

EGO: A Personalized Multimedia Management and Retrieval Tool

Jana Urban,* Joemon M. Jose†

*Department of Computing Science, University of Glasgow,
17 Lilybank Gardens, Glasgow, G12 8RZ, UK*

The problems of content-based image retrieval (CBIR) systems can be attributed to the *semantic gap* between the low-level data representation and the high-level concepts the user associates with images, on the one hand, and the time-varying and often vague nature of the underlying information need, on the other. These problems can be addressed by improving the interaction between the user and the system. In this article, we sketch the development of CBIR interfaces and introduce our view on how to solve some of the problems these interfaces present. To address the semantic gap and long-term multifaceted information needs, we propose a “*retrieval in context*” system, EGO. EGO is a tool for the management of image collections, supporting the user through personalization and adaptation. We will describe how it learns from the user’s personal organization, allowing it to recommend relevant images to the user. The recommendation algorithm is described, which is based on relevance feedback techniques. Additionally, we provide results of a performance analysis of the recommendation system and of a preliminary user study. © 2006 Wiley Periodicals, Inc.

1. INTRODUCTION

Most research in the field of information retrieval (IR) focuses on the performance of the retrieval system. In an effort to broaden the horizon of future search systems, researchers attribute increasingly more importance to the human–computer interaction aspect of IR. Based on studies of information-seeking behavior^{1,2} and user emotions and psychologies,³ new interfaces for search systems move into focus. Belkin has recently pointed out some grand challenges for information system design.⁴ 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) designing a system that supports a variety of interactions, and
- (2) personalizing the support of information interaction.

*Author to whom all correspondence should be addressed: e-mail: jana@dcs.gla.ac.uk.

†e-mail: jj@dcs.gla.ac.uk.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 21, 725–745 (2006)
© 2006 Wiley Periodicals, Inc. Published online in Wiley InterScience
(www.interscience.wiley.com). • DOI 10.1002/int.20157



In this line, the aim of our work is to design an adaptive content-based image retrieval (CBIR) system. The above issues act as the primary design goals for its development. We argue 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 for is a holistic application for retrieval, organization, and annotation, as well as any other activity a user might wish to perform. Moreover, the system should have the ability to adapt to the user by learning from user interaction. This facilitates personalization and improves system performance, because the system is able to learn from the knowledge and interaction of individual users.

In this article we introduce our system, EGO (Effective Group Organization), which is currently being developed with these principles in mind. The system addresses many of the shortcomings of traditional CBIR systems, which we identify and discuss in the following sections. In Section 2 we provide our motivations for researching novel interaction strategies. Section 3 reviews a representative selection of CBIR systems to trace their development. The remaining sections are devoted to the EGO system. Section 4 provides a detailed description of its interface and some of the implementation details. The system's evaluation is summarized in Section 5, followed by a list of features still in development in Section 6 and a concluding discussion in Section 7.

2. MOTIVATION

CBIR is an active research area focused on reducing the effort of making the vast amount of available images more manageable. Within this field one can observe a constant progress toward more and more intelligent systems.⁵ The benefits this provides are overshadowed by problems caused by the interaction with a typical CBIR user interface. The most pertinent questions are:

- What is the meaning of an image? It can be difficult to grasp the *meaning* of an image. As a work of art—similar to poems—an image's meaning cannot be pinpointed with universal consensus. Current CBIR technology has difficulties in extracting the major objects contained in an image, let alone its meaning. The trend to overcome the *semantic gap* between the system and the user is to learn semantic concepts in order to move closer to decoding meaning.^{6,7} Because there is no consensus on 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.^{7,8}
- How can the users be assisted in communicating their information need? The query formulation problem has emerged as an IR problem in general.⁹ The internal representation of documents is optimized for indexing efficiency and retrieval performance, but more often than not proves alien to the user. Hence, users may be taught to speak the language of the system, or even better the system taught to speak the language of its users.

Relevance feedback is an automatic process of improving the initial query based on relevance judgments provided by the user.¹⁰⁻¹³ 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, because they are only requested

to label the returned images as either relevant or not. However, it is still an ongoing research challenge to accurately learn the information need from the user based on a few relevance judgments.¹⁴

- How can the time-varying nature of information needs be modeled in the system? It is often the case that the initial idea of an image the user has in mind before starting a search session deviates from the final results they will choose.¹⁵ Whatever the reasons for this change, it shows that it is hard to guess an ideal query from the initial query and consequent relevance feedback. Rather it should be attempted to trace the actions over the course of the interaction with the system to detect changes in the information need.^{15,16}
- Although there are some approaches that model the dynamic nature of information needs,^{15,17} more work still needs to be done to this end. Current relevance feedback techniques treat the relevance judgments 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 judgments were *traced* rather than *collected*. In this way, new feedback can be compared to previous feedback, with the goal to detect changes over iterations. The proposed system, EGO, allows us to collect traces of actions. However, such methods have not been incorporated in the retrieval algorithms presented in this article.

To be able 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. Without the users' knowledge of the world and their superior visual system, CBIR system capabilities are rather limited. Moreover, user satisfaction greatly depends on subjective judgments of image contents as well as relevance. It is impossible to accommodate the huge diversity of users, yet systems can adjust to individual users by learning their preferences.

From the user's perspective, however, searching for and performing a selection of images is usually embedded in other tasks, and thus it is at least equally important to understand and capture the work flow.^{18,19} Therefore, a solution to accommodate the needs of users must be flexible, support multiple tasks, and allow exchanges or even seamless integration with other applications used for the work tasks. Moreover, the search process often takes place in a collaborative context, in which people work together and are inspired by and learn from each other's activities.

What is needed is a *holistic view* of personal image organization and retrieval. Satisfactory solutions to the uncertainty of image meaning, the query formulation problem, and the time-varying needs of the user can only be found in this environment. A "retrieval in context system" offers a great opportunity for learning, adaptation, and personalization. We now review the related work that has led us to the design of EGO.

3. RELATED WORK

In this section we study existing solutions for user interface support for CBIR. As we will see shortly, existing systems are predominantly search systems. They lack the ability to support the user in organizing their results in a meaningful and time-persistent way, thus losing the opportunity of learning and adapting to the user- and task-dependent context.

The interface is the mediator between the user and the search system. From the perspective of the user, it is the entry point to the system. A properly designed interface assists users with meaningful and intuitive ways of communicating their information need to the system and displays results in ways that stimulate users and enhance performance. In this section we introduce approaches for creating more meaningful result displays and review interfaces to see how they deal with the interaction metaphor. The major innovation for the former 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.^{20,21} These multidimensional displays aim at revealing relationships between images by visualizing mutual similarities between any two images. The axes either represent feature dimensions, such as color or textures, or are a result of dimension reduction methods, such as principal component analysis, mapping the cardinality of the feature space down to the two or three most discriminative dimensions. The goal of these visualization 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.²² has pointed to the benefits of a display organized by similarity for image browsing. The remainder of this section serves as an outline of the development of CBIR systems on the basis of their interface design.

1. QBIC: This is one of the earliest image retrieval systems with CBIR query facilities, developed by IBM.²³ QBIC supports the retrieval of images based on a number of primitive image features, including color, texture, and shape. The query component is the most important aspect of the interface. It allows the user to compose a query by drawing the rough shapes and choosing the color of objects 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. This interface requires intuitive and meaningful query composition facilities and is reliant on the user's ability to map from the high-level concepts they have in mind when querying to the low-level visual attributes the system understands for retrieval. It hardly assists the user in this task and does not learn from user interaction. To relieve the user from the query formulation problem, it additionally implements the *query-by-example* approach, in which the query features are automatically extracted from a user-supplied example image.

2. MARS: Recent systems have emphasized an interactive result refinement strategy made possible through relevance feedback. These systems usually implement the query-by-example approach to bootstrap the search. After the first iteration, the user is asked to specify the relevance of images in the result set. In MARS¹² this feedback can be given by changing the value of a slider of any image indicating the degree of (ir)relevance. The system responds with a new result set, which could be improved through a suitable learning strategy from the experience gained from the relevance feedback. This process is iterated until the user is satisfied with the results. Hence, interaction takes place between the system and the user in which the user responds to the result 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 for any of 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 a selected image has on the new results, and so forth. As a result, the user is forced to make decisions without having enough knowledge about the effects of his or her actions. Because the actions are usually irreversible, this can have detrimental effects on the perceived performance of the system.

3. **IMAGEGROUPER**: This system presents novel interaction strategies for CBIR.¹³ The major emphasis lies on group-based search, and this system combines the tasks of searching, annotating, and organizing images by groups. Image retrieval in this interface 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 results display from the workspace. The workspace serves as the organization and storage area. Images can easily be dragged from the results panel onto the workspace, and consequently be organized 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 organization are always visible. By dragging images around the workspace, that is, 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 labeling the groups (simple popup menu). Through a simple interaction strategy, the user gives relevance feedback information without having to think in terms of the system's internal representation. The organization of images into groups is more natural to the user and matches more closely to the process of accomplishing the task. The *trial-and-error* approach ensures that actions are reversible, which is necessary due to the inferior capabilities of current CBIR technology of matching human similarity judgments. Nonetheless, **IMAGEGROUPER** 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.

In summary, the representation of information has traditionally been confined to those suitable for retrieval. Thus, in image retrieval systems, the interface was focused on the provision of query components to specify the appropriate image features used for retrieval. However, to support the way information is used and managed, the interface has to include better result handling and personalization techniques. **IMAGEGROUPER** moves some way toward this goal; nevertheless, it is still mainly a *search* interface. Our aim is to develop a tool, **EGO**, that places emphasis on the long-term management and personalized access to the image (or multimedia) collection. The long-term usage provides additional search clues such

as usage histories of images and groups that should be combined with the low-level image features. Similar to the FETCH²⁴ system for organizing web documents, EGO provides the means to describe a long-term multifaceted information need. To achieve this, the user and the system interactively 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. By placing them on a workspace, the user leaves trails of her actions for her or others to inspect and follow. The process is incremental and dynamic: An organization is built up and changes by usage. A semantic organization emerges that reflects the user's mental model and the work tasks. These are the two most important influences on the organization of personal media recognized in Ref. 25: "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" (p. 764). As will be described in the following sections, EGO is a personalized "retrieval in context" system that allows users to effectively manage and search their images. 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.

4. EGO: EFFECTIVE GROUP ORGANIZATION

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 organization processes take place and are interleaved with each other. In this section we provide the general concepts of the system and describe how it would typically be used and how the system adapts based on the interaction.

The prototype interface for EGO is depicted in Figure 1. In EGO the user will be involved in an organization process, in which the user and the system interactively group images. As a starting point, the system provides a query panel (in the top left-hand corner in Figure 1), in which traditional query-by-example queries can be issued. The search results will be displayed in the panel beside. The user can then drag images from the results into groups on the workspace. This forms the start of the interactive group creation. For the currently selected group, the system provides recommendations of new images based on the images already contained in the group at certain time intervals (or on user request). The system's suggestions appear as a popup (yellow rectangle) below the currently 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.

4.1. The Interface

The EGO interface comprises a query editor, results display area, and workspace. By providing these facilities, different types of requirements are catered to,

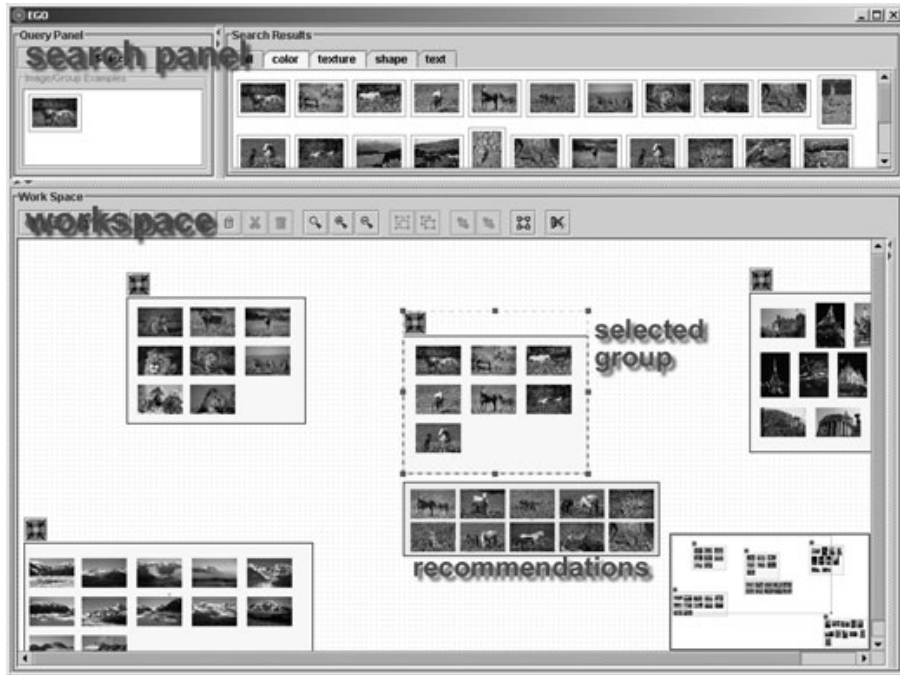


Figure 1. The annotated interface.

enabling the user to both search and organize results effectively. In the following, the main components are discussed in more detail.

1. *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. It should be noted that the size of all main components in the interface can be changed or even hidden on demand. In the query panel, the user can issue a search by choosing example query images. At the moment, the query-by-example strategy is the only one supported in EGO. Having argued for a flexible interface, other querying modes will be provided in the future. This might include keyword search (if annotation is available; see below in Section 6) or other query construction facilities based on semantic concepts (e.g., Ref. 26). The search results are displayed in the results panel beside the query construction component. It allows for different views of the results based on the features supported. The user can choose to view all results or results for only one feature category (color, texture, shape, or text, respectively). Again, we have only implemented a linear result display, but other visualization techniques, such as the ones mentioned above,^{15,20,21} would 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 organizing the collection.

2. *The Workspace*. Similar to the IMAGEGROUPER¹³ and FETCH²⁴ systems, the main component of the interface is the workspace panel provided in EGO. The workspace serves as an organization ground for the user to construct groupings of images. Groups can be created by right-clicking anywhere on the workspace, which opens a context menu in which the option can be selected. Traditional drag-and-drop techniques allow the user to drag images into a group and reposition the group on the workspace or the images within a group. It has to be noted that, unlike in conventional file systems, an image can belong to multiple groups simultaneously. The workspace is designed as a potentially infinite 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 in the bottom right-hand corner of the interface. It provides an overview, in which the whole workspace is visible, and a sense of location by marking the position of the current view. Additionally, a fish-eye view could be beneficial to provide a view of the whole organization and reduce clutter.

The act of grouping information is a natural means of managing information to support diverse, complex, and often simultaneous tasks.²⁴ This metaphor allows the user to resort to traditional problem-solving techniques, freeing her from the necessity of query formulation, which should ultimately create a natural and enhanced information-seeking environment. In addition, varying types of information need can be supported. Short-term needs can be satisfied by locating previously created groups that best match a user's need. (Although an automatic location of groups based on a query is still in the process of being implemented, the user can still manually locate groups in the workspace.) If there are no matching groups, the user can still resort to the traditional query facility. Furthermore, groups can be created and populated over time, reflecting long-term, time-varying needs.

To assist the user in this task, 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. 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 to the recommendation system that arise from the application. First, no image that is already contained in the group should be recommended again. Second, because organization 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 workspace (see Figure 1). So as not to burden the user, the number of recommended images is based on the standard cognitive limits of 7 ± 2 .²⁷

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. Because 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 we will describe next is based on learning the similarities of images. Future

work will improve recommendation quality by using contextual and usage information to better capture semantic information.

In brief, the advantages of EGO 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.
- The users leave trails of their actions behind, that the system can not only exploit for adaptation but that also other people can trace. Hence, EGO is ideal in a collaborative work context.

The next sections will deal with the implementation details of EGO: the way images are represented in the system and the matching and recommendation algorithms.

4.2. Image Representation

The images are represented according to the hierarchical object model proposed in Ref. 10. This model makes a distinction between the various visual features extracted. Rather than representing an image by a single stacked feature vector, it is composed of a set of feature vectors, one for each distinct feature implemented.

1. *Distance Measure.* The distance between an object x in the database and a given query representation q is computed in two steps. First, the individual feature distances g_i (for i in $1 \dots I$, where I is the number of features) are computed by the generalized Euclidean distance

$$g_i = (\vec{q}_i - \vec{x}_i)^T W_i (\vec{q}_i - \vec{x}_i) \tag{1}$$

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 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 $\vec{g} = [g_1, \dots, g_I]^T$ and a feature weight vector \vec{u}

$$d = \vec{u}^T \vec{g} \tag{2}$$

2. *Implemented Features.* We use the following six low-level color, texture, and shape features (feature dimension): average RGB (3), color moments (9),²⁸ co-occurrence (20), autocorrelation (25), edge frequency (25),²⁹ and invariant moments (7).³⁰

4.3. The Recommendation System

The recommendation system is based on a relevance feedback algorithm that attempts to learn the best query representation and feature weighting for a selected group of images. As far as the learning system is concerned, each group image is

regarded as a positive training sample. The proposed group-based learning scheme involves (1) updating the system's matching parameters, (2) creating a multipoint query representation and computing a ranked list for each query point based on the learned parameters, and (3) combining the individual result lists for the new recommendations. We will focus on each of these steps in more detail in the following.

1. *Learning the Feature Weights.* The parameter adaptation is achieved by finding new feature weights based on the feedback samples. We adopt the optimized framework for learning the feature weights proposed in Ref. 11. Due to the hierarchical object model, it distinguishes between intra- and interfeature weights. The optimal intrafeature component weights are given by an optimal feature space transformation matrix W_i . W_i is calculated as

$$W_i = \det(C_i)^{1/K_i} C_i^{-1} \quad (3)$$

where C_i is the *weighted covariance matrix* of the N positive examples according to the i th feature. W_i takes the form of a full matrix if N is larger than the dimensionality of the feature; otherwise only the diagonal entries are considered. The optimal interfeature weights $\vec{u} = [u_1, \dots, u_I]$ are the weights that best capture the intersimilarity between the training samples. The \vec{u}_i s are solved by

$$u_i = \sum_{j=1}^I \sqrt{\frac{f_j}{f_i}} \quad (4)$$

where $f_i = \sum_{n=1}^N g_{ni}$. The optimal intrafeature weights W_i and the optimal interfeature weights u are used in Equations 1 and 2, respectively, to calculate the total distance between a database object and the query representation.

2. *Computing the Query Representation.* Instead of calculating one overall query representation as in Ref. 11, our scheme relies on a form of query expansion. The chosen query representation for a group is a multipoint query,¹² 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.³¹

The number of clusters is a parameter in the hierarchical clustering algorithm, which is important in order to arrive at homogeneous clusters. Because we do not know how visually similar the images in the group are (or how homogeneous the group is), the ideal number of clusters can hardly be set in advance. Therefore, we employ the algorithm to automatically estimate this number presented in Ref. 32. The ideal number of clusters is often considered the "knee" (or point of maximum curvature) in an evaluation graph that plots the number of clusters versus some measure of quality or error of a clustering consisting of x clusters. In a hierarchical clustering approach, the value of the merge distance of the two most similar clusters at that point is a suitable measure of error. Salvador and Chan³² propose the L-Method to find the "knee" of the graph (or the ideal number of clusters) as the intersection between the two straight lines that most closely fit the curve. The L-Method is shown to be both an efficient and effective way to

automatically estimate the parameter for hierarchical clustering, unless you work with a very small data size.

The query points are selected as the image closest to each cluster centroid. Each query point is associated with a weight relative to the cluster size, that is, $w_i = n_i/N$, where n_i is the number of images in cluster i , and N is the total number of images in the group.

3. *List Combination*. When issuing the multipoint query to the system, a separate result list will be returned for each query point, and these need to be combined. An ongoing investigation of several combination strategies has led us to choose the rank-based *voting approach (VA)*. In this approach each query representative is treated as a voter producing its own individual orderings of candidates (images) based on the similarity to this query. The combined list is computed based on an adaptation of the *median rank aggregation* method as in Ref. 33, which sorts the database objects with respect to their median of the ranks they receive from the voters. This algorithm is very 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. To incorporate the query-point weights, w_i , determined above, 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, s_c , is accumulated: $s_c = \sum_{i=1}^l w_i$, where $l \leq L$. The majority criterion is fulfilled if $s_c > 0.5$ (this candidate is seen in the weighted majority of lists). Further, the lists are sorted in descending order of their weights, as this algorithm is sensitive to the sequence in which they are processed.

5. EVALUATION

So far, we have evaluated two aspects of the system:

- (1) a quantitative performance evaluation of the recommendation system based on low-level features and
- (2) a user study comparing the EGO interface to a traditional relevance feedback interface.

We will summarize the main results of these evaluations below. The user study is still ongoing, but we can already provide some intermediate trends here.

5.1. Performance of the Recommendation Algorithm

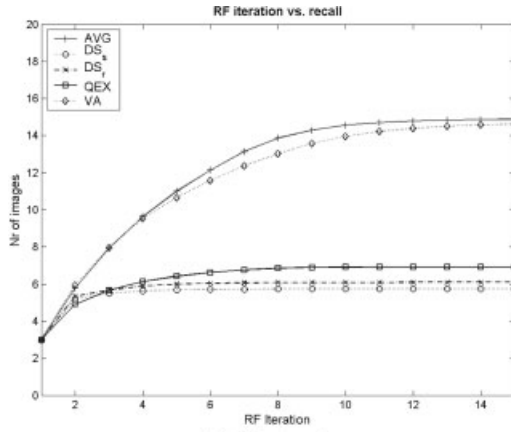
We have chosen the recommendation algorithm just described based on a performance evaluation of various alternative techniques, which can be found in Ref. 34. Experiments are conducted on photo CD 7 of the Corel image collection containing 23,796 images. Domain experts have organized the collection into 238 high-level semantic categories, from which we have selected 10 categories of 100 images each 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. User interaction is simulated by starting with groups of three randomly chosen images from a given category and performing relevance feedback from the top 10 returned images. In each feedback iteration the simulated user adds all relevant images to the current group. A query run terminates if no more relevant images can be found.

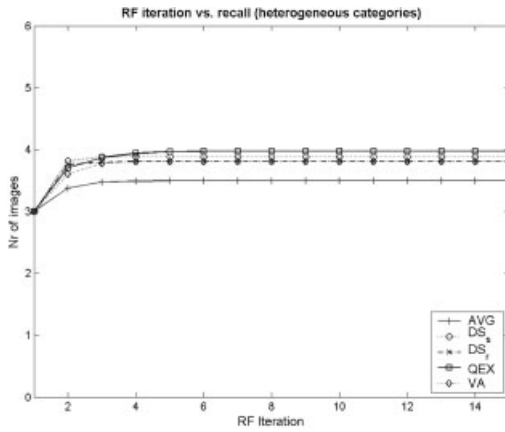
Results show that the proposed algorithm (*VA*) performs significantly better than the multipoint query technique (*QEX*) of Ref. 12, which simply combines the individual scores linearly, and also two variants of the Dempster–Shafer³⁵ combination (a ranked-based (DS_r) as well as a score-based (DS_s) version). *VA* is comparable with the baseline method (*AVG*) of Ref. 11, where only one query point is determined as the total average of all group images. Figure 2a shows the results for the simulated run just described. The graph depicts the average number of images found in each iteration, based on 50 queries per category.

Analyzing 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 (e.g., “roses”), whereas heterogeneous categories contain visually less similar images and are not easily distinguishable from other categories (e.g., “tribal people”). Our sample categories contained five of each. The results for the heterogeneous categories are displayed in Figure 2b, and Figure 2c depicts the homogeneous categories. It shows that *AVG* performs very well on homogeneous categories, whereas it performs slightly worse than the multipoint queries on heterogeneous categories. However, *VA* manages to capture a group’s query representation well in both circumstances. It should be noted that the majority of queries (about one-third) in heterogeneous categories do not manage to find any relevant images in the first iteration, where they start with only three images. *AVG* is particularly poor in almost 40% of these queries, compared to 35%, 29%, 28%, and 32% for *QEX*, DS_s , DS_r , and *VA*, respectively.

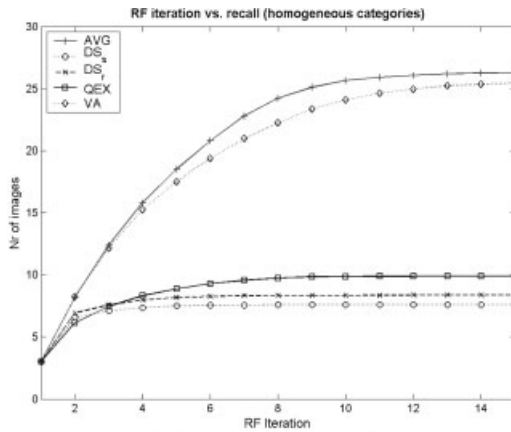
In the experiments described in Ref. 34, we have conducted an investigation of the various fusion methods based on (a) variations in the group-size, (b) using a weighting mechanism for the query points, and (c) the length of the individual lists to be combined. We can only summarize the overall conclusions from this study here. These included the observation that multipoint 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. On the other hand, *VA* has exhibited stable performance and is the only fusion method with performance comparable to *AVG* under various settings. In general, multipoint queries perform better than a single-point query in heterogeneous groups, where the images will indeed form multiple distinct clusters. In contrast, a single query point is sufficient to describe homogeneous groups. In addition, from a sufficiently large group size, a single query representation can be employed in any case.



(a) all categories



(b) heterogeneous categories



(c) homogeneous categories

Figure 2. Average number of images found per simulated relevance feedback iteration.

5.2. User Study

To test the usability of the EGO interface, we are in the process of running a task-oriented, user-centered evaluation.² Our experimental hypothesis is that EGO can be used for a variety of search tasks by supporting users in communicating their information need effectively. A traditional relevance feedback interface serves as the baseline system in this evaluation, because it is currently the most widespread interface in image retrieval research.

5.2.1. Experimental Methodology

In our evaluative study, we adopt a randomized within-subjects design, in which 24 searchers use the two systems. The independent variable is system type; two sets of values of a variety of dependent variables indicative of acceptability or user satisfaction are to be determined through questionnaires. To counterbalance the effect of learning from one system to the other, the order of the systems and tasks is rotated according to a Latin square design. For the purpose of the experiment we employ a subset of the Corel collection (CD 1 and CDs 4–6 of the Corel 1.6M data set), containing 12,800 photographs in total.

1. *Hypothesis.* We have broken up the experimental hypothesis into the following, more manageable, subhypotheses:

- The interplay of query components helps the user to select the most effective strategy for different search tasks or points in the search.
- Grouping search results on the workspace incites users to organize results for their search/work task, which in turn helps users to solve the task. (Organization as a secondary notation in support of memory/information seeking.)
- The iterative organization process helps to overcome the query formulation problem, because it is closer to real-life search strategies. In particular, the recommendation system can be used effectively to engage the user in an interactive organization.

2. *Systems.* We investigate user performance on EGO (referred to as *WS* for Workspace System in the following) compared to a standard relevance feedback interface (referred to as *CS* for Checkbox System). Both systems employ the same retrieval algorithms, which are described in Sections 4.2 and 4.3. For comparability, both the recommendation system in *WS* and the relevance feedback learning in *CS* use the *AVG* method described in Section 5.1. Every query-by-example search in *WS* and *CS* returns the top 100 results. In *CS*, the user can browse through the results and select relevant images for feedback, which will be used to update the internal retrieval parameters and return an improved set of 100 images. The recommendation system in *WS* offers the top 10 results on the workspace for a selected group.

3. *Tasks.* To place our participants in a real work task scenario, we use a simulated work task situation as conducted in Ref. 36. This scenario allows the users to evolve their information needs in just the same dynamic manner as such needs might be observed to do in participants' real working lives. As part of our experimental hypothesis, we want to investigate how EGO supports the user in a

variety of tasks. For this reason, we have created two different tasks, namely one resembling category search (i.e., users were asked to find as many images as possible from a given topic) and the other resembling an open-ended design task, where they have to search for and make a choice of three to five images. For the first task, we have chosen six topics, which differ in their complexity: three topics are very focused (“mountainous landscapes,” “elephants,” “tigers”) whereas the remaining three are broader, covering various aspects (“animals in the snow,” “African wildlife,” “underwater world”). The first task is set on both systems, *CS* and *WS* (with a different topic on each), whereas the latter one is performed on the *WS* system only. The objective behind setting these two very different tasks on *WS* is to get an idea of how the organization of search results is dependent on the nature of the task. A maximum time is set for all tasks to limit the total time spent on the experiment. This is 10 min for the category search, and 20 min for the design task.

4. *Participants and Procedure.* Because we want to test the system in a realistic usage scenario, we aim for a sample user population consisting of 24 post-graduate design students. The experimental procedure is as follows. We start with an introductory session, in which we explain the procedure and tasks and gather some information on their prior image search experience. In the first stage of the experiment, they are set two category search tasks, one to be performed on each system. Before commencing a search task, there is a training session to allow the users to become familiar with the system. A postsearch questionnaire is administered after each search task to get a glimpse of the search experience from the user’s perspective. The final stage comprises the design task on the *WS* system, followed by a questionnaire.

5.2.2. Results Analysis

So far, we have tested 12 users. The general trend is that people prefer *WS* over *CS*: six participants found *WS* more effective for the tasks compared to four who preferred *CS* and two who were undecided; eight participants, however, liked *WS* the best compared to three votes for *CS* and one undecided. All statistically significant results presented below were obtained using the nonparametric Wilcoxon paired-sample test.³⁷

1. *User Satisfaction.* The participants were asked to rate the system on a number of semantic differentials on a scale from 1 to 5. Figure 3 shows the means for these differentials. *CS* was considered significantly *easier* than *WS*, whereas *WS* was considered to be significantly more *stimulating*, *flexible*, and *novel*. The other differentials showed no significant differences between the systems.

In addition, people found *CS* significantly easier to *learn to use*, whereas there was no difference between *using* them. However, the display and features offered in *WS* were considered more *helpful*, *useful*, and significantly more *effective*. Furthermore, people thought *WS* helped them to explore the collection better, as well as analyze the task better. The responses to these two statements showed a significant level. They also considered the search process significantly more *interesting*.

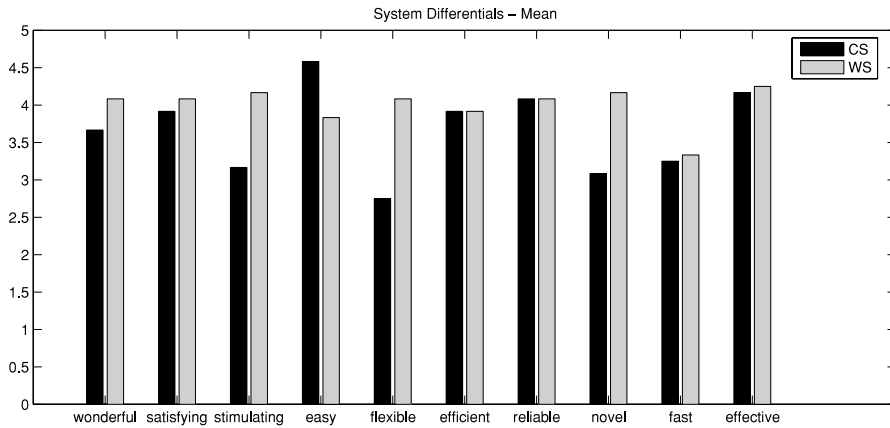


Figure 3. Semantic differential means for system part.

2. *Quantitative Results on Task Performance.* Although the participants' own judgment on usability and task performance favored *WS*, this is not reflected in the actual task performance. Table I depicts the average recall results per task. The number of relevant images found per task were generally higher in *CS* than in *WS*. Based on the analysis of the users' behavior (analyzing the usage logs) and their responses in the questionnaires, the reason for this appears to be that the selection of relevant images is much faster than the dragging of images. Also, users spent time on creating groups of images and moving images between groups in the *WS* system. Because we have set a maximum time limit, the number of images found was generally higher in *CS*, where users were not "distracted" by managing their search results.

In addition, the failure of the recommendation system has most probably contributed to these results. Analyzing the users' comments, we could identify that many people thought the recommendation system could potentially have been a very useful feature, but was not employed due to its inability to recommend relevant images. Our initial hypothesis, namely that the recommendation system helped to overcome the query formulation problem, could not be verified directly. On the other hand, when analyzing the way the users manually created the queries, we could observe an interesting pattern. They usually started off with a small number of

Table I. Total number of relevant images found per task.

System	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	All
CS recall	13.0%	15.8%	18.0%	24.8%	5.8%	8.5%	14.3%
WS recall	7.6%	8.8%	11.7%	15.2%	3.1%	7.2%	8.9%
AVG recall	10.3%	12.3%	14.8%	20.0%	4.5%	7.8%	11.6%
Total # relevant images	549	114	103	220	865	402	375.5

example images (from the given items and some initial results). Once they had created a group on the workspace that contained a number of 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 in that case. However, because this was not the case, they had to resort to the manual facility of finding more similar images for the group.

In any case, the users in this evaluation all started from scratch, that is, with no predefined groups or classes. We suppose that if the number of existing classes exceeds a certain threshold or “critical mass,” the usefulness of the recommendation system will eventually outweigh the additional effort required to create groups in the first instance.

3. *Analysis of Search Result Organization Depending on the Nature of the Task.* The objective of the design task set to round up the experiment is to find out how people make use of the groupings and organize their workspace. We have investigated the number of groups they created in the design task. In analyzing this number we could identify two different types of behavior: About half the people saved a large number of candidate images on the workspace organized into a variety of groups (between four and nine) reflecting different aspects of the task, before making the final selection; the others only added a small number of images on the workspace, mostly all in the same group. 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.

It is also very interesting to highlight differences in their behavior in the design task and the category search task. The average number of groups per task are 1.5, 1.0, 1.0, 7.0, 2.0, and 4.0 for topics 1–6 in the category search task, respectively, and 4.0 in the design task. We can clearly see a dependency between the number of groups and the nature of the task. Topics 1–3 are very focused (e.g., “mountainous landscapes”), whereas Topics 4–6 are multifaceted (e.g., “African wildlife”).

When we compare the responses to the questionnaires, we see that the participants had a (slightly) clearer idea of the kind of images that were relevant for the category search task, in both systems, compared to the design task. Also, the organization of images into groups seems to be more helpful in the design task than in the category search tasks. The average of the responses to the statement whether the system organization of images into groups helps them express different aspects of the task is 4.42 and 3.92 for the design task and category search task, respectively. The difference is even more pronounced comparing the different task groups for the category search tasks. The average response is 3.0 for the focused tasks and 4.83 for the more complex tasks. Based on this evidence, we can conclude that, although the organization of search results is helpful in general, it is dependent on the nature of the task.

5.2.3. Discussion

Overall, the users liked the way of selecting relevant images. They thought this was easier and quicker than dragging images on the workspace. At the same

time, they liked the ability to create groups and organize the search results. Among the advantages of *WS*, they stated its flexibility and greater control. The main disadvantage of *WS* was the poor quality of the recommendations. In *CS* people criticized that they did not have enough control over their search and that the interface was not very intuitive.

We believe these preliminary results back our claim that the possibility of placing search results on a workspace helps users organize their search results conceptually (a result that was reflected in the number of groups created for each task). Although the analysis of users' performance and their responses shows that relevance assessment might be easier to use (quicker), people felt less in control of their searches. Relevance assessment is less transparent to the user. It does not take into account changes in information need (or breaking up the information need in various aspects), as opposed to user-selected groupings saved on the workspace, where the user can dictate the various facets of the search task. In summary, this study has provided evidence that *EGO* helps users in communicating their information need. To improve its effectiveness, we still have to be able to improve the system's retrieval algorithm (and the features it is based on), and the recommendation system in particular. We will summarize some ideas of how this can be addressed in the next section.

6. FUTURE WORK

The development of *EGO* is still in its early stages. We are currently working on various aspects of the system that are needed to create a versatile, adaptive, personalized multimedia management tool. The following features are work in progress:

- **Enhanced search facilities:** As mentioned before, the search component could be enhanced by supporting additional query strategies and alternative result visualization techniques.
- **Textual annotations:** Textual annotation is desirable as a means to narrow the semantic gap. The interface has potential for a group-based annotation operation, which is both simple and efficient, as has been shown in *IMAGEGROUPER*.¹³ Starting from these user-provided annotations, a learning framework as in Ref. 7 or that of Ref. 38 can be employed to propagate labels.
- **Contextual information:** Collecting usage and contextual information is one of the main objectives of *EGO*. Usage information will include access counts of images and groups, change rates of groups, and so forth. The contextual information under consideration arises from the fact that images can belong to multiple groups. The context of the groups an image belongs to will be used in the recommendation system. Also, when used in a collaborative context with more than one user creating groups, we have to address the question of how to fuse the individual groupings. This work is currently in progress.
- **Linkage of groups:** So far, the system described has few browsing facilities. To remedy this shortcoming, we envisage developing a browsable workspace by creating links between groups. The links will be based on similarity of groups (both feature similarity as well as usage similarity) and will be visualized by arrows connecting two groups. The strength of a link (i.e., similarity score) will be reflected by the arrow's thickness

and length. The resulting network of linked groups provides the user the possibility to navigate the “group-space.”

- **Extension to other data formats:** Finally, an extension to multimedia data could be possible. This could include an integration of text documents based on the mechanisms studied in FETCH,²⁴ and video data by adopting more general representation strategies of multimedia retrieval tools.

7. DISCUSSION AND CONCLUSIONS

We have proposed and developed EGO, a tool for personalized multimedia management. In this article, we have described its interface and the recommendation system that allows it to adapt to its users. The mechanism underlying the recommendation system is based on learning multipoint queries and a voting approach used to combine the evidence from the query points. If we reflect back on the issues mentioned as the primary motivations for the design process, we can 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 recognize the importance of context. This context is not contained within the retrieval engine, but is determined by the tasks and work environment. To truly make an effort toward understanding the meaning, the image has to be placed in the wider context. In EGO the semantics of the images are conveyed through groupings that the user creates over the course of time. The organization resulting from the long-term interactions reflect the usage of the collection in the user’s context. From this organization, it is easier for the system to infer the intended semantic meaning.
- “How can the users be assisted in communicating their information need?” EGO does not require the user to think in terms of the system (e.g., how to formulate a query, how a search works, etc.) but engages in an interactive organization process that iteratively defines 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.
- “How can the time-varying nature of information needs be modeled in the system?” EGO invites users to create groups according to the multiple facets of their need. The groups are created and changed over multiple sessions, so that they capture aspects of the user’s long-term need. Organized on a 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.

To conclude, the design of EGO as a tool to create a task-specific organization of images reflecting an individual’s mental model will overcome many of the problems of traditional CBIR systems.

References

1. Bates MJ. The design of browsing and berrypicking techniques for the online search interface. *Online Review* 1989;13:407–424.
2. Ingwersen P. *Information retrieval interaction*. London: Taylor Graham; 1992.
3. Picard RW. *Affective computing*. Cambridge, MA: MIT Press; 1997.
4. Belkin NJ. Interface techniques for making searching for information more effective. In: CHI’2003, Best Practices and Future Visions for Search UIs: A Workshop; 2003.

5. Smeulders AW, Worring M, Santini S, Gupta A, Jain R. Content-based image retrieval at the end of the early years. *IEEE Trans Patt Anal Mach Intell* 2000;22:1349–1380.
6. Jeon J, Lavrenko V, Manmatha R. Automatic image annotation and retrieval using cross-media relevance models. In: *Proc Ann Int ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR'03)*; 2003.
7. Zhou XS, Huang TS. Unifying keywords and visual contents in image retrieval. *IEEE Multimedia* 2002;9:23–33.
8. Santini S, Gupta A, Jain R. Emergent semantics through interaction in image databases. *IEEE Trans Knowl Data Eng* 2001;13:337–351.
9. ter Hofstede AHM, Proper HA, van der Weide TP. Query formulation as an information retrieval problem. *Comput J* 1996;39:255–274.
10. Rui Y, Huang TS, Ortega M, Mehrotra S. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans Circ Syst Video Technol* 1998;8:644–655.
11. Rui Y, Huang TS. Optimizing learning in image retrieval. In: *IEEE Proc Conf on Computer Vision and Pattern Recognition (CVPR-00)*, June 2000, Los Alamitos: IEEE Computer Society Press; 2000. pp 236–245.
12. Porkaew K, Chakrabarti K, Mehrotra S. Query refinement for multimedia similarity retrieval in MARS. In: *Proc ACM Int Conf on Multimedia*, Orlando, Florida; 1999. pp 235–238.
13. Nakazato M, Manola L, Huang TS. ImageGrouper: Search, annotate and organize images by groups. In: *Proc Fifth Int Conf on Visual Information Systems (VISual 2002)*. Lecture Notes in Computer Science, Vol. 2314. Berlin: Springer Verlag; 2002. pp 129–142.
14. Zhou XS, Huang T. Relevance feedback in image retrieval: A comprehensive review. *ACM Multimedia Syst J* 2003;8:536–544.
15. Urban J, Jose JM, van Rijsbergen CJ. An adaptive approach towards content-based image retrieval. In: *Proc Third Int Workshop on Content-Based Multimedia Indexing (CBMI'03)*, Rennes, France, September 2003. pp 119–126.
16. White RW, Jose JM, Ruthven I. An approach for implicitly detecting information needs. In: *Proc Int Conf on Information and Knowledge Management (CIKM 2003)*, November 2003. pp 504–507.
17. Campbell I, van Rijsbergen CJ. The ostensive model of developing information needs. In: *Proc Int Conf on Conceptions of Library and Information Science*; 1996. pp 251–268.
18. Garber SR, Grunes MB. The art of search: A study of art directors. In: *Proc ACM Int Conf on Human Factors in Computing Systems (CHI'92)*; 1992. pp 157–163.
19. Markkula M, Sormunen E. End-user searching challenges indexing practices in the digital newspaper photo archive. *Inform Retriev* 2000;1:259–285.
20. Santini S, Jain R. Integrated browsing and querying for image databases. *IEEE Trans Multimedia* 2000;7:26–39.
21. Chen JY, Bouman CA, Dalton JC. Hierarchical browsing and search of large image databases. *IEEE Trans Image Process* 2000;9:442–455.
22. Rodden K, Basalaj W, Sinclair D, Wood K. Does organisation by similarity assist image browsing? In: *Proc ACM Int Conf on Human Factors in Computing Systems*; 2001. pp 190–197.
23. Flickner M, Sawhney H, Niblack W, Ashley J, Huang Q, Dom B, Gorkani M, Hafner J, Lee D, Petkovic D, Steele D, Yanker P. Query by image and video content: The QBIC system. *Computer* 1995;28:23–32.
24. Martin I, Jose JM. Fetch: A personalised information retrieval tool. In: *Proc Conf Recherche d'Information Assistée par Ordinateur (RIAO'2004)*; 2004. pp 405–419.
25. Kang H, Sheiderman B. Mediafinder: An interface for dynamic personal media management with semantic regions. In: *Proc ACM Int Conf on Human Factors in Computing Systems (CHI'2003)*; 2003. pp 764–765.
26. Lim JH. Explicit query formulation with visual keywords. In: *Proc Eighth Int ACM Conf on Multimedia (Multimedia-00)*, October 30–November 4, 2000. New York: ACM Press; 2001. pp 407–409.

27. Miller G. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psych Rev* 1956;63:81–97.
28. Stricker M, Orengo M. Similarity of color images. In: *Proc SPIE: Storage and Retrieval for Image and Video Databases*, February 1995. pp 381–392.
29. Sonka M, Hlavac V, Boyle R. *Image processing, analysis, and machine vision*, 2nd ed. Toronto: Thomson Engineering; 1998.
30. Hu MK. Visual pattern recognition by moment invariants. *IEEE Trans Inform Theor* 1962;8:179–187.
31. Theodoridis S, Koutroumbas K. *Pattern recognition*. San Diego, CA: Academic Press; 1999.
32. Salvador S, Chan P. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. Technical Report CS-2003-18. Melbourne, FL: Florida Institute of Technology; 2003.
33. Fagin R, Kumar R, Sivakumar D. Efficient similarity search and classification via rank aggregation. In: *Proc ACM SIGMOD Int Conf on Management of Data*. New York: ACM Press; 2003. pp 301–312.
34. Urban J, Jose JM. Evidence combination for multi-point query learning in content-based image retrieval. In: *Proc IEEE Sixth Int Symp on Multimedia Software Engineering (ISMSE'04)*, December 2004. pp 583–586.
35. Jose JM, Harper DJ. A retrieval mechanism for semi-structured photographic collections. In: Hameurlain A, Tjoa A, editors. *Proc Int Conf on Database and Expert Systems Applications*, Toulouse, France, September 1997. pp 276–292.
36. Jose JM, Furner J, Harper DJ. Spatial querying for image retrieval: A user-oriented evaluation. In: *Proc Ann Int ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR'98)*. New York: ACM Press; 1998. pp 232–240.
37. Zar J. Paired-sample testing by ranks. In: Kurtz B, editor. *Biostatistical analysis*. Englewood Cliffs, NJ: Prentice-Hall; 1984. pp 153–156.
38. Bissol S, Mulhem P, Chiaramella Y. Towards personalized retrieval. In: *Proc Second Int Workshop on Adaptive Multimedia Retrieval*; 2004. pp 89–102.