  $j$  appears in  $D_i$ ),  $idf_j = \log_2 \frac{N}{df_j}$  the inverse document frequency,  $df_i$  the document frequency (how often the term appears in the whole collection) and  $N$  the number of documents in the collection. In the peer index,  $tf_{i,j}$  measures how often two images  $i$  and  $j$  co-occur, while  $idf_j$  measures the general importance of image  $j$  depending on how many times this image co-occurs with other images in the collection. The term frequency  $tf_{i,j}$  is equivalent to the corresponding entry  $m_{i,j}$  in the peer matrix  $M$ .

The similarity between two documents (images) is traditionally computed based on the cosine between their term-vectors. The *cosine similarity* between a query vector  $Q$  and a document  $D_i$  is defined as:

$$sim(Q, D_i) = \frac{\sum_{j=0}^V w_{Q,j} \times w_{i,j}}{\sqrt{\sum_{j=0}^V (w_{Q,j})^2 \sum_{j=0}^V (w_{i,j})^2}} \quad (7.2)$$

where  $V$  is the total number of terms and  $w_{Q,j}$  is the weight of term  $j$  in the query.

However, since the vocabulary in our case is very large ( $V = N$ ) and the vectors can be expected to be very sparse, the exact vector-space model is expensive to implement. The text IR community again has found a solution to this problem in the form of an inverted index, in which each term is stored with its postings list. The postings list for a particular term  $j$  is a list of documents that contain this term (together with the term weight in the document  $i$ ,  $tf_{i,j}$ ).

So instead of storing the whole matrix  $M$ , we create an inverted index. The posting list in our case contains a reference to all peers of a given image. Instead of having to compare  $N$  vectors given a particular query, the inverted index facilitates a fast computation of the relevant results, since we only have to iterate over the query images rather than all images in the collection. The querying process is specified in Algorithm 1.

The normalisation in line 7 in Algorithm 1 discards the effect of document length on document scores. The exact normalisation of scores is extremely expensive again, since it requires access to *every* term (see denominator in Equation 7.2). To approximate the effect of normalisation, we can base it on the number of terms in a document (Lee et al. 1997). The document scores in Algorithm 1 are then normalised by:

$$score(d) = \frac{score(d)}{\sqrt{\text{number of terms in } d}} \quad (7.3)$$

Since the peer index is symmetric, the number of terms in a document  $i$  (that is the number of peers of image  $i$ ) is equal to the number of documents containing the term, which is given by the postings list size.

---

**Algorithm 1** Query processing with inverted index

---

- 1: **for** every query image  $q$  in  $Q$  **do**
  - 2:   retrieve the postings list for  $q$  from the inverted index
  - 3:   **for** each peer  $d$  indexed in the postings list **do**
  - 4:      $score(d) = score(d) + tf_{d,q} \times idf_q$
  - 5:   **end for**
  - 6: **end for**
  - 7: Normalise scores.
  - 8: Sort documents according to normalised scores.
-

### Relevance Feedback

In an interactive search session, the user can provide feedback in the form of relevant and irrelevant images for a particular query (in the case of manual queries) or group (in the case of recommendations). In the interactive setting in *EGO* as described in Section 6.3, we have adopted an implicit negative feedback strategy for groups (cf Section 6.3), where an image is considered as negative feedback after having been ignored from the recommendation set (top 10 returned images) in three consecutive turns. However, for the purpose of the experiments described in this chapter, we employ a simulated setting. Therein, we explicitly use both relevant and non-relevant images, as determined from the ground-truth (cf Section 7.4.3), amongst the top 10 results (recommendation set) as positive and negative feedback, respectively.

This section describes how we have implemented short-term learning with the peer feature. In addition, we describe how the peer index is updated when images are added permanently to a group, which represents the system's long-term learning capability.

**Short-term Learning in an Inverted Index** Rocchio's algorithm is widely used in text retrieval to incorporate relevance feedback (Rocchio 1971). Rocchio's formula for the query vector is updated based on the set of positive examples  $P$  and negative examples  $N$ :

$$Q' = \alpha Q + \beta \left( \frac{1}{|P|} \sum_{x \in P} x \right) - \gamma \left( \frac{1}{|N|} \sum_{x \in N} x \right) \quad (7.4)$$

where the parameters  $\alpha$ ,  $\beta$ ,  $\gamma$  are typically chosen experimentally. The objective of this method is to move the query point closer towards relevant documents and further away from non-relevant documents.

In the case of an inverted index Rocchio's method can be implemented efficiently by adjusting the weights for each query term  $t \in Q \cup P \cup N$ :

$$w'_{Q,t} = \alpha w_{Q,t} + \beta \left( \frac{1}{|P|} \sum_{t \in P} w_{Q,t} \right) - \gamma \left( \frac{1}{|N|} \sum_{t \in N} w_{Q,t} \right) \quad (7.5)$$

These updated weights are used to compute the document scores in Algorithm 1 (cf Equation 7.1).

**Long-term Learning** In addition to adjusting the query term weights on a per-query basis, the overall peer weights in the inverted index are updated. So whenever an image is added to a group, the new image is added to the postings list of all group images and all group images are added to the postings list of the new image. If an image already exists in a postings list, its weight is incremented by the relevance score (typically 1). In reverse, if an image is considered a negative example to the current group, the weights between the negative example and the group images are decremented. Parallel to the long-term learning strategy in the ICG (cf Section 7.3.6) other negative strategies could be implemented.

### 7.2.2 Textual Feature

Some Corel images contain annotations obtained from (BerkleyCorel 2005). We use both the keyword as well as the description fields to annotate the images.

Similarly to the peer feature, the textual annotations are stored in an inverted index. The querying and relevance feedback process are the same as for the peer feature, with the exception of the long-term learning facility. The long-term facility in the peer index is implemented by adding or removing images from the postings list when they are selected as positive or negative feedback, respectively. In the term index, we have refrained from changing the index structure in the long-term, which would result in changing the annotations of the images. Rather, we have assumed that the given annotations are accurate albeit possibly incomplete. Learning and updating annotations is a whole research area in itself, as was discussed in Section 2.2.4.

### 7.2.3 Visual Feature

The visual feature model has been elaborated in Chapter 5. Section 5.1.1 details how images are represented and Section 5.1.2 describes the relevance feedback learning technique applied for visual features.

### 7.2.4 Combination of Results

In the simple retrieval model three separate result lists are computed and then combined to arrive at the final results. The evaluation in Chapter 5 has shown that the *voting approach* is a robust technique to combine various heterogeneous result lists (see Section 5.2.1). Therefore, it is also employed here. Please note, however, that the late fusion of results is an intricate topic in itself (Lee 1997, TrecVid 2005, Iyengar et al. 2005, Tong et al. 2005). Rather than studying better fusion algorithms, we have concentrated our efforts on finding an integral approach, as will be discussed in the following.

## 7.3 The Image Context Graph – ICG

Recall from Chapter 4 that the missing link in the holistic view implemented in *EGO* is to define a higher-level semantic feature (cf Section 4.5). We have argued before that such a feature should be based on user context. The context, as we have defined it here, is gathered from the user's groupings. The groupings define the context in which images are used and can thus capture semantic relationships between images.

In the individualist approach described above this is addressed by the peer index. It models semantic relations by counting co-occurrences of images. The problem with the individualist approach is that the three features (peer, textual and visual) are treated independently. Thus, any interdependence between features is essentially lost. We can now encode semantic concepts as understood by the user, but cannot leverage or generalise this information in any way. For example, an image can only be retrieved if it either has been recorded as a peer in the peer index previously, or if it is similar in (visual and textual) content to the query images. What we cannot easily do, however, is retrieve an image that is similar in content to *other* images that share semantic concepts



with the query images. This problem has been addressed by Yang et al. (2004), for example, by introducing a pseudo relevance step, in which images that have peer links to the query image are used to expand the visual query examples. This approach, however, constitutes only one step of generalisation.

We propose to go further than this and tackle the problem as a whole. In order to capture the interplay between features and support greater generalisation capabilities, they need to be modelled in context. The idea is to represent images and all their attributes (features) in a graph. The graph consists of a number of levels of vertices: vertices for all images in the collection, and one level of vertices per implemented feature. These levels will contain both visual and textual features. There are two different types of edges connecting vertices: edges between the image vertices and their attributes (“attribute edges”); and edges representing the similarity amongst vertices in the same layer (“similarity edges”). These edges are constructed based on the similarity between features (similarity between visual feature vectors, similarity between terms) or semantic relationships/co-occurrences of images. Thus the graph represents the images in context and in the following it is referred to as the *Image Context Graph*, or *ICG*. An example graph containing three image nodes ( $I_1, \dots, I_3$ ), four term nodes ( $t_1, \dots, t_4$ ) and two types of visual features ( $f_1, f_2$ ) is depicted in Figure 7.1.

Moreover, the edges in this graph can be weighted. Conceptually, the weight of an edge defines the probability of moving from a node  $x$  to some other node  $y$ . For the first type of edges, these weights represent the current feature weights (ie textual versus visual versus peer weight). The second type of edges are associated with a weight proportional to the similarity between the two vertices (similarity of two visual features, similarity between terms, number of co-occurrences between images).

The general recommendation problem (or retrieval problem for that matter) can be stated as: Given a query, consisting of image examples and/or terms, compute the most similar images to recommend to the user. In the ICG, this translates to: given a start set of vertices in the graph, compute those image vertices that are most likely to be reached starting from the start set.

A solution to this problem can be found in the theory of Random Walks. The likelihood of passing a node in the ICG is given by calculating the stationary distribution of the Markov chain induced by the ICG. By setting the restart vector to the nodes representing the query items, we can stage a Random Walk with Restarts on the ICG. This is equivalent to computing a query-biased “PageRank” of the ICG as will be explained in the following section.

### 7.3.1 Related Work

The theory of Random Walks (Lovasz 1993) has been applied to information retrieval in the form of Google’s famous PageRank algorithm invented by Brin & Page (1998). The idea can be sketched as follows. Imagine a random surfer on the Web choosing to follow a link on each page at random. Occasionally, the surfer gets stuck in a dead end or in cycles, or simply gets bored. At these points, he may randomly jump to another page on the Web and not follow a links. The goal of a page’s PageRank score is to reflect its quality depending on the number of other pages linking to it based on the random surfer model. The PageRank algorithm can be viewed as a Random Walk on the Web graph. The mathematical details will be elaborated in Section 7.3.2.

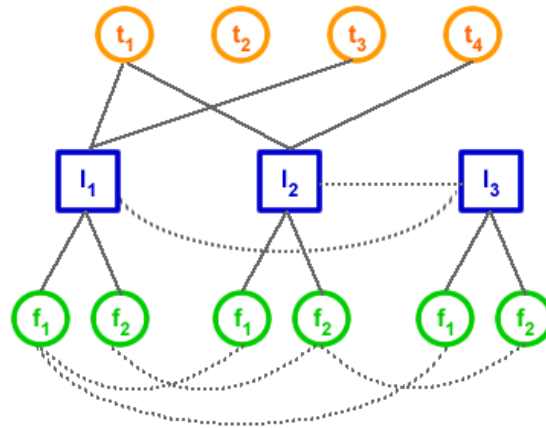


Figure 7.1: An example image-context graph

### Random Walks in the Image Domain

Graph-based modelling techniques have recently found their way into the image domain. The two most closely related approaches include its application for relevance feedback learning (Han et al. 2005, He et al. 2005) and for image captioning (Pan et al. 2004). Han et al. (2005) have proposed modelling the relationships between images based on their co-selection in relevance feedback sessions. The ratio of the frequency of two image being labelled as positive examples in the same retrieval session over the total frequency of them having been selected together (as positive or negative samples) determines the weight of the link between these two images. The calculation of a semantic similarity measure between two images is based on the overall correlation as determined by analysing the resulting graph (referred to as the image link network). An overall similarity measure is defined as a weighted linear combination of the semantic similarity and the low-level feature similarity. In contrast, the theory of Random Walks is explicitly employed on an image graph in which links between image nodes are also constructed from relevance feedback information by He et al. (2005). Here the graph is constructed by adding two special nodes to the graph: a positive absorbing node and a negative absorbing node. Each positively labelled image receives a link to the positive absorbing node, while negative examples are directly linked to the negative absorbing node. As this approach is not discriminating between query sessions, it can only be used for short-term learning.

The second application of Random Walks in the image domain is to automatically learn annotations for previously unlabelled images as proposed by Pan et al. (2004). A graph, called GCap, is constructed, which contains one node per image, a node for each image region per image and a node for all terms in the vocabulary. Images are connected to its region nodes and the terms it is annotated with. Further, regions are linked to their  $k$ -nearest neighbours. Given an unlabelled image, ie an image node  $I_i$  that does not have any links to a term node in the graph, a Random Walk is performed to compute the most probable terms for this image. These are found by calculating the long-term (stationary) probabilities that a Random Walker finds himself at a particular node given that he randomly restarts the walk from  $i$ . The top  $t$  terms with the highest stationary probability are returned as the suggested labels.

Finally, Microsoft Research Asia has been employing graph theory for multimedia retrieval and clustering (Tong et al. 2005, Wang et al. 2005). They have concentrated on creating independent graphs for the feature modalities under consideration (for example visual, textual and link information in (Wang et al. 2005)) and then fusing the results in (Tong et al. 2005) or selecting a training set for a classifier in (Wang et al. 2005). Furthermore, they do not consider encoding usage information at all. On the contrary, Lin et al. (2005) use a content-graph in combination with two relevance feedback graphs (one for positive and negative relations, respectively) to learn a reduced dimensionality space to represent images. Again, the information is encoded in independent graphs.

The semantic link approaches (Han et al. 2005, He et al. 2005) only model the information gained from relevance feedback which has to be combined with feature-based similarity values in a further step, while the image captioning approach (Pan et al. 2004) only models image-feature similarities without any ability to adaptation to relevance feedback. We propose to model both the image-feature relations as well as inter-image (or semantic) relations together. Hence there are two vital ingredients to our approach: the feature integration of semantic as well as low-level features using a graph-model; and a learning strategy in the graph model. The latter incorporates two levels of feedback to implement short- and long-term learning from user feedback. By adding links between images that are grouped together the semantic network is iteratively constructed and enforced by using adaptive link weights, thus implementing a *long-term learning* strategy. Further, we show how *short-term learning* can be achieved by introducing feature weights to ensure that those links to feature nodes with a strong feature weight are favoured over feature links with small weights given a particular query.

### 7.3.2 Mathematical Background

A Random Walk is a finite-state Markov chain that is time-reversible<sup>7-1</sup>. Markov chains are frequently used to model physical and conceptual processes that evolve over time, for example the spread of disease within a population or the modelling of gambling. An introduction to Random Walks and Markov chains can be found in (Lovasz 1993).

Let the Markov chain  $\mathcal{M}$  consist of a finite number of states, say  $N = \{1, 2, \dots, n\}$ , and probabilities of a transition occurring between states at discrete time steps. The (one-step) *transition probability*  $p_{ij}$ , denotes the conditional probability that  $\mathcal{M}$  will be in state  $j$  at time  $t + 1$  given that it was observed in state  $i$  at time  $t$ . In general,  $p_{ij}^k$  denotes the probability that  $\mathcal{M}$  proceeds from state  $i$  to state  $j$  after  $k$  transitions. The *transition probability matrix*  $P = [p_{ij}]$  is often used to represent  $\mathcal{M}$ . The stationary distribution  $\pi^T = [\pi_1, \pi_2, \dots, \pi_n]$  represents the long-run proportion of time the chain  $\mathcal{M}$  spends in each state.  $\pi$  is also referred to as the steady state probability vector. Markov chains are often represented as a graph, or state transition diagram  $G$ . Finally to make the connection to PageRank: the PageRank scores are equivalent to the stationary distribution  $\pi$  of the Markov chain associated with the Web graph.

<sup>7-1</sup>The time-reversibility criterion implies that a Random Walk considered backwards is also a Random Walk. More formally, a Markov chain is time-reversible if  $\forall \text{state } i, j: \pi(i)p_{ij} = \pi(j)p_{ji}$ , that is in a stationary walk we step as often from  $i$  to  $j$  as from  $j$  to  $i$ .

### Calculating $\pi$

In general the stationary distribution,  $\pi$ , of a Markov chain can be found by solving the following eigenvector problem:

$$\pi = \bar{P}^T * \pi \quad (7.6)$$

A unique stationary distribution is guaranteed to exist, iff  $\bar{P}$  is a stochastic, irreducible matrix (Langville & Meyer 2004).

In the PageRank model, a transition probability matrix  $P$  is built from the hyperlink structure of the Web. To create a stochastic, irreducible matrix, Brin and Page suggested to eliminate dangling pages (pages with no outlinks) by linking them to all other pages in the Web (Brin & Page 1998). This is achieved by replacing  $0^T$  rows of the sparse matrix  $P$  with dense vectors, that is the uniform vector  $\frac{1}{n}e^T$  initially or a more general probability distribution over all pages  $v^T$ . The updated matrix,  $\bar{P}$ , that does not contain any dangling nodes is defined as:

$$\bar{P} = P + a v^T \quad (7.7)$$

where  $a$  is a vector whose elements  $a_i = 1$  if row  $i$  in  $P$  corresponds to a dangling node, and 0 otherwise; and  $v$  representing a general probability distribution over the nodes—often referred to as the personalisation or restart vector. In order to ensure that a stationary distribution *exists*, the chain further has to be irreducible, that is the corresponding graph is bipartite and strongly connected. This can be guaranteed by directly connecting every node to every other node, making sure that the additional edges receive a very small but nonzero transition probability. This second stochastic fix can be modelled by the following transformation (Langville & Meyer 2004):

$$\bar{\bar{P}} = (1 - \alpha)\bar{P} + \alpha e v^T \quad (7.8)$$

where  $e$  the vector of all 1s; and  $0 \leq \alpha \leq 1$ . Substituting  $\bar{\bar{P}}$  in Equation 7.6 then leads to:

$$\pi = ((1 - \alpha)P + ((1 - \alpha)a + \alpha e)v^T)^T \pi \quad (7.9)$$

$$\pi = (1 - \alpha)(P + a v^T)^T \pi + \alpha v \quad (7.10)$$

with the constraint that  $\pi$  is normalised, such that  $|\pi| = 1$  and thus  $e^T \pi = 1$ .  $\alpha$  is then the probability of restarting the Random Walk from any of the nodes in  $v$ .

### Parameters of the PageRank Model

$\alpha$ : The value of  $\alpha$  denotes the probability of a surfer choosing to jump to a new Web page (teleportation), while they choose to click on hyperlinks with probability  $(1 - \alpha)$ . A small  $\alpha$  places more emphasis on the hyperlink structure of the graph and much less on the teleportation tendencies, and also slows convergence of the iterative computation of PageRank. Originally  $\alpha = 0.15$  was proposed (Brin & Page 1998).

In the image annotation graph of Pan et al. (2004) a value of  $\alpha = 0.65$  was found to be better suited, which they could explain by a relationship to the estimated diameter of the graph.

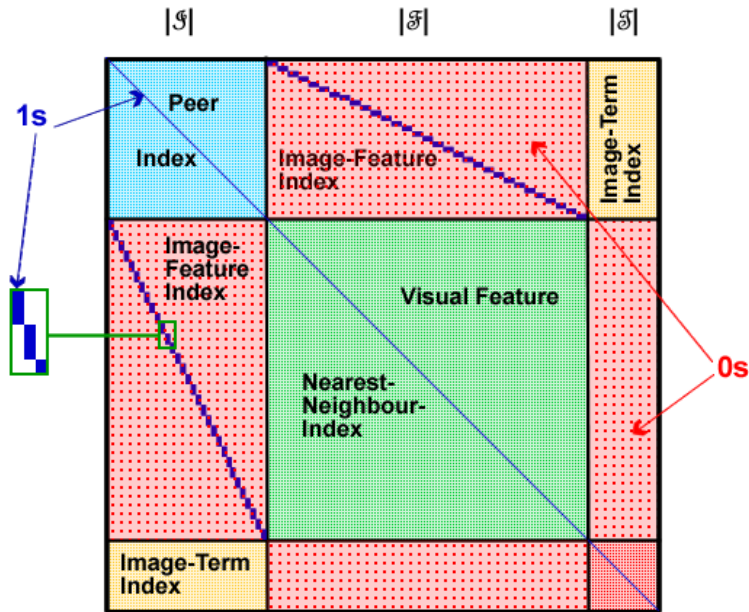


Figure 7.2: The adjacency matrix for the ICG

**The personalisation vector  $v^T$ :** Instead of the uniform distribution  $\frac{1}{n}e^T$ , a more general distribution  $v^T$  can be used in its place.  $v^T$  is often referred to as *personalisation vector* or *restart vector* in Random Walk terms.

The personalisation vector also allows PageRank to be made query-sensitive. The original PageRank assigns a score to a page proportional to the number of times a *random* surfer would visit that page, if they surfed indefinitely, following all outlinks with equal probability or occasionally jumping to a random new page chosen with equal probability. If we change the probability distribution given by the personalisation vector  $v^T$  then we can introduce a certain bias that the surfer jumps to pages with high probability in  $v^T$ .

### 7.3.3 Constructing the ICG

Let  $G$  be the ICG and  $V$  the set of vertices in  $G$  and  $E$  the set of edges. Then  $G = (V, E)$ . The graph will be stored in the form of its adjacency matrix  $M$  (see Figure 7.2).

#### The Nodes

There are three types of nodes: image nodes; term nodes; and feature nodes.

**Image Nodes:** Let  $\mathcal{I}$  denote the set of all image nodes in  $G$ . Add one node per image to the set of image nodes.  $I_i$  denotes the node for image  $i$ .

**Term Nodes:** Let  $\mathcal{T}$  denote the set of all term nodes in  $G$ . Add one node for every term in the vocabulary to  $\mathcal{T}$ .  $t_i$  denotes the node for term  $i$ .

**Visual Feature Nodes:** Construct the set of visual feature nodes  $\mathcal{F}$  by adding one node per low-level visual feature for each image. If the number of implemented visual features is  $F$  (which is 6 in our case), then  $|\mathcal{F}| = F \times |\mathcal{I}|$ .  $f_{ij}$  denotes the node for the  $j$ -th feature of image  $i$ .

Then  $V = \mathcal{I} \cup \mathcal{T} \cup \mathcal{F}$ .

### The Edges

There are two types of edges: attribute edges and similarity edges. The first type of edges link images to their attributes, the second type of edge links nodes of the same feature type (term and visual feature nodes) based on the similarity between these nodes. A special type of similarity edges are peer edges. These are edges between image nodes themselves, which are created based on users' groupings of images (user defined similarity).

**Attribute Edges:** Each image node  $I_i$  is linked to all its features. Thus an edge is created to each of its visual feature nodes  $f_{i1}, \dots, f_{iF}$ . For the textual features, an edge is created between an image node  $I_i$  and a term node  $t_j$ , if image  $i$  is annotated with term  $j$ .

**Similarity Edges:** Similar to (Pan et al. 2004, Lin et al. 2005), we propose to create edges between visual features based on their nearest neighbours. Consider a feature node  $f_{il}$  representing the  $l$ -th feature of image  $i$ , then compute the top  $k$  nearest neighbours by calculating the similarity score between the feature vector  $\vec{f}_{il}$  and the feature vector  $\vec{f}_{jl}$  for all other images  $j$  ( $0 < j < |\mathcal{I}|$ ). This allows for an adaptive definition of closeness without having to fix a threshold value.

A similar idea could be applied to the term nodes by choosing a similarity measure between terms based on relationships between terms (eg using WordNet) or a collection-based analysis. Since the number of terms contained in an image (annotations) is typically very low (compared to text documents), a collection-based analysis is probably not very significant. Instead we adopt a simple similarity measure  $sim(t_i, t_j) = 1$  if  $i = j$  and 0 otherwise. Using this similarity measure, we will obtain an edge that links each term node to itself.

**Peer Edges:** Finally, the edges between the image nodes themselves are based on user feedback. For each group created by a user, edges are created connecting all the images in that group. An edge between two images  $i$  and  $j$  has a weight, which generally reflects the frequency of these images co-occurring in groups. However, the weight can also be reduced by negative feedback (see below). These edges represent high-level semantic relationships between images based on their usage.

#### 7.3.4 Maintaining the ICG

The graph has to be maintained when new nodes are added and edges are updated. The former only takes place when new terms or images are inserted in the collection or a new visual feature is indexed from scratch. Updating edges occurs more frequently to reflect both short and long-term learning.

##### Adding new Nodes

**Adding a new Term** This is the easiest of all cases, assuming we add a term that no image is annotated with. In this case, a new node is created for the term and only one edge linking the term to itself is added.

**Adding a new Visual Feature** Assume a new low-level visual feature should be added to the graph. Then for each image, the new feature must be extracted and stored in vector format. A new node is created for each feature vector and the  $k$ -nearest neighbours for the new feature nodes are computed. The number of nodes to be added is proportional to the images already contained in the graph,  $|\mathcal{I}|$ , and the number of new links is  $k \times |\mathcal{I}|$ .

**Adding a new Image** Adding a new image to the graph is computationally expensive, since the nearest neighbours for *all* features have to be recomputed. We need to add a new image node,  $I_n$ , and create a link to all terms that the image is annotated with (if a new term appears we also have to add this term to the graph). Then, the visual features have to be extracted and added to the graph as nodes with attribute links from  $I_n$ . Finally, all nearest neighbour links have to be deleted and recomputed. To speed up this process, we can temporarily only compute the nearest neighbours for the new image's feature nodes, accepting that this graph is only an approximation (since the new feature nodes might be nearest neighbours to some of the existing features). Occasionally, the graph can be brought up to date by recomputing the nearest neighbours for the whole graph offline.

### Updating Edges

**Updating Peer Edges** The image nodes and the edges between them form the semantic network of the feature-context graph. The relationships between images are encoded in these edges with a certain strength reflected by its weight. Peer edges are updated in response to relevance feedback received from the user, similarly to the long-term learning strategy implemented in the peer index (cf Section 7.2.1). Each time an image is added or removed from a group, the semantic network is updated. Further negative feedback can be incorporated by a discount factor to increase link weights if an image is regarded as a negative example for a given group of images. In the current implementation of *EGO*, negative feedback is implicitly gathered when an image has been ignored three times from the recommendation set, or when an image is explicitly removed from a group. As explained in Section 7.2.1, however, the implicit negative feedback strategy is replaced by an explicit one for the purpose of the simulated experiments described later in this chapter.

Peer edges are created or their weight is updated as follows;

- If an image  $i$  is added to a group, a new link is added between  $I_i$  and all other images in the group. Similarly, links are added in the reverse direction (from the images in the group to  $I_i$ ). If a link between two images already exists, its weight is incremented by 1.
- If an image  $i$  is explicitly removed from a group, the link weight between  $I_i$  and the remaining images in the group is decreased by 1. Similarly, the reverse link weights are decreased. (The user simply changes his mind about an image.)
- If an image  $i$  is considered as negative sample for a particular group, the link weights (of links in both directions) are divided by a discount factor  $d$  (eg  $d = 5$ ). If the resulting weight is below 1, the link is removed completely. The larger  $d$ , the more negative feedback affects the overall structure of the semantic network (negative feedback outweighs positive feedback if two images receive contradictory feedback). See also Section 7.3.6.

---

**Algorithm 2** Calculating the query results based on a Random Walk on ICG

---

**Require:** Query consisting of image examples and query terms;  $M$  the adjacency matrix of the ICG; constant  $0 < \alpha < 1$ ;

**Ensure:**  $\|\pi\|_1 = 1$  ( $L_1$  norm of  $\pi$ )

- 1: Initialise personalisation vector  $v$ .
  - 2:  $M' = \text{normalise}(M)$ .
  - 3: Initialise  $\pi^0 = v$
  - 4: Set  $k = 0$  the number of iterations.
  - 5: **while** not converged **do**
  - 6:    $\pi^k = (1 - \alpha) * M' * \pi^{k-1} - \alpha * v$
  - 7:   Normalise  $\pi^k$ .
  - 8:    $k = k + 1$
  - 9: **end while**
  - 10: **return** Image documents sorted by their  $\pi$  values after convergence.
- 

This update strategy reflects the long-term learning capability of the system. Over time, semantic relations between images are created, enforced and memorised by the system.

**Updating Feature-Attribute Edges** The edges between an image and its features (low-level and terms) are part of the fixed structure of the graph. Unless a new low-level feature is added, or an image receives new term annotations, these edges remain unchanged.

However, to implement short-term learning from relevance feedback, the weights of these edges can be scaled (temporarily) to reflect the overall weighting between the visual features, terms and peers (see below in Section 7.3.6).

**Updating Similarity Edges** Again, these edges are part of the fixed structure of the graph. They only have to be recomputed if the similarity measure between the features changes. This event is not catered for in the current implementation. A change in the similarity measure would mean that the graph would need to be completely reconstructed.

### 7.3.5 Evaluating a Query

The objective of retrieval in the graph is to find those image nodes  $\in \mathcal{S}$  that are closest (or best connected) to the query nodes. The overview of the algorithm is as follows. First, the restart vector is built from the query nodes. Then, a Random Walk with Restarts is performed on the graph to estimate the stationary probability distribution  $\pi$ . Finally, the image nodes are returned to the user sorted in descending order by their steady state probability scores. Algorithm 2 shows an overview of these steps.

**Construction of the restart/personalisation vector** Assume a query contains a number of image examples and a set of terms. The personalisation vector  $v$  is initialised, such that  $v(u) = \frac{1}{q}$  for all nodes  $u$  representing the image examples and terms, where  $q$  is the size of the query. The remaining elements are set to 0. In the event that a query example comes from outside the collection, it will be replaced by its nearest neighbour image from within the collection. Choosing



the personalisation vector this way ensures that these nodes are favoured in the following Random Walk computation.

**Calculating  $\pi$**  Recall from Section 7.3.2 (cf Equation 7.6) that the stationary distribution,  $\pi$ , of a Markov chain can be found by solving the eigenvector problem:  $\pi = \overline{P}^T * \pi$ . In the ICG, there are no dangling nodes due to the way the ICG is constructed, so the transformation to create a stochastic, irreducible matrix representing the ICG (cf Equation 7.8) can be simplified to:

$$\overline{P} = (1 - \alpha)P + \alpha e v^T \quad (7.11)$$

And the calculation of  $\pi$  can be achieved by:

$$\pi = (1 - \alpha)M'\pi + \alpha v \quad (7.12)$$

where  $M' (= P^T)$  is the column normalised adjacency matrix of the ICG.  $\alpha$  is the probability of restarting the Random Walk from any of the nodes in  $v$ .

The estimation of  $\pi$  is solved in the iterative algorithm detailed in Algorithm 2. The algorithm converges if two consecutive estimates  $\pi^k$  and  $\pi^{k+1}$  are reasonably close together, ie  $|\pi^k - \pi^{k+1}| < T$ . The threshold,  $T$ , is set to  $10^{-6}$ .

**Returning the query results** Finally, we choose the top  $r$  image nodes (ie the elements,  $\pi(u_i)$ , from  $\pi$ , where  $1 \leq i \leq |\mathcal{I}|$ ) and present them to the user.

#### Issuing a query with images not contained in the database

We have three possibilities: (1) add the new image node to the graph before computing the steady-state vector (see Section 7.3.4); (2) we extract the features for the new image, determine the top  $k$  nearest neighbour feature nodes per feature and use these as the starting nodes; or (3) simply compute the most similar image overall to the query image and use this image in its place. The third method is favoured in the current implementation.

### 7.3.6 Relevance Feedback

In this section, we show how both long- and short-term learning can be implemented in the ICG to create a retrieval system that adapts to its users. On the one hand, relevance feedback is used to build up the semantic or peer network (the subgraph consisting of image nodes and the edges between them) over time. On the other hand, short-term learning is implemented by computing a set of feature weights used to adapt the transition probabilities from image nodes to feature nodes on a per-query basis.

#### Long-Term Learning: Adding Peer Links

In our application feedback is provided in terms of grouping images. Images which are put in the same group receive a co-occurrence edge in the graph, thus the weight of an edge between two particular images reflects the frequency of these images being grouped together. Similarly if an

image is selected as a negative example for a particular group, the weight of the edges between the negative image and all other images in the group will be discounted. Hence, over time semantic relations are strengthened which reflect the usage context of the images.

*Positive Feedback* is always a result of adding an image to a group. Therefore, the newly added image receives a new link to all other images in the group (and vice versa). If a link already exists between the new image and another group image, its link weight is increased by 1. With usage over time, the link weights between semantically related images are strengthened.

*Negative Feedback* is incurred if either an image is specifically labelled as not belonging to a certain group, implicitly ignored in the recommendation set for a group in three consecutive turns, or deleted from a group in which it previously resided.

There are two strategies to deal with negative feedback: discount the link weight by a constant factor, eg  $\frac{1}{5}$ , or decrement the link weight by 1. The former changes the weights drastically in favour of the most recent feedback. For instance, if two images are in the same 10 groups together but are regarded as unrelated in the 11th group, the resulting link weight is 2 ( $10 * \frac{1}{5}$ ). This might not be a desirable outcome with regards to the long-term learning capabilities of the ICG. However, one could also implement an overlay of the graph in which the most recent feedback changes the current link weights by a high discount factor so that it will take a short-term view. Only the original link weights of the ICG will be stored and the overlay discarded after a query session.

### Short-Term Learning: Adjusting Link Weights

**Visual Feature Weights** Given the ICG is constructed from  $F$  visual features, then each image node is connected to exactly  $F$  feature nodes (in addition to possible term and other image nodes). Now, the importance of the specific visual feature will depend on the current query. Therefore, the probability of moving from an image node to a visual feature node (and vice versa) should change depending on the importance of that particular feature.

In Rui & Huang (2000) an optimised framework is presented for calculating visual feature weights, when only positive feedback is considered (also used in the baseline system). The same inter-feature learning algorithm can be employed in the ICG to change the link weights between image and feature nodes. Essentially, an optimal solution for the visual feature weights  $\vec{u}$  is derived that minimises the summed distances between positive feedback examples and the query. The feature weights are indirectly proportional to the sum of weighted distances<sup>7-2</sup> between the query and all relevant images under feature  $j$  (for  $j = 1, \dots, F$ ):

$$u_j \propto \frac{1}{\sqrt{\sum_{i=1}^P \text{rel}_i d_j(p_i, q)}} \quad (7.13)$$

where  $\text{rel}_i \in [0, 1]$  is the relevance score of the  $i$ -th example. The feature weights are subject to normalisation, ie  $\sum_{j=1}^F u_j = 1$ .

During the normalisation phase before calculating  $\pi$ , all outgoing links from an image node to their feature nodes are re-weighted with the corresponding link weight, such that  $l' = u_j * l$ , where  $l$  is the link weight.

<sup>7-2</sup>As the link weights are probabilities (ie the higher its weight the more likely it is that the link will be followed) the distances used in the computation of feature weights are converted to similarities by  $1/d$ .

**Overall Feature Weights** A final modification on link weights is made to influence the overall importance of the three feature modalities: visual, textual and peer. Assume there are weights  $w_v$ ,  $w_t$  and  $w_p$  for the three high-level features. To implement overall feature weighting, all outgoing links from an *image* node are weighted with the corresponding feature weight depending on their link type:  $l' = w_i * l$ , where  $l$  is the link weight and  $i$  is the feature type.

These weights could either be specified explicitly by the user or obtained automatically. There are two possibilities of how to calculate the weights automatically on a per-query basis. First, the weights can be calculated based on the similarity between the query items considering the three feature modalities separately. For this, construct a visual, term and peer query based on the image examples and terms provided in the query (cf Section 7.2). The visual feature weight is then proportional to the sum of similarity scores between these queries and the query items. The disadvantage of this method is that it uses the original indices and similarity computations rather than the graph representation to determine feature weights. In addition, the peer-, feature- and term-scores are not readily comparable, which has to be addressed for example by a min-max normalisation of the scores.

Alternatively, the graph structure could be used to determine the similarity between query nodes based on the three individual features. A solution would be to perform a Random Walk for each type of feature: one in which the restart vector is set to the term nodes contained in the query for the term feature; then to the image nodes for the peer feature; and finally to the visual feature nodes connected to the query images for visual feature. Again the weights would be proportional to the sum of resulting ranks (ie similarity scores). Let  $\pi_t$  be the stationary distribution of the Random Walk started from the term nodes and let  $sim_t$  denote the overall term similarity of the query. Define  $\pi_v$ ,  $\pi_p$  and  $sim_v$ ,  $sim_p$  similarly. Then  $sim_t = \sum_{u \in Q} \pi_t$ , and  $w_t = \frac{sim_t}{sim_t + sim_v + sim_p}$ . The obvious problem with this approach is that three additional Random Walks have to be calculated before the actual Random Walk is calculated.

In the experiments, we present results using a set of predefined feature weights to determine the effect of such weights. We also investigate the influence of adaptive weights using the former approach.

## 7.4 Experimental Setup

We chose a simulated experiment as the most suitable for the purpose of evaluating the underlying retrieval techniques, since they allow algorithmic issues to be isolated from interface design or user issues. However, we are aware that this analysis is dependent on how we decide to simulate the user (see also Section 2.5). In particular, this includes our choice of relevance feedback strategy, which assumes that the user judges all items in the recommendation set (top 10 returned images) as described in Section 7.5.2. However, users' behaviour is very hard to predict, and therefore the results presented here should be interpreted with these considerations in mind. Evaluating different feedback strategies, or ideally asking 'real' users to perform these tasks, remains as future work.

### 7.4.1 Dataset

The experiments are performed on the same subset containing 12,800 images, derived from photo CD 1, CD 4, CD 5 and CD 6 of the Corel 1.6M dataset, that has been used in the previous user experiments.

### 7.4.2 Features

**Visual Features** As in all previous experiments, we use the following 6 low-level colour, texture and shape features (feature dimension in brackets):

**Colour** Average RGB (3), Colour Moments (9) Stricker & Orengo (1995)

**Texture** Co-occurrence (20), Autocorrelation (25) and Edge Frequency (25) Sonka et al. (1998)

**Shape** Invariant Moments (7) Hu (1962)

**Textual Feature** The textual feature is composed of annotations obtained from BerkleyCorel (2005) also previously used in Experiment 2 (cf Section 6.3) of the user evaluation. We use both the keyword as well as the description fields to annotate the images. Further, terms are porter-stemmed and stop words are removed. Stemming involves stripping words of common endings, and the universally used algorithm for the English language was proposed by Porter (1997). In IR, the use of a stop word list<sup>7-3</sup> is common, which lists those words that are too general to be useful such as “the”, “an”, and “be”. The stop words are not indexed by the system.

**Peer Feature** The peer feature is the sum of all feedback received from 24 users performing a variety of tasks in Experiments 1 and 2 in Chapter 6. For the detailed description of which tasks were performed please consult the previous chapter, specifically Section 6.2.2 for the tasks performed in Experiment 1 and Section 6.3.2 for the tasks in Experiment 2. Please note that this constitutes realistic usage information (based on task-based evaluations) not tailored for the experiments reported in this chapter.

### 7.4.3 Tasks

To evaluate the different retrieval techniques under various circumstances, we have chosen a variety of tasks that reflect different high-level concepts. Table 7.1 summarises the 10 tasks that have been chosen and manually labelled for this evaluation. Tasks 2 and 3 are almost identical to existing Corel categories with annotations. Tasks 7 and 8 are also derived from Corel categories, but none of their images are annotated. Further, Tasks 1–6 have been used in Experiment 1 of the user evaluation described in Section 6.2.2.

Some statistics of the chosen tasks are compiled in Table 7.1. The table lists the tasks along with the number of ground-truth images they contain. It further shows the percentage of images that contain annotations per task, and the average number of annotation terms per image. These two numbers reflect the strength that can be expected from the textual baseline. The table also

<sup>7-3</sup>Available for example at [http://ir.dcs.gla.ac.uk/resources/linguistic\\_utils/stop\\_words](http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words).

Table 7.1: Tasks and their properties

Task	Description	#images	#annotated	#terms	#peers	#peers same group
Task1	Mountainous landscapes	549	67%	4.8	40.3	40.1
Task2	Elephants	113	98%	6.0	43.5	27.9
Task3	Tigers	103	100%	5.9	25.4	11.1
Task4	Animals in the snow	220	99%	5.7	30.8	30.5
Task5	African wildlife	865	99%	5.9	24.2	22.3
Task6	Underwater world	402	50%	2.7	41.4	41.4
Task7	Skiing	100	0%	0.0	0.7	0.0
Task8	Caves	200	0%	0.0	0.0	0.0
Task9	Snowy mountains	244	70%	4.9	43.5	20.7
Task10	Autumn trees	203	62%	3.3	0.8	0.0

shows the average number of peer links per image for each task. Images belonging to the tasks used in the user evaluation (Tasks 1–6) contain on average 35 peer links. Finally, we provide the average number of peer links to images belonging to the same task in the last column. This number reflects the coherence or complexity of a task. This shows for example that around 50% of the peer links of images in Tasks 2, 3 and 9 are to other images not considered relevant for these particular tasks. This can be explained by the fact that Task 2 and 3 can be considered subcategories of Task 5, while Task 9 is a subcategory of Task 1. We have introduced short-term learning in our approach (cf Sections 7.2.1 and 7.3.6) to discount semantic links irrelevant for the current task in order to better capture the current context. However, it is a big challenge to learn the current level of abstraction the user has in mind, and we do not claim we have solved this so far.

To summarise, the tasks are chosen to investigate the importance of the three features to different degrees. Some tasks benefit from annotations, some from peers, while others cannot use either.

#### 7.4.4 Performance Measures

Like in the simulated experiments described in Chapter 5, the performance of these experiments is based on the traditional *precision* and *recall* measures. Recall from Section 5.3.3 that they are defined as:

$$Precision = \frac{\# \text{ relevant images retrieved}}{\# \text{ retrieved images}} \quad (7.14)$$

$$Recall = \frac{\# \text{ relevant images retrieved}}{\# \text{ relevant images in the database}} \quad (7.15)$$

In our application, only the top  $r$  (where  $r \ll N$ ) images are ranked and returned to the user. Also, the query images will not reappear in this ranking, because they are already contained in the group.  $P(r)$  and  $R(r)$  values are in the range  $[0, 1]$  (corresponding to 0-100% of precision and recall at rank  $r$ , respectively). In addition, we present  $P(NR)$ , that is the precision at the rank of the total number of available relevant images for a task, and  $R(P05)$ , that is recall at the 0.5 precision level, for the initial runs where all images are ranked to allow scientific comparison. In the later runs including RF only the top 100 images are ranked.

Table 7.2: Description of methods and their variations

	Name	Description
Individual approach	$IND_v$	visual feature only
	$IND_t$	textual feature only
	$IND_p$	peer feature only
	$IND_{tv}$	visual and textual results combined
	$IND$	visual, textual and peer results combined
	$IND_a$	all three features combined with adaptive feature weights
Image Context Graph	$ICG$	Basic ICG without peer links
	$ICG_p$	ICG with peer links
	$ICG_{pd}$	ICG with peers and negative discount feedback
	$ICG_{pv}$	ICG with peers, using visual weights
	$ICG_w$	ICG with peers, using overall feature weights
	$ICG_{w:a}$	ICG with peers, using overall adaptive feature weights

## 7.5 Results

To establish the retrieval effectiveness of the proposed approach, we compare different variations of the ICG to the separatist approach. The individual baselines are referred to as  $IND_v$ ,  $IND_t$ ,  $IND_p$  for the visual, textual and peer features, respectively.  $IND_{tv}$  denotes the combination of visual and textual features, while the combination of all three features is referred to as  $IND$  (see Table 7.2). The parameters of the ICG are fixed to  $alpha = 0.6$  and  $k = 25$  in these experiments, based on some initial runs to establish the influence of the parameters whose results are reported in Appendix E.1.

### 7.5.1 Initial Runs without Relevance Feedback

The results are based on the average performance over 2999 queries in total (one query per ground-truth item per task). Table 7.3 compiles these results based on precision at rank 10, 20, 50 and at the rank of the number of relevant images per task,  $P(NR)$ , and recall at rank 10, 50, 100 and at 0.5 precision,  $R(P05)$ .

#### Individual Features

First of all, the individual baselines  $IND_v$ ,  $IND_t$  and  $IND_p$  were considered. The textual feature on its own outperforms all other features both in terms of precision as well as recall<sup>7-4</sup>. The visual features are especially poor.

The strengths and weaknesses of the approaches under investigation come to light if we analyse the performance for each task separately. The task-based results are provided in Tables 7.4 and 7.5, showing precision at 10 and recall at 100, respectively (Additional results based on the remaining performance measures for these runs can be found in Appendix E.2). Precision at 10 is best for Tasks 2–5 when using the textual feature only,  $IND_t$ . This reflects our expectations, since the images contained in these tasks are almost exclusively annotated. (cf Table 7.1). On the other hand, the text feature does not produce any relevant results (amongst the top 100) for Tasks

<sup>7-4</sup>The dominance of textual features has also been shown on the TrecVid corpus (TrecVid 2005)

7 and 8, which are purely visual tasks since no annotations are available. Just like the text feature, the peer feature is dependent on previously encoded information. It performs well for Tasks 1–6, for which relevance feedback was collected in the user evaluation of the *EGO* interface. The remaining tasks cannot be solved by the peer feature alone with the exception of Task 9 (snowy mountains), which can benefit from information recorded in Task 1 (mountainous landscapes).

Figure 7.3 shows the distribution of P(NR) values. It is interesting to see that the individual baselines contain many outliers. This shows that they can perform really well for some queries (eg queries whose result set is annotated or ones that were previously captured by peer information), which confirms the task-based analysis from above. However, if this is not the case, the individual feature cannot contribute any relevant results and a combination is necessary.

### Comparison to ICG

Next, the baseline is compared to the ICG when no peer information is available (ie no previous interaction has been recorded). The results in Table 7.3 show that while  $IND_{tv}$ 's performance is initially better than ICG's performance, ICG outperforms  $IND_{tv}$  when considering a larger result set (P(NR), R(100) and R(P05)). Also note that on average, the textual feature alone always outperforms the combination between visual and text,  $IND_{tv}$ .

Comparing the performance for the individual tasks again, we find that Tasks 3, 5 and 7 result in more precise results amongst the top 10 (in Table 7.4), while  $IND_{tv}$  significantly outperforms ICG only in Task 10. As Table 7.5 shows ICG always retrieves more relevant images amongst the top 100.

These results show that if we want to achieve high-precision results amongst the top 10, we are mostly better off just consulting the textual index apart for some tasks in which the text feature fails altogether. The ICG, however, manages comparatively high-precision results amongst the top 10 and becomes even better with a larger window of results, regardless of the task. Therefore, we conclude the ICG is more versatile in the sense that its overall ranking is more reliable.

### Adding Peer Information

Finally, the peer information is added to the baseline and the ICG. The results are shown in the last two columns of Tables 7.3-7.5. This time  $ICG_p$  significantly<sup>7-5</sup> outperforms the baseline. The results in Table 7.3 reveal that the combination of all three features,  $IND$ , is finally comparable to the text-only baseline,  $IND_t$ .  $ICG_p$ , however, outperforms both  $IND_t$  and  $IND$ . Also note that ICG *without* peer information eventually manages to retrieve more relevant images than the baseline  $IND$  *with* peers (P(NR), R(P05)), although the baseline is more precise up to the top 20 results (P(10), P(20)).

Looking at the influence of the peer information in more detail, we see that it does not help improve performance for Tasks 7, 8 and 10 ( $ICG$  versus  $ICG_p$  in Tables 7.4 and 7.5). Again this is expected, since these tasks have not been used in the user-evaluation and therefore there are no peer links to be exploited. However, recall at 100 is also worse for Tasks 2 and 3 when peer information is included in the ICG. These two tasks contain many irrelevant peer links, since they

<sup>7-5</sup>Statistical significance was calculated with the paired-sample t-test (Maxwell & Delaney 1990), which resulted in a significant difference of  $p < 0.01$  between  $ICG_p$  and all other methods.

are a subset of Task 5, but not all images for Task 5 are also relevant for Tasks 2 and 3. Therefore, some of these links are misleading. This problem of misleading links is addressed by the negative feedback strategy investigated below, which discounts peer links upon negative feedback received from the user. Also, Tasks 2 and 3 are textually very compact in the sense that they are almost exclusively annotated with the same set of terms for each image (eg almost all images in Task 3 contain the term “tiger”). We will see later how feature weights can help to guide the retrieval in the ICG to improve performance in this case.

Moreover, we would like to draw attention to a hypothetical comparison to the GCap approach proposed by Pan et al. (2004). An extension of GCap for retrieval has already been discussed by Pan et al., but they have not shown how it would perform experimentally. If we consider that the basic graph construction before any user interaction is recorded is almost analogous with the GCap graph (apart from the fact that regions of images are used as visual nodes, while we use the individual global features), we can also consider *ICG* as the baseline. Our results show that *ICG* (and thus GCap) performs well in comparison to the separatist approach. However, adding peer information causes a dramatic increase in performance over all baselines. The long-term learning facility is thus essential for improving retrieval effectiveness.

### 7.5.2 Performance with Relevance Feedback

After having compared the performance of one-shot queries, the next objective is to study retrieval performance over multiple relevance feedback iterations. The setup of these runs is the following: for each task 200 queries consisting of 3 example images are issued to the system and relevance feedback is performed over a total of 20 relevance feedback iterations. All of the images in the recommendation set, ie those amongst the top 10 retrieval results, are chosen for feedback. Both positive as well as negative feedback is used. The peer information is reset after each query. Also, only the top 100 images are retrieved and merged in the case of *IND* to keep the computational costs involved down (see the discussion in Section 7.5.5).

The main results are compiled in Figures 7.4, 7.5 and 7.6, showing the development of P(10), P(100) and R(100) over RF iterations. In terms of precision, *ICG<sub>p</sub>* outperforms the baseline *IND*. However, recall can only be improved by a significant margin after the 15th iteration. Even if the users might not always be prepared to ask for this many recommendations, the same effect will be noticed once the group size becomes sufficiently large. In this simulated setup, only a small number of images are added to the group in each iteration. In reality, the user can populate groups much faster resulting in larger groups for which *ICG<sub>p</sub>* will return the better results. The effect of group size (number of query images) is discussed in the following section. Nevertheless, *ICG* improves the average number of images retrieved by about 10 images over *IND*, and *ICG<sub>p</sub>* by almost 15 images as can be seen from Table 7.6, which reveals the average number of images found after 20 iterations. Note that the maximum group size that can theoretically be reached at this point is the minimum of 193 (3 initial images plus 10 images per iteration) and the number of relevant images in the ground-truth for a particular task.

*ICG* without peers exhibits an interesting behaviour. While its initial performance is close to *ICG<sub>p</sub>* in the P(100) graph and even better in the R(100) graph, it quickly drops off after the 10th iteration. Initially, both methods are able to find similar images based on annotations or visual



Table 7.3: Comparison between baselines and ICG with and without peer information

Method	IND <sub>v</sub>	IND <sub>t</sub>	IND <sub>p</sub>	IND <sub>tv</sub>	ICG	IND	ICG <sub>p</sub>
P(10)	0.18	0.58	0.29	0.50	0.42	0.58	0.62
P(20)	0.16	0.56	0.28	0.44	0.41	0.54	0.59
P(50)	0.14	0.51	0.28	0.36	0.40	0.48	0.57
P(NR)	0.09	0.24	0.02	0.21	0.29	0.26	0.39
R(10)	0.01	0.02	0.01	0.01	0.01	0.02	0.02
R(50)	0.01	0.03	0.02	0.02	0.02	0.03	0.03
R(100)	0.03	0.14	0.07	0.08	0.12	0.12	0.15
R(P05)	0.00	0.18	0.02	0.10	0.24	0.18	0.36

Table 7.4: P(10) for individual tasks for baselines and ICG

P10	IND <sub>v</sub>	IND <sub>t</sub>	IND <sub>p</sub>	IND <sub>tv</sub>	ICG	IND	ICG <sub>p</sub>
Task1	0.20	0.42	0.35	0.44	0.42	0.58	0.61
Task2	0.16	0.97	0.57	0.81	0.79	0.86	0.89
Task3	0.06	0.97	0.40	0.55	0.83	0.73	0.86
Task4	0.15	0.76	0.43	0.66	0.60	0.74	0.75
Task5	0.25	0.92	0.30	0.71	0.81	0.77	0.86
Task6	0.18	0.46	0.38	0.45	0.44	0.59	0.63
Task7	0.13	0.00	0.00	0.13	0.19	0.13	0.19
Task8	0.10	0.00	0.00	0.10	0.09	0.10	0.09
Task9	0.13	0.42	0.27	0.32	0.30	0.42	0.48
Task10	0.09	0.17	0.00	0.20	0.11	0.20	0.13

Table 7.5: R(100) for individual tasks for baselines and ICG

R100	IND <sub>v</sub>	IND <sub>t</sub>	IND <sub>p</sub>	IND <sub>tv</sub>	ICG	IND	ICG <sub>p</sub>
Task1	0.03	0.05	0.06	0.05	0.05	0.09	0.10
Task2	0.06	0.85	0.33	0.29	0.69	0.50	0.58
Task3	0.03	0.89	0.16	0.18	0.77	0.31	0.59
Task4	0.04	0.19	0.17	0.16	0.18	0.23	0.30
Task5	0.02	0.09	0.04	0.05	0.08	0.07	0.09
Task6	0.03	0.10	0.09	0.08	0.11	0.13	0.16
Task7	0.05	0.00	0.00	0.05	0.06	0.05	0.06
Task8	0.03	0.00	0.00	0.03	0.03	0.03	0.03
Task9	0.03	0.09	0.07	0.07	0.08	0.11	0.13
Task10	0.03	0.04	0.00	0.06	0.04	0.06	0.04

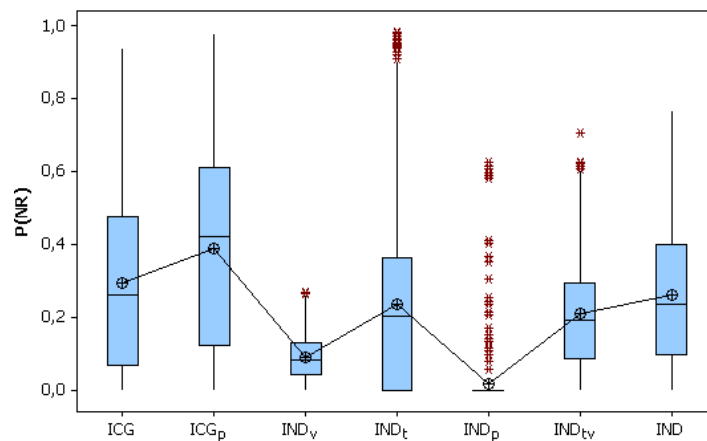


Figure 7.3: P(NR) for ICG and baselines

Table 7.6: Average group size after 20 RF iterations

GS	IND	IND <sub>a</sub>	ICG	ICG <sub>p</sub>	ICG <sub>pd</sub>	ICG <sub>pv</sub>	ICG <sub>w:p</sub>	ICG <sub>w:t</sub>	ICG <sub>w:v</sub>	ICG <sub>w:a</sub>
Task1	152.31	110.84	126.13	189.10	189.10	189.06	188.87	189.30	189.28	189.26
Task2	102.28	75.58	111.05	73.18	73.18	73.16	73.16	74.08	73.18	73.22
Task3	87.72	57.55	99.52	50.61	50.60	50.66	50.41	65.08	50.60	50.93
Task4	113.31	80.42	127.28	146.12	146.33	146.29	125.74	153.13	145.82	144.98
Task5	162.26	117.30	160.02	185.74	185.68	185.92	185.86	186.42	185.98	185.81
Task6	149.46	98.06	185.41	189.22	189.22	188.78	188.93	190.08	189.22	189.18
Task7	8.86	10.96	34.96	36.84	36.14	43.26	37.68	36.96	36.40	36.58
Task8	7.66	7.84	34.03	42.48	42.48	28.94	43.73	42.99	41.92	42.23
Task9	82.72	86.87	74.46	88.94	88.92	88.64	88.94	88.84	88.95	88.94
Task10	18.57	12.81	32.84	27.80	27.42	27.16	27.96	27.54	27.85	28.22
Avg	88.52	65.82	98.57	103.00	102.91	102.19	101.13	105.44	102.92	102.93

features. After that the peer links are particularly useful to navigate to related images which are not necessarily similar and can therefore continue to retrieve relevant results.

Table 7.6 also lists the task-based results, which allows us to see that the graph-based approaches,  $ICG$  and  $ICG_p$  succeed in quadrupling the number of relevant images found for the two visual tasks (Task 7 and 8). Task 2 and 3, the two textually compact tasks, are best if the peer information is ignored as in  $ICG$ , corroborating our previous observations. The interested reader can again refer to Appendix E.3 for the task-based results for P(10), P(100) and R(100) after the first, fifth and tenth iteration and the average over all 20 iterations.

### Alternative Negative Feedback Strategy

We also explored an alternative feedback strategy: one which discounts negative feedback links by a factor of 5 (see Section 7.3.6), referred to as  $ICG_{pd}$ . This variation does not have a noticeable effect compared to the decrementing feedback strategy implemented in  $ICG_p$  as can be seen from the various graphs in Figures 7.4, 7.5 and 7.6. Our previous assumption that Tasks 2 and 3 would benefit from a more drastic feedback strategy is therefore not satisfied. This is probably due to the fact that the peer information is still relatively sparse so that peer links do not have large weights in the first place. In this case, decrementing by one or dividing by a factor both have a similar effect.

### 7.5.3 Variations of Group Size

The results of the previous RF runs have suggested that  $ICG_p$ 's performance increases over  $IND$  with a growing group size. To verify this assumption the group size was varied from 5 to 50 (in steps of 5) in these runs. The results are averaged over 200 queries of the specified group size per task. No relevance feedback was performed. Table 7.7 compiles the P(10) results for  $IND$  and Table 7.8 for  $ICG_p$ . As before, the remaining results are provided in Appendix E.4.

The difference between  $IND$  and  $ICG_p$  is about 10% points in favour of  $ICG_p$ . While  $IND$ 's performance peaks at a group size of 10,  $ICG_p$  performs best at a group size of 25. The performance for the visual tasks, Tasks 7 and 8, in  $ICG_p$  with growing group size is especially remarkable. For Task 8,  $ICG_p$  manages a precision of 60% for 50 query images. Compared to that,

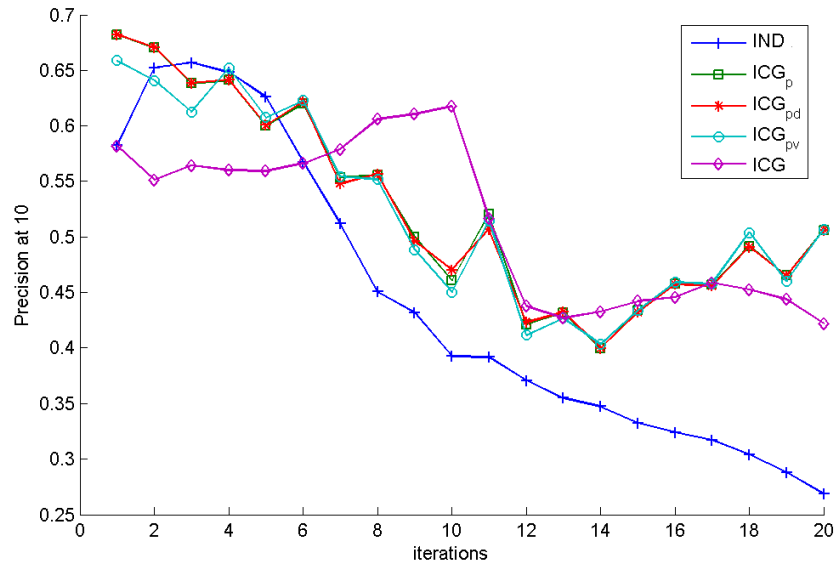


Figure 7.4: P(10) over RF iterations

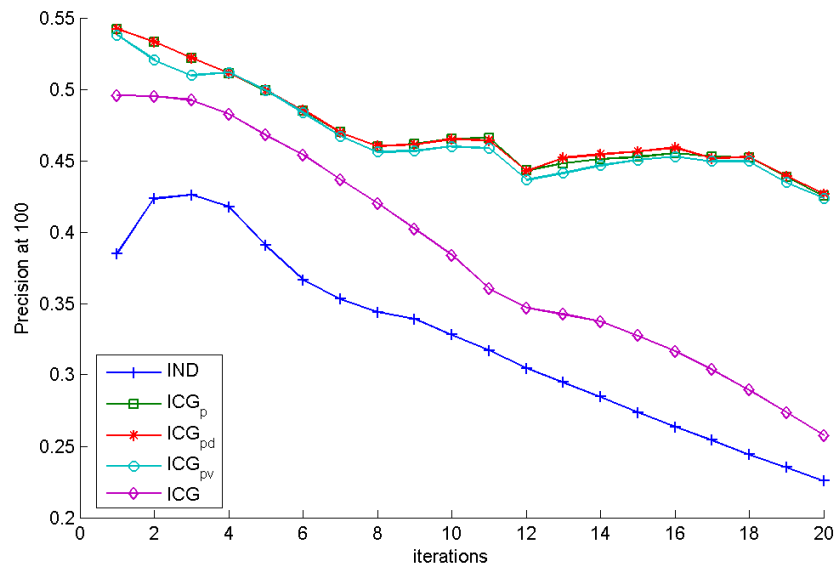


Figure 7.5: P(100) over RF iterations

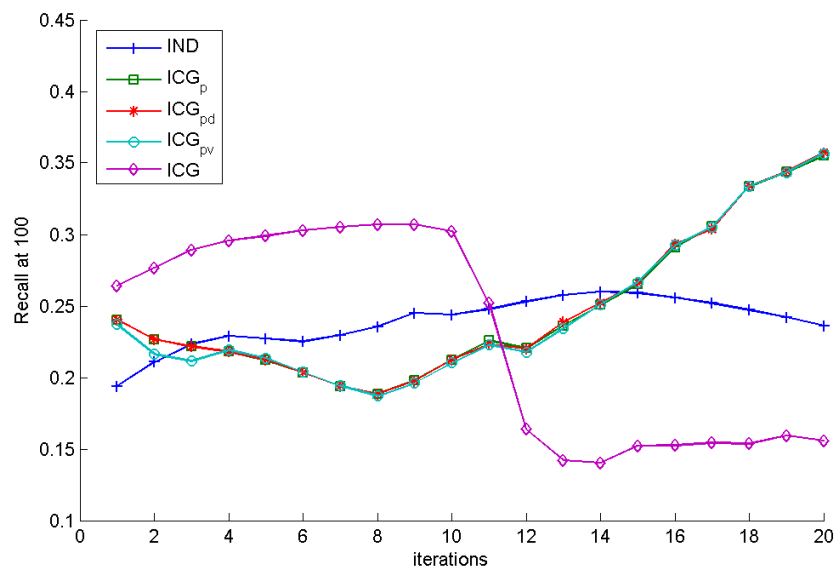


Figure 7.6: R(100) over RF iterations

Table 7.7: P(10) for *IND* for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.88	0.96	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	0.98
Task2	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task3	0.91	0.96	0.93	0.91	0.88	0.86	0.87	0.87	0.87	0.86	0.89
Task4	0.79	0.91	0.96	0.96	0.97	1.00	1.00	1.00	1.00	1.00	0.96
Task5	0.90	0.94	0.95	0.95	0.88	0.89	0.89	0.87	0.88	0.87	0.90
Task6	0.93	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Task7	0.14	0.10	0.13	0.12	0.06	0.11	0.10	0.09	0.09	0.08	0.10
Task8	0.07	0.08	0.07	0.05	0.09	0.09	0.09	0.09	0.08	0.08	0.08
Task9	0.50	0.43	0.38	0.36	0.26	0.36	0.36	0.37	0.34	0.34	0.37
Task10	0.17	0.18	0.17	0.13	0.10	0.07	0.08	0.09	0.10	0.11	0.12
Average	0.63	0.66	0.66	0.65	0.62	0.64	0.64	0.64	0.64	0.63	

Table 7.8: P(10) for *ICG<sub>p</sub>* for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.92	0.98	0.98	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Task2	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task3	0.93	0.97	0.98	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.98
Task4	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task5	0.98	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task6	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.37	0.45	0.47	0.50	0.47	0.44	0.39	0.36	0.34	0.30	0.41
Task8	0.19	0.26	0.34	0.38	0.44	0.47	0.50	0.54	0.57	0.60	0.43
Task9	0.69	0.66	0.67	0.64	0.64	0.63	0.62	0.61	0.60	0.59	0.63
Task10	0.18	0.20	0.22	0.23	0.23	0.22	0.20	0.20	0.19	0.16	0.20
Average	0.72	0.75	0.77	0.77	0.78	0.78	0.77	0.77	0.77	0.76	

*IND*'s performance remains relatively constant with varying group size, and reaches a maximum precision of only 9%. These results suggest that *ICG* can better handle larger group sizes.

It always poses a challenge to pick a suitable query representative from a large number of query images, which contain a selection of images each relevant to that group for a variety of reasons (or feature modalities). If we simply choose the average representation there is the danger of losing important feature variations, which is especially true for visual features (cf Chapter 5). The *ICG*, on the other hand, treats every query item individually, since the starting set (or restart vector) is formed from the collection of query items. This is the reason why *ICG* is better for larger groups.

#### 7.5.4 Introducing Feature Weights

As has been elaborated in Section 7.3.6, short-term learning can be implemented by adjusting link weights in the *ICG*. In the following, we present the results for feature weights on two different levels: visual feature weights for weighting the importance of the visual nodes; and overall feature weights between the three feature modalities represented in the graph.

### Visual Feature Weights

The results (based on the same setup as for the RF runs) of incorporating visual feature weights, referred to as  $ICG_{pv}$ , are plotted in Figures 7.4, 7.5 and 7.6. Overall, the visual feature weighting does not have a noticeable impact in these graph. Therefore, we turn our attention to the performance for the individual tasks in Table 7.6 revealing the average number of images found after 20 iterations. These results show that for the two visual tasks, Task 7 and 8, the weighting actually influences the performance, albeit in different ways for the two tasks. The weights lead to an increase in the group size for Task 7, while the group size drops in comparison to  $ICG_p$  for Task 8. The same phenomenon was observed for the other performance measures. The interested reader can refer to these results in Appendix E.3. Task 7 (skiing) can be better captured by the implemented visual features, since the images are very homogeneous in colour. Nevertheless, it seems that the graph structure is the crucial factor in the Random Walk computation.

### Overall Feature Weights

The final modification on link weights can be made to influence the overall importance of the three high-level features: peers, textual and visual.

**Fixed Weights** We experimented with three sets of feature weights:  $ICG_{w:p}$  denotes the weight ratio 3:1:1 between peer, text and visual features,  $ICG_{w:t}$  the ratio 1:3:1 and  $ICG_{w:v}$  the ratio 1:1:3. The results are shown in Figures 7.7 and 7.8. It can be seen that the weights do not influence the performance early on in the feedback session. Later, the text feature is dominant, while strengthening the visual weights does not have an impact on performance either way. However, if the peer weights are emphasised it actually hurts the performance in the long run. The peer information is collected over time, involving a number of different users performing different tasks. Hence, it does not capture exactly the semantic relationships relevant for a new task and therefore can also lead the Random Walker in the wrong direction after the relevant peers have been found. The tasks for which this was the case here are Tasks 3 and 4 as is conveyed by the per-task results in Table 7.6. Other performance measures (provided in Appendix E.3) indicate that the textual weights can also boost the performance of Task 2. For example, the average precision at 100 after 20 iterations is 0.27 for  $ICG_{w:t}$ , compared to 0.19 for  $ICG_{w:p}$  and 0.22 for  $ICG_p$  (cf Table E.34). Tasks 2–4 contain almost exclusively images with annotations, and therefore it is not surprising that relevant images can be found quicker by increasing the textual feature weight. Furthermore, Tasks 2 and 3 contain a lot of irrelevant peer links (since they are a subset of Task 5, but not all images relevant for Task 5 are also relevant for Tasks 2 and 3). All other tasks show little differences in performance.

**Adaptive Feature Weights** It is desirable to determine the importance of features automatically on a per-query basis. In order to study the effect of adaptive overall weights, we chose to calculate them based on the similarity between the query items considering the three features separately. Therefore, a visual, term and peer query is constructed based on the image examples and terms provided in the query (cf Section 7.2). The visual feature weight is then proportional to the sum of similarity scores between these queries and the query items. Let  $sim_t$  denote the

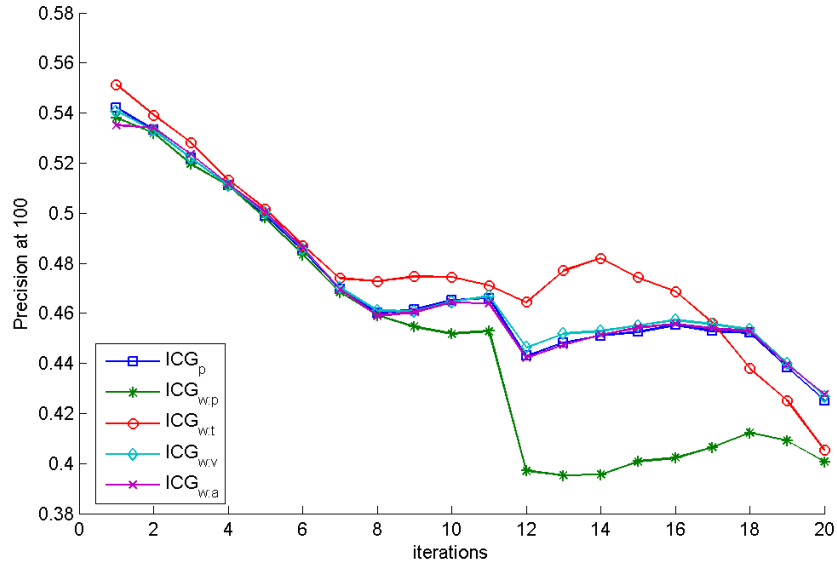


Figure 7.7: P(100) of weighted ICG over RF iterations

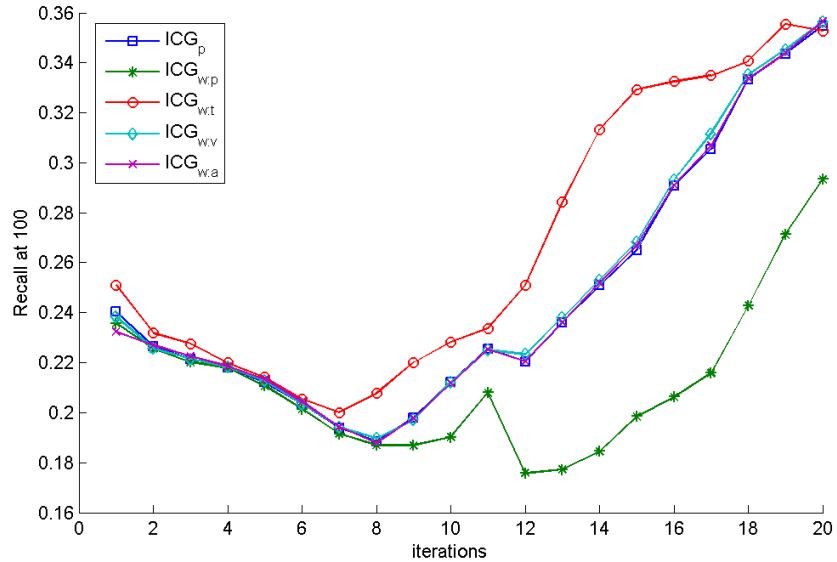


Figure 7.8: R(100) of weighted ICG over RF iterations

overall term similarity of the query ( $sim_v$ ,  $sim_p$  similarly for the visual and peer similarity). Then

$$w_t = \frac{sim_t}{sim_t + sim_v + sim_p}.$$

One major problem with determining the feature weights from the individual feature indices is the comparability of scores across the features. In general, we need a normalisation technique in order to make these scores comparable. There are various normalisation techniques, for instance min-max normalisation or a Gaussian normalisation. We have compare these normalisation techniques applied to the retrieval methods studied here. The results are provided in Appendix E.5.1. In addition, initial runs without relevance feedback were executed whose results are collected in Appendix E.5.2. Based on these results a Gaussian normalisation (with the visual feature weight divided by 2 to deemphasise its impact) is used for the following runs. The adaptive feature weights will be referred to as  $ICG_{w;a}$  when employed in  $ICG_p$  and  $IND_a$  when employed in  $IND$ . Note that in the voting approach, the individual lists can be weighted during the list combination

Table 7.9: Average iteration number and time to solve the Random Walk ( $k=25$ )

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Iterations	68.3	41.2	28.7	21.5	16.5	13.0	10.0	8.0	6.0
Solving time in ms	1,713	1,069	1,186	930	638	479	376	303	159
Total query time in ms	3,809	3,118	2,888	2,727	2,549	2,572	2,168	2,101	2,039

as described in Section 5.4.1.

The graphs for the development of  $P(100)$  and  $R(100)$  over RF iterations in Figures 7.7 and 7.8 display the adaptive weighing strategy in ICG alongside the fixed weight sets and  $ICG_p$  without weighting. They show that incorporating the adaptive weighting strategy does not change  $ICG_p$ 's performance. The group size results in Table 7.6 confirm this observation. However, using the adaptive weights in  $IND_a$  results in a significantly smaller group size at the end of the RF iterations over  $IND$  without weights as can be seen from this table. This suggests that the automatic computation of weights is not ideal in the first place. However, this also implies that the effectiveness of ICG is not affected by poorly chosen weights.

These results show that knowledge about the tasks and dataset is necessary if one wants to exploit the relative importance of high-level features. An adaptive or interactive learning strategy for setting weights is very desirable. Further study is needed in order to find a solution to this problem. Nevertheless, the results in the previous sections have shown that, even without short-term learning capabilities, the graph represents the similarities and relationships between images very well, as its performance is significantly better than without the peer information and also than the baselines.

### 7.5.5 Computational Comparison

Another issue worth mentioning is the computational costs involved with these methods. Retrieval on the ICG requires: normalisation of the graph matrix; and solving the Random Walk to find the stationary distribution. The former takes about 0.7sec on average on a quad 3.2Ghz Xeon processor system with 4GB of RAM (The machine was also supporting three other processes simultaneously during most of the total experiment run.). The latter varies with the parameter  $\alpha$ : the larger  $\alpha$  is, the quicker the algorithm converges. For  $\alpha = 0.1$  it takes on average 1.7sec, for  $\alpha = 0.6$  the solving time is 0.5sec, and for  $\alpha = 0.9$  it comes down to 0.2sec (see Table 7.9). The total query time is around 2sec for the ICG with  $\alpha$  set to 0.6, which is the same as  $IND$  if only the top 100 results are merged. The costs for  $IND$  increase substantially if we attempt to merge more results. For example merging the complete result set, ie 12,800 images, took approximately 200sec.

Preliminary runs on a much larger collection consisting of almost 40,000 images (created by adding Corel CDs 7 and 8) suggest that the query time of  $ICG$  ( $\alpha = 0.6$ ,  $k = 10$ ) increases to approximately 22sec (1sec for normalisation and 21sec for solving). However, for this collection size there is a considerable increase in retrieval effectiveness, too. For instance,  $IND$  achieves 0.21 precision at 10, compared to 0.45 for  $ICG_p$  for the 10 tasks used previously. Also recall at 100 more than doubles from 0.07 to 0.17. The superior performance justifies a further investigation of optimised algorithms for computing the Random Walk. This has been studied in the Web domain

extensively, considering that there the algorithm has to deal with billions of documents (Langville & Meyer 2004).

### Computational Complexity

Since the exact running time depends largely on the available hardware, a short discussion of ICG's run-time costs in terms of the algorithm's computational complexity will follow.

The normalisation of the graph matrix requires the summation of all row entries per column and consequent division of all entries by the calculated sums. Since we only have to consider non-zero entries in the matrix, the computational costs are dependent on the number of edges in the graph. Hence, the normalisation can be done in  $O(2 \times |E|) = O(|E|)$  time, where  $|E|$  is the number of edges.

The calculation of the stationary distribution is also linear in the number of edges. In the current implementation,  $\pi$  is estimated iteratively until the difference between two consecutive stationary distributions is smaller than a set threshold value of  $10^{-6}$ . Each iteration requires a sparse matrix multiplication between the normalised graph matrix and  $\pi$  (cf Equation 7.12). The multiplication requires  $2 \times |E|$ . Table 7.9 shows that the number of iterations is typically small, ie in the order of  $O(1)$ . The overall costs of the retrieval algorithm in ICG is  $O(2 \times |E|) + O(1) \times O(2 \times |E|)$ , which reduces to  $O(|E|)$ .

### Improving Efficiency

Practically, there are some issues we can address to reduce the running time of the retrieval algorithm. First of all, the number of edges can be kept small by reducing  $k$ , the number of nearest neighbours. The results in Appendix E.1 suggest that the effectiveness is not affected dramatically if we choose a small  $k$ .

Secondly, we can attempt to reduce the number of iterations either by choosing a larger convergence threshold or choosing a larger  $\alpha$ . As can be seen from Table 7.9, the number of iterations decreases substantially with increasing  $\alpha$ .

In any case, handling the adjacency matrix is a bottleneck of this approach. Also, there is a trade-off between fast normalisation time and fast solving time. Note that the sparse graph matrix is currently stored in a row-wise format, ie each row is represented by a vector of all non-zero column entries. This facilitates a faster computation of the matrix-vector product in Equation 7.12. However, the row-wise storage penalises the normalisation, since this is done column-wise. Normalisation alone takes about 1-2 seconds, which is currently repeated before each query. If query time is crucial and enough storage is available, the normalised adjacency matrix can be stored alongside the original (note that both these matrices are sparse). Updating the normalised matrix is then only required if the graph has changed (or even collect a number of changes until a certain threshold is exceeded before updating the normalised matrix). This would bring down the query time close to the raw solving time.

Last but not least, we can choose faster solutions to linear systems. For instance, using the Gauss-Seidel method for solving a linear system instead of the standard Power Iteration (or Jacobi method) can reduce the number of iterations needed (Arasu et al. 2002). Langville & Meyer (2004)



discuss some other techniques that result in faster convergence.

## 7.6 Future Work

As mentioned previously, we have not succeeded in finding a suitable adaptive weighting strategy. It would be desirable to use the graph structure directly to determine the importance of features for a particular query, but we have refrained from implementing the technique suggested in Section 7.3.6 for efficiency reasons. In addition, we can also envisage a more drastic weighting technique instead of simply using the feature weights as a scaling factor for updating link weights. To ensure that a peer link is chosen with probability  $w_p$ , a feature-attribute link with probability  $w_v$  and a term-attribute link with probability  $w_t$ , all link weights of a particular feature  $i$  have to sum to the feature weight  $w_i$ . Therefore, during the normalisation stage of the adjacency matrix of the ICG, the first  $|\mathcal{S}|$  columns of the feature-context matrix are normalised, such that all peer links sum to  $w_c$ . Similarly, all feature-attribute links sum to  $w_f$ , and all term-attribute links sum to  $w_t$  (in the first  $|\mathcal{S}|$  columns). The remaining columns sum to 1.

Evidence combination is a whole field of research in itself, thus providing a plethora of techniques to choose from and compare the ICG against. For example, Tong et al. (2005) propose to use a graph-based approach for learning in the multimedia domain. In this model each feature is represented in a separate graph, and the learning task is formulated as inference problem from the constraints in every graph. The authors investigate a linear and a sequential scheme to fuse the constraints given by the individual graphs. Another interesting approach is pursued by Iyengar et al. (2005), which is based on statistical techniques to model relationships between features in a probabilistic framework.

Most important of all, however, the ICG could be employed in a different context, including other media and features. This chapter has revealed its success in integrating various features. This should be a big advantage for video data, as videos are typically represented by a plethora of features, including audio, text, visual and motion-based. Therefore, it would be interesting to employ the ICG for the TrecVid collections (TrecVid 2005, 2006) as it has the additional benefit of having associated queries and manually labelled ground-truth data.

## 7.7 Summary and Conclusions

The goal of this chapter was to find a contextual feature to implement a long-term learning strategy for an improved recommendation system in *EGO*. For this purpose, we introduced a model to learn relationships between images obtained from user interaction as a contextual feature that allows long-term learning in an image retrieval and management environment. The relationships are mined from user interaction, which results in a personalised, semantic feature—the peer feature.

In the simple, individualist approach the visual, textual and peer features are stored in separate indices. The combined retrieval results are obtained by merging the individual lists using the rank-based voting approach. Furthermore, we proposed and explored a graph-based model, the ICG, that encodes all three features together. The theory of Random Walks is employed to compute retrieval results in the ICG.

Results of a simulated experiment showed that the ICG was successful at integrating various features that are otherwise difficult to compare for adaptive image retrieval. The ICG generally outperformed the baseline methods, which treat each feature separately for retrieval and then merge the final results. The ICG was more versatile than the individual baselines and their combination in the sense that its overall ranking was more reliable for a variety of tasks. In particular, the graph structure was very good when only visual information was available. In the relevance feedback scenario, this led to a fourfold increase in performance.

Most importantly, including the peer information significantly improved retrieval performance. This was shown for the case of one-shot queries, as well as in the relevance feedback setting. In the relevance feedback runs we could observe that, initially, both methods, *ICG* and *ICG<sub>p</sub>*, performed equally well. From a certain stage, after having retrieved close-by images based on annotations or visual features, the peer links were particularly useful to navigate to related images which are not necessarily similar and could therefore continue to retrieve relevant results. The long-term learning capability in the form of *ICG<sub>p</sub>* is therefore an improvement over the graph model proposed by Pan et al. (2004) for learning image annotations that does not use any feedback information at all.

Moreover, we discovered that the ICG was better able to handle larger group sizes (number of query images) than the individual baseline. By collecting each query item in the restart vector, it treats each item individually instead of averaging over them. In particular, visual tasks benefitted greatly from the overall graph-structure if we have many query images. This corroborates our findings in the relevance feedback runs.

Further, we proposed and experimented with various short-term learning strategies that influence the link weights in the graph for the current query session. This again improved the retrieval effectiveness for certain tasks. For instance, incorporating individual visual feature weights resulted in better performance for the visually homogeneous task. Also, when emphasising the textual feature over the other two overall features we could witness an improvement for the text-based tasks. However, our attempts to adjust these weights automatically were unsuccessful. While the retrieval performance was unaltered for adaptive weights employed in the ICG, the performance of the weighted combination in IND actually suffered. This has led us to the conclusion that the adaptive weights were poorly chosen. Therefore, the question still remains of how such weights could be adapted automatically.

To conclude, there are two main benefits of the ICG. First, it is able to incorporate long-term learning in the form of a contextual feature, which takes into account the usage context and the user's interpretation of semantic relationships between images. This is the basis of a personalised, contextual recommendation system. Second, it successfully integrates features at various semantic levels, such as low-level visual features, high-level textual annotations representing "all-purpose" semantics and the proposed peer feature defining user semantics. As the experimental results suggested some features were better than others for specific tasks. Nevertheless, the graph-based representation performs well under most circumstances. The proposed model is an elegant formulation of a retrieval technique based on features at different semantic levels and therefore the last piece of the puzzle in our quest to define a holistic environment for image management and retrieval.

---

## CONCLUSIONS AND FUTURE WORK

---

The broad objective of my research was to formulate a holistic retrieval and management environment and show that it can address the intrinsic problems of image retrieval: the image meaning (semantic gap), the query formulation problem and time-varying information needs. This chapter summarises the work towards this end and draws conclusions on the success of the formulation of a holistic approach. Finally, I discuss avenues of future research that could complement the work described within this dissertation.

### 8.1 Summary and Contributions

The failure of pure content-based image retrieval techniques can be attributed to the lack of a semantic representation of images. This manifests itself in:

- Image features that do not capture the user’s needs (cf Sections 2.2, 3.2.1, 3.3 and 6.2.4);
- Relevance feedback techniques that attempt to close this gap but are ultimately constrained by the underlying representation (cf Sections 2.3, 6.2.4 and 6.5); and
- Poor interface support that either forces the user to represent needs in a query based on the low-level features (query formulation problem) (cf Sections 3.2.2 and 3.3), or becomes too simplistic by hiding all internals of the retrieval mechanism based on low-level features often leaving the user confused and lost (eg relevance feedback systems) (cf Sections 6.2.4 and 6.5).

However, semantics are about interpretation and, as such, user and context dependent. Hence, a “semantic” feature should be based on the user’s views. In order to elicit this information from the user, we need to provide an appropriate interface in which the user can interact with the images and give plentiful feedback, albeit not explicit. Finally, we need an appropriate learning technique that improves with the feedback collected from the user. These three parts together—a semantic feature representation, an intuitive interface and an adaptive retrieval algorithm—form the basis for a semantic retrieval system. To this end a “retrieval in context” approach was proposed, which is based on two key components:

1. An intuitive interface that engages the user in an interactive organisation process to help users solve their tasks and, as a by-product, captures the context in which the images are used.
2. A powerful recommendation system that takes into account the usage context and incorporates it as a semantic feature with traditional image features for an improved adaptive retrieval framework.

### 8.1.1 The Interface

A novel image search interface, *EGO*, described in Chapter 4, has been developed as a “retrieval in context” system. Its design is based on cognitive ideas, previous user studies and interviews of design professionals. *EGO* provides an environment where the user is invited to organise their search results into groups created on a workspace. This has several advantages:

- It helps the user in overcoming the query formulation problem, since they can concentrate on organising rather than formulating queries.
- It helps the user in conceptualising their tasks better because they are assisted in the task of breaking up the search task into related concepts.
- The grouping information is the basis for a contextual feature learnt from the user.
- A personalised view of the collection can be provided, so that the users can go back to groupings previously created or explore trails of other users.

In order to investigate whether *EGO* provides these benefits from the user’s perspective a user evaluation was conducted. The experimental methodology was based on a collection of realistic and practical design-oriented tasks proposed in this dissertation. The evaluation is a comprehensive study of the effectiveness and use of a workspace for image retrieval. It includes an analysis of the extent of the query formulation problem in image retrieval interfaces, an analysis of task-dependent search strategies and an analysis of organisation patterns on the workspace.

The results described in Chapter 6 revealed interesting usage and search patterns. The workspace in *EGO* was used to organise search results into different semantic facets of the task. It was deemed most useful when the underlying information need was vague or when the task was complex or multi-faceted. For open, exploratory searches the interface was able to support the user in exploring the collection and analysing their task. This led the users to discover more aspects of the task than initially anticipated. For complex, multi-faceted tasks, it enabled users to break up their overall task into a small set of individual search tasks. People often referred to the individual groups as different “search threads”. This process assisted the users in the conceptualisation of their tasks.

The grouping process allowed people to pursue a progressive search strategy by following multiple search threads simultaneously, while maintaining a constant overview of intermediate results and searches on the workspace. The overview of search results also provided the user the opportunity to compare selected images. Hence, people were more inclined to select good quality

images and if a number of images was to be selected, they were trying to find images complementing each other. For this reason people were more satisfied with their performance when given the opportunity to organise their results and the system was perceived as more effective.

Last but not least, the evaluation provided evidence that the interactive grouping mechanism helped to overcome the query formulation problem. In the relevance feedback system serving as a baseline comparison for *EGO*, the users were often unsure about which images to select for feedback and were confused about the results returned by the system. In *EGO*, by contrast, they found it easier to categorise images into task aspects and interpret the system's results accordingly. Therefore, they did not need to worry about the internals of the retrieval system.

Including the preliminary user study presented in Section 3.1, this work provides a comprehensive critique of four different image search interfaces: *EGO*, a traditional relevance feedback approach, the Ostensive Browser (Campbell & van Rijsbergen 1996), and a manual system providing query-by-example and query-by-keyword. In addition, I have proposed and evaluated a new adaptive query learning scheme for visual and textual features in the Ostensive Browser (cf Section 3.1.3 and Appendix A).

### 8.1.2 The Recommendation System

The recommendation system provides the seamless integration between retrieval and management that was a key design goal of *EGO*. It assists the user in the interactive organisation process by recommending images relevant to a selected group. From the system's perspective it allows adaptive retrieval by learning semantic relationships between images.

In the first instance, the recommendation system was based on content-based features only. I had already anticipated that CBIR techniques were limiting, since images grouped together by a user share semantic concepts, but are not necessarily similar in the feature space. Therefore, a multi-point query learning strategy was used to exploit a powerful learning algorithm, particularly suitable for small sample sizes and online computation, for clusters of visually similar images. The challenge posed by multi-point queries is how to combine the results from the various query points, which was addressed in the simulated experiment described in Chapter 5. This experiment established an encompassing comparative evaluation of visual retrieval algorithms focussing on evidence combination of multi-point queries. It compared a single query representation to multi-point queries using several list combination techniques. The results showed that multi-point queries were suitable for heterogeneous groups while a single representation might be favoured for homogeneous groups. Moreover, I found that the right choice of combination strategy was vital for multi-point queries to provide a benefit over a single query representative. The most stable performance was achieved by a rank-based combination, the voting approach, which I am the first to have used in this context. The voting approach performed well for both homogeneous and heterogeneous categories.

However, relevance feedback algorithms relying solely on visual features tend to quickly converge after a few iterations even if there are a lot more undetected relevant images. The reason for this is that most learning algorithms are aimed at narrowing down a query rather than generalising. Moreover, the user experiments from Chapter 6 also drew my attention to the problems associated with a retrieval system based on low-level features only. Based on an analysis of how people or-

ganised images I formulated a contextual feature based on semantic relationships between images mined from user interaction. This feature, apart from encoding user-based concepts, also enables long-term learning in the system.

Finally, Chapter 7 introduced the improved recommendation system that takes into account the contextual feature learnt from the user and combines it with content- and text-based representations. Integrating various features is an open research problem. A novel model was proposed, which allows: a straight-forward way of representing the contextual feature; and a way of integrating various features. This is achieved in the Image-Context Graph (ICG)—a graph-based representation where all images along with their low-level features and annotation terms are represented as nodes. The images are linked to their features, and features are linked amongst each other based on the similarity between them. The semantic feature is implemented by directly connecting images that belong to the same group. Querying in the ICG is implemented by computing a Random Walk (Lovasz 1993) on the graph, which determines the retrieval score of an image based on the long-term probability of navigating to an image node given a set of query items as the starting points. This method turns out to be highly effective as determined in a simulated experimental setup. The setup ensured a comprehensive test of a variety of parameters of the proposed model. Results showed the value of using the contextual feature and the superiority of the ICG over the traditional approach of a late combination of the retrieval results from different feature modalities.

## 8.2 Analysis and Conclusions

Recall from Chapter 1 that the underlying thesis of this work was stated as:

*A “retrieval in context” framework will help overcome the intrinsic problems of Content-Based Image Retrieval, such as query formulation, the semantic gap and time-varying information needs, by providing an integrated environment for image search and management in order to create and capture the context in which the images are used. This integration is achieved by the addition of a workspace to interactively group retrieval results, which supports the user, specifically where the user’s task is creative, and leads to a more effective system and increased user satisfaction.*

In order to show whether I achieved this goal, I would like to revisit the issues mentioned as the primary motivations for the design process in Chapter 3 one last time. The following discussion highlights how the questions raised previously are addressed in *EGO*, considering both the interaction strategies afforded by its interface (user perspective) and the underlying retrieval system (system perspective).

### *“What is the meaning of an image?”*

**User perspective** The most important point to recognise is that the meaning of an image is always determined by the user, their tasks and work environment. In *EGO* the semantics in the images are conveyed through groupings that the *user* creates while pursuing a certain work task. The organisation resulting from the long-term interaction reflects the usage of the

collection in the user's context. An analysis of the groups that were created during the user evaluation revealed that they were based on semantic concepts. The resulting organisation reflected different task aspects. From this organisation, it is easier for the system to infer the intended semantic meaning.

**System perspective** The problems from the system's perspective are the choice of semantic feature representation and the formulation of a retrieval technique based on features at different semantic levels. The proposed model, the ICG, is an elegant solution to these problems. It incorporates all features in a single graph in order to take advantage of the interdependence between features. The semantic feature is implemented by direct links between images rather than a separate entity. Experimental results have shown that this model can successfully exploit the semantic information.

*“How can the user be assisted in communicating their information need?”*

**User perspective** *EGO* engages its users in an organisation process to iteratively define their semantic needs. The participants of the user experiments stated that the organisation process helped them to conceptualise and explore their tasks. The recommendation system helped them to find more related images and populate their groups. This interactive process does not require the user to think in terms of the system (ie how to formulate a query, how a search works, etc.). Instead they can concentrate on exploring the task, iteratively and interactively reformulating their need on the way.

**System perspective** The user experiments have shown that it is not enough to abstract from the underlying representation and provide an over-simplified query input mechanism, as was the case in the relevance feedback system I studied. Although the relevance feedback system was simple and easy to use, people had more problems with it because they did not understand what was going on behind the scenes. This was not the case in *EGO*. On the one hand, the recommendation system is assisted by allowing the user to break up their search tasks into related facets, which usually results in more coherent search requests. On the other hand, the system's long-term learning facility is improved by incorporating the information gained from the user created groups as a semantic feature.

*“How can the time-varying nature of information needs be supported by the system?”*

**User perspective** Users are known to change their needs while interacting with a retrieval system. *EGO* invites the user to create groups according to the multiple facets of their need. The usage patterns in the evaluations have revealed that people tend to create new groups when detecting new facets or changing their need. Therefore, they let the system implicitly know that their need has changed. The grouping process further helps to develop vague needs as multiple trains of thought can be pursued simultaneously.

In addition, long-term needs are supported, since the groups can be created and changed over multiple sessions. A set of groups can therefore capture aspects of the user's long-term need. Finally, the system returns whole groups as retrieval results, enabling quick access to previously created groups.

**System perspective** The problem of time-varying information needs does not need to be modelled explicitly in *EGO*, since the users implicitly make changes in their needs known to the system. Therefore, we can avoid automatically detecting changes in the information need, as attempted in the Ostensive Browser (Campbell & van Rijsbergen 1996) for instance, and still provide a more intuitive and effective system.

Long-term access is additionally facilitated by a memory function implemented by the semantic feature. This allows the system to retrieve or re-retrieve images that have been deemed relevant under similar circumstances before.

In conclusion, the results of both user experiments—evaluating its interface—and simulated experiments—evaluating its retrieval system—have proven that *EGO* manages to overcome many of the problems of traditional image retrieval systems. From the user’s perspective it provides a better way of expressing complex, multifaceted and dynamic needs, communicating these needs and therefore solving their tasks. From the system’s perspective, the additional context provides a retrieval system that can adapt to its users. Altogether, *EGO* creates an environment, in which the meaning of an image is interactively defined, the query formulation problem is mitigated, and time-varying information needs can be expressed. Hence, it is a user-centred approach that comes close to bridging the semantic gap.

## 8.3 Future Work

It has been an exciting process to develop ideas and implement techniques to formulate an adaptive retrieval approach. Unfortunately, not all ideas could be investigated in this work. In this section, I will sketch a few suggestions that merit further investigation in future work.

### 8.3.1 Study of Long-term Effects from the User’s Perspective

I have repeatedly argued that *EGO* is suited for long-term management. The reasoning behind this is that groups, which reflect the user’s trains of thought (cf Section 6.5), are stored permanently on the workspace. They leave behind trails of actions used by the system to adapt to the user’s need and enabling users to trace and reflect on their actions. Therefore, their searches can be pursued over multiple sessions by accessing and retrieving groups when required again at a later stage.

The obvious next step to follow through is to conduct a user experiment that studies the long-term effects of grouping retrieval results. Ideally this should be done in the form of a longitudinal study, over a period of say three months, where users are asked to repeatedly use the system. This would allow us to observe if and how groups are changed over time. Moreover, my studies showed that the ICG is very good for long-term learning. However, this was established in simulated experiments and should be verified by real users.

### 8.3.2 *EGO* in a Collaborative Context

When using *EGO* the users leave trails of their actions—in the form of arrangements of groups—behind, which can also be examined by other people. Hence, *EGO* is suitable for a collaborative work context. The following scenarios would suggest supporting a collaborative environment:



- Design-oriented projects are often assigned to a team rather than a single person.
- It becomes more and more popular for online communities of users to share their ideas, thoughts, photographs, etc (eg blogs, Flickr).

On a collaborative workspace, people can easily pick up where other people have left off. However, other problems arise in a collaborative environment. Issues such as networking (client-server communication), privacy and access control have to be addressed then.

### 8.3.3 Browsing the Group-Space

In this work so far I have mainly addressed the creation and population of groups in *EGO*. Lightweight operations to create groups and the interactive recommendation system supporting a simple way to populate groups ensure it to be a successful tool for image management. However, the more groups are created the more difficult it will become to retain an overview of existing groups and locate groups on the workspace. One solution to this problem is to provide a tool to browse the group-space. This can be facilitated by creating links between groups, similar to hyperlinks between documents. The links can be based on time, eg all groups are linked that were created in the same session, which essentially creates some sort of history functionality. Another way of automatically creating links can be based on a similarity measure between groups. Alternatively, we can allow the user to explicitly state relationships between groups. Hence, a browsable space is created, in which navigation between groups is possible through a combination of hyperlinks and history functionality.

### 8.3.4 Applications to other Domains

*EGO* and the ICG have been developed primarily for interactive image retrieval. In theory, the same ideas can be employed to other types of media, especially for multimedia domains. For example, the conceptual ideas behind *EGO*, ie that categorising results helps solve tasks, have been recently explored in the text retrieval domain by Harper & Kelly (2006). They studied an interface which allowed users to categorise their results into a number of buckets and found that this process helped users in conceptualising their tasks.

In particular, the ICG can be employed as a general framework:

- to represent a variety of features for which it is not always obvious how individual features are best to be combined;
- to improve retrieval effectiveness by additionally encoding relationships between data objects or features.

This could be exploited in multimedia or even mixed-media domains. Multimedia, such as video, is naturally represented by a mixture of audio, text, visual and time-based features. All we need to encode these in the ICG is a similarity measure for each feature. Each feature is then represented as a layer in the graph, with k-NN links between nodes on that layer as defined by the particular similarity measure.

Mixed-media domains can exploit a synergy between the different types of media as long as some of the features are shared. I could envisage a joint modelling of image and text documents for example. Navigation from text to image nodes in the graph is then possible if at least some of the images have associated annotations. This would allow interesting operations, such as “return images that are suitable for illustrating a given piece of text”.

### 8.3.5 Minor Ramifications

Those chapters with experimental studies have pointed to areas that require improvement or further study. The main issues in the first recommendation algorithm discussed in Chapter 5 boiled down to the clustering algorithm used to obtain the multi-point queries. Although the clustering algorithm was not the focus of attention, it could have had a large impact on the retrieval performance of the multi-point query approach. Further it was discussed that an automatic detection mechanism of group homogeneity would help to decide when to use multiple query representatives over a single representative. Finally, I suggested exploring a feature weighting strategy for individual clusters rather than the whole group.

The user evaluation of Chapter 6 could be continued in multiple ways. I have already identified the possibility of studying the long-term effects of *EGO* in Section 8.3.1 and in a collaborative environment in Section 8.3.2. It would also be interesting to investigate different negative feedback strategies in *EGO*, in particular a comparison of implicit negative feedback strategies, as was employed here, and explicit ones.

Chapter 7 evaluated the improved recommendation system based on the ICG. In the ICG I aimed to address both long- and short-term learning. However, the short-term learning strategies that aimed at adapting link weights in the graph did not deliver the improvements I had hoped for. In addition, I would have liked to compare the ICG against other techniques that address the issue of feature integration. In the experiments I only explored the voting approach for combining the individual results in the baseline. This choice was based on the experiments of combination strategies in Chapter 5. However, other combination techniques or even completely different approaches to feature integration exist and should be considered for comparison (eg, Tong et al. 2005, Iyengar et al. 2005). As mentioned previously, I would also like to scrutinise the proposed model for other types of media.

### 8.3.6 Additional Ideas

#### Image annotation

The group-based environment is ideal for collecting annotations for images. Users are known to be reluctant to provide annotations for each image. It is not nearly as tedious to provide a label for a whole group of images. On the other hand, automatic image annotation provides an alternative route towards solving the problems associated with the semantic gap. Many works that attempt to learn associations between visual features and labels in order to predict new labels for unseen images have been discussed in Section 2.2.4. In particular, Pan et al. (2004) have already shown how a graph-based representation can be used to implement automatic label propagation. Together

with the workspace, *EGO* can provide an environment to collect explicit annotations from the user aided by automatically propagating labels to unlabelled images implemented by the ICG.

### **Alternative Results Presentation and Navigation Facilities**

The retrieval results are currently displayed in a conventional grid where images are arranged by descending similarity to the query. A simple enhancement would be to provide a clustered representation of results to make the user aware of inherent relationships between returned images.

In addition, *EGO* may benefit from navigation and browsing facilities. The browsable workspace has already been discussed as an example. However, the draw-back of this facility is that it will only allow browsing to groups that have already been created and therefore requires a substantial ground-work by the users. Sometimes, it would be sufficient to simply provide the user a facility to browse the entire collection. Adding groups obtained by pre-clustering the entire collection in addition to user-created groups on the workspace might provide a nice solution. Alternative techniques like ostensive browsing (Campbell & van Rijsbergen 1996) could be supported as well, which provide the user with a guided browser for exploratory searches (cf Section 3.1). By introducing two different modes—query and browse mode—the query panels in *EGO* can be replaced by an Ostensive Browser when switching to the browse mode. The user still has the option of storing images permanently in groups on the workspace, when coming across interesting images in the browsing phase.

### **Automatic or Semi-Automatic Facet Detection**

Finally, I can envisage a system that automatically detects facets given a particular query in order to assist the user in breaking up their tasks. I realise that this is an ambitious endeavour, however. It might be simpler to use the existing groups on the workspace and attempt to break up the query results according to these facets. In this scenario, the user will be presented with recommendations for each of the groups on their workspace (or a subset thereof), instead of an overall list of results.

---

## BIBLIOGRAPHY

---

- Aigrain, P., Zhang, H.-J. & Petkovic, D. (1996), 'Content-based representation and retrieval of visual media: A state-of-the-art review', *Multimedia Tools and Applications* **3**(3), 179–202.
- Arasu, A., Novak, J., Tomkins, A. & Tomlin, J. (2002), PageRank computation and the structure of the web: Experiments and algorithms, in 'Proc. of the 11th Int. World Wide Web Conference, Poster Track'.
- Armitage, L. H. & Enser, P. G. (1996), 'Analysis of user need in image archives', *Journal of Information Science* **23**(4), 287–299.
- Bates, M. J. (1989), 'The design of browsing and berrypicking techniques for the online search interface', *Online Review* **13**(5), 407–424.
- Bauer, D., Fastrez, P. & Hollan, J. (2004), Computationally-enriched 'piles' for managing digital photo collections, in 'Proc. of the IEEE Symposium on Visual Languages and Human Centric Computing (VLHSS'04)', IEEE Computer Society, Los Alamitos, CA, USA, pp. 193–195.
- Beaulieu, M. & Jones, S. (1998), 'Interactive searching and interface issues in the Okapi best match probabilistic retrieval system', *Interacting with Computers* **10**(3), 237–248.
- Bederson, B. B. (2001), Photomesa: a zoomable image browser using quantum treemaps and bubblemaps, in 'Proc. of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST '01)', ACM Press, New York, NY, USA, pp. 71–80.
- Belkin, N. J. (2003), Interface techniques for making searching for information more effective, in 'CHI'2003, Best Practices and Future Visions for Search UIs: A Workshop'.
- Belkin, N. J., Oddy, R. N. & Brooks, H. M. (1982), 'Ask for information retrieval: Part i — background and theory', *Journal of Documentation* **38**(2), 61–71.
- Benchathlon (n.d.), 'The Benchathlon Network', Home of CBIR Benchmarking.  
**URL:** <http://www.benchathlon.net/>
- BerkleyCorel (2005), 'Berkley's list of Corel CD names, images keywords and captions', Berkley's Digital Library Project.  
**URL:** <http://elib.cs.berkeley.edu/photos/corel/>
- Black, Jr., J. A., Fahmy, G. & Panchanathan, S. (2002), A method for evaluating the performance of content-based image retrieval systems based on subjectively determined similarity between images, in 'CIVR 2002, LNCS 2383', Springer-Verlag, pp. 356–366.
- Böhm, C., Berchtold, S. & Keim, D. A. (2001), 'Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases', *ACM Comput. Surv.* **33**(3), 322–373.
- Borlund, P. (2003a), 'The concept of relevance in ir', *Journal of the American Society for Information Science* **54**(10), 913–925.
- Borlund, P. (2003b), 'The IIR evaluation model: A framework for evaluation of interactive information retrieval systems', *Information Research* **8**(3).
- Borlund, P. & Ingwersen, P. (1997), 'The development of a method for the evaluation of interactive information retrieval systems', *Journal of Documentation* **53**(3), 225–250.

## BIBLIOGRAPHY

---

- Bradshaw, B. (2000), Semantic based image retrieval: A probabilistic approach, in 'Proc. of the ACM Int. Conf. on Multimedia (Multimedia-00)', ACM Press, New York, NY, USA, pp. 167–176.
- Brin, S. & Page, L. (1998), 'The anatomy of a large-scale hypertextual Web search engine', *Computer Networks and ISDN Systems* **30**(1–7), 107–117.
- Campbell, I. (2000a), 'Interactive evaluation of the Ostensive Model, using a new test-collection of images with multiple relevance assessments', *Information Retrieval* **2**(1), 89–114.
- Campbell, I. (2000b), The ostensive model of developing information needs, PhD thesis, University of Glasgow.
- Campbell, I. & van Rijsbergen, C. J. (1996), The ostensive model of developing information needs, in 'Proc. of the Int. Conf. on Conceptions of Library and Informaion Science', pp. 251–268.
- Carneiro, G. & Vasconcelos, N. (2005), A database centric view of semantic image annotation and retrieval, in 'Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'05)', ACM Press, New York, NY, USA, pp. 559–566.
- Chalmers, M., Rodden, K. & Brodbeck, D. (1998), 'The order of things: Activity-centred information access', *Computer Networks and ISDN Systems* **30**(1–7), 359–367.
- Chang, E., Goh, K., Sychay, G. & Wu, G. (2003), 'CBSA: Content-based soft annotation for multimodal image retrieval using bayes point machines', *IEEE Trans. Circuits Syst. Video Technol. (Special Issue on Conceptual and Dynamical Aspects of Multimedia Content Description)* **13**(1), 26–38.
- Chen, J.-Y., Bouman, C. A. & Dalton, J. C. (2000), 'Hierarchical browsing and search of large image databases', *IEEE Trans. Image Processing* **9**(3), 442–455.
- COREL (n.d.), 'COREL clipart and photos'.  
**URL:** <http://www.corel.com/>
- Cousins, S. B., Paepcke, A., Winograd, T., Bier, E. A. & Pier, K. (1997), The digital library integrated task environment (DLITE), in 'Computer-Human Interactions '97', pp. 142–151.
- Cox, I. J., Miller, M. L., Minka, T. P., Papatthomas, T. V. & Yianilos, P. N. (2000), The bayesian image retrieval retrieval system, PicHunter: Theory, implementation and psychophysical experiments, in 'IEEE Trans. Image Processing', Vol. 9, pp. 20–37.
- Cox, I. J., Miller, M. L., Omohundro, S. M. & Yianilos, P. N. (1996), PicHunter: Bayesian relevance feedback for image retrieval, in 'Proc. of the Int. Conf. on Pattern Recognition', pp. 361–369.
- Cunningham, S. J., Bainbridge, D. & Masoodian, M. (2004), How people describe their image information needs: a grounded theory analysis of visual arts queries, in 'Proc. of the 2004 Joint ACM/IEEE Conf. on Digital Libraries', pp. 47–48.
- Datta, R., Li, J. & Wang, J. Z. (2005), Content-based image retrieval: approaches and trends of the new age, in 'Proc. of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'05)', ACM Press, New York, NY, USA, pp. 253–262.
- Davis, M., King, S., Good, N. & Sarvas, R. (2004), From context to content: leveraging context to infer media metadata, in 'Proc. of the ACM Int. Conf. on Multimedia (MULTIMEDIA '04)', ACM Press, New York, NY, USA, pp. 188–195.
- de Freitas, N., Brochu, E., Barnard, K., Duygulu, P. & Forsyth, D. (2002), 'Bayesian models for massive multimedia databases: a new frontier', 7th Valencia International Meeting on Bayesian Statistics/2002 ISBA International Meeting.
- Del Bimbo, A. (1999), *Visual Information Retrieval*, Morgan Kaufmann Publishers.
- Dimitrova, N., Zhang, H.-J., Shahraray, B., Sezan, I., Huang, T. & Zakhor, A. (2002), 'Applications of video-content analysis and retrieval', *IEEE Multimedia* **9**(3), 42–55.

## BIBLIOGRAPHY

---

- Drucker, S. M., Wong, C., Roseway, A., Glenner, S. & Mar, S. D. (2004), Mediabrowser: reclaiming the shoebox, in 'Proc. of the Working Conf. on Advanced Visual Interfaces (AVI '04)', ACM Press, New York, NY, USA, pp. 433–436.
- Duda, R. O., Hart, P. E. & Stork, D. G. (2001), *Pattern Classification*, 2nd edn, Wiley-Interscience.
- Dunlop, M. (2000), 'Reflections on Mira: interactive evaluation in information retrieval', *Journal of the American Society for Information Science* **51**(14), 1269–1274.
- Duygulu, P., Barnard, K., de Freitas, N. & Forsyth, D. (2002), Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in 'Seventh European Conference on Computer Vision', Springer-Verlag, pp. IV:97–112.
- Dwork, C., Kumar, R., Naor, M. & Sivakumar, D. (2001), Rank aggregation methods for the web, in 'Proc. of the Int. World Wide Web Conf.', ACM Press, pp. 613–622.
- Eakins, J. P. (2001), Trademark image retrieval, in Lew (2001), pp. 319–350.
- Eakins, J. P. (2002), 'Towards intelligent image retrieval', *Pattern Recognition* **35**(1), 3–14.
- Eakins, J. P. & Graham, M. (1999), 'Content-based image retrieval', JISC Technology Applications Report 39.
- Enser, P. G. (2000), 'Visual information retrieval: seeking the alliance of concept-based and content-based paradigms', *Journal of Information Science* **26**(4), 199–210.
- Fagin, R., Kumar, R. & Sivakumar, D. (2003), Efficient similarity search and classification via rank aggregation, in 'Proc. of the ACM SIGMOD Int. Conf. on Management of Data', ACM Press, New York, NY, USA, pp. 301–312.
- Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. & Yanker, P. (1995), 'Query by image and video content: The QBIC system', *Computer* **28**(9), 23–32.
- Forsyth, D. A. (2001), Benchmarks for storage and retrieval in multimedia databases, in 'Proc. of SPIE - The International Society for Optical Engineering', Vol. 4676, pp. 240–247.
- Forsyth, D. A. & Ponce, J. (2003), *Computer Vision: A Modern Approach*, Prentice-Hall, New Jersey.
- Garber, S. R. & Grunes, M. B. (1992), The art of search: A study of art directors, in 'Proc. of the ACM Int. Conf. on Human Factors in Computing Systems (CHI'92)', pp. 157–163.
- Gentner, D. & Nielsen, J. (1996), 'The anti-mac interface', *Communications of the ACM* **39**(8), 70–82.
- Gentner, D. & Stevens, A. L., eds (1983), *Mental Models*, Lawrence Erlbaum Associates, Hillsdale, NJ, USA.
- Gevers, T. (2001), Color-based retrieval, in Lew (2001), pp. 11–49.
- Ghoshal, A., Ircing, P. & Khudanpur, S. (2005), Hidden markov models for automatic annotation and content-based retrieval of images and video, in 'Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'05)', ACM Press, New York, NY, USA, pp. 544–551.
- Girgensohn, A., Adcock, J., Cooper, M., Foote, J. & Wilcox, L. (2003), Simplifying the management of large photo collections, in 'Proc. of the Int. Conf. on Human-Computer Interaction (INTERACT'03)', pp. 196–203.
- Grant, K. D., Graham, A., Nguyen, T., Paepcke, A. & Winograd, T. (2003), Beyond the shoe box: Foundations for flexibly organizing photographs on a computer, Technical report, Computer Science Department, Stanford University.
- Grosky, W. I., Fotouhi, F., Sethi, I. K. & Capatina, B. (1994), 'Using metadata for the intelligent browsing of structured media objects', *SIGMOD Record (ACM Special Interest Group on Management of Data)* **23**(4), 49–56.

## BIBLIOGRAPHY

---

- Han, J., Li, M., Zhang, H. & Guo, L. (2005), 'A memory learning framework for effective image retrieval', *IEEE Trans. Image Processing* **14**(4), 511–524.
- Harper, D. J. & Kelly, D. (2006), Contextual relevance feedback, in 'IiX: Proc. of the 1st Int. Conf. on Interaction in Context', ACM Press, New York, NY, USA, pp. 129–137.
- He, J., Tong, H., Li, M., Ma, W.-Y. & Zhang, C. (2005), Multiple random walks and its application in content-based image retrieval, in 'Proc. of the 7th ACM SIGMM Int. Workshop on Multimedia Information Retrieval (MIR'05)', ACM Press, pp. 151–158.
- He, X., King, O., Ma, W.-Y., Li, M. & Zhang, H.-J. (2003), 'Learning a semantic space from user's relevance feedback for image retrieval', *IEEE Trans. Circuits Syst. Video Technol.* **13**(1), 39–48.
- Heesch, D. & Rüger, S. (2004a),  $NN^k$  networks for content-based image retrieval, in 'Proc. of the European Conf. on Information Retrieval', Vol. 2997 of *Lecture Notes in Computer Science*, Springer-Verlag, Heidelberg, Germany, pp. 253–266.
- Heesch, D. & Rüger, S. (2004b), Three interfaces for content-based access to image collections, in 'Proc. of the Int. Conf. on Image and Video Retrieval', Vol. 3115 of *Lecture Notes in Computer Science*, Springer-Verlag, Heidelberg, Germany, pp. 491–499.
- Henderson, Jr., D. A. & Card, S. (1986), 'Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface', *ACM Trans. Graph.* **5**(3), 211–243.
- Hendry, D. G. (1996), Extensible Information-Seeking Environments, PhD thesis, The Robert Gordon University, Aberdeen.
- Hendry, D. G. (2006), 'Workspaces for search', *Journal of the American Society for Information Science* **57**(6), 800–802.
- Hendry, D. G. & Harper, D. J. (1997), 'An informal information-seeking environment', *Journal of the American Society for Information Science* **48**(11), 1036–1048.
- Hu, M.-K. (1962), 'Visual pattern recognition by moment invariants', *IEEE Trans. Information Theory* **8**(2), 179–187.
- ImageCLEF (n.d.), 'The CLEF Cross Language Image Retrieval Track (ImageCLEF)'.  
**URL:** <http://ir.shef.ac.uk/imageclef/>
- Ingwersen, P. (1992), *Information Retrieval Interaction*, Taylor Graham, London, England.
- Ingwersen, P. (1996), 'Cognitive perspectives of information retrieval interaction', *Journal of Documentation* **52**(1), 3–50.
- Ishikawa, Y., Subramanya, R. & Faloutsos, C. (1998), MindReader: Querying databases through multiple examples, in A. Gupta, O. Shmueli & J. Widom, eds, 'Proc. of the 24th Int. Conf. on VLDB', Morgan Kaufmann Publishers, New York, NY, USA, pp. 218–227.
- Iyengar, G., Duygulu, P., Feng, S., Ircing, P., Khudanpur, S. P., Klakow, D., Krause, M. R., Manmatha, R., Nock, H. J., Petkova, D., Pytlik, B. & Virga, P. (2005), Joint visual-text modeling for automatic retrieval of multimedia documents, in 'Proc. of the ACM Int. Conf. on Multimedia (MULTIMEDIA'05)', ACM Press, New York, NY, USA, pp. 21–30.
- Jaimes, A., Christel, M., Gilles, S., Sarukkai, R. & Ma, W.-Y. (2005), Multimedia information retrieval: what is it, and why isn't anyone using it?, in 'Proc. of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'05)', ACM Press, New York, NY, USA, pp. 3–8.
- Jain, A. K. & Vailaya, A. (1998), 'Shape-based retrieval: A case study with trademark image databases', *Pattern Recognition* **31**(9), 1369–1390.
- Jain, R. (2003), Semantics in multimedia systems, in 'Proc. of Int. Conf. on Multimedia Modeling'. Key note talk.

## BIBLIOGRAPHY

---

- Järvelin, K. & Wilson, T. (2003), 'On conceptual models for information seeking and retrieval research', *Information Research* **9**(1), paper 163.
- Jeon, J., Lavrenko, V. & Manmatha, R. (2003), Automatic image annotation and retrieval using cross-media relevance models, in 'Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR03)', ACM Press, New York, NY, USA, pp. 119–126.
- Johnson-Laird, P., Girotto, V. & Legrenzi, P. (1998), 'Mental models: a gentle guide for outsiders', The Interdisciplinary Committee on Organizational Studies, University of Michigan.  
**URL:** <http://www.si.umich.edu/ICOS/gentleintro.html>
- Jose, J. M. (1998), An Integrated Approach for Multimedia Information Retrieval, PhD thesis, The Robert Gordon University, Aberdeen.
- Jose, J. M., Furner, J. & Harper, D. J. (1998), Spatial querying for image retrieval: A user-oriented evaluation, in 'Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'98)', ACM Press, New York, NY, USA, pp. 232–240.
- Jose, J. M. & Harper, D. J. (1997), A retrieval mechanism for semi-structured photographic collections, in A. Hameurlain & A. Tjoa, eds, 'Proc. of the Int. Conf. on Database and Expert Systems Applications', Toulouse, France, pp. 276–292. (Also in Lecture Notes Computer Science num. 1308).
- Kang, H. & Shneiderman, B. (2003), Mediafinder: An interface for dynamic personal media management with semantic regions, in 'Proc. of the ACM Int. Conf. on Human Factors in Computing Systems CHI'2003', pp. 764–765.
- Karger, D. R., Bakshi, K., David, H., Quan, D. & Sinha, V. (2005), Haystack: A general purpose information management tool for end users of semistructured data, in 'Proc. of the Second Biennial Conf. on Innovative Data Systems Research (CIDR 2005)', pp. 13–26.
- Kim, D.-H. & Chung, C.-W. (2003), Qcluster: relevance feedback using adaptive clustering for content-based image retrieval, in 'Proc. of the ACM SIGMOD Int. Conf. on Management of Data', ACM Press, pp. 599–610.
- Kirsch, D. (1995), 'The intelligent use of space', *Artificial Intelligence* **73**, 31–68.
- Kuchinsky, A., Pering, C., Creech, M. L., Freeze, D., Serra, B. & Gwizdka, J. (1999), Fotofile: a consumer multimedia organization and retrieval system, in 'Proc. of the ACM Int. Conf. on Human Factors in Computing Systems (CHI '99)', ACM Press, New York, NY, USA, pp. 496–503.
- Langville, A. N. & Meyer, C. D. (2004), 'Deeper inside PageRank', *Internet Mathematics* **1**(3), 335–400.
- Lee, D. L., Chuang, H. & Seamons, K. (1997), 'Document ranking and the vector-space model', *IEEE Softw.* **14**(2), 67–75.
- Lee, J. H. (1997), Analyses of multiple evidence combination, in N. J. Belkin, A. D. Narasimhalu & P. Willett, eds, 'Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR 97)', ACM Press, pp. 267–276.
- Lew, M. S., ed. (2001), *Principles of Visual Information Retrieval*, Advances in Pattern Recognition, Springer-Verlag.
- Lewis, J. P. & Trail, A. (1999), *Statistics Explained*, Addison-Wesley Lonhman Limited, Harlow, England.
- Lim, J.-H. (1999), Learning visual keywords for content-based retrieval, in 'IEEE Proc. of Int. Conf. on Multimedia Computing and Systems', Vol. 2, pp. 169–173.
- Lim, J.-H. (2000), Explicit query formulation with visual keywords, in 'Proc. of the ACM Int. Conf. on Multimedia (Multimedia-00)', ACM Press, N. Y., pp. 407–409.
- Lin, Y.-Y., Liu, T.-L. & Chen, H.-T. (2005), Semantic manifold learning for image retrieval, in 'Proc. of the ACM Int. Conf. on Multimedia (MULTIMEDIA '05)', ACM Press, New York, NY, USA, pp. 249–258.



## BIBLIOGRAPHY

---

- Loncaric, S. (1998), 'A survey of shape analysis techniques', *Pattern Recognition* **31**(8), 983–1001.
- Lovasz, L. (1993), 'Random walks on graphs: A survey', *Combinatorics, Paul Erdos is Eighty* **2**, 353–398.
- Low, A. (1999), A folder-based graphical interface for an information retrieval system, Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science.
- Lu, G. (1999), 'Design issues of multimedia information indexing and retrieval systems', *Journal of Network and Computer Applications (Academic Press)* **22**(3), 175–198.
- Ma, W.-Y. & Manjunath, B. S. (1998), 'A texture thesaurus for browsing large aerial photographs', *Journal of the American Society for Information Science* **49**(7), 633–648.
- Ma, W.-Y. & Manjunath, B. S. (1999), 'Netra: A toolbox for navigating large image databases', *ACM Multimedia Systems Journal* **7**(3), 184–198.
- MacLean, A., Bellotti, V., Young, R. & Moran, T. (1991), Reaching through analogy: a design rationale perspective on roles of analogy, in 'Proc. of the ACM Int. Conf. on Human Factors in Computing Systems (CHI'91)', ACM Press, New York, NY, USA, pp. 167–172.
- Malone, T. W. (1983), 'How do people organize their desks?: Implications for the design of office information systems', *ACM Trans. Office Information Systems* **1**(1), 99–112.
- Manjunath, B. S. & Ma, W.-Y. (1996), 'Texture features for browsing and retrieval of image data', *IEEE Trans. Pattern Analysis and Machine Intelligence* **18**(8), 837–842.
- Manjunath, B. S., Ohm, J.-R., Vasudevan, V. V. & Yamada, A. (2001), 'Color and texture descriptors', *IEEE Trans. Circuits Syst. Video Technol.* **11**(6), 703–715.
- Manjunath, B. S., Wu, P., Newsam, S. & Shin, H. D. (2000), 'A texture descriptor for browsing and similarity retrieval', *Signal Processing: Image Communication (Elsevier)* **16**(1–2), 33–43.
- Markkula, M. & Sormunen, E. (2000), 'End-user searching challenges indexing practices in the digital newspaper photo archive', *Information Retrieval* **1**(4), 259–285.
- Markkula, M., Tico, M., Sepponen, B., Nirkkonen, K. & Sormunen, E. (2001), 'A test collection for the evaluation of content-based image retrieval algorithms—A user and task-based approach', *Inf. Retr.* **4**(3–4), 275–293.
- Maxwell, S. E. & Delaney, H. D. (1990), *Designing Experiments and Analysing Data*, Wadsworth Publishing Company, Belmont, CA, USA.
- McDonald, S., Lai, T.-S. & Tait, J. (2001), Evaluating a content based image retrieval system, in 'Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval', New Orleans, Louisiana, USA, pp. 232–240.
- Meilhac, C. & Nastar, C. (1999), Relevance feedback and category search in image databases, in 'IEEE Proc. of Int. Conf. on Multimedia Computing and Systems', Vol. 1, pp. 512–517.
- Miller, G. (1956), 'The magical number seven, plus or minus two: Some limits on our capacity for processing information', *The Psychological Review* **63**, 81–97.
- Minka, T. P. & Picard, R. W. (1996), Interactive learning using a 'society of models', in 'IEEE Proc. of Conf. on Computer Vision and Pattern Recognition(CVPR-1996)', pp. 447–452.
- Moëllic, P.-A., Hède, P., Grefenstette, G. & Millet, C. (2005), 'Evaluating content based image retrieval techniques with the one million images clic testbed', *Trans. Engineering, Computing and Technology* **V4**, 171–174.
- Mulhem, P. & Lim, J.-H. (2002), Symbolic photograph content-based retrieval, in 'Proc. of the ACM Int. Conf. on Information and Knowledge Management (CIKM'02)', ACM Press, New York, NY, USA, pp. 94–101.

## BIBLIOGRAPHY

---

- Müller, H. (2002), User Interaction and Performance Evaluation in Content-Based Visual Information Retrieval, PhD thesis, Université de Genève.
- Müller, H., Marchand-Maillet, S. & Pun, T. (2002), The truth about Corel - evaluation in image retrieval, in 'CIVR 2002, LNCS 2388', Springer-Verlag, pp. 38–49.
- Nakakoji, K., Yamamoto, Y., Takada, S. & Reeves, B. N. (2000), Two-dimensional spatial positioning as a means for reflection in design, in 'Proc. of the Conf. on Designing Interactive Systems (DIS'00)', ACM Press, New York, NY, USA, pp. 145–154.
- Nakazato, M., Dagli, C. & Huang, T. S. (2003), Evaluating group-based relevance feedback for content-based image retrieval, in 'IEEE Proc. of Int. Conf. on Image Processing', Vol. 2, pp. 599–602.
- Nakazato, M., Manola, L. & Huang, T. S. (2002), ImageGrouper: Search, annotate and organize images by groups, in 'Proc. of the Fifth Int. Conf. on Visual Information Systems (VISual 2002)', Vol. 2314 of LNCS, Springer-Verlag, Heidelberg, Germany, pp. 129–142.
- Nakazato, M., Manola, L. & Huang, T. S. (2003), 'ImageGrouper: A group-oriented user interface for content-based image retrieval and digital image arrangement', *Journal of Visual Languages and Computing* **14**, 363–386.
- Naphade, M. R., Kennedy, L., Kender, J. R., Chang, S.-F., Smith, J. R., Over, P. & Hauptmann, A. (2005), A light scale concept ontology for multimedia understanding for TRECVID 2005, Technical Report RC23612 (W0505-104), IBM Research.
- Norman, D. A. (1988), *The Design of Everyday Things*, The MIT Press, London, England.
- O'Hare, N., Gurrin, C., Lee, H., Murphy, N., Smeaton, A. F. & Jones, G. J. (2005), My digital photos: where and when?, in 'Proc. of the ACM Int. Conf. on Multimedia (MULTIMEDIA '05)', ACM Press, New York, NY, USA, pp. 261–262.
- Oliva, A. & Torralba, A. (2001), 'Modeling the shape of the scene: A holistic representation of the spatial envelope', *Int. Journal of Computer Vision* **42**(3), 145–175.
- Page, L., Brin, S., Motwani, R. & Winograd, T. (1998), The pagerank citation ranking: Bringing order to the web, Technical report, Stanford Digital Library Technologies Project.
- Pan, J.-Y., Yang, H.-J., Faloutsos, C. & Duygulu, P. (2004), GCap: Graph-based automatic image captioning, in 'Proc. of the 4th Int. Workshop on Multimedia Data and Document Engineering (MDDE 04), in conjunction with CVPR'04', pp. 146–155.
- Pečenović, Z., Do, M., Vetterli, M. & Pu, P. (2000), Integrated browsing and searching of large image collections, in 'Visual Information and Information Systems', pp. 279–289.
- Peng, J., Bhanu, B. & Qing, S. (1999), 'Probabilistic feature relevance learning for content-based image retrieval', *Computer Vision and Image Understanding* **75**(1/2), 150–164.
- Pentland, A., Picard, R. W. & Sclaroff, S. (1993), Photobook: Content-Based Manipulation of Image Databases, Technical Report 255, MIT Media Laboratory Perceptual Computing Computing.
- Pentland, A., Picard, R. W. & Sclaroff, S. (1994), Photobook: Tools for content-based manipulation of image databases, in 'Proc. Storage and Retrieval for Image and Video Databases II', Vol. 2185 of *SPIE Storage and Retrieval for Image and Video Databases II*, SPIE, San Jose, CA, USA, pp. 34–47.
- Picard, R. W. (1997), *Affective Computing*, MIT Press, Cambridge, MA.
- Picard, R. W. & Liu, F. (1994), A new Wold ordering for image similarity, in 'Proc. of the IEEE Conf. Acoustics Speech and Signal Processing', Adelaide, Australia, pp. 129–132.
- Porkaew, K., Chakrabarti, K. & Mehrotra, S. (1999), Query refinement for multimedia similarity retrieval in MARS, in 'Proc. of the ACM Int. Conf. on Multimedia', Orlando, Florida, pp. 235–238.
- Porter, M. F. (1997), 'An algorithm for suffix stripping', pp. 313–316.

## BIBLIOGRAPHY

---

- Puzicha, J., Rubner, Y., Tomasi, C. & Buhmann, J. M. (1999), Empirical evaluation of dissimilarity measures for color and texture, *in* 'IEEE Proc. of Int. Conf. on Computer Vision (ICCV'99)', pp. 1165–1173.
- Rath, T. M., Manmatha, R. & Lavrenko, V. (2004), A search engine for historical manuscript images, *in* 'Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '04)', ACM Press, New York, NY, USA, pp. 369–376.
- Ravela, S. & Luo, C. (2000), Appearance-based global similarity of images, *in* B. W. Croft, ed., 'Advances in Information Retrieval - Recent Research from the Center for Intelligent Information', Kluwer Academic Publishers, pp. 267–303.
- Ravela, S. & Manmatha, R. (1997), Image retrieval by appearance, *in* 'Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval', Image Retrieval, pp. 278–285.
- Rocchio, J. J. (1971), Relevance feedback in information retrieval, *in* G. Salton, ed., 'The SMART retrieval system: experiments in automatic document processing', Prentice-Hall, Englewood Cliffs, US, pp. 313–323.
- Rodden, K. (1999), How do people organise their photographs?, *in* 'Proc. of the 21st BCS IRSG Colloquium on IR, Electronic Workshops in Computing'.  
**URL:** <http://www.ewic.org.uk>
- Rodden, K., Basalaj, W., Sinclair, D. & Wood, K. (2001), Does organisation by similarity assist image browsing?, *in* 'Proc. of the ACM Int. Conf. on Human Factors in Computing Systems', Sensible Navigation Search, pp. 190–197.
- Rodden, K. & Wood, K. R. (2003), How do people manage their digital photographs?, *in* 'Proc. of the ACM Int. Conf. on Human Factors in Computing Systems (CHI '03)', ACM Press, New York, NY, USA, pp. 409–416.
- Rubner, Y. (1999), Perceptual Metrics for Image Database Navigation, PhD thesis, Stanford University, USA.
- Rui, Y. & Huang, T. S. (2000), Optimizing learning in image retrieval, *in* 'IEEE Proc. of Conf. on Computer Vision and Pattern Recognition (CVPR-00)', IEEE Computer Society Press, Los Alamitos, pp. 236–245.
- Rui, Y., Huang, T. S. & Chang, S.-F. (1999), 'Image retrieval: Current techniques, promising directions and open issues', *Journal of Visual Communication and Image Representation (Academic Press)* **10**(4), 39–62.
- Rui, Y., Huang, T. S. & Mehrotra, S. (1997), Content-based image retrieval with relevance feedback in MARS, *in* 'IEEE Proc. of Int. Conf. on Image Processing (ICIP'97)', pp. 815–818.
- Rui, Y., Huang, T. S., Ortega, M. & Mehrotra, S. (1998), 'Relevance feedback: A power tool for interactive content-based image retrieval', *IEEE Trans. Circuits Syst. Video Technol.* **8**(5), 644–655. Special Issue on Segmentation, Description, and Retrieval of Video Content.
- Rummukainen, M., Laaksonen, J. & Koskela, M. (2003), An efficiency comparison of two content-based image retrieval systems, GIFT and PicSOM, *in* 'Proc. of the Int. Conf. on Image and Video Retrieval', pp. 500–510.
- Ruthven, I. (2000), 'Incorporating aspects of information use into relevance feedback', *Journal of Information Retrieval* **2**(1), 83–88.
- Ruthven, I. (2005), *Integrating approaches to relevance*, Vol. 19 of *Information Retrieval Series*, Springer-Verlag, chapter 4, pp. 61–80.
- Salton, G. & Buckley, C. (1990), 'Improving retrieval performance by relevance feedback', *Journal of the American Society for Information Science* **41**(4), 288–297.
- Salton, G. & McGill, M. J. (1983), *Introduction to Modern Information Retrieval*, McGraw-Hill, Tokio.

## BIBLIOGRAPHY

---

- Salvador, S. & Chan, P. (2003), Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms, Technical Report CS-2003-18, Florida Institute of Technology.
- Santini, S., Gupta, A. & Jain, R. (2001), 'Emergent semantics through interaction in image databases', *IEEE Trans. Knowledge and Data Engineering* **13**(3), 337–351.
- Santini, S. & Jain, R. (2000), 'Integrated browsing and querying for image databases', *IEEE Trans. Multimedia* **7**(3), 26–39.
- Sebe, N. & Lew, M. S. (2001), Texture features for content-based retrieval, in Lew (2001), pp. 51–85.
- Sharma, M., Markou, M. & Singh, S. (2001), Evaluation of texture methods for image analysis, in 'Proc. of the 7th Australian and New Zealand Intelligent Information Systems Conference', pp. 117–121.
- Shneiderman, B. & Kang, H. (2000), Direct annotation: A drag-and-drop strategy for labeling photos, in 'Fourth Int. Conf. on Information Visualisation (IV'00)', IEEE Computer Society Press, pp. 88–95.
- Smeulders, A. W., Worring, M., Santini, S., Gupta, A. & Jain, R. (2000), 'Content-based image retrieval at the end of the early years', *IEEE Trans. Pattern Analysis and Machine Intelligence* **22**(12), 1349–1380.
- Smith, J. R. & Chang, S.-F. (1996), Local color and texture extraction and spatial query, in 'IEEE Proc. of Int. Conf. on Image Processing (ICIP-96)', Vol. 3, Lausanne, Switzerland, pp. 1011–1014.
- Snoek, C. G., Worring, M., van Gemert, J. C., Geusebroek, J.-M. & Smeulders, A. W. (2006), The challenge problem for automated detection of 101 semantic concepts in multimedia, in 'Proc. of the ACM Int. Conf. on Multimedia (MULTIMEDIA '06)', ACM Press, New York, NY, USA.
- Sonka, M., Hlavac, V. & Boyle, R. (1998), *Image Processing, Analysis, and Machine Vision*, 2nd edn, Thomson-Engineering, Toronto, Canada.
- Squire, D. M. & Pun, T. (1998), 'Assessing agreement between human and machine clustering of image databases', *Pattern Recognition* **31**(12), 1905–1919.
- Srikanth, M., Varner, J., Bowden, M. & Moldovan, D. (2005), Exploiting ontologies for automatic image annotation, in 'Proc. of the Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '05)', ACM Press, New York, NY, USA, pp. 552–558.
- Stricker, M. & Orengo, M. (1995), Similarity of color images, in 'Proc. of the SPIE: Storage and Retrieval for Image and Video Databases', Vol. 2420, pp. 381–392.
- Su, Z. & Zhang, H.-J. (2002), Relevance feedback in CBIR, in X. Zhou & P. Pu, eds, 'Sixth Working Conference on Visual Database Systems (VDB'02), May 29-31, 2002, Brisbane, Australia', Vol. 216 of *IFIP Conference Proceedings*, Kluwer Academic Publishers, pp. 21–35.
- Swain, M. J. & Ballard, D. H. (1991), 'Color indexing', *Int. Journal of Computer Vision* **7**(1), 11–32.
- Tamura, H. & Yokoya, N. (1984), 'Image database systems: A survey', *Pattern Recognition* **17**(1), 29–43.
- ter Hofstede, A. H. M., Proper, H. A. & van der Weide, T. P. (1996), 'Query formulation as an information retrieval problem', *The Computer Journal* **39**(4), 255–274.
- Theodoridis, S. & Koutroumbas, K. (1999), *Pattern Recognition*, Academic Press.
- Tieu, K. & Viola, P. (2000), Boosting image retrieval, in 'IEEE Proc. of Conf. on Computer Vision and Pattern Recognition (CVPR-00)', IEEE, Los Alamitos, pp. 228–235.
- Tong, H., He, J., Li, M., Zhang, C. & Ma, W.-Y. (2005), Graph based multi-modality learning, in 'Proc. of the ACM Int. Conf. on Multimedia (MULTIMEDIA'05)', ACM Press, New York, NY, USA, pp. 862–871.
- Tong, S. & Chang, E. (2001), Support vector machine active learning for image retrieval, in 'Proc. of the ACM Int. Conf. on Multimedia', ACM Press, pp. 107–118.

## BIBLIOGRAPHY

---

- TrecVid (2003), *Proc. of the TREC Video Retrieval Evaluation Conference (TRECVID2003)*, Gaithersburgh, MD, USA.  
**URL:** <http://www-nlpir.nist.gov/projects/tv2003/>
- TrecVid (2005), *Proc. of the TREC Video Retrieval Evaluation Conference (TRECVID2005)*, Gaithersburgh, MD, USA.  
**URL:** <http://www-nlpir.nist.gov/projects/tv2005/>
- TrecVid (2006), *Proc. of the TREC Video Retrieval Evaluation Conference (TRECVID2006)*, NIST, Gaithersburgh, MD, USA.  
**URL:** <http://www-nlpir.nist.gov/projects/tv2006/>
- TREC (n.d.), 'Text REtrieval Convergence', National Institute for Standards and Technology Retrieval Group.  
**URL:** <http://trec.nist.gov/>
- Truran, M., Goulding, J. & Ashman, H. (2005), Co-active intelligence for image retrieval, in 'Proc. of the ACM Int. Conf. on Multimedia (MULTIMEDIA'05)', ACM Press, New York, NY, USA, pp. 547–550.
- T.V., A., Gupta, R. & Ghosal, S. (2002), Adaptable similarity search using non-relevant information, in 'Proc. of the Int. Conf. on Very Large Data Bases (VLDB'02)', pp. 47–58.
- Urban, J. & Jose, J. M. (2004a), EGO: A personalised multimedia management tool, in 'Proc. of the 2nd Int. Workshop on Adaptive Multimedia Retrieval (AMR'04)', pp. 3–17.
- Urban, J. & Jose, J. M. (2004b), Evidence combination for multi-point query learning in content-based image retrieval, in 'Proc. of the IEEE Sixth Int. Symposium on Multimedia Software Engineering (ISMSE'04)', pp. 583–586.
- Urban, J. & Jose, J. M. (2005), Exploring results organisation for image searching, in 'Proc. of the Tenth IFIP TC13 Int. Conf. on Human-Computer Interaction (INTERACT 2005)', LNCS 3585, Springer-Verlag, pp. 958–961.
- Urban, J. & Jose, J. M. (2006a), Adaptive image retrieval using a graph model for semantic feature integration, in 'Proc. of the 8th ACM SIGMM Int. Workshop on Multimedia Information Retrieval (MIR'06)', ACM Press.
- Urban, J. & Jose, J. M. (2006b), Can a workspace help to overcome the query formulation problem in image retrieval?, in 'Proc. of the European Conf. on Information Retrieval (ECIR 2006)', Vol. 3936 of LNCS, Springer-Verlag, pp. 385–396.
- Urban, J. & Jose, J. M. (2006c), 'EGO: A personalised multimedia management and retrieval tool', *International Journal of Intelligent Systems (IJIS)*, Special Issue on 'Intelligent Multimedia Retrieval' **21**(7), 725–745.
- Urban, J. & Jose, J. M. (2006d), 'Evaluating a workspace's usefulness for image retrieval', *ACM Multimedia Systems Journal (Special Issue on User-Centered Multimedia)*.  
**URL:** <http://dx.doi.org/10.1007/s00530-006-0051-z>
- Urban, J. & Jose, J. M. (2006e), An explorative study of interface support for image searching, in 'Adaptive Multimedia Retrieval: User, Context, and Feedback: Third International Workshop, AMR 2005', Vol. 3877 of LNCS, Springer-Verlag, pp. 207–221.
- Urban, J., Jose, J. M. & van Rijsbergen, C. J. (2003), An adaptive approach towards content-based image retrieval, in 'Proc. of the Third International Workshop on Content-Based Multimedia Indexing (CBMI'03)', Rennes, France, pp. 119–126.
- Urban, J., Jose, J. M. & van Rijsbergen, C. J. (2005), 'An adaptive technique for content-based image retrieval', *Multimedia Tools and Applications*.  
**URL:** <http://dx.doi.org/10.1007/s11042-006-0035-1>
- van Rijsbergen, C. J. (1979), *Information Retrieval*, 2nd edn, Butterworth, London.

## BIBLIOGRAPHY

---

- Vasconcelos, N. & Kunt, M. (2001), Content-based retrieval from image databases: Current solutions and future directions, in 'IEEE Proc. of Int. Conf. on Image Processing', Vol. 3, pp. 6–9.
- Vasconcelos, N. & Lippman, A. (2000), Bayesian relevance feedback for content-based image retrieval, in 'IEEE Proc. of Workshop on Content-based Access of Image and Video Libraries', pp. 63–67.
- von Ahn, L. & Dabbish, L. (2004), Labeling images with a computer game, in 'CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems', ACM Press, New York, NY, USA, pp. 319–326.
- Wang, J. Z., Li, J. & Wiederhold, G. (2001), 'SIMPLIcity: Semantics-sensitive integrated matching for picture Llibraries', *IEEE Trans. Pattern Analysis and Machine Intelligence* **23**(9), 947–963.
- Wang, X.-J., Ma, W.-Y., Zhang, L. & Li, X. (2005), Multi-graph enabled active learning for multimodal web image retrieval, in 'Proc. of the 7th ACM SIGMM Int. Workshop on Multimedia Information Retrieval (MIR'05)', ACM Press, New York, NY, USA, pp. 65–72.
- Wenyin, L., Chen, Z., Lin, F., Zhan, H. & Ma, W.-Y. (2003), 'Ubiquitous media agents: a framework for managing personally accumulated multimedia files', *Multimedia Systems* **9**, 144–156.
- White, R. W. (2004), Implicit Feedback for Interactive Information Retrieval, PhD thesis, Department of Computing Science, University of Glasgow, Glasgow, UK.
- Wood, M. E. J., Thomas, B. T. & Campbell, N. W. (1998), Iterative refinement by relevance feedback in content-based digital image retrieval, in 'ACM Multimedia 98', ACM Press, Bristol, UK, pp. 13–20.
- Yang, C. C. (2004), 'Content-Based Image Retrieval: A Comparison between Query by Example and Image Browsing Map Approaches', *Journal of Information Science* **30**(3), 254–267.
- Yang, J., Li, Q. & Zhuang, Y. (2004), 'Towards data-adaptive and user-adaptive image retrieval by peer indexing', *International Journal of Computer Vision, Special Issue on Content-Based Image Retrieval* **56**(1), 47–63.
- Yavlinsky, A., Schofield, E. & Rüger, S. (2005), Automated image annotation using global features and robust nonparametric density estimation, in 'Proc. of the Int. Conf. on Image and Video Retrieval', LNCS 3568, pp. 507–517.
- Yin, X., Li, M., Zhang, L. & Zhang, H. (2003), Semantic image clustering using relevance feedback, in 'Proc. of the Int. Symposium on Circuits and Systems (ISCAS '03)', Vol. 2, IEEE Computer Society Press, pp. 904–907.
- Zhao, R. & Grosky, W. I. (2000), From features to semantics: Some preliminary results, in 'IEEE Proc. of Int. Conf. on Multimedia and EXPO (II)', Vol. 2, IEEE Computer Society Press, pp. 679–682.
- Zhao, R. & Grosky, W. I. (2001), Bridging the semantic gap in image retrieval, in T. Shih, ed., 'Distributed Multimedia Databases: Techniques and Applications', Idea Group Publishing, Hershey, Pennsylvania, pp. 14–36.
- Zhou, X. S. & Huang, T. (2003), 'Relevance feedback in image retrieval: A comprehensive review', *ACM Multimedia Systems Journal, Special Issue on CBIR* **8**(6), 536–544.
- Zhou, X. S. & Huang, T. S. (2002), 'Unifying keywords and visual contents in image retrieval', *IEEE Multimedia* **9**(2), 23–33.

---

## QUANTITATIVE EVALUATION OF THE OSTENSIVE MODEL

---

In Chapter 3, we have discussed and evaluated the Ostensive Browser from the user’s perspective. While the usability of a system depends largely on its interface, the performance of the underlying algorithms cannot be neglected for judging a system’s overall effectiveness. The retrieval performance of the OM-based query learning scheme is better judged in comparison to other relevance feedback techniques in a more objective quantitative evaluation. A simulated evaluation we have conducted to this end showed that performance can be increased in the ostensive browsing scenario. This appendix will present these preliminary results.

### A.1 Introduction

We have set up a simulated comparative evaluation to measure the retrieval performance of the Ostensive Model (OM). In this experiment, we are interested in how well the OM performs in terms of the number of images found in a category search. The number of relevant images retrieved is an indication of the overall level of recall, ie the number of relevant images retrieved divided by the total number of relevant images for a category. The number of iterations until a session converges (when the system is not able to return any new relevant images) gives an indication of user effort to retrieve all these images. In the ideal case, while maximising recall the iteration number should be low, meaning that the retrieval system succeeds in returning all relevant images early in the session.

The query learning scheme proposed by Rui & Huang (2000) serves as baseline. Rui & Huang’s scheme is essentially a relevance feedback technique, which represents a query as the average over all positive examples in addition to a feature re-weighting scheme (cf Section 5.1). To make the comparison fair, the same query learning and feature re-weighting is employed for the OM. The idea behind the learning scheme is still the same as the one proposed in Section 3.1.4. Similar to the query representation in Equation 3.2, the new query is computed as the *weighted* (with the ostensive relevance weights) average of the path images in the OM. Instead of using the Dempster-Shafer theory, however, the visual features are linearly combined using the feature weights computed according to Rui & Huang’s scheme. The details of the feature representation and re-weighting scheme can be found in (Rui & Huang 2000).

The evaluation is performed on a subset of the Corel dataset (Photo CD 4), containing 24 categories of 100 images each. We only use content-based features for this evaluation. The 6 low-level colour, texture and shape features implemented are (feature dimension): Average RGB (3), Colour Moments (9) (Stricker & Orengo 1995); Co-occurrence (20), Autocorrelation (25) and Edge Frequency (25) (Sonka et al. 1998); Invariant Moments (7) (Hu 1962) (cf Appendix C).

## A.2 The Simulation Setup

We simulated user interaction to find as many relevant images from a given category as possible. An image is considered relevant if it belongs to the same category as the initial query image. The simulation for the baseline system is as follows. Starting with one image from the given category, the system returns the 20 most similar images (images already in the query are not returned again so as to maximise the system’s ability of collecting a large number of distinct relevant images). From this set, the simulated user selects at most  $n$  relevant images to add to the query and the system recomputes the top 20 images. The process is iterated until there are no more relevant images in the returned set. This simulation resembles the traditional relevance feedback process. We report results from two variations of  $n$ : a “realistic” scenario where  $n=3$ , referred to as  $RFS_3$ , and a “greedy” scenario,  $RFS_g$ , where  $n=20$ .

The simulation setup for the OM is slightly different. Starting with one query image as the root image, the system returns the top  $k$  ( $6 \leq k \leq 12$ ) candidates. The user selects the first relevant image from the returned set and adds it to the path. The process is repeated until there are no more relevant images in the latest candidates. At this point, the user backs up along the path and continues with the closest image, which has unprocessed relevant candidates. This corresponds to a depth-first traversal of the ostensive tree. The session continues until there are no more new relevant images in the ostensive tree. There are two assumptions being made about the user’s actions. First, the simulated user only selects relevant images and second, once a relevant image has been selected in one path, it will not be pursued in a different branch again. (There can be duplicate images in different branches of the ostensive tree, even though the candidates will not contain any image already on the current path.) The simulation scheme will be referred to as  $OMS_k$ .

## A.3 Results Analysis

We have chosen five categories that contain visually similar images. Other categories are very difficult for CBIR, so that the majority of queries would not return any relevant images in the first iteration. The selected categories are: ‘lions’, ‘elephants’, ‘horses’, ‘mountains’, ‘orchids’. Every image in each category serves as the first query image for both schemes, RFS and OMS, resulting in a total of 100 queries per category.

Figure A.1(a) shows the average number of (unique) relevant images found for all five categories in  $RFS_3$ ,  $RFS_g$  as well as  $OMS_k$  for various candidate sizes  $k$ . The performance of  $RFS_3$  and  $RFS_g$  is very similar, with  $RFS_3$  results being slightly better. It can be seen that for  $k = 9$ ,  $OMS_9$  succeeds in finding approximately the same number of relevant images as both RF scenar-



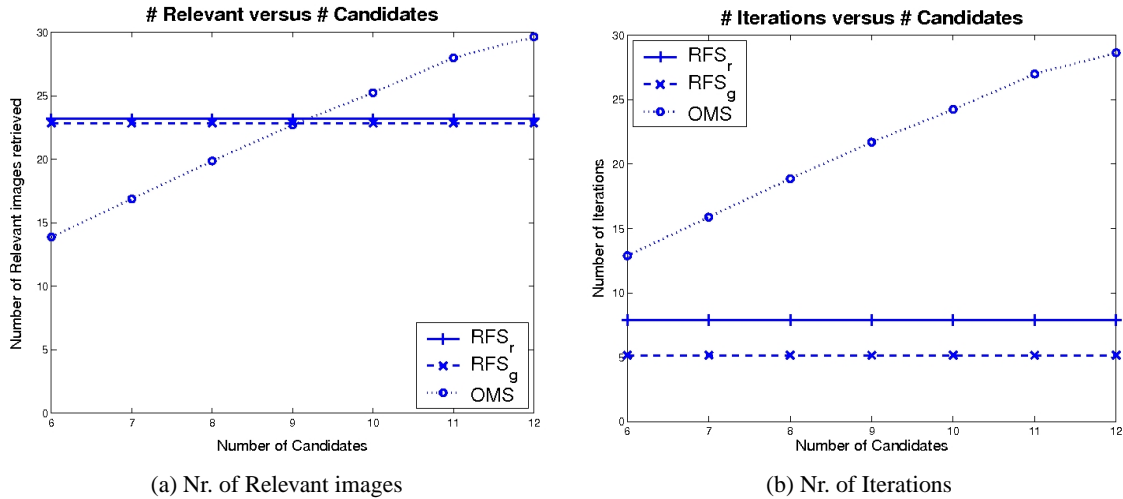


Figure A.1: Nr. of relevant images and nr. of iterations vs candidate size

	RFS <sub>3</sub>	RFS <sub>g</sub>	OMS <sub>6</sub>	OM <sub>7</sub>	OM <sub>8</sub>	OM <sub>9</sub>	OM <sub>10</sub>	OM <sub>11</sub>	OM <sub>12</sub>
R	23.19	22.86	13.88	16.88	19.87	22.71	25.24	27.99	29.65
I	7.89	5.15	12.88	15.88	18.87	21.71	24.24	26.99	28.65
R/I	2.94	4.44	1.08	1.06	1.05	1.05	1.04	1.04	1.04

Table A.1: Average results for nr. of relevant images retrieved (R), nr. of iterations (I) and nr. of relevant per iteration (R/I)

ios. Increasing  $k$  results in  $OMS_k$  outperforming RFS in terms of the level of recall. (An example of how to display a larger number of candidates in the interface is displayed in Figure A.2.) However, as the number of candidates increases and more relevant images can be found, the number of iterations until convergence increases with it. (Note, that in the OM simulation the iteration number is always one more than the number of unique relevant images found, since each relevant image will be selected exactly once, plus one for the final iteration, which fails to return any relevant images). The iteration number of  $OMS_6$  is already higher than the baseline, as can be seen in Figure A.1(b). Table A.1 summarises these results. It can be seen that RFS<sub>3</sub> converges after approximately 8 iterations, while the greedy strategy only needs 5 iterations to achieve a similar level of recall. Although RFS apparently converges faster than OMS, this does not necessarily mean that RFS requires less user effort (in terms of mouse clicks for example). Keeping in mind that the number of relevant images for feedback in RFS is  $n$ , ie the user has to click up to  $n+1$  times ( $n$  for feedback, 1 for new search) whereas in the OM scenario only 1 click is required to initiate a new iteration. The average click count for RFS <sub>$n$</sub>  in the simulation was 28, which interestingly is very similar to the iteration number of  $OMS_{12}$ .

## A.4 Limitations of the Study

The evaluation presented is merely a preliminary study, and a lot of additional factors besides the recommendation size can be considered, such as for example the choice of ostensive relevance profile (cf Section 3.1.4). Also our assumptions about the user's actions might not necessarily be

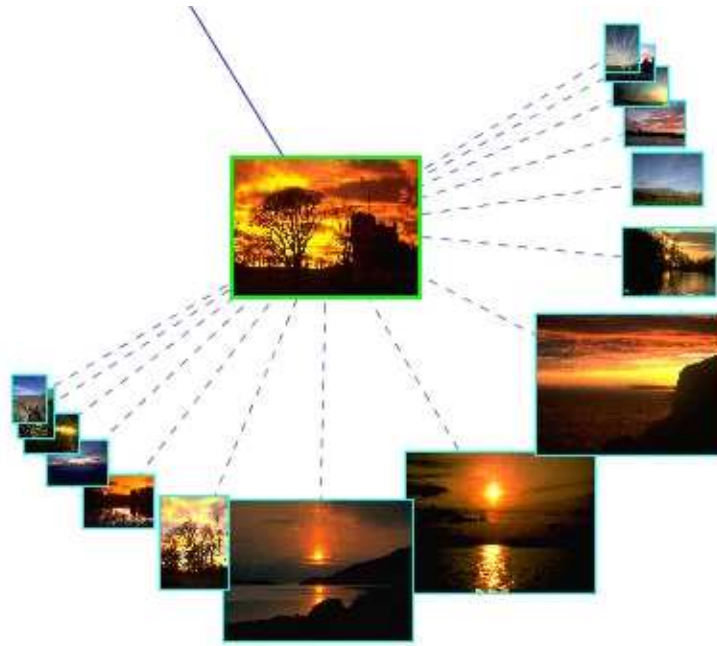


Figure A.2: Example of fisheye display for candidate size of 15

realistic. Does a user always select relevant information? Does a user rather proceed deep along a path (depth-first traversal as modelled here) or rather select all relevant options first (breadth-first)? In a future study, a proper user model should be constructed or even better, real users should conduct the searches.

Another favourable point is that the interaction possibilities in an Ostensive Browser allow for more flexibility than in a traditional relevance feedback system. In an RF scenario, all images selected relevant will be accumulated and added to the query. In contrast, the Ostensive Browser gives the user more control over which images are relevant for a query by moving back and forth in one path and branching off into different directions. The effect of the selection strategy is a very interesting point to consider. In a future evaluation we could compare various user models in the simulation.

## A.5 Conclusions

In Chapter 3, we have developed and described an adaptive retrieval approach towards CBIR based on the concept of Ostension. The underlying idea is to mine and interpret the information from the user's interaction in order to understand the user's needs. The system's interpretation is used for suggesting new images to the user. A user-centred, work-task oriented evaluation demonstrated the value of our technique by comparing it to a traditional CBIR interface. In addition, the results presented in this appendix showed that the OM-based query learning strategy showed favourable retrieval performance in comparison to a standard relevance feedback technique in a simulated quantitative evaluation.

---

## ARCHITECTURE AND IMPLEMENTATION OF *EGO*

---

The system is implemented purely in Java, and as such is platform independent. It has been tested on Microsoft Windows and Linux. To run the system, we recommend a machine with at least 512MB of RAM and a processor of 2.0 GHz or above. A screen resolution of at least 1024 x 768 is recommended, although the image icon sizes in each panel can be adjusted in a properties file to accommodate smaller screens.

The system was built from scratch over a period of three years and consists of more than 500 Java classes (more than half of them implement the interface). The visual features implemented in *EGO* are taken from the *Discovir* project.<sup>2-1</sup> The workspace is based on JGraph<sup>2-2</sup>, a powerful and standards-compliant open source graph component available for Java. JGraph provides an implementation of standard graph nodes, zooming, layout algorithms and much more. The system is organised into four main packages:

### Server-side packages

- `ego.data` for the data representation including classes for the available document types, a class representing a group, a class representing a collection, etc.
- `ego.feature` for the visual feature extractors
- `ego.ir` for all IR related classes, most notably the `DatabaseManager` that manages the various indices, and the `RetrievalEngine` that communicates with the `DatabaseManager` and several `Query` classes for the various query types

### Client-side packages

- `ego.ui` for all interface-related objects

The communication between client and server takes place exclusively through a communication manager class (`ego.CommunicationManager`). In the future, we would like to completely separate the client and the server side to allow them to be run on different machines. Using a centralised

---

<sup>2-1</sup><http://www.cse.cuhk.edu.hk/~miplab/discovir/>

<sup>2-2</sup><http://www.jgraph.com>

server would make collaborative usage possible and could possibly speed up retrieval time with a dedicated server and a fast network.

---

## IMPLEMENTED IMAGE FEATURES

---

In EGO, the study of visual features has not been one of the main objectives. The visual features implemented in *EGO* are taken from the *Discovir* project available at <http://www.cse.cuhk.edu.hk/~miplab/discovir/>. These features were chosen to construct a rapid initial prototype implemented purely in Java because they were readily available in Java. The retrieval system, however, does not rely on this specific set of features. In fact, future improvements should incorporate the descriptors proposed for the MPEG-7 standard<sup>3-1</sup>, since those features have proven successful for a variety of retrieval tasks. This would also allow better comparison of retrieval techniques.

### C.1 Overview of Implemented Features

Name	Dim
<b>Colour</b>	
Average RGB	3
Colour Moments	9
<b>Texture</b>	
Edge Frequency	25
Cooccurrence Matrix	20
Auto-correlation	25
<b>Shape</b>	
Invariant Moments	7

### C.2 Colour Features

#### Global Average RGB

**Category** Colour feature extraction

---

<sup>3-1</sup><http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

**Abstract** The Average RGB descriptor consists of the average values in the R, G and B channel of the pixels in an image.

<b>Notation</b>	$I$	an image
	$w$	width of image $I$
	$h$	height of image $I$
	$I(x, y)$	the pixel of image $I$ at row $y$ , column $x$
	$R(p), G(p), B(p)$	the red, green and blue colour component of pixel $p$
	$ra, ga, ba$	the average red, green and blue component of image $I_a$

### Description

- The three equations to compute the average R, G, and B component of an image  $I$  are:

$$r = \frac{\sum_{x=1, y=1}^{x=w, y=h} R(I(x, y))}{w \times h}$$

$$g = \frac{\sum_{x=1, y=1}^{x=w, y=h} G(I(x, y))}{w \times h}$$

$$b = \frac{\sum_{x=1, y=1}^{x=w, y=h} B(I(x, y))}{w \times h}$$

- Feature Dimension: 3

### Colour Moment

**Category** Colour feature extraction

**Abstract** Colour moments represent the dominant colour features instead of storing the complete or quantised (as in histograms) colour distribution. For each image in the database, the first three moments of each colour channel are stored. For an HSV image this would result in a vector containing nine values per image. The three moments typically used for image retrieval are: average, variance and skewness.

<b>Notation</b>	$p_{ij}$	The value of the $i$ -th colour channel (H,S, or V) at the $j$ -th image pixel
	$N$	Number of image pixels of an image
	$r$	Number of colour channels
	$w$	User specified weights

### Description

- Average Moment:

$$E_i = \frac{1}{N} \sum_{j=1}^N p_{ij}$$

defines the average moment of a specified image at colour channel  $i$ .

- Variance:

$$\sigma_i = \left( \frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^2 \right)^{\frac{1}{2}}$$

defines the variance of a specified image at colour channel  $i$

- Skewness

$$s_i = \left( \frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^3 \right)^{\frac{1}{3}}$$

defines the skewness of a specified image at colour channel  $i$ .

- Feature Dimension: 9 (3\*3)

**Authors/References** (Stricker & Orengo 1995)

### C.3 Texture Features

#### Edge Frequency

**Category** Texture Feature Extraction

**Abstract** Coarse textures are represented by a large number of neighbouring pixels with the same grey level, whereas a small number represents a fine texture. A primitive is a continuous set of pixels in the same direction that have the same grey level. Each primitive is defined by its grey level, length and direction.

<b>Notation</b>	$B(a, r)$	the number of primitives
	$r$	the number of primitives of all directions having length
	$a$	grey level
	$M, N$	image dimensions
	$L$	number of grey levels
	$Nr$	the maximum primitive length in the images
	$K$	the total number of runs

#### Description

- Short primitive emphasis:

$$\frac{1}{K} \sum_{a=1}^L \sum_{r=1}^{Nr} \frac{B(a, r)}{r^2}$$

- Long primitive emphasis

$$\frac{1}{K} \sum_{a=1}^L \sum_{r=1}^{Nr} B(a, r) r^2$$

- Grey level uniformity:

$$\frac{1}{K} \sum_{a=1}^L \sum_{r=1}^{N_r} [B(a, r)r^2]^2$$

- Primitive length uniformity:

$$\frac{1}{K} \sum_{a=1}^L \sum_{r=1}^{N_r} [B(a, r)]^2$$

- Primitive percentage:

$$\frac{K}{\sum_{a=1}^L \sum_{r=1}^{N_r} rB(a, r)} = \frac{K}{MN}$$

- Feature Dimension: 25

**Authors/References** (Sonka et al. 1998)

### Co-occurrence Matrices

**Category** Texture Feature Extraction

**Abstract** Co-occurrence matrix is a statistical method using second order statistics to model the relationships between pixels within the region by constructing Spatial Grey Level Dependency (SGLD) matrices. The Grey-level co-occurrence matrix is the two dimensional matrix of joint probabilities  $P_{d,r}(i, j)$  between pairs of pixels, separated by a distance,  $d$ , in a given direction,  $r$ . It is popular in texture description and based on the repeated occurrence of some grey level configuration in the texture; this configuration varies rapidly with distance in fine textures, slowly in coarse textures.

If the texture is coarse and distance  $d$  is small compared to the size of the texture elements, the pairs of points at distance  $d$  should have similar grey levels. Conversely, for a fine texture, if distance  $d$  is comparable to the texture size, then the grey levels of points separated by distance  $d$  should often be quite different, so that the values in the SGLD matrix should be spread out relatively uniformly.

Hence, a good way to analyse texture coarseness would be, for various values of distance  $d$ , some measure of scatter of the SGLD matrix around the main diagonal. Similarly, if the texture has some direction, ie is coarser in one direction than another, then the degree of spread of the values about the main diagonal in the SGLD matrix should vary with the direction. Thus texture directionality can be analysed by comparing spread measures of SGLD matrices constructed at various distances  $d$ . From SGLD matrices, a variety of features may be extracted.

<b>Notation</b>	$P_{d,r}(i, j)$	joint probabilities between pairs of pixels in a given direction
	$d$	distance between pairs of pixels in a given direction
	$r$	a given direction

**Description** Finding texture features from grey-level co-occurrence matrix for texture classification are based on these criteria:



- Energy

$$\sum_i \sum_j P_{d,r}^2(i, j)$$

- Entropy

$$\sum_i \sum_j P_{d,r}(i, j) \log P_{d,r}(i, j)$$

- Contrast (typically  $k = 2, \lambda = 1$ )

$$\sum_i \sum_j |i - j|^k P_{d,r}^\lambda(i, j)$$

- Homogeneity

$$\sum_i \sum_j \frac{P_{d,r}(i, j)}{|i - j|}$$

- Feature Dimension: 20

**Authors/References** (Sonka et al. 1998)

### Auto-correlation

**Category** Texture Feature Extraction

**Abstract** Autocorrelation measures the coarseness of an image by evaluating the linear spatial relationships between texture primitives. Large primitives give rise to coarse texture (eg rock surface) and small primitives give rise to fine texture (eg silk surface).

If the primitives are large, the autocorrelation function decreases slowly with increasing distance whereas it decreases rapidly if texture consists of small primitives. However, if the primitives are periodic, then the autocorrelation function increases and decreases periodically with distance.

<b>Notation</b>	$f(i, j)$	the grey level value of the pixel in row $i$ and column $j$
	$M, N$	image dimensions
	$p, q$	positional difference in $i, j$ direction

**Description** A set of autocorrelation coefficients is derived from the following autocorrelation function and used as texture features:

- Autocorrelation function

$$C_f f(p, q) = \frac{MN}{(M-p)(N-q)} \frac{\sum_{i=1}^{M-p} \sum_{j=1}^{N-q} f(i, j) f(i+p, j+q)}{\sum_{i=1}^M \sum_{j=1}^N f^2(i, j)}$$

Usually,  $(p, q)$  are varied from  $(0, 0)$  to  $(8, 8)$  in a step of two, which results in a total of 25 features.

- Feature Dimension: 25

**Authors/References** (Sonka et al. 1998)

## C.4 Shape Features

### Invariant Moments

We use the seven invariant shape moments as discussed by Hu (1962), which include six absolute orthogonal invariants and one skew orthogonal invariant computed from the second and third order moments.

- Feature Dimension: 7

**Authors/References** (Hu 1962)

# APPENDIX D

---

## EXPERIMENTAL DOCUMENTS

---

This appendix contains the experimental documents used for Experiment 1 and Experiment 2 described in Chapter 6. I am grateful to Dr. Ryen White for the templates (White 2004).

### D.1 Experiment 1

#### D.1.1 Tasks

**TASK DESCRIPTION**



**Project:** A Comparative Study of Two Interfaces for Image Searching  
**Researcher:** Jana Urban

**Task Scenario**

Imagine you are a designer with responsibility for the design of leaflets on various subjects for the Wildlife Conservation (WLC). The leaflets are intended to raise awareness among the general public for endangered species and the preservation of their habitats. These leaflets are, generally A4/3 i.e. an A4 sheet folded into three consisting of a body of text interspersed with up to 4-5 images selected on the basis of their appropriateness to the use to which the leaflets are put.

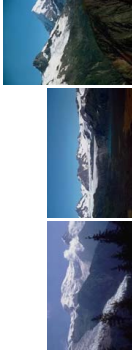


**Category Search Task (Task A/B):**

You will be given a leaflet topic from the list overleaf. Your task involves searching for as many images as you are able to find on the given topic, suitable for presentation in the leaflet. In order to perform this task, you have the opportunity to make use of an image retrieval system, the operation of which will be demonstrated to you. You have **10 minutes** to attempt this task.




**Design Task (Task C):**

This time, you're asked to select images for a leaflet for WLC presenting the organisation and a selection of their activities (some of WLC's activities are listed overleaf but feel free to consider other topics they might be involved in). Your task is to search for suitable images and then make a pre-selection of 3-5 images for the leaflet. You have **20 minutes** to attempt this task.

**Category A**

<b>1</b>	<p><b>Mountainous Landscapes</b>                      Leaflet Theme: Ecoregions—The Alps</p> 
<b>2</b>	<p><b>Elephants</b>                      Leaflet Theme: Poaching for Elephant Ivory                      Puts Species at Risk</p> 
<b>3</b>	<p><b>Tigers</b>                      Leaflet Theme: Endangered Species—Tigers</p> 

**Category B**

<b>4</b>	<p><b>Animals in the Snow</b>                      Leaflet Theme: Climate Change—Animals in Danger</p> 
<b>5</b>	<p><b>African Wildlife</b>                      Leaflet Theme: Ecoregions—Africa</p> 
<b>6</b>	<p><b>Underwater World</b>                      Leaflet Theme: Marine Life and Corals</p> 

D.1.2 Information Sheet and Consent Form

**INFORMATION SHEET**

**Project:** A Comparative Study of Two Interfaces for Image Searching

**Researcher:** Jana Urban



UNIVERSITY  
of  
GLASGOW

You are invited to take part in a research study. Before you decide to do so, it is important for you to understand why the research is being done and what it will involve. Please take time to read the following information carefully. Ask me if anything is not clear or if you would like more information.

The aim of this experiment is to investigate the relative effectiveness of two different image search interfaces. We cannot determine the value of search systems unless we ask those people who are likely to be using them, which is why we need to run experiments like these. Please remember that it is the interfaces, not you, that are being evaluated.

It is up to you to decide whether or not to take part. If you decide to take part you will be given this information sheet to keep and asked to sign a consent form. You are free to withdraw at any time without giving a reason. You also have the right to withdraw retrospectively any consent given, and to require that any data gathered on you be destroyed.

The experiment will last around two hours and you will receive a reward of £10 upon completion. You will be given a chance to learn how to use the two interfaces before we begin. At this time, you will also be asked to complete an introductory questionnaire. You will perform three tasks in total. The first part of the experiment involves two tasks, one with each interface. Each task should take about 10 minutes to complete. All of your interactions (e.g. mouse movements and clicks and key presses) will also be logged. You are encouraged to comment on each interface as you use it, which I will take notes on. Please ask questions if you need to and please let me know when you are finished with the task. After each task you are asked to fill in a questionnaire about your experience during the search. After completing both tasks in the first set of the experiment, you will be asked some questions about the tasks and systems. Finally, you will be given a third task to be performed on one of the systems, which should take about 20 minutes to complete. Again, I will ask you to fill in a short questionnaire on your experience. If you opt out after the first part of the experiment, you will still be rewarded £7 for your effort.

All information collected about you during the course of this study will be kept strictly confidential. You will be identified by an ID number and all information about you will have your name and contact details removed so that you cannot be recognised from it. Data will be stored for analysis, and then destroyed.

The results of this study will be used for my PHD research. The results are likely to be published in late 2004. You can request a summary of the results in the consent form. You will not be identified in any report or publication that arises from this work.

This study is being funded by the Research Student Committee at the Department of Computing Science, University of Glasgow. This project has been reviewed by the Faculty of Information and Mathematical Sciences Ethics Committee.

For further information about this study please contact:

Jana Urban  
Department of Computing Science, University of Glasgow  
17 Lilybank Gardens  
Glasgow, G12 8RZ  
Email: [jana@cdcs.gha.ac.uk](mailto:jana@cdcs.gha.ac.uk)  
Tel.: 0141 330 5006



**CONSENT FORM**

**Project:** A Comparative Study of Two Interfaces for Image Searching

**Researcher:** Jana Urban



UNIVERSITY  
of  
GLASGOW

Please tick box

1. I confirm I have read and understand the information sheet for the above study and have had the opportunity to ask questions.
2. I understand that my permission is voluntary and that I am free to withdraw at any time, without giving any reason, without my legal rights being affected.
3. I agree to take part in the above study.
4. I would like to receive a summary sheet of the experimental findings

If you wish a summary, please leave an email address \_\_\_\_\_

Name of Participant \_\_\_\_\_ Date \_\_\_\_\_ Signature \_\_\_\_\_

Researcher \_\_\_\_\_ Date \_\_\_\_\_ Signature \_\_\_\_\_

**D.1.3 Questionnaires**

**Entry Questionnaire**

**Post-Search Questionnaire for WS**

**CS Part of Post-Search Questionnaire**

**Post-Design-Search Questionnaire for WS**

**Exit Questionnaire**

**ENTRY QUESTIONNAIRE**

This questionnaire will provide us with background information that will help us analyse the answers you give in later stages of this experiment. You are not obliged to answer a question, if you feel it is too personal.

UserID:

Please place a TICK  in the square that best matches your opinion.

**Part 1: PERSONAL DETAILS**

This information is kept completely confidential and no information is stored on computer media that could identify you as a person.



**UNIVERSITY  
of  
GLASGOW**

1. Please provide your AGE:

**2. Please indicate your GENDER:**

Male.....

Female.....

**3. Please indicate the HAND you use to CONTROL the MOUSE:**

Right.....

Left.....

**4. Are you COLOUR BLIND?**

No.....

Yes.....

**5. Please provide your current OCCUPATION:**

YEAR:

**Part 2: SEARCH EXPERIENCE**

**Experience with Images**

7. How often do you deal with photographs or images in your work, study or spare time?

Never	<input type="checkbox"/>	Once or twice a year	<input type="checkbox"/>	Once or twice a month	<input type="checkbox"/>	Once or twice a week	<input type="checkbox"/>	Once or twice a day	<input type="checkbox"/>	More often	<input type="checkbox"/>
-------	--------------------------	----------------------	--------------------------	-----------------------	--------------------------	----------------------	--------------------------	---------------------	--------------------------	------------	--------------------------

**8. How often do you take photographs in your work, study or spare time?**

Never	<input type="checkbox"/>	Once or twice a year	<input type="checkbox"/>	Once or twice a month	<input type="checkbox"/>	Once or twice a week	<input type="checkbox"/>	Once or twice a day	<input type="checkbox"/>	More often	<input type="checkbox"/>
-------	--------------------------	----------------------	--------------------------	-----------------------	--------------------------	----------------------	--------------------------	---------------------	--------------------------	------------	--------------------------

**9. How often do you carry out image searches at home or work?**

Never	<input type="checkbox"/>	Once or twice a year	<input type="checkbox"/>	Once or twice a month	<input type="checkbox"/>	Once or twice a week	<input type="checkbox"/>	Once or twice a day	<input type="checkbox"/>	More often	<input type="checkbox"/>
-------	--------------------------	----------------------	--------------------------	-----------------------	--------------------------	----------------------	--------------------------	---------------------	--------------------------	------------	--------------------------

**Image Search Experience**

10. Please indicate which online search services you use to search for IMAGES (mark AS MANY as apply)

Google (<http://www.google.com>) .....  1

Yahoo (<http://www.yahoo.com>) .....  2

Altavista (<http://www.altavista.com>) .....  3

AlltheWeb (<http://www.alltheweb.com>) .....  4

Others (please specify) .....  5

11. Using the image search services you chose in question 10 is GENERALLY:

easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	difficult	<input type="checkbox"/>	N/A	<input type="checkbox"/>
stressful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	relaxing	<input type="checkbox"/>		
simple	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	complex	<input type="checkbox"/>		
satisfying	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	frustrating	<input type="checkbox"/>		

12. You find what you are searching for on any kind of image search service...

Never	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Always	<input type="checkbox"/>	N/A	<input type="checkbox"/>
	1	2	3	4	5				

13. Have you ever used STOCK PHOTOGRAPHY services?

Yes.....  1 No.....  2

If **yes**, please indicate which services you have used (mark AS MANY as apply):

Corel Stock Images .....  1

Corbis .....  2

Getty Images .....  3

Others (please specify) .....  4

14. Using the stock photography search services you chose in question 13 is GENERALLY:

easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	difficult	<input type="checkbox"/>	N/A	<input type="checkbox"/>
stressful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	relaxing	<input type="checkbox"/>		
simple	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	complex	<input type="checkbox"/>		
satisfying	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	frustrating	<input type="checkbox"/>		

15. You find what you are searching for on any kind of stock photography service....

Never	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Always	<input type="checkbox"/>	N/A	<input type="checkbox"/>
	1	2	3	4	5				

16. Please indicate which systems you use to MANAGE your images (mark AS MANY as apply)

None (I just create directories and files on my computer) .....  1

Adobe Album.....  2

Picasa (Google).....  3

iView Multimedia (Mac).....  4

ACDSee.....  5

Others (please specify) .....  6

17. Using the image management tools you chose in question 16 is GENERALLY:

easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	difficult	<input type="checkbox"/>	N/A	<input type="checkbox"/>
stressful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	relaxing	<input type="checkbox"/>		
simple	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	complex	<input type="checkbox"/>		
satisfying	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	frustrating	<input type="checkbox"/>		

18. It is easy to find a particular image that you have saved previously on your computer...

Never	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Always	<input type="checkbox"/>	N/A	<input type="checkbox"/>
	1	2	3	4	5				

19. What do you expect from an image search service?

20. Explain how an image search and management tool could be helpful to you.

20. What sort of features would you expect in such a tool?

## POST-SEARCH QUESTIONNAIRE

To evaluate the system you have just used, we now ask you to answer some questions about it. Take into account that we are interested in knowing your opinion: answer questions freely, and consider there are no right or wrong answers.

Please remember that we are evaluating the system you have just used and not you.

User ID:  System:  Task:  Order:

Please place a TICK  in the square that best matches your opinion. Please answer all questions.

### Part 1: TASK

In this section we ask about the search task you have just attempted.

1.1 The task we asked you to perform was:

unclear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	clear
simple	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	complex
unfamiliar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	familiar

1.2 The search I have just performed was:

stressful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	relaxing
interesting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	boring
firing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	restful
easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	difficult

1.3 I had enough time to do an effective search.

Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Disagree
	5	4	3	2	1					

1.4 I believe I have succeeded in my performance of the task.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
	1	2	3	4	5					

1.5 Do you have any further comments about the task you have just attempted?

### Part 2: RETRIEVED IMAGES

In this section we ask you about the images you found/selected.

2.1 The images I have received through the searches are:

relevant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	not relevant
inappropriate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	appropriate
complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	incomplete

2.2 I had an idea of which kind of images fit in this category before starting the search.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
	1	2	3	4	5					

2.3 During the search I have discovered more aspects of the category than initially anticipated.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
	1	2	3	4	5					

2.4 I am satisfied with my search results.

Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Disagree
	5	4	3	2	1					

### Part 3: SYSTEM

3.1 Overall reaction to the system:

terrible	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	wonderful
satisfying	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	frustrating
cull	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	stimulating
easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	difficult
rigid	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	flexible
efficient	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	inefficient
unreliable	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	reliable
novel	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	standard
slow	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	fast
effective	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ineffective



**Part 4: SYSTEM SPECIFIC FEATURES (Workspace System)**

**Search Facilities**

In the Workspace System you could search for images by composing your own query (query-by-example) or asking for recommendations for selected groups/images (recommendations). In the following we will ask you about each of these facilities in turn.

**Query Composition**

In the system you used you had the option of creating a query by selecting items. The following questions ask you about how useful you found this tool to pursue your task.

4.1 How you constructed queries was (i.e. selecting appropriate example images):

difficult	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	easy
effective	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ineffective
not useful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	useful

4.2 How you constructed queries made you feel:

comfortable	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	uncomfortable
not in control	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	in control

4.3 The result images returned by the system were:

irrelevant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	relevant
useful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	not useful
appropriate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	inappropriate

4.4 Do you have any further comments about the query composition?

---

**Groups/Recommendations**

In this section we would like to know how useful you found the groups and recommendations.

4.5 The grouping of images (creation of groups and adding images to existing groups) was:

difficult	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	easy
effective	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ineffective
not useful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	useful

3.2 When interacting with the system, I felt:

in control	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	not in control
uncomfortable	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	comfortable
confident	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	unconfident

3.3 How easy was it to LEARN TO USE the system?

Not at all

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Extremely
1	2	3	4	5		

3.4 How easy was it to USE the system?

Extremely

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Not at all
5	4	3	2	1		

3.5 The system helped me to explore the collection better.

Agree

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Disagree
5	4	3	2	1		

3.6 The system helped me to analyse the task better.

Disagree

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
1	2	3	4	5		

4.6 The grouping of images made you feel:

comfortable	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	uncomfortable
not in control	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	in control

4.7 I only put images on the workspace that were relevant to the task.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
	1	2	3	4	5			

4.8 The workspace helped me organise the images I found for the task.

Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Disagree
	5	4	3	2	1			

4.9 The images recommended by the system were:

irrelevant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	relevant
useful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	not useful
appropriate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	inappropriate

4.10 You accepted any recommended images because (mark AS MANY as apply):

they were relevant for the selected group(s) .....  1

they were relevant for other groups on the workspace .....  2

they represented new ideas (i.e. not part of your original request) .....  3

other (please specify) .....  4

4.11 I would trust the system to choose images for me.

Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Disagree
	5	4	3	2	1			

4.12 Do you have any further comments about the workspace or the images that were recommended?

**Part 5: FEATURES/DISPLAY**  
 Finally, we would like to know your opinion about the system's features and display of components.

5.1 Did you find the screen layout/arrangements of components:

unhelpful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	helpful
useful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	not useful
ineffective	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	effective
clear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	unclear

5.2 Which features of the system did you like most?

5.3 Which features of the system did you dislike most?

5.4 Which features would you like to add to the system?

5.5 Any other comments about the system's features or display?

## Part 4: SYSTEM SPECIFIC FEATURES (Checkbox System)

### Search Facilities

In the Checkbox System you could search for images by composing your own query (query-by-example) or selecting relevant images (relevance assessment). In the following we will ask you about each of these facilities in turn.

### Query Composition

In the system you used you had the option of creating a query by selecting items. The following questions ask you about how useful you found this tool to pursue your task.

4.1 How you constructed queries was (i.e. selecting appropriate example images):

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
difficult							
effective							
not useful							
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
easy							
ineffective							
useful							

4.2 How you constructed queries made you feel:

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
comfortable							
not in control							
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
uncomfortable							
in control							

4.3 The result images returned by the system were:

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
irrelevant							
useful							
appropriate							
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
relevant							
not useful							
inappropriate							

4.4 Do you have any further comments about the query composition?

### Relevance Assessment

In this section we would like to know more about your opinion of the relevance assessment facility.

4.5 How you conveyed relevance of images to the system (i.e. ticking boxes) was:

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
difficult							
effective							
not useful							
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
easy							
ineffective							
useful							

4.6 How you conveyed relevance to the system made you feel:

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
comfortable							
not in control							
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
uncomfortable							
in control							

4.7 The result images returned by the system were:

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
irrelevant							
useful							
appropriate							
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
relevant							
not useful							
inappropriate							

4.8 Selecting relevant images usually improved the search results.

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agree							
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Disagree							

4.9 Do you have any further comments about the relevance assessment?



**POST-DESIGN SEARCH QUESTIONNAIRE**

To evaluate the system you have just used, we now ask you to answer some questions about it. Take into account that we are interested in knowing your opinion: answer questions freely, and consider there are no right or wrong answers. This is the last questionnaire in this evaluation.

Please remember that we are evaluating the system you have just used and not you.

User ID:  System:  Condition:

Please place a TICK  in the square that best matches your opinion. Please answer all questions.

**Part 1: TASK**

In this section we ask about the search task you have just attempted.

1.1 The task we asked you to perform was:

unclear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	clear
simple	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	complex
unfamiliar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	familiar

1.2 The search I have just performed was:

stressful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	relaxing
interesting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	boring
firing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	restful
easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	difficult

1.3 I had enough time to do an effective search.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
1	2	3	4	5	6	7	8	9	10	

1.4 I believe I have succeeded in my performance of the task.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
1	2	3	4	5	6	7	8	9	10	

1.5 Do you have any further comments about the task you have just attempted?

**Part 2: RETRIEVED IMAGES**

In this section we ask you about the images you found/selected.

2.1 The images I have received through the searches are:

relevant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	irrelevant
useful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	not useful
inappropriate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	appropriate
complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	incomplete

2.2 I had an exact idea of the type of images I wanted before starting the search.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
1	2	3	4	5	6	7	8	9	10	

2.3 I believe that I have seen all possible images that satisfy my requirement.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
1	2	3	4	5	6	7	8	9	10	

2.4 I am satisfied with my search results.

Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Disagree
1	2	3	4	5	6	7	8	9	10	

**Part 3: SYSTEM and WORKSPACE**

3.1 The system helped me explore the collection better.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
1	2	3	4	5	6	7	8	9	10	

3.2 The system helped me to analyse the task better.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
1	2	3	4	5	6	7	8	9	10	

3.3 I have changed my initial idea of the type of images I wanted while using the system.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agree
1	2	3	4	5	6	7	8	9	10	

3.1.1 Do you have any further comments about the workspace?

**Part 4: FACILITIES**

For this task you could search for images by composing your own query (query-by-example), asking for recommendations for selected groups/images (recommendations), or locate already existing groups on the workspace (existing groups). In this section we ask you which of these facilities you found most useful for the task.

4.1 Which of the system's facilities did you find easier to use?

Query-by-example.....	<input type="checkbox"/>	1
Recommendations.....	<input type="checkbox"/>	2
Existing groups.....	<input type="checkbox"/>	3
No difference.....	<input type="checkbox"/>	4

4.2 Which of the system's facilities did you find more EFFECTIVE for the task you performed?

Query-by-example.....	<input type="checkbox"/>	1
Recommendations.....	<input type="checkbox"/>	2
Existing groups.....	<input type="checkbox"/>	3
No difference.....	<input type="checkbox"/>	4

4.3 Which of the system's facilities did you LIKE BEST overall?

Query-by-example.....	<input type="checkbox"/>	1
Recommendations.....	<input type="checkbox"/>	2
Existing groups.....	<input type="checkbox"/>	3
No difference.....	<input type="checkbox"/>	4

4.4 Do you have any other suggestions for the system?

Please take note of my email address and let me know if you have any further questions.

**Thank you for your help!**

3.4 The workspace helped me organise the images I found for the task.

Agree	<input type="checkbox"/>	5	<input type="checkbox"/>	4	<input type="checkbox"/>	3	<input type="checkbox"/>	2	<input type="checkbox"/>	1	Disagree
-------	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	----------

3.5 I only put images on the workspace that were relevant to the task.

Disagree	<input type="checkbox"/>	1	<input type="checkbox"/>	2	<input type="checkbox"/>	3	<input type="checkbox"/>	4	<input type="checkbox"/>	5	Agree
----------	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	-------

3.6 The organisation of relevant images into groups helped me express different aspects of the task.

Agree	<input type="checkbox"/>	5	<input type="checkbox"/>	4	<input type="checkbox"/>	3	<input type="checkbox"/>	2	<input type="checkbox"/>	1	Disagree
-------	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	----------

3.7 The existing groups on the workspace helped me for the task.

Disagree	<input type="checkbox"/>	1	<input type="checkbox"/>	2	<input type="checkbox"/>	3	<input type="checkbox"/>	4	<input type="checkbox"/>	5	Agree	N/A	<input type="checkbox"/>
----------	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	-------	-----	--------------------------

3.8 I believe someone else might benefit from my groupings for other tasks.

Agree	<input type="checkbox"/>	5	<input type="checkbox"/>	4	<input type="checkbox"/>	3	<input type="checkbox"/>	2	<input type="checkbox"/>	1	Disagree	N/A	<input type="checkbox"/>
-------	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	--------------------------	---	----------	-----	--------------------------

3.9 Which tools do you think are (would be) helpful in locating information on the workspace (mark AS MANY as apply):

manual zoom facility.....	<input type="checkbox"/>	1
bird's eye view.....	<input type="checkbox"/>	2
highlighting of same images on workspace.....	<input type="checkbox"/>	3
automatic search of specific groups on workspace.....	<input type="checkbox"/>	4
browsing groups on workspace by links between them.....	<input type="checkbox"/>	5
history of groups (e.g. last 10 groups you have 'touched').....	<input type="checkbox"/>	6
other (please specify).....	<input type="checkbox"/>	7

3.10 Explain why/how the tool selected in the previous question are (would be) helpful.

### EXIT QUESTIONNAIRE

The aim of this experiment was to investigate the relative effectiveness of two different image search interfaces. Please consider the entire search experience that you just had when you respond to the following questions.



**UNIVERSITY**  
*of*  
**GLASGOW**

User ID:

Please place a TICK  in the square that best matches your opinion. Please answer the questions as fully as you feel able to.

#### Part 1: TASK EXPERIENCE

1.1 To what extent did you understand the nature of the searching task?

Not at all	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Completely
	1	2	3	4	5		

1.2 To what extent did you find the task similar to other searching tasks you typically perform?

Completely	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Not at all
	5	4	3	2	1		

#### Part 2: SYSTEM EXPERIENCE

2.1 How different did you find the systems from one another?

Not at all	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Completely
	1	2	3	4	5		

2.2 Which of the systems did you find easier to LEARN TO USE?

Checkbox.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Workspace.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
No difference.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.3 Which of the systems did you find easier to USE?

Checkbox.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Workspace.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
No difference.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.4 Which of the systems did you find more EFFECTIVE for the tasks you performed?

Checkbox.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Workspace.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
No difference.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.5 Which of the systems did you LIKE BEST overall?

Checkbox.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Workspace.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
No difference.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.6 What did you LIKE about each of the systems?

CS:

WS:

2.7 What did you DISLIKE about each of the systems?

CS:

WS:

#### Part 3: COMMENTS

3. Do you have any further comments or questions about the systems or experiment (please continue on the back of this page)?

## D.2 Experiment 2

### D.2.1 Tasks

#### TASK D

##### D.1

##### Task Description

Look at the three images provided below. They all share a common theme. Your task is to find and select a fourth image complementing the set.



#### TASK D

##### D.2

##### Task Description

Look at the three images provided below. They all share a common theme. Your task is to find and select a fourth image complementing the set.



<p><b>TASK E</b></p> <p><b>E.1</b></p> <p><b>Task Description</b></p> <p>Imagine you are the web designer for an online travel agency called PerfectHoliday. In order to gain more customers, they have decided to hold a competition entitled "Win your dream holiday". They have provided you with the details of the competition (see below) and have asked you to select some images to illustrate the text.</p> <p>Your task is to find one main and two additional images that you would place on the webpage along with the competition details. The images should draw people's attention and spark their imagination.</p> <div data-bbox="927 1128 1374 1861" style="border: 1px solid black; padding: 5px;"> <p><b>Win your dream holiday!</b></p> <p>What if you could make you dream holiday become reality? Where would you go and what would you do? PerfectHoliday is giving you the chance to win that dream! We will be giving away £2000 to the lucky winner for the holiday of their dreams! What would you do with the money? Swim with the dolphins? Stay in a French castle or sail the Mediterranean on a luxurious sailboat? Do you imagine yourself white water rafting in the Alps? Or would a secluded beach with pearly white sands be for you? No matter what your dream holiday looks like, we will make your dreams come true.</p> <p>To enter this competition, simply send us a description of the perfect holiday illustrated with a picture before midnight on [...]</p> <p>So don't hesitate! Send your details to [...] and you could be packing your bags!</p> </div>	<p><b>TASK E</b></p> <p><b>E.2</b></p> <p><b>Task Description</b></p> <p>Imagine you are hired by a company called PEdu, who offer a large variety of interesting and challenging courses for adults. They are in process of publishing their new course catalogue for next year. In order to draw people's attention to particular courses, they would like to illustrate the course description with images.</p> <p>Your task is to find three images that fit the course description below. The images should show the diversity of the course's topics.</p> <div data-bbox="852 327 1433 1055" style="border: 1px solid black; padding: 5px;"> <p><b>Tropical Marine Ecology</b></p> <p>Tropical Marine Ecology serves as an intensive field-based introduction to the ecology of estuarine and marine environments.</p> <p>The primary goal of the course is to immerse you in field experiences that link with readings, lectures, discussions, labs, and discovery-oriented investigations of tropical environments. Several topics will be covered in depth. These include ecology and geology of these environments, marine ecology, coral reef ecology, intertidal zonation, grassbed ecology, taxonomy of vertebrates and invertebrates of coral reefs, lagoons, and tidal flats, statistical analyses of data, astronomy, and group projects concerning biological and physical analyses of select marine habitats.</p> <p><b>INTERESTED?</b> Contact [...] for more information. Places are strictly limited. Apply today to secure your place in this course!</p> <p>The images below should provide a hint, a mere glimpse, into the beauty and complexity of a wide variety of ecosystems. So, enjoy!</p> </div>
---	--



<p><b>TASK E</b></p> <p><b>E.3</b></p> <p><b>Task Description</b></p> <p>Imagine you are an employee of ADdictive – a company providing marketing services. You are working on a project to launch a new advertising campaign for one of your customers. The company in question is PowerHouse, who produce renewable energy. The following advertising slogan representing the company’s business has already been selected by your team:</p> <p><b>“PowerHouse - In tune with nature all around the world!”</b></p> <p>Your task is now to find suitable images to go along with this slogan.</p>	<p><b>TASK F</b></p> <p><b>F.1</b></p> <p><b>Task Description</b></p> <p>Imagine you want to take part in a photo competition, where you could win £100 for a picture that depicts the following theme:</p> <p><b>Dynamic</b></p> <p>In order to get ideas for the competition, you want to look for already existing photographs conveying the same theme. Your task is to select at least one image that represents the theme well.</p>
<p><b>TASK E</b></p> <p><b>E.4</b></p> <p><b>Task Description</b></p> <p>Imagine you are an employee of ADdictive – a company providing marketing services. You are working on a project to launch a new advertising campaign for one of your customers. The company in question is Flash, who manufactures sports clothing and equipment. The following advertising slogan representing the company’s business has already been selected by your team:</p> <p><b>“Flash - Unleash the animal inside!”</b></p> <p>Your task is now to find suitable images to go along with this slogan.</p>	<p><b>TASK F</b></p> <p><b>F.2</b></p> <p><b>Task Description</b></p> <p>Imagine you want to take part in a photo competition, where you could win £100 for a picture that depicts the following theme:</p> <p><b>Cute</b></p> <p>In order to get ideas for the competition, you want to look for already existing photographs conveying the same theme. Your task is to select at least one image that represents the theme well.</p>

**D.2.2 Questionnaires**

**Entry Questionnaire**

**Post-Search Questionnaire for WS**

**Post-Search Questionnaire for CS**

**Exit Questionnaire/Interview**

**ENTRY QUESTIONNAIRE**

This questionnaire will provide us with background information that will help us analyse the answers you give in later stages of this experiment. You are not obliged to answer a question, if you feel it is too personal.

User ID:

Please place a TICK  in the square that best matches your opinion.



**Part 1: PERSONAL DETAILS**

This information is kept completely confidential and no information is stored on computer media that could identify you as a person.

1. Please provide your AGE:

2. Please indicate your GENDER: Male  1 Female  2

3. Please provide your current OCCUPATION:  YEAR:

4. What is your FIELD of work or study?

**Part 2: SEARCH EXPERIENCE**

**Experience with Images**

Circle the number closest to your experience.

How often do you...	Never	Once or twice a year	Once or twice a month	Once or twice a week	Once or twice a day	More often
5. deal with photographs or images in your work, study or spare time?	1	2	3	4	5	6
6. take photographs in your work, study or spare time?	1	2	3	4	5	6
7. carry out image searches at home or work?	1	2	3	4	5	6

**Image Search Experience**

8. Please indicate which online search services you use to search for IMAGES (mark AS MANY as apply)

Google (<http://www.google.com>) .....  1

Yahoo (<http://www.yahoo.com>) .....  2

Alla Vista (<http://www.allavista.com>) .....  3

AlltheWeb (<http://www.alltheweb.com>) .....  4

Others (please specify).....



POST-SEARCH QUESTIONNAIRE

To evaluate the system you have just used, we now ask you to answer some questions about it. Take into account that we are interested in knowing your opinion: answer questions freely, and consider there are no right or wrong answers. Please remember that we are evaluating the system you have just used and not you.

User ID:  System:  WS  Task:  Order:

Please place a TICK  in the square that best matches your opinion. Please answer all questions.

Part 1: TASK

In this section we ask about the search task you have just attempted.

1.1. The task we asked you to perform was:

unclear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
simple	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
unfamiliar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
clear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
difficult	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
familiar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

1.2. It was easy to formulate queries on this topic.

Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

1.3. The search I have just performed was:

stressful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
interesting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tiring	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
relaxing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
boring	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
restful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Part 2: RETRIEVED IMAGES

In this section we ask you about the images you found/selected.

2.1. The images I have received through the searches are:

relevant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
inappropriate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
not relevant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
appropriate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
incomplete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.2. I had an idea of which kind of images were relevant for the topic before starting the search.

Not at all	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vague	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.3. During the search I have discovered more aspects of the topic than initially anticipated.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.4. The image(s) I chose in the end match what I had in mind before starting the search.

Exactly	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
some-what	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Not at all	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.5. I believe I have seen all possible images that satisfy my requirement.

Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.6. I am satisfied with my search results.

Very	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
some-what	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Not at all	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Part 3: SYSTEM & INTERACTION

In this section we ask you some general questions about the system you have just used. You can skip this part if you have used the same system for a different task before.

3.1. Overall reaction to the system:

terrible	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
satisfying	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dull	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
rigid	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
efficient	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
novel	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
wonderful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
frustrating	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stimulating	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
difficult	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
flexible	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
inefficient	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
standard	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4.9. What was the most useful tool to support your search strategy?

4.10. What was the least useful tool to support your search strategy?

4.11. Do you have any other comments on the workspace or the recommendations?  
 e.g. a) How did you organise the images for your task?  
 b) What do you think are the benefits of grouping (short-term and long-term)?  
 c) What could be improved?

And finally:  
 4.12. I believe I have succeeded in my performance of the task.

Disagree  1  2  3  4  5 Agree

What are the issues/problems that affected your performance?

	Agree	Disagree
4.13. I didn't understand the task.	1 2 3 4 5	1 2 3 4 5
4.14. I image collection didn't contain the images I wanted.	1 2 3 4 5	1 2 3 4 5
4.15. The system didn't return relevant images.	1 2 3 4 5	1 2 3 4 5
4.16. I didn't have enough time to do an effective search.	1 2 3 4 5	1 2 3 4 5
4.17. I was often unsure of what action to take next.	1 2 3 4 5	1 2 3 4 5

3.2. When interacting with the system, I felt:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
in control							not in control
uncomfortable							comfortable
confident							unconfident

3.3. How easy was it to LEARN TO USE the system?

Not at all  1  2  3  4  5 Extremely

3.4. How easy was it to USE the system?

Extremely  1  2  3  4  5 Not at all

**Part 4: INTERFACE SUPPORT & SEARCH STRATEGY**

In this section we ask you more detailed questions about the interface and your search strategy.

4.1. The interface was effective for solving the task.

Agree  1  2  3  4  5 Disagree

Because it helped me to...

	Disagree	Agree
4.2. analyse the task.	1 2 3 4 5	1 2 3 4 5
4.3. explore the collection.	1 2 3 4 5	1 2 3 4 5
4.4. find relevant images.	1 2 3 4 5	1 2 3 4 5
4.5. organise the images I found for the task.	1 2 3 4 5	1 2 3 4 5
4.6. detect and express different aspects of the task.	1 2 3 4 5	1 2 3 4 5

4.7. The grouping of images (creation of groups and adding images to existing groups) was:

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
difficult							easy
effective							ineffective
not useful							useful

4.8. The interface supported my own style of searching.

Disagree  1  2  3  4  5 Agree

### POST-SEARCH QUESTIONNAIRE

To evaluate the system you have just used, we now ask you to answer some questions about it. Take into account that we are interested in knowing your opinion: answer questions freely, and consider there are no right or wrong answers. Please remember that we are evaluating the system you have just used and not you.



**UNIVERSITY**  
*of*  
**GLASGOW**

User ID:  System:  CS Task:  Order:

Please place a TICK  in the square that best matches your opinion. Please answer all questions.

#### Part 1: TASK

In this section we ask about the search task you have just attempted.

1.1. The task we asked you to perform was:

unclear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
simple	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
unfamiliar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
clear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
difficult	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
familiar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

1.2. It was easy to formulate queries on this topic.

Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

1.3. The search I have just performed was:

stressful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
interesting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tiring	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
relaxing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
boring	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
restful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

#### Part 2: RETRIEVED IMAGES

In this section we ask you about the images you found/selected.

2.1. The images I have received through the searches are:

relevant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
inappropriate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
not relevant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
appropriate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
incomplete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.2. I had an idea of which kind of images were relevant for the topic before starting the search.

Not at all	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vague	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.3. During the search I have discovered more aspects of the topic than initially anticipated.

Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.4. The image(s) I chose in the end match what I had in mind before starting the search.

Exactly	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
some-what	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Not at all	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.5. I believe I have seen all possible images that satisfy my requirement.

Agree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Disagree	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.6. I am satisfied with my search results.

Very	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
some-what	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Not at all	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

#### Part 3: SYSTEM & INTERACTION

In this section we ask you some general questions about the system you have just used. You can skip this part if you have used the same system for a different task before.

3.1. Overall reaction to the system:

terrible	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
satisfying	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dull	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
easy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
rigid	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
efficient	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
novel	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
wonderful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
frustrating	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stimulating	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
difficult	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
flexible	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
inefficient	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
standard	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4.9. What was the most useful tool to support your search strategy?

4.10. What was the least useful tool to support your search strategy?

4.11. Do you have any other comments on the system?  
 e.g. a) Did selecting images usually improve the results?  
 b) What could be improved?

And finally:  
 4.12. I believe I have succeeded in my performance of the task.

Disagree  1  2  3  4  5 Agree

What are the issues/problems that affected your performance?

	Agree					Disagree				
4.13. I didn't understand the task.	1	2	3	4	5	1	2	3	4	5
4.14. I image collection didn't contain the images I wanted.	1	2	3	4	5	1	2	3	4	5
4.15. The system didn't return relevant images.	1	2	3	4	5	1	2	3	4	5
4.16. I didn't have enough time to do an effective search.	1	2	3	4	5	1	2	3	4	5
4.17. I was often unsure of what action to take next.	1	2	3	4	5	1	2	3	4	5

3.2. When interacting with the system, I felt:

in control	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	not in control
uncomfortable	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	comfortable
confident	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	unconfident

3.3. How easy was it to LEARN TO USE the system?

Not at all  1  2  3  4  5 Extremely

3.4. How easy was it to USE the system?

Extremely  1  2  3  4  5 Not at all

**Part 4: INTERFACE SUPPORT & SEARCH STRATEGY**

In this section we ask you more detailed questions about the interface and your search strategy.

4.1. The interface was effective for solving the task.

Agree  1  2  3  4  5 Disagree

Because it helped me to...

	Disagree					Agree				
4.2. analyse the task.	1	2	3	4	5	1	2	3	4	5
4.3. explore the collection.	1	2	3	4	5	1	2	3	4	5
4.4. find relevant images.	1	2	3	4	5	1	2	3	4	5
4.5. organise the images I found for the task.	1	2	3	4	5	1	2	3	4	5
4.6. detect and express different aspects of the task.	1	2	3	4	5	1	2	3	4	5

4.7. How you conveyed relevance to the system (i.e., ticking boxes) was:

difficult	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	easy
effective	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ineffective
not useful	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	useful

4.8. The interface supported my own style of searching.

Disagree  1  2  3  4  5 Agree

## EXIT QUESTIONNAIRE/INTERVIEW

The aim of this experiment was to investigate the relative effectiveness of two different image search interfaces. Please consider the entire search experience that you just had when you respond to the following questions.



UNIVERSITY  
of  
GLASGOW

User ID:

Please place a TICK  in the square that best matches your opinion. Please answer the questions as fully as you feel able to.

### Part 1: TASKS and SEARCH STRATEGY

1.1. To what extent did you find the tasks similar to other searching tasks you typically perform?

Not at all  1  2  3  4  5 Completely

1.2. How did the search tasks fit into your normal work tasks?

- a) What sort of work tasks do you need to perform?
- b) What sort of search tasks do you perform in order to fulfil your work tasks?

1.3. Describe your natural search strategy (faking a typical search task into consideration)?

- a) Your problem solving strategy?
- b) Is it dependent on the search/work task?
- c) In an ideal scenario (when you have the necessary tools), how could a system support your search strategy?

1.4. Which of the two systems supported your strategy better?

- a) How?
- b) Why?
- c) What did you have to do in each case to adapt your search strategy to the system?
- d) In an ideal scenario (when you have the necessary tools) would you be following the same search strategy?

1.5. How important was it for you to organise the images / manage your results?

Not at all  1  2  3  4  5 Completely

1.6. How did you organise the images?

- a) Did it help you clarify/conceptualise/analyse the task?
- b) Did it help in making a selection?
- c) Would you have liked to retrieve whole semantic groups as search results (instead of just single images)?
- d) Do you think other people (or yourself) will benefit from the groupings you created?



**Part 2: TASKS and INFORMATION NEED DEVELOPMENT**

2.1. How clear did you find the tasks and how well-defined was your initial information need?

Task	Unclear					Clear					IN					Not at all					Completely													
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5									
Task 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Task 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Task 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Task 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Task 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2.2. How did your need develop?

a) Did you get new ideas/discover new aspects of the task during the search?  
 b) What caused you to change your initial idea?  
 c) How did the system support/trigger changes?  
 d) Which of the systems was more helpful in developing your information need?

**PART 3: SYSTEM EXPERIENCE**

Which of the systems did you...

	Checkbox	Workspace	No difference
3.1. ... find easier to LEARN TO USE?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.2. ... find easier to USE?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.3. ... find more EFFECTIVE for the tasks you performed?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.4. ... LIKE BEST overall?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3.5. What did you LIKE about each of the systems?

CS:

WS:

3.6. What did you DISLIKE about each of the systems?

CS:

WS:

---

## ADDITIONAL RESULTS FOR THE ICG EVALUATION

---

In Chapter 7 the improved recommendation system based on a semantic feature was discussed and evaluated. Additional results that could not be presented there are compiled in this appendix.

### E.1 Testing the Parameters of the ICG

Before comparing the ICG approach to the individual baselines several experimental runs have been performed to evaluate the influence of the parameters of ICG. The two parameters that influence the computation of the stationary distribution of ICG are:

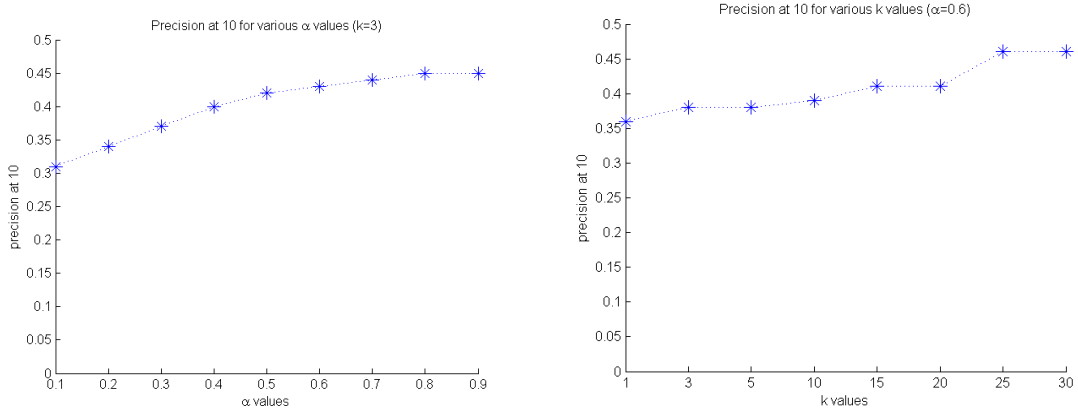
- $\alpha$ : The factor that favours the starting nodes (in the personalisation vector) over browsing the graph structure (see Formula 7.9).
- $k$ : The number of nearest neighbours that determines the number of links between feature nodes.

The results in this section are based on the 10 tasks used in the main experimental runs (cf Section 7.4.3). For each task, each image belonging to the specified category in turn is selected as query image and the results are averaged over these runs (so the total number of queries run per task is the number of images that category contains). The ICG in these runs is constructed without using any peers, that is there are no direct links between image nodes in the graph.

#### $\alpha$ Test

The value of  $\alpha$  influences the importance of the query images (and possibly terms) as opposed to the graph structure when computing the random walk on the ICG. The bigger  $\alpha$ , the more emphasis is on the query images (see Equation 7.9). In the following,  $k$  is fixed at 3 while  $\alpha$  is varied from 0.1 to 0.9.

The results are compiled in Tables E.2–E.9. Further, Figure E.1 shows the development of precision at 10 for various  $\alpha$  values graphically. The results show that the larger  $\alpha$ , the higher the precision. However, variations in  $\alpha$  do not influence recall very much. This means that for a

Figure E.1: P(10) for various values of  $\alpha$  ( $k=3$ ) Figure E.2: P(10) for various values of  $k$  ( $\alpha=0.6$ )Table E.1: P(10) for  $k = 25$ 

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg
Task1	0.41	0.41	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
Task2	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
Task3	0.85	0.84	0.84	0.84	0.83	0.83	0.82	0.82	0.82	0.83
Task4	0.58	0.59	0.59	0.60	0.60	0.60	0.60	0.60	0.60	0.60
Task5	0.80	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81
Task6	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44	0.44
Task7	0.20	0.20	0.20	0.20	0.20	0.19	0.19	0.18	0.18	0.19
Task8	0.10	0.11	0.10	0.10	0.10	0.09	0.09	0.09	0.09	0.10
Task9	0.36	0.37	0.37	0.37	0.37	0.38	0.38	0.37	0.37	0.37
Task10	0.12	0.12	0.12	0.13	0.13	0.13	0.13	0.14	0.14	0.13
Average	0.46	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	

higher  $\alpha$  value, the images that are close to the query items are favoured, which seem to be most relevant for these tasks, as well.

However, if we consider a larger  $k$ , say 25, we observe that for some tasks precision actually peaks for  $\alpha = 0.4$ – $0.6$  as is revealed in Table E.1. In order to place equal emphasis on the graph-structure and the query items, as well as maximising retrieval performance, we therefore suggest using  $\alpha = 0.6$ .

### K-NN Test

When creating the graph, a decision has to be made on how many nearest neighbour links should be added between feature nodes. The parameter  $k$  determining the number of nearest neighbours to be linked is varied between 1 and 30 in the following runs. Note that the maximum number of links between two feature nodes can be  $2k$ , since the edges are undirected.

Having fixed  $\alpha$  at 0.6, we observe that a larger  $k$  results in better precision scores (Tables E.10–E.13). The graph in Figure E.2 visualises the increase in P(10) performance with growing  $k$ . Again, recall is only influenced slightly (Tables E.14–E.17). Therefore, we have chosen to create the ICG with 25 nearest neighbours.

Table E.2: P(10) for  $k = 3$ 

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg
Task1	0.31	0.33	0.36	0.37	0.39	0.40	0.40	0.41	0.41	0.38
Task2	0.52	0.58	0.64	0.70	0.74	0.76	0.79	0.79	0.79	0.70
Task3	0.51	0.57	0.61	0.66	0.70	0.74	0.79	0.82	0.82	0.69
Task4	0.37	0.42	0.48	0.54	0.57	0.59	0.60	0.60	0.60	0.53
Task5	0.51	0.60	0.67	0.72	0.76	0.79	0.80	0.81	0.81	0.72
Task6	0.32	0.34	0.35	0.36	0.37	0.39	0.40	0.42	0.42	0.38
Task7	0.12	0.12	0.11	0.11	0.11	0.10	0.10	0.10	0.10	0.11
Task8	0.07	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Task9	0.26	0.28	0.30	0.32	0.34	0.35	0.36	0.36	0.36	0.33
Task10	0.09	0.10	0.10	0.10	0.11	0.12	0.12	0.13	0.13	0.11
Average	0.31	0.34	0.37	0.40	0.42	0.43	0.44	0.45	0.45	

Table E.3: P(20) for  $k = 3$ 

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg
Task1	0.27	0.30	0.32	0.33	0.35	0.36	0.36	0.37	0.37	0.34
Task2	0.46	0.54	0.60	0.66	0.69	0.72	0.74	0.74	0.74	0.65
Task3	0.50	0.57	0.62	0.69	0.72	0.77	0.82	0.83	0.84	0.71
Task4	0.30	0.35	0.40	0.44	0.47	0.49	0.50	0.50	0.50	0.44
Task5	0.49	0.57	0.64	0.69	0.73	0.75	0.77	0.78	0.78	0.69
Task6	0.25	0.26	0.28	0.29	0.31	0.34	0.37	0.39	0.39	0.32
Task7	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.09	0.09	0.10
Task8	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Task9	0.23	0.25	0.27	0.28	0.30	0.31	0.32	0.32	0.32	0.29
Task10	0.08	0.08	0.09	0.09	0.10	0.10	0.11	0.11	0.11	0.10
Average	0.27	0.31	0.34	0.36	0.38	0.40	0.41	0.42	0.42	

Table E.4: P(50) for  $k = 3$ 

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg
Task1	0.26	0.28	0.29	0.31	0.32	0.32	0.32	0.33	0.33	0.31
Task2	0.44	0.51	0.58	0.63	0.68	0.71	0.74	0.74	0.74	0.64
Task3	0.51	0.59	0.62	0.65	0.69	0.72	0.76	0.78	0.78	0.68
Task4	0.29	0.33	0.35	0.38	0.40	0.41	0.43	0.44	0.44	0.39
Task5	0.51	0.57	0.61	0.65	0.69	0.72	0.74	0.74	0.75	0.66
Task6	0.23	0.27	0.30	0.33	0.36	0.39	0.41	0.42	0.43	0.35
Task7	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.07	0.08
Task8	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05	0.06
Task9	0.20	0.21	0.23	0.24	0.25	0.25	0.25	0.26	0.26	0.24
Task10	0.07	0.08	0.08	0.08	0.09	0.09	0.09	0.09	0.10	0.08
Average	0.26	0.30	0.32	0.34	0.36	0.37	0.39	0.39	0.39	

Table E.5: P(NR) for  $k = 3$ 

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg
Task1	0.22	0.22	0.22	0.23	0.23	0.23	0.23	0.23	0.23	0.23
Task2	0.58	0.63	0.65	0.67	0.69	0.72	0.74	0.75	0.75	0.69
Task3	0.52	0.59	0.62	0.64	0.67	0.72	0.76	0.78	0.78	0.67
Task4	0.29	0.30	0.31	0.32	0.32	0.33	0.33	0.34	0.35	0.32
Task5	0.40	0.41	0.41	0.41	0.41	0.41	0.42	0.42	0.42	0.41
Task6	0.27	0.27	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
Task7	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Task8	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Task9	0.14	0.15	0.15	0.16	0.16	0.16	0.16	0.16	0.17	0.16
Task10	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.07
Average	0.26	0.27	0.28	0.28	0.29	0.30	0.31	0.31	0.31	

Table E.6: R(10) for  $k = 3$ 

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg
Task1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task2	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.07	0.07	0.06
Task3	0.05	0.06	0.06	0.06	0.07	0.07	0.08	0.08	0.08	0.07
Task4	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.02
Task5	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task6	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task7	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Task9	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01
Task10	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Average	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02

Table E.7: R(50) for  $k = 3$ 

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg
Task1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task2	0.08	0.10	0.11	0.12	0.12	0.13	0.13	0.13	0.13	0.12
Task3	0.10	0.11	0.12	0.14	0.14	0.15	0.16	0.16	0.16	0.14
Task4	0.03	0.03	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.04
Task5	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task6	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Task7	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task9	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.02
Task10	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Average	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.04

Table E.8: R(100) for  $k = 3$ 

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg
Task1	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Task2	0.52	0.56	0.58	0.60	0.62	0.65	0.67	0.68	0.68	0.62
Task3	0.51	0.57	0.60	0.63	0.66	0.70	0.74	0.76	0.77	0.66
Task4	0.13	0.14	0.15	0.15	0.16	0.16	0.17	0.17	0.18	0.16
Task5	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.08
Task6	0.08	0.09	0.09	0.10	0.10	0.11	0.11	0.11	0.11	0.10
Task7	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Task8	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task9	0.07	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.09	0.08
Task10	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Average	0.15	0.17	0.17	0.18	0.19	0.20	0.20	0.21	0.21	0.18

Table E.9: R(P05) for  $k = 3$ 

$\alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Avg
Task1	0.03	0.05	0.05	0.06	0.07	0.07	0.08	0.07	0.08	0.06
Task2	0.84	0.87	0.87	0.87	0.87	0.87	0.87	0.90	0.90	0.87
Task3	0.92	0.94	0.94	0.95	0.95	0.95	0.95	0.95	0.95	0.95
Task4	0.14	0.18	0.20	0.22	0.23	0.25	0.25	0.28	0.31	0.23
Task5	0.33	0.34	0.35	0.35	0.35	0.36	0.36	0.36	0.37	0.35
Task6	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
Task7	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01
Task8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Task9	0.02	0.02	0.03	0.04	0.05	0.05	0.05	0.06	0.06	0.04
Task10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00
Average	0.25	0.26	0.27	0.27	0.28	0.28	0.28	0.29	0.29	0.27

Table E.10: P(10) for  $\alpha = 0.6$ 

<b>k</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>	<b>30</b>	<b>Avg</b>
Task1	0.35	0.40	0.41	0.41	0.42	0.42	0.42	0.42	0.41
Task2	0.63	0.63	0.63	0.63	0.63	0.63	0.79	0.79	0.67
Task3	0.60	0.60	0.60	0.60	0.60	0.60	0.83	0.83	0.66
Task4	0.48	0.48	0.48	0.48	0.60	0.60	0.60	0.60	0.54
Task5	0.65	0.79	0.81	0.81	0.81	0.81	0.81	0.81	0.79
Task6	0.34	0.34	0.34	0.43	0.43	0.43	0.44	0.44	0.40
Task7	0.10	0.10	0.10	0.10	0.10	0.10	0.19	0.19	0.12
Task8	0.06	0.06	0.06	0.06	0.08	0.08	0.09	0.09	0.08
Task9	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
Task10	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
Average	0.36	0.38	0.38	0.39	0.41	0.41	0.46	0.46	

Table E.11: P(20) for  $\alpha = 0.6$ 

<b>k</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>	<b>30</b>	<b>Avg</b>
Task1	0.33	0.36	0.37	0.37	0.37	0.37	0.38	0.38	0.37
Task2	0.63	0.63	0.63	0.63	0.63	0.63	0.74	0.74	0.66
Task3	0.69	0.69	0.69	0.69	0.69	0.69	0.84	0.84	0.73
Task4	0.42	0.42	0.42	0.42	0.50	0.50	0.50	0.50	0.46
Task5	0.66	0.75	0.78	0.78	0.78	0.78	0.78	0.78	0.76
Task6	0.30	0.30	0.30	0.40	0.40	0.40	0.41	0.41	0.36
Task7	0.09	0.09	0.09	0.09	0.09	0.09	0.14	0.14	0.10
Task8	0.05	0.05	0.05	0.05	0.07	0.07	0.08	0.08	0.07
Task9	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
Task10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
Average	0.36	0.37	0.37	0.38	0.39	0.39	0.43	0.43	

Table E.12: P(50) for  $\alpha = 0.6$ 

<b>k</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>	<b>30</b>	<b>Avg</b>
Task1	0.31	0.32	0.33	0.33	0.34	0.34	0.34	0.34	0.33
Task2	0.63	0.63	0.63	0.63	0.63	0.63	0.74	0.74	0.66
Task3	0.70	0.70	0.70	0.70	0.70	0.70	0.79	0.79	0.72
Task4	0.39	0.39	0.39	0.39	0.44	0.44	0.44	0.44	0.42
Task5	0.67	0.72	0.74	0.75	0.75	0.75	0.75	0.75	0.73
Task6	0.36	0.36	0.36	0.43	0.43	0.43	0.43	0.43	0.40
Task7	0.04	0.04	0.04	0.04	0.04	0.04	0.09	0.09	0.05
Task8	0.03	0.03	0.03	0.03	0.06	0.06	0.06	0.06	0.04
Task9	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
Task10	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
Average	0.35	0.35	0.36	0.36	0.37	0.37	0.40	0.40	

Table E.13: P(NR) for  $\alpha = 0.6$ 

<b>k</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>	<b>30</b>	<b>Avg</b>
Task1	0.23	0.23	0.22	0.23	0.23	0.23	0.23	0.23	0.23
Task2	0.71	0.71	0.71	0.71	0.71	0.71	0.75	0.75	0.72
Task3	0.72	0.72	0.72	0.72	0.72	0.72	0.79	0.79	0.74
Task4	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
Task5	0.42	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
Task6	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
Task7	0.03	0.03	0.03	0.03	0.03	0.03	0.06	0.06	0.04
Task8	0.01	0.01	0.01	0.01	0.05	0.05	0.05	0.05	0.03
Task9	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16
Task10	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07
Average	0.29	0.29	0.29	0.29	0.30	0.30	0.31	0.31	

Table E.14: R(10) for  $\alpha = 0.6$ 

<b>k</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>	<b>30</b>	<b>Avg</b>
Task1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task2	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.06
Task3	0.06	0.06	0.06	0.06	0.06	0.06	0.08	0.08	0.06
Task4	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.02
Task5	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task6	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task7	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.01
Task8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Task9	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task10	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Average	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02

Table E.15: R(50) for  $\alpha = 0.6$ 

<b>k</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>	<b>30</b>	<b>Avg</b>
Task1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task2	0.11	0.11	0.11	0.11	0.11	0.11	0.13	0.13	0.12
Task3	0.14	0.14	0.14	0.14	0.14	0.14	0.17	0.17	0.14
Task4	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.04
Task5	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task6	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Task7	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.02
Task8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task9	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task10	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Average	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.04

Table E.16: R(100) for  $\alpha = 0.6$ 

<b>k</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>	<b>30</b>	<b>Avg</b>
Task1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Task2	0.63	0.63	0.63	0.63	0.63	0.63	0.69	0.69	0.65
Task3	0.70	0.70	0.70	0.70	0.70	0.70	0.77	0.77	0.72
Task4	0.16	0.16	0.16	0.16	0.18	0.18	0.18	0.18	0.17
Task5	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
Task6	0.10	0.10	0.10	0.11	0.11	0.11	0.11	0.11	0.11
Task7	0.03	0.03	0.03	0.03	0.03	0.03	0.06	0.06	0.04
Task8	0.01	0.01	0.01	0.01	0.03	0.03	0.03	0.03	0.02
Task9	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
Task10	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Average	0.19	0.19	0.19	0.19	0.19	0.19	0.21	0.21	0.19

Table E.17: R(P05) for  $\alpha = 0.6$ 

<b>k</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>15</b>	<b>20</b>	<b>25</b>	<b>30</b>	<b>Avg</b>
Task1	0.08	0.07	0.07	0.08	0.08	0.08	0.08	0.08	0.08
Task2	0.87	0.87	0.87	0.87	0.87	0.87	0.90	0.90	0.88
Task3	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
Task4	0.26	0.26	0.26	0.26	0.28	0.28	0.28	0.28	0.27
Task5	0.37	0.36	0.36	0.35	0.36	0.36	0.36	0.36	0.36
Task6	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22
Task7	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.02	0.01
Task8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Task9	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Task10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average	0.28	0.28	0.28	0.28	0.28	0.28	0.29	0.29	0.28

## **E.2 Additional Results of Runs without Relevance Feedback**

Tables E.18–E.23 show additional task-based results for the runs discussed in Section 7.5.1. Please refer to this section for the setup of these runs.

## **E.3 Runs With Relevance Feedback**

Tables E.24–E.35 show additional results for the runs discussed in Section 7.5.2. Please refer to this section for the setup of these runs. Tables E.24–E.26 show the results for P10, P100, and R100 after the first RF iteration. The same results after the fifth iteration are shown in Tables E.27–E.29, and after the tenth iteration in Tables E.30–E.32. Finally, Tables E.33–E.35 average the results over all 20 iterations.

## **E.4 Variations of Group Size**

Tables E.36–E.45 show additional results for the runs discussed in Section 7.5.3. Please refer to this section for the setup of these runs.



Table E.18: P(20) for baselines and ICG

<b>P20</b>	<b>IND<sub>v</sub></b>	<b>IND<sub>t</sub></b>	<b>IND<sub>p</sub></b>	<b>IND<sub>tv</sub></b>	<b>ICG</b>	<b>IND</b>	<b>ICG<sub>p</sub></b>
Task1	0.19	0.40	0.35	0.40	0.38	0.56	0.59
Task2	0.12	0.97	0.56	0.68	0.74	0.81	0.83
Task3	0.05	0.96	0.37	0.39	0.84	0.63	0.82
Task4	0.13	0.70	0.43	0.56	0.50	0.68	0.70
Task5	0.23	0.91	0.30	0.64	0.78	0.73	0.83
Task6	0.16	0.45	0.38	0.42	0.41	0.57	0.61
Task7	0.10	0.00	0.00	0.10	0.14	0.10	0.14
Task8	0.08	0.00	0.00	0.08	0.08	0.08	0.08
Task9	0.11	0.38	0.23	0.28	0.28	0.38	0.41
Task10	0.08	0.14	0.00	0.17	0.10	0.17	0.11
Average	0.13	0.49	0.26	0.37	0.43	0.47	0.51

Table E.19: P(50) for baselines and ICG

<b>P50</b>	<b>IND<sub>v</sub></b>	<b>IND<sub>t</sub></b>	<b>IND<sub>p</sub></b>	<b>IND<sub>tv</sub></b>	<b>ICG</b>	<b>IND</b>	<b>ICG<sub>p</sub></b>
Task1	0.17	0.34	0.35	0.35	0.34	0.52	0.57
Task2	0.08	0.97	0.54	0.46	0.74	0.73	0.80
Task3	0.04	0.95	0.27	0.26	0.79	0.50	0.72
Task4	0.11	0.56	0.43	0.45	0.44	0.59	0.69
Task5	0.20	0.86	0.31	0.53	0.75	0.63	0.80
Task6	0.14	0.43	0.38	0.36	0.43	0.54	0.63
Task7	0.07	0.00	0.00	0.07	0.09	0.07	0.09
Task8	0.07	0.00	0.00	0.07	0.06	0.07	0.06
Task9	0.09	0.28	0.19	0.22	0.24	0.31	0.35
Task10	0.07	0.11	0.00	0.14	0.09	0.14	0.09
Average	0.10	0.45	0.25	0.29	0.40	0.41	0.48

Table E.20: P(NR) for baselines and ICG

<b>PNR</b>	<b>IND<sub>v</sub></b>	<b>IND<sub>t</sub></b>	<b>IND<sub>p</sub></b>	<b>IND<sub>tv</sub></b>	<b>ICG</b>	<b>IND</b>	<b>ICG<sub>p</sub></b>
Task1	0.12	0.20	0.00	0.19	0.23	0.25	0.38
Task2	0.07	0.91	0.33	0.31	0.75	0.51	0.60
Task3	0.03	0.90	0.16	0.18	0.79	0.31	0.60
Task4	0.07	0.35	0.00	0.29	0.34	0.35	0.50
Task5	0.12	0.26	0.00	0.31	0.41	0.33	0.51
Task6	0.09	0.10	0.00	0.20	0.26	0.28	0.44
Task7	0.05	0.00	0.00	0.05	0.06	0.05	0.06
Task8	0.05	0.00	0.00	0.05	0.05	0.05	0.05
Task9	0.06	0.17	0.00	0.13	0.16	0.19	0.28
Task10	0.05	0.08	0.00	0.10	0.07	0.10	0.07
Average	0.07	0.30	0.05	0.18	0.31	0.24	0.35

Table E.21: R(10) for baselines and ICG

<b>R10</b>	<b>IND<sub>v</sub></b>	<b>IND<sub>t</sub></b>	<b>IND<sub>p</sub></b>	<b>IND<sub>tv</sub></b>	<b>ICG</b>	<b>IND</b>	<b>ICG<sub>p</sub></b>
Task1	0.00	0.01	0.01	0.01	0.01	0.01	0.01
Task2	0.01	0.09	0.05	0.07	0.07	0.08	0.08
Task3	0.01	0.10	0.04	0.05	0.08	0.07	0.08
Task4	0.01	0.03	0.02	0.03	0.03	0.03	0.03
Task5	0.00	0.01	0.00	0.01	0.01	0.01	0.01
Task6	0.00	0.01	0.01	0.01	0.01	0.01	0.02
Task7	0.01	0.00	0.00	0.01	0.02	0.01	0.02
Task8	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Task9	0.01	0.02	0.01	0.01	0.01	0.02	0.02
Task10	0.00	0.01	0.00	0.01	0.01	0.01	0.01
Average	0.01	0.03	0.01	0.02	0.02	0.03	0.03

Table E.22: R(50) for baselines and ICG

<b>R50</b>	<b>IND<sub>v</sub></b>	<b>IND<sub>t</sub></b>	<b>IND<sub>p</sub></b>	<b>IND<sub>tv</sub></b>	<b>ICG</b>	<b>IND</b>	<b>ICG<sub>p</sub></b>
Task1	0.01	0.01	0.01	0.01	0.01	0.02	0.02
Task2	0.02	0.17	0.10	0.12	0.13	0.14	0.15
Task3	0.01	0.19	0.07	0.08	0.17	0.12	0.16
Task4	0.01	0.06	0.04	0.05	0.05	0.06	0.06
Task5	0.01	0.02	0.01	0.01	0.02	0.02	0.02
Task6	0.01	0.02	0.02	0.02	0.02	0.03	0.03
Task7	0.02	0.00	0.00	0.02	0.03	0.02	0.03
Task8	0.01	0.00	0.00	0.01	0.01	0.01	0.01
Task9	0.01	0.03	0.02	0.02	0.02	0.03	0.03
Task10	0.01	0.01	0.00	0.02	0.01	0.02	0.01
Average	0.01	0.05	0.03	0.04	0.05	0.05	0.05

Table E.23: R(P05) for baselines and ICG

<b>RP05</b>	<b>IND<sub>v</sub></b>	<b>IND<sub>t</sub></b>	<b>IND<sub>p</sub></b>	<b>IND<sub>tv</sub></b>	<b>ICG</b>	<b>IND</b>	<b>ICG<sub>p</sub></b>
Task1	0.01	0.09	0.00	0.05	0.08	0.14	0.31
Task2	0.01	0.96	0.33	0.25	0.90	0.51	0.69
Task3	0.00	0.96	0.13	0.08	0.95	0.25	0.67
Task4	0.00	0.27	0.00	0.18	0.28	0.29	0.55
Task5	0.00	0.19	0.00	0.17	0.36	0.22	0.51
Task6	0.01	0.11	0.00	0.13	0.22	0.23	0.43
Task7	0.01	0.00	0.00	0.01	0.02	0.01	0.02
Task8	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Task9	0.00	0.06	0.08	0.03	0.05	0.07	0.14
Task10	0.00	0.01	0.00	0.01	0.00	0.01	0.00
Average	0.00	0.26	0.05	0.09	0.29	0.17	0.33

Table E.24: P(10) after the first RF iteration

<b>P10</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.79	0.81	0.59	0.87	0.87	0.87	0.87	0.87	0.88	0.87
Task2	0.98	0.93	0.92	0.97	0.97	0.97	0.97	0.97	0.96	0.96
Task3	0.88	0.92	0.90	0.93	0.93	0.93	0.93	0.93	0.92	0.92
Task4	0.80	0.83	0.68	0.94	0.94	0.94	0.94	0.94	0.94	0.94
Task5	0.78	0.87	0.80	0.91	0.91	0.91	0.91	0.91	0.91	0.91
Task6	0.82	0.83	0.71	0.91	0.91	0.90	0.91	0.91	0.91	0.90
Task7	0.12	0.12	0.20	0.32	0.32	0.12	0.33	0.32	0.31	0.31
Task8	0.09	0.09	0.08	0.14	0.14	0.12	0.16	0.15	0.14	0.14
Task9	0.43	0.32	0.49	0.67	0.67	0.67	0.66	0.67	0.66	0.65
Task10	0.15	0.11	0.19	0.17	0.17	0.17	0.18	0.17	0.17	0.17
Average	0.58	0.58	0.56	0.68	0.68	0.66	0.68	0.68	0.68	0.68

Table E.25: P(100) after the first RF iteration

<b>P100</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.51	0.61	0.44	0.86	0.86	0.85	0.86	0.85	0.86	0.86
Task2	0.71	0.61	0.91	0.66	0.66	0.68	0.64	0.71	0.66	0.64
Task3	0.53	0.39	0.87	0.52	0.52	0.54	0.49	0.59	0.51	0.47
Task4	0.48	0.50	0.52	0.86	0.86	0.86	0.87	0.85	0.87	0.87
Task5	0.57	0.65	0.79	0.85	0.85	0.85	0.85	0.84	0.85	0.86
Task6	0.58	0.62	0.74	0.91	0.91	0.91	0.91	0.91	0.91	0.91
Task7	0.06	0.05	0.07	0.11	0.11	0.06	0.11	0.11	0.10	0.10
Task8	0.05	0.05	0.05	0.08	0.08	0.07	0.08	0.08	0.08	0.08
Task9	0.29	0.24	0.33	0.46	0.46	0.46	0.46	0.45	0.45	0.46
Task10	0.07	0.06	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
Average	0.38	0.38	0.48	0.54	0.54	0.54	0.54	0.55	0.54	0.54

Table E.26: R(100) after the first RF iteration

<b>R100</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.09	0.11	0.08	0.16	0.16	0.16	0.16	0.16	0.16	0.16
Task2	0.65	0.55	0.83	0.60	0.60	0.62	0.59	0.65	0.60	0.58
Task3	0.53	0.39	0.87	0.52	0.52	0.54	0.49	0.59	0.51	0.47
Task4	0.22	0.23	0.24	0.40	0.40	0.39	0.40	0.39	0.40	0.40
Task5	0.07	0.08	0.09	0.10	0.10	0.10	0.10	0.10	0.10	0.10
Task6	0.15	0.16	0.18	0.23	0.23	0.23	0.23	0.23	0.23	0.23
Task7	0.06	0.06	0.07	0.11	0.11	0.06	0.11	0.11	0.11	0.11
Task8	0.03	0.03	0.02	0.04	0.04	0.03	0.04	0.04	0.04	0.04
Task9	0.12	0.10	0.14	0.19	0.19	0.19	0.19	0.19	0.19	0.19
Task10	0.03	0.03	0.06	0.06	0.06	0.06	0.06	0.05	0.06	0.05
Average	0.19	0.17	0.26	0.24	0.24	0.24	0.24	0.25	0.24	0.23

Table E.27: P(10) after the fifth RF iteration

<b>P10</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.99	0.98	0.54	0.98	0.98	0.98	0.98	0.99	0.99	0.99
Task2	1.00	0.96	1.00	0.92	0.92	0.93	0.92	0.93	0.92	0.91
Task3	0.77	0.74	0.94	0.20	0.20	0.22	0.25	0.20	0.22	0.23
Task4	0.97	0.99	0.47	0.99	1.00	1.00	0.99	1.00	0.99	0.99
Task5	0.91	0.86	0.87	0.91	0.91	0.92	0.92	0.92	0.92	0.91
Task6	0.96	0.93	0.97	0.99	0.99	0.99	0.98	0.98	0.99	0.99
Task7	0.03	0.04	0.17	0.26	0.26	0.32	0.26	0.25	0.26	0.26
Task8	0.02	0.02	0.11	0.20	0.20	0.18	0.19	0.20	0.18	0.18
Task9	0.52	0.51	0.27	0.40	0.40	0.40	0.40	0.40	0.40	0.40
Task10	0.08	0.06	0.16	0.15	0.15	0.15	0.15	0.14	0.16	0.15
Average	0.63	0.61	0.55	0.60	0.60	0.61	0.60	0.60	0.60	0.60

Table E.28: P(100) after the fifth RF iteration

<b>P100</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.59	0.32	0.51	0.98	0.98	0.98	0.98	0.98	0.98	0.98
Task2	0.52	0.31	0.71	0.29	0.29	0.29	0.29	0.30	0.29	0.29
Task3	0.39	0.10	0.58	0.17	0.16	0.16	0.15	0.17	0.17	0.18
Task4	0.55	0.24	0.51	0.83	0.84	0.83	0.83	0.84	0.83	0.83
Task5	0.71	0.27	0.72	0.95	0.95	0.96	0.96	0.96	0.96	0.95
Task6	0.69	0.32	0.95	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Task7	0.02	0.03	0.07	0.10	0.10	0.14	0.10	0.10	0.10	0.10
Task8	0.02	0.01	0.07	0.12	0.12	0.10	0.13	0.12	0.12	0.12
Task9	0.35	0.36	0.29	0.45	0.45	0.44	0.45	0.45	0.45	0.45
Task10	0.05	0.02	0.12	0.11	0.11	0.11	0.11	0.11	0.11	0.11
Average	0.39	0.20	0.45	0.50	0.50	0.50	0.50	0.50	0.50	0.50

Table E.29: R(100) after the fifth RF iteration

<b>R100</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.12	0.06	0.10	0.19	0.19	0.19	0.19	0.19	0.19	0.19
Task2	0.75	0.44	0.97	0.40	0.40	0.41	0.40	0.42	0.41	0.40
Task3	0.61	0.15	0.94	0.22	0.22	0.21	0.20	0.22	0.22	0.23
Task4	0.31	0.14	0.27	0.47	0.47	0.47	0.47	0.47	0.47	0.47
Task5	0.09	0.03	0.09	0.12	0.12	0.12	0.12	0.12	0.12	0.12
Task6	0.19	0.09	0.26	0.28	0.28	0.27	0.27	0.27	0.28	0.28
Task7	0.02	0.04	0.09	0.12	0.12	0.16	0.12	0.12	0.12	0.12
Task8	0.01	0.01	0.03	0.06	0.06	0.05	0.07	0.07	0.06	0.06
Task9	0.16	0.16	0.13	0.21	0.21	0.20	0.21	0.21	0.21	0.21
Task10	0.03	0.01	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Average	0.23	0.11	0.29	0.21	0.21	0.21	0.21	0.21	0.21	0.21

Table E.30: P(10) after the tenth RF iteration

<b>P10</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.88	0.66	0.61	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task2	0.09	0.02	1.00	0.10	0.15	0.09	0.10	0.21	0.16	0.10
Task3	0.29	0.06	0.97	0.03	0.08	0.03	0.04	0.03	0.08	0.03
Task4	0.47	0.10	0.71	0.45	0.45	0.45	0.38	0.54	0.45	0.44
Task5	0.88	0.67	0.82	0.97	0.97	0.97	0.96	0.98	0.97	0.97
Task6	0.75	0.49	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.02	0.03	0.17	0.17	0.16	0.20	0.17	0.16	0.17	0.16
Task8	0.02	0.01	0.19	0.22	0.22	0.13	0.24	0.24	0.22	0.22
Task9	0.46	0.42	0.30	0.53	0.54	0.54	0.53	0.54	0.54	0.54
Task10	0.07	0.05	0.16	0.13	0.13	0.12	0.13	0.12	0.14	0.15
Average	0.39	0.25	0.59	0.46	0.47	0.45	0.45	0.48	0.47	0.46

Table E.31: P(100) after the tenth RF iteration

<b>P100</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.57	0.14	0.57	0.98	0.98	0.98	0.98	0.98	0.98	0.98
Task2	0.29	0.00	0.21	0.07	0.07	0.06	0.06	0.07	0.07	0.07
Task3	0.33	0.01	0.12	0.21	0.21	0.21	0.07	0.29	0.21	0.20
Task4	0.35	0.03	0.52	0.69	0.69	0.69	0.69	0.69	0.69	0.68
Task5	0.71	0.19	0.66	0.97	0.97	0.97	0.97	0.98	0.97	0.97
Task6	0.64	0.07	0.96	0.98	0.98	0.98	0.98	0.98	0.98	0.98
Task7	0.02	0.03	0.07	0.07	0.07	0.09	0.07	0.07	0.07	0.07
Task8	0.02	0.01	0.09	0.15	0.15	0.09	0.15	0.15	0.15	0.15
Task9	0.31	0.33	0.36	0.43	0.43	0.43	0.43	0.43	0.43	0.43
Task10	0.05	0.02	0.11	0.10	0.10	0.10	0.10	0.11	0.10	0.11
Average	0.33	0.08	0.37	0.47	0.47	0.46	0.45	0.47	0.46	0.46

Table E.32: R(100) after the tenth RF iteration

<b>R100</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.12	0.03	0.12	0.21	0.21	0.21	0.21	0.21	0.21	0.21
Task2	0.82	0.01	0.91	0.13	0.13	0.13	0.13	0.15	0.13	0.13
Task3	0.75	0.01	0.83	0.35	0.35	0.35	0.12	0.49	0.35	0.34
Task4	0.24	0.02	0.33	0.54	0.54	0.54	0.54	0.53	0.53	0.53
Task5	0.09	0.02	0.08	0.13	0.13	0.13	0.12	0.13	0.13	0.12
Task6	0.20	0.02	0.30	0.32	0.32	0.32	0.31	0.32	0.31	0.31
Task7	0.02	0.03	0.09	0.10	0.10	0.12	0.10	0.10	0.10	0.10
Task8	0.01	0.01	0.05	0.08	0.08	0.05	0.09	0.08	0.08	0.08
Task9	0.16	0.17	0.17	0.22	0.22	0.22	0.22	0.22	0.22	0.22
Task10	0.03	0.01	0.06	0.05	0.05	0.05	0.05	0.06	0.05	0.06
Average	0.24	0.03	0.30	0.21	0.21	0.21	0.19	0.23	0.21	0.21

Table E.33: Average P(10) over 20 RF iterations

<b>P10</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.77	0.55	0.65	0.96	0.96	0.96	0.96	0.96	0.96	0.96
Task2	0.51	0.36	0.54	0.36	0.36	0.36	0.36	0.36	0.36	0.36
Task3	0.44	0.28	0.48	0.29	0.29	0.29	0.28	0.32	0.28	0.29
Task4	0.56	0.40	0.65	0.74	0.74	0.74	0.66	0.76	0.74	0.74
Task5	0.83	0.58	0.83	0.96	0.96	0.96	0.96	0.97	0.96	0.96
Task6	0.76	0.48	0.95	0.98	0.98	0.98	0.98	0.99	0.98	0.98
Task7	0.03	0.04	0.17	0.17	0.17	0.21	0.18	0.17	0.17	0.17
Task8	0.02	0.02	0.17	0.21	0.21	0.14	0.21	0.21	0.20	0.21
Task9	0.41	0.44	0.38	0.45	0.45	0.45	0.45	0.45	0.45	0.45
Task10	0.08	0.05	0.16	0.13	0.13	0.13	0.13	0.13	0.13	0.13
Average	0.44	0.32	0.50	0.53	0.52	0.52	0.52	0.53	0.52	0.53

Table E.34: Average P(100) over 20 RF iterations

<b>P100</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.54	0.20	0.54	0.92	0.92	0.92	0.92	0.93	0.92	0.92
Task2	0.35	0.14	0.32	0.22	0.22	0.22	0.19	0.27	0.22	0.22
Task3	0.33	0.06	0.27	0.34	0.33	0.34	0.23	0.36	0.34	0.34
Task4	0.31	0.11	0.50	0.64	0.65	0.64	0.56	0.63	0.64	0.64
Task5	0.70	0.21	0.71	0.89	0.89	0.89	0.89	0.89	0.89	0.89
Task6	0.65	0.17	0.74	0.96	0.96	0.96	0.96	0.96	0.96	0.96
Task7	0.02	0.03	0.07	0.07	0.07	0.09	0.08	0.07	0.07	0.07
Task8	0.02	0.02	0.09	0.13	0.13	0.09	0.14	0.13	0.13	0.13
Task9	0.27	0.31	0.37	0.44	0.44	0.44	0.44	0.44	0.44	0.44
Task10	0.05	0.02	0.11	0.10	0.10	0.10	0.10	0.10	0.10	0.11
Average	0.32	0.13	0.37	0.47	0.47	0.47	0.45	0.48	0.47	0.47

Table E.35: Average R(100) over 20 RF iterations

<b>R100</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG</b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pd</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.12	0.04	0.11	0.20	0.20	0.20	0.20	0.21	0.21	0.20
Task2	0.82	0.17	0.52	0.35	0.35	0.35	0.29	0.48	0.35	0.35
Task3	0.77	0.08	0.54	0.55	0.54	0.55	0.35	0.62	0.56	0.55
Task4	0.19	0.06	0.34	0.47	0.48	0.47	0.38	0.47	0.47	0.47
Task5	0.09	0.03	0.09	0.11	0.11	0.11	0.11	0.11	0.11	0.11
Task6	0.21	0.05	0.23	0.32	0.32	0.32	0.32	0.32	0.32	0.32
Task7	0.02	0.03	0.09	0.10	0.09	0.12	0.10	0.10	0.09	0.09
Task8	0.01	0.01	0.05	0.08	0.08	0.05	0.08	0.08	0.08	0.08
Task9	0.14	0.16	0.18	0.23	0.23	0.23	0.23	0.23	0.23	0.23
Task10	0.03	0.01	0.06	0.05	0.05	0.06	0.06	0.06	0.06	0.06
Average	0.24	0.06	0.22	0.25	0.25	0.25	0.21	0.27	0.25	0.25

Table E.36: P(20) for  $IND$  for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.82	0.87	0.94	0.94	0.96	0.99	1.00	1.00	1.00	1.00	0.95
Task2	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task3	0.88	0.93	0.91	0.89	0.82	0.82	0.84	0.83	0.82	0.79	0.85
Task4	0.66	0.74	0.86	0.85	0.93	0.99	0.99	0.99	0.99	0.99	0.90
Task5	0.85	0.87	0.88	0.87	0.81	0.80	0.78	0.76	0.76	0.76	0.81
Task6	0.91	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Task7	0.12	0.09	0.10	0.10	0.05	0.10	0.10	0.09	0.09	0.07	0.09
Task8	0.07	0.07	0.06	0.05	0.10	0.11	0.10	0.10	0.09	0.09	0.08
Task9	0.49	0.44	0.42	0.43	0.33	0.36	0.36	0.36	0.35	0.34	0.39
Task10	0.14	0.14	0.13	0.12	0.10	0.08	0.08	0.09	0.09	0.09	0.11
Average	0.59	0.61	0.63	0.62	0.61	0.63	0.62	0.62	0.62	0.61	

Table E.37: P(20) for  $ICG_p$  for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.93	0.98	0.98	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Task2	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task3	0.88	0.91	0.94	0.95	0.95	0.96	0.97	0.96	0.94	0.90	0.94
Task4	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task5	0.96	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Task6	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.31	0.36	0.38	0.40	0.37	0.35	0.31	0.28	0.26	0.23	0.33
Task8	0.16	0.22	0.28	0.31	0.35	0.37	0.39	0.42	0.44	0.47	0.34
Task9	0.59	0.58	0.57	0.57	0.56	0.55	0.54	0.53	0.52	0.52	0.55
Task10	0.16	0.18	0.18	0.19	0.19	0.19	0.18	0.19	0.18	0.15	0.18
Average	0.69	0.72	0.73	0.74	0.74	0.74	0.74	0.74	0.73	0.73	

Table E.38: P(50) for  $IND$  for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.66	0.68	0.72	0.73	0.72	0.87	0.86	0.86	0.85	0.86	0.78
Task2	0.98	0.99	0.99	0.99	0.98	0.99	0.96	0.92	0.88	0.83	0.95
Task3	0.69	0.68	0.65	0.63	0.60	0.62	0.61	0.59	0.56	0.54	0.62
Task4	0.48	0.53	0.64	0.63	0.71	0.79	0.78	0.76	0.74	0.71	0.68
Task5	0.75	0.76	0.76	0.75	0.70	0.69	0.67	0.67	0.67	0.66	0.71
Task6	0.78	0.85	0.88	0.89	0.87	0.91	0.92	0.92	0.92	0.91	0.88
Task7	0.09	0.06	0.07	0.07	0.04	0.08	0.09	0.08	0.07	0.07	0.07
Task8	0.06	0.06	0.06	0.05	0.09	0.11	0.10	0.10	0.09	0.09	0.08
Task9	0.42	0.38	0.36	0.35	0.31	0.36	0.37	0.38	0.37	0.37	0.37
Task10	0.10	0.10	0.10	0.10	0.08	0.07	0.08	0.08	0.08	0.07	0.09
Average	0.50	0.51	0.52	0.52	0.51	0.55	0.54	0.53	0.52	0.51	

Table E.39: P(50) for  $ICG_p$  for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.93	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Task2	0.93	0.97	0.99	0.98	0.97	0.94	0.90	0.85	0.80	0.75	0.91
Task3	0.67	0.65	0.63	0.61	0.58	0.55	0.52	0.48	0.44	0.40	0.55
Task4	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task5	0.93	0.94	0.97	0.99	0.99	0.99	0.99	1.00	1.00	1.00	0.98
Task6	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.20	0.24	0.26	0.27	0.24	0.23	0.20	0.18	0.17	0.14	0.21
Task8	0.12	0.16	0.19	0.22	0.25	0.26	0.28	0.29	0.30	0.31	0.24
Task9	0.51	0.52	0.52	0.51	0.51	0.50	0.49	0.48	0.48	0.46	0.50
Task10	0.13	0.16	0.16	0.16	0.16	0.15	0.15	0.15	0.14	0.13	0.15
Average	0.64	0.66	0.67	0.67	0.67	0.66	0.65	0.64	0.63	0.62	



Table E.40: R(10) for *IND* for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task2	0.09	0.10	0.10	0.11	0.11	0.12	0.13	0.14	0.15	0.16	0.12
Task3	0.09	0.10	0.11	0.11	0.11	0.12	0.13	0.14	0.15	0.16	0.12
Task4	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.05
Task5	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task6	0.02	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Task7	0.01	0.01	0.01	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.01
Task8	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.00
Task9	0.02	0.02	0.02	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Task10	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.01	0.01	0.01	0.01
Average	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	

Table E.41: R(10) for *ICG<sub>p</sub>* for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task2	0.09	0.10	0.10	0.11	0.11	0.12	0.13	0.14	0.15	0.16	0.12
Task3	0.10	0.10	0.11	0.12	0.13	0.14	0.15	0.16	0.17	0.19	0.14
Task4	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.05
Task5	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task6	0.02	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Task7	0.04	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
Task8	0.01	0.01	0.02	0.02	0.03	0.03	0.03	0.03	0.04	0.04	0.03
Task9	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
Task10	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Average	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	0.06	0.06	

Table E.42: R(50) for  $IND$  for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Task2	0.18	0.19	0.20	0.22	0.23	0.24	0.26	0.27	0.29	0.32	0.24
Task3	0.18	0.20	0.21	0.21	0.21	0.22	0.25	0.26	0.28	0.30	0.23
Task4	0.06	0.07	0.08	0.08	0.10	0.10	0.11	0.11	0.11	0.12	0.09
Task5	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task6	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.05
Task7	0.02	0.02	0.02	0.02	0.01	0.03	0.03	0.03	0.03	0.03	0.03
Task8	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task9	0.04	0.04	0.04	0.04	0.03	0.03	0.03	0.04	0.03	0.03	0.04
Task10	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Average	0.06	0.06	0.07	0.07	0.07	0.08	0.08	0.08	0.09	0.09	

Table E.43: R(50) for  $ICG_p$  for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
Task2	0.18	0.19	0.20	0.22	0.23	0.24	0.26	0.27	0.29	0.32	0.24
Task3	0.18	0.20	0.21	0.23	0.24	0.26	0.29	0.31	0.32	0.34	0.26
Task4	0.09	0.10	0.10	0.10	0.10	0.11	0.11	0.11	0.11	0.12	0.10
Task5	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task6	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.06	0.06	0.05
Task7	0.06	0.08	0.09	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.09
Task8	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.04
Task9	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Task10	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Average	0.07	0.08	0.08	0.09	0.09	0.09	0.10	0.10	0.11	0.11	

Table E.44: R(100) for  $IND$  for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.10	0.11	0.12	0.12	0.12	0.13	0.12	0.12	0.12	0.13	0.12
Task2	0.68	0.70	0.71	0.73	0.76	0.78	0.79	0.81	0.82	0.84	0.76
Task3	0.53	0.54	0.55	0.56	0.58	0.63	0.66	0.69	0.73	0.77	0.62
Task4	0.19	0.21	0.25	0.25	0.27	0.29	0.30	0.30	0.30	0.30	0.27
Task5	0.08	0.08	0.08	0.08	0.08	0.07	0.07	0.07	0.08	0.07	0.08
Task6	0.17	0.18	0.19	0.19	0.19	0.20	0.20	0.20	0.21	0.21	0.19
Task7	0.08	0.06	0.07	0.07	0.04	0.10	0.11	0.11	0.12	0.11	0.09
Task8	0.03	0.03	0.03	0.03	0.04	0.06	0.06	0.05	0.05	0.05	0.04
Task9	0.14	0.14	0.13	0.13	0.13	0.14	0.14	0.14	0.14	0.14	0.14
Task10	0.04	0.04	0.04	0.05	0.04	0.04	0.04	0.05	0.04	0.04	0.04
Average	0.20	0.21	0.22	0.22	0.23	0.24	0.25	0.25	0.26	0.27	

Table E.45: R(100) for  $ICG_p$  for various group sizes

Group size	5	10	15	20	25	30	35	40	45	50	Avg
Task1	0.17	0.18	0.19	0.19	0.19	0.19	0.19	0.20	0.20	0.20	0.19
Task2	0.61	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.63	0.63	0.62
Task3	0.49	0.46	0.45	0.45	0.45	0.45	0.45	0.44	0.45	0.45	0.45
Task4	0.42	0.44	0.45	0.46	0.47	0.48	0.49	0.50	0.51	0.52	0.47
Task5	0.10	0.11	0.11	0.11	0.12	0.12	0.12	0.12	0.12	0.12	0.11
Task6	0.25	0.25	0.26	0.26	0.27	0.27	0.27	0.28	0.28	0.28	0.27
Task7	0.14	0.19	0.21	0.23	0.23	0.23	0.22	0.20	0.20	0.19	0.20
Task8	0.05	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.14	0.15	0.11
Task9	0.20	0.21	0.21	0.22	0.22	0.22	0.23	0.23	0.23	0.24	0.22
Task10	0.06	0.07	0.07	0.08	0.08	0.08	0.08	0.09	0.09	0.08	0.08
Average	0.25	0.26	0.26	0.27	0.27	0.28	0.28	0.28	0.28	0.29	

## E.5 Adaptive Feature Weights to Implement Short-term Learning

The objective of feature weights is to influence the importance of the three overall features used: visual, textual and peer features. In order to calculate the weights automatically on a per-query basis, they are estimated based on the similarity between the query items considering the three features separately. Therefore, a visual, term, and peer query is constructed based on the image examples and terms provided in the query (cf Section 7.2). The visual feature weight is then proportional to the sum of similarity scores between these queries and the query items. Let  $sim_t$  denote the overall term similarity of the query ( $sim_v$ ,  $sim_p$  similarly for the visual and peer similarity). Then  $w_t = \frac{sim_t}{sim_t + sim_v + sim_p}$ .

### E.5.1 Normalisation of Weights

One major obstacle of this approach is that the similarity scores are not readily comparable and therefore need to be normalised. We have experimented with two normalisation techniques in order to address this problem: a min-max normalisation and Gaussian normalisation.

In the min-max normalisation, scores are normalised to lie in the range from zero to one, by subtracting the minimum value and dividing it by the range between the minimum and maximum values. Thus the normalised score  $s'$  is obtained by  $s' = \frac{s - \min}{\max - \min}$ .

Another normalisation technique is Gaussian normalisation, which results in a set of scores whose mean is zero and standard deviation is one. The statistical mean,  $\mu$ , and standard deviation,  $\sigma$ , of the scores are calculated and each score is normalised by  $s' = \frac{s - \mu}{\sigma}$ .

The feature weights obtained with these two normalisation techniques are compiled in Tables E.46 and E.47. The scores received from the visual feature generally show less variability (are in a closer range) and therefore receive a very high weight after normalisation. The Gaussian normalisation technique addresses this issue, but still the visual feature is most emphasised. From the results presented in the main chapter, we know however that the visual feature is the worst performing while the textual feature performs best. Hence, we experimented with deemphasising the visual feature weight by dividing it by two and/or emphasising the textual feature weight by multiplying it by two.

The results for the various normalisation techniques are shown in Tables E.48–E.53 for the *ICG* and the baseline *IND*. Note that in the Voting Approach, the individual lists can be weighted during the list combination as described in Section 5.4.1.

### E.5.2 Performance of Adaptive Weights compared to Baselines

Choosing the Gaussian normalisation with the feature weight additionally divided by 2 ( $G(v)$  from above), we now compare these adaptive weighting strategy to the baseline performances without weighting and to the fixed weight sets as discussed in Section 7.5.4. The adaptive weighting strategy employed in *IND* is referred to  $IND_a$ , while the same strategy employed in  $ICG_p$  is referred to  $ICG_{w:a}$ .

Initially, the weighing strategies have been evaluated for 200 queries per task. No relevance feedback was performed. The average results are shown in Tables E.54–E.54.

Table E.46: Adaptive weights from individual indices using min-max normalisation

<b>weights</b>	<b>visual</b>	<b>text</b>	<b>peer</b>
Task1	0.59	0.28	0.13
Task2	0.45	0.37	0.18
Task3	0.45	0.42	0.13
Task4	0.46	0.37	0.17
Task5	0.51	0.38	0.11
Task6	0.61	0.23	0.16
Task7	0.90	0.05	0.05
Task8	0.91	0.04	0.04
Task9	0.57	0.29	0.14
Task10	0.67	0.30	0.03
Average	0.61	0.27	0.12

Table E.47: Adaptive weights from individual indices using Gaussian normalisation

<b>weights</b>	<b>visual</b>	<b>text</b>	<b>peer</b>
Task1	0.47	0.30	0.23
Task2	0.38	0.37	0.24
Task3	0.40	0.40	0.20
Task4	0.40	0.40	0.20
Task5	0.37	0.37	0.26
Task6	0.50	0.25	0.26
Task7	0.78	0.05	0.17
Task8	0.88	0.06	0.06
Task9	0.46	0.32	0.21
Task10	0.57	0.34	0.09
Average	0.52	0.27	0.19

The same adaptive weighting strategies are employed for the RF runs discussed in Section 7.5.2. These results are added to Tables E.24–E.35 in Section E.3 of this appendix.

Table E.48: P(10) for *IND* and *ICG* with adaptive weights under various normalisation techniques

P10	<b>IND</b>	MM	Gauss	G(v)	G(t)	G(v,t)	<b>ICG<sub>p</sub></b>	MM	Gauss	G(v)	G(t)	G(v,t)
Task1	0.96	0.34	0.65	0.91	0.81	0.66	0.98	0.98	0.99	0.98	0.98	0.97
Task2	1.00	0.87	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task3	0.96	0.81	0.97	0.99	1.00	1.00	0.97	0.97	0.97	0.98	0.99	0.99
Task4	0.91	0.72	0.90	0.57	0.30	0.30	1.00	1.00	1.00	1.00	1.00	1.00
Task5	0.94	0.49	0.95	0.97	0.97	0.97	0.99	0.98	0.99	0.99	0.99	0.99
Task6	0.98	0.28	0.49	0.96	0.89	0.98	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.10	0.11	0.10	0.10	0.10	0.10	0.45	0.43	0.43	0.43	0.43	0.43
Task8	0.08	0.08	0.08	0.08	0.08	0.08	0.26	0.25	0.25	0.25	0.25	0.25
Task9	0.43	0.21	0.35	0.40	0.33	0.31	0.66	0.67	0.67	0.67	0.68	0.68
Task10	0.18	0.11	0.10	0.15	0.15	0.18	0.20	0.21	0.20	0.20	0.20	0.20
Average	0.66	0.40	0.56	0.61	0.56	0.56	0.75	0.75	0.75	0.75	0.75	0.75

Table E.49: P(20) for *IND* and *ICG* with adaptive weights under various normalisation techniques

P20	<b>IND</b>	MM	Gauss	G(v)	G(t)	G(v,t)	<b>ICG<sub>p</sub></b>	MM	Gauss	G(v)	G(t)	G(v,t)
Task1	0.87	0.33	0.64	0.89	0.79	0.63	0.98	0.99	0.99	0.98	0.98	0.98
Task2	1.00	0.86	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task3	0.93	0.77	0.92	0.96	1.00	1.00	0.91	0.91	0.91	0.92	0.92	0.93
Task4	0.74	0.70	0.87	0.49	0.25	0.25	1.00	1.00	1.00	1.00	1.00	1.00
Task5	0.87	0.47	0.92	0.95	0.96	0.96	0.98	0.97	0.98	0.98	0.97	0.98
Task6	0.97	0.25	0.48	0.96	0.89	0.97	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.09	0.09	0.08	0.08	0.08	0.08	0.36	0.36	0.35	0.35	0.35	0.35
Task8	0.07	0.07	0.07	0.07	0.07	0.07	0.22	0.21	0.21	0.21	0.21	0.21
Task9	0.44	0.19	0.34	0.39	0.33	0.32	0.58	0.59	0.58	0.58	0.59	0.59
Task10	0.14	0.09	0.09	0.10	0.10	0.11	0.18	0.18	0.18	0.18	0.18	0.18
Average	0.61	0.38	0.54	0.59	0.55	0.54	0.72	0.72	0.72	0.72	0.72	0.72

Table E.50: P(50) for *IND* and *ICG* with adaptive weights under various normalisation techniques

P50	<b>IND</b>	MM	Gauss	G(v)	G(t)	G(v,t)	<b>ICG<sub>p</sub></b>	MM	Gauss	G(v)	G(t)	G(v,t)
Task1	0.68	0.30	0.62	0.81	0.77	0.55	0.98	0.99	0.99	0.98	0.98	0.98
Task2	0.99	0.85	0.99	0.99	1.00	1.00	0.97	0.97	0.97	0.97	0.97	0.98
Task3	0.68	0.44	0.49	0.75	0.99	0.99	0.65	0.68	0.67	0.68	0.72	0.73
Task4	0.53	0.73	0.93	0.33	0.15	0.15	1.00	1.00	1.00	1.00	1.00	1.00
Task5	0.76	0.45	0.90	0.88	0.91	0.91	0.94	0.94	0.94	0.95	0.94	0.95
Task6	0.85	0.22	0.45	0.92	0.88	0.91	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.06	0.07	0.06	0.06	0.06	0.06	0.24	0.24	0.24	0.24	0.24	0.24
Task8	0.06	0.06	0.06	0.06	0.06	0.06	0.16	0.16	0.16	0.16	0.16	0.16
Task9	0.38	0.18	0.36	0.37	0.33	0.25	0.52	0.52	0.52	0.52	0.52	0.52
Task10	0.10	0.08	0.07	0.07	0.07	0.07	0.16	0.16	0.16	0.16	0.16	0.16
Average	0.51	0.34	0.49	0.52	0.52	0.50	0.66	0.67	0.66	0.67	0.67	0.67

Table E.51: R(10) for *IND* and *ICG* with adaptive weights under various normalisation techniques

R10	<b>IND</b>	MM	Gauss	G(v)	G(t)	G(v,t)	<b>ICG<sub>p</sub></b>	MM	Gauss	G(v)	G(t)	G(v,t)
Task1	0.02	0.01	0.01	0.02	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Task2	0.10	0.08	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
Task3	0.10	0.09	0.10	0.11	0.11	0.11	0.10	0.10	0.10	0.11	0.11	0.11
Task4	0.04	0.03	0.04	0.03	0.01	0.01	0.05	0.05	0.05	0.05	0.05	0.05
Task5	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task6	0.03	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.03
Task7	0.01	0.01	0.01	0.01	0.01	0.01	0.05	0.05	0.05	0.05	0.05	0.05
Task8	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01
Task9	0.02	0.01	0.02	0.02	0.01	0.01	0.03	0.03	0.03	0.03	0.03	0.03
Task10	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Average	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04

Table E.52: R(50) for *IND* and *ICG* with adaptive weights under various normalisation techniques

R50	<b>IND</b>	MM	Gauss	G(v)	G(t)	G(v,t)	<b>ICG<sub>p</sub></b>	MM	Gauss	G(v)	G(t)	G(v,t)
Task1	0.03	0.01	0.02	0.03	0.03	0.02	0.04	0.04	0.04	0.04	0.04	0.04
Task2	0.19	0.17	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
Task3	0.20	0.16	0.20	0.21	0.21	0.21	0.20	0.20	0.20	0.20	0.20	0.20
Task4	0.07	0.07	0.08	0.05	0.02	0.02	0.10	0.10	0.10	0.10	0.10	0.10
Task5	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task6	0.05	0.01	0.02	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Task7	0.02	0.02	0.02	0.02	0.02	0.02	0.08	0.08	0.08	0.08	0.08	0.08
Task8	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Task9	0.04	0.02	0.03	0.03	0.03	0.03	0.05	0.05	0.05	0.05	0.05	0.05
Task10	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Average	0.06	0.05	0.06	0.06	0.06	0.06	0.08	0.08	0.08	0.08	0.08	0.08

Table E.53: R(100) for *IND* and *ICG* with adaptive weights under various normalisation techniques

R100	<b>IND</b>	MM	Gauss	G(v)	G(t)	G(v,t)	<b>ICG<sub>p</sub></b>	MM	Gauss	G(v)	G(t)	G(v,t)
Task1	0.11	0.05	0.11	0.14	0.14	0.09	0.18	0.18	0.18	0.18	0.18	0.18
Task2	0.70	0.54	0.62	0.65	0.94	0.94	0.62	0.63	0.62	0.63	0.64	0.65
Task3	0.54	0.28	0.27	0.64	0.97	0.97	0.46	0.52	0.49	0.51	0.56	0.57
Task4	0.21	0.33	0.43	0.12	0.04	0.04	0.44	0.44	0.44	0.44	0.44	0.44
Task5	0.08	0.05	0.11	0.09	0.09	0.09	0.11	0.11	0.11	0.11	0.11	0.11
Task6	0.18	0.05	0.11	0.22	0.22	0.20	0.25	0.25	0.25	0.25	0.25	0.25
Task7	0.06	0.06	0.05	0.05	0.05	0.05	0.19	0.18	0.18	0.18	0.18	0.18
Task8	0.03	0.03	0.03	0.03	0.03	0.03	0.07	0.06	0.06	0.06	0.06	0.06
Task9	0.14	0.07	0.14	0.14	0.12	0.08	0.21	0.21	0.21	0.21	0.21	0.21
Task10	0.04	0.04	0.04	0.03	0.03	0.03	0.07	0.07	0.07	0.07	0.07	0.07
Average	0.21	0.15	0.19	0.21	0.26	0.25	0.26	0.27	0.26	0.27	0.27	0.27

Table E.54: P(10) for *IND* and *ICG* with various feature weighting strategies

<b>P10</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.96	0.91	0.98	0.96	0.96	0.98	0.99	0.98
Task2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task3	0.96	0.99	0.97	0.97	0.97	0.98	0.96	0.98
Task4	0.91	0.57	1.00	1.00	1.00	1.00	1.00	1.00
Task5	0.94	0.97	0.99	0.99	0.99	0.99	0.99	0.99
Task6	0.98	0.96	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.10	0.10	0.45	0.07	0.47	0.45	0.44	0.43
Task8	0.08	0.08	0.26	0.21	0.28	0.26	0.25	0.25
Task9	0.43	0.40	0.66	0.66	0.65	0.67	0.66	0.67
Task10	0.18	0.15	0.20	0.21	0.20	0.21	0.20	0.20
Average	0.66	0.61	0.75	0.71	0.75	0.75	0.75	0.75

Table E.55: P(20) for *IND* and *ICG* with various feature weighting strategies

<b>P20</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.87	0.89	0.98	0.97	0.98	0.98	0.99	0.98
Task2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Task3	0.93	0.96	0.91	0.93	0.91	0.92	0.89	0.92
Task4	0.74	0.49	1.00	1.00	1.00	1.00	1.00	1.00
Task5	0.87	0.95	0.98	0.99	0.98	0.98	0.98	0.98
Task6	0.97	0.96	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.09	0.08	0.36	0.06	0.38	0.37	0.36	0.35
Task8	0.07	0.07	0.22	0.18	0.23	0.22	0.21	0.21
Task9	0.44	0.39	0.58	0.58	0.58	0.59	0.58	0.58
Task10	0.14	0.10	0.18	0.18	0.18	0.18	0.18	0.18
Average	0.61	0.59	0.72	0.69	0.72	0.72	0.72	0.72

Table E.56: P(50) for *IND* and *ICG* with various feature weighting strategies

<b>P50</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.68	0.81	0.98	0.98	0.98	0.98	0.99	0.98
Task2	0.99	0.99	0.97	0.98	0.97	0.97	0.97	0.97
Task3	0.68	0.75	0.65	0.67	0.65	0.69	0.63	0.68
Task4	0.53	0.33	1.00	1.00	1.00	1.00	1.00	1.00
Task5	0.76	0.88	0.94	0.95	0.94	0.95	0.94	0.95
Task6	0.85	0.92	1.00	1.00	1.00	1.00	1.00	1.00
Task7	0.06	0.06	0.24	0.05	0.25	0.24	0.24	0.24
Task8	0.06	0.06	0.16	0.14	0.17	0.17	0.16	0.16
Task9	0.38	0.37	0.52	0.52	0.52	0.52	0.52	0.52
Task10	0.10	0.07	0.16	0.16	0.16	0.16	0.16	0.16
Average	0.51	0.52	0.66	0.64	0.66	0.67	0.66	0.67



Table E.57: R(10) for *IND* and *ICG* with various feature weighting strategies

<b>R10</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task2	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
Task3	0.10	0.11	0.10	0.10	0.10	0.11	0.10	0.11
Task4	0.04	0.03	0.05	0.05	0.05	0.05	0.05	0.05
Task5	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Task6	0.03	0.02	0.03	0.03	0.03	0.03	0.03	0.03
Task7	0.01	0.01	0.05	0.01	0.05	0.05	0.05	0.05
Task8	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01
Task9	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.03
Task10	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Average	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04

Table E.58: R(50) for *IND* and *ICG* with various feature weighting strategies

<b>R50</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04
Task2	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
Task3	0.20	0.21	0.20	0.20	0.20	0.20	0.19	0.20
Task4	0.07	0.05	0.10	0.10	0.10	0.10	0.10	0.10
Task5	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Task6	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Task7	0.02	0.02	0.08	0.01	0.08	0.08	0.08	0.08
Task8	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Task9	0.04	0.03	0.05	0.05	0.05	0.05	0.05	0.05
Task10	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.02
Average	0.06	0.06	0.08	0.07	0.08	0.08	0.08	0.08

Table E.59: R(100) for *IND* and *ICG* with various feature weighting strategies

<b>R100</b>	<b>IND</b>	<b>IND<sub>a</sub></b>	<b>ICG<sub>p</sub></b>	<b>ICG<sub>pv</sub></b>	<b>ICG<sub>w:p</sub></b>	<b>ICG<sub>w:t</sub></b>	<b>ICG<sub>w:v</sub></b>	<b>ICG<sub>w:a</sub></b>
Task1	0.11	0.14	0.18	0.18	0.18	0.18	0.18	0.18
Task2	0.70	0.65	0.62	0.63	0.61	0.65	0.61	0.63
Task3	0.54	0.64	0.46	0.50	0.43	0.54	0.43	0.51
Task4	0.21	0.12	0.44	0.44	0.44	0.44	0.44	0.44
Task5	0.08	0.09	0.11	0.11	0.11	0.11	0.11	0.11
Task6	0.18	0.22	0.25	0.25	0.25	0.25	0.26	0.25
Task7	0.06	0.05	0.19	0.04	0.19	0.19	0.18	0.18
Task8	0.03	0.03	0.07	0.06	0.07	0.07	0.06	0.06
Task9	0.14	0.14	0.21	0.21	0.21	0.21	0.21	0.21
Task10	0.04	0.03	0.07	0.07	0.07	0.07	0.07	0.07
Average	0.21	0.21	0.26	0.25	0.26	0.27	0.26	0.27