ENHANCING CBIR THROUGH FEATURE OPTIMIZATION, COMBINATION AND SELECTION

Xavier Hilaire and Joemon Jose

University of Glasgow
Department of Computing Science
17 Lilybank Gardens
Glasgow G12 8QQ, United Kingdom
{hilaire,jj}@dcs.gla.ac.uk

ABSTRACT

We present a Content-Based Image Retrieval (CBIR) method based on the combination and selection of several image features. The novelty of our approach over existing methods is threefold: we provide a statistical optimization of the similarity distance for each feature; we replace certain features by a selection in a non-linear expansion of them; and we perform a linear combination of the features. We demonstrate superior capabilities of our method in certain cases over support vector machines (SVM) on a COREL image collection.

1. INTRODUCTION

Content-based image retrieval (CBIR) is concerned with the problem of searching a database for images that match a user query. Good surveys covering the topic may be found in [13, 5]. Usually, the query includes a short textual description of the user expectation (for instance, "find pictures of Tony Blair"), and a small set of image examples. The system may or may not support user interaction during the search.

Two critical aspects inhering in all retrieval systems are feature selection, and similarity metric. These aspects also depend on application scenarios. In this paper, we shall examine the case where the query consists of a small set of image examples (20 at most), and where no interaction is permitted with the user during the search. Three assumptions therefore directed our work:

- The database is assumed to be large ($N > 100000$ images). The images are keyframes extracted from video files, and their resolution is therefore rather low (typically 350x240).
- In comparison, the initial query set of images is small ($n \approx 10$ per query). Such query images can be selected, for example, as part of the query in TRECVid or by the use of relevance feedback techniques.
- The set of query images are passed to the system all at once, and they are moreover assumed mutually exclusive one to each other (a query image relevant for its query is assumed irrelevant for any other one). In other words, we assume that negative sample images for a given query can be obtained by randomly sampling the set of other queries.

These assumptions are valid for searching in a database of keyframes as well as of still images. We simply found it more suitable to demonstrate its capabilities on COREL image database because of its diversity and its widespread use.

The rest of this paper is organized as follows. In section 2, we give a brief overview of our system and detail the three optimization steps it involves. In section 3, we compare the results obtained on COREL images with and without optimization and provide an analysis. We finally close the paper with a discussion in section 4.

2. PROPOSED APPROACH

2.1. Outline of the method

An overview of the system we propose is given in Fig. 1. It is quite conventional: from m example images $I_1, ..., I_m$, we predict an invariant f on N features $\phi_1, ..., \phi_N$; as a real-valued function, f is then used to sort the p images $K_1, ..., K_p$ of the database.

The originality of our approach, however, comes from the way we compute f, as detailed in Fig. 2. In a first step, we compute N functions $x_1, ..., x_N$, called similarity metrics, one per feature; these metrics are then optimized individually, and independently one to each other, so as to minimize the integral square classification error. In a second step, we expand the previous set of metrics to its power set,

The first author is now with ESIEE Paris, France.

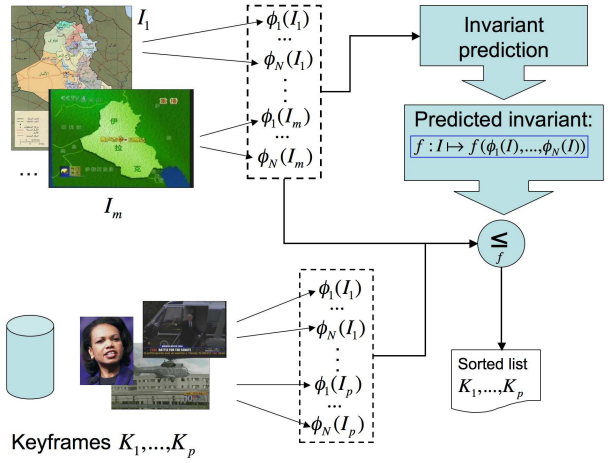

Fig. 1. An overview of the proposed CBIR system.

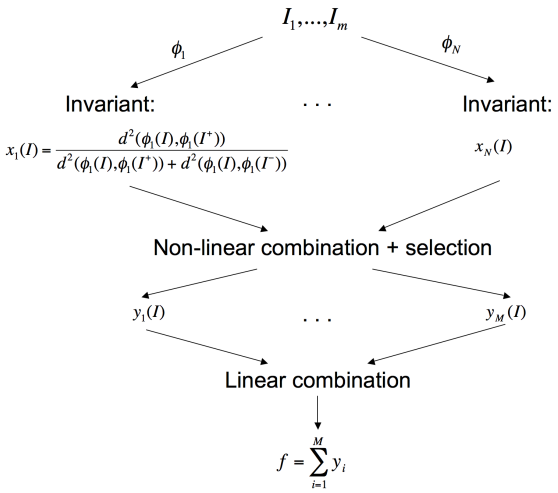

Fig. 2. Determination of the final image invariant.

and we select then $q \leq N$ most discriminating ones as new functions y_i. The last step performs a linear discriminant analysis of the y_i's, and determines the final invariant as the optimal linear combination of them.

In the next subsections, we detail each of these steps separately.

2.2. Step 1: Similarity metrics

One of the most serious problems encountered in image retrieval is probably the sparseness of the images once mapped in the feature space. For most of features, one may not expect to observe any clear separation between positive and negative samples, shall it be for training or testing data, as depicted in Fig. 3.

Indeed, points in the feature space may be so sparse, and the centers of classes so close one to each other that talking

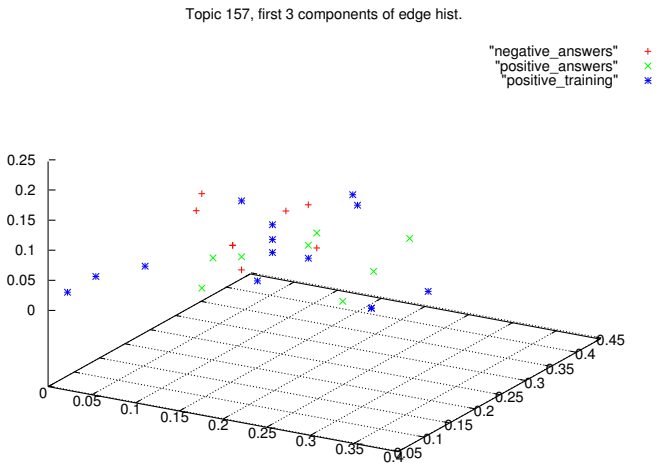

Fig. 3. Three first components of the MPEG-7 edge histogram descriptor obtained for different images (topic 157 of TRECVid 2005 training data).

about separability is meaningless. In such cases, it appears more natural to consider each point as a center of class itself rather than a member of any other one, and rely on a nearest neighbor selection. This is, at least, the conclusion which came out of our experimental work on TRECVid collections.

Following this idea, let I be a query image, $\mathcal{I}^+$ and $\mathcal{I}^-$ the sets of positive and negative answers to the current query, ϕ_i a feature, and finally I^+ and I^- the closest positive and negative neighbours of I in the feature space w.r.t the Euclidean distance:

$$I^+ = \arg \min_{X \in S^+} d(\phi_i(X), \phi_i(I))$$
$$I^- = \arg \min_{X \in S^-} d(\phi_i(X), \phi_i(I))$$

To assess the likelihood that an image I is a positive answer according to feature ϕ_i, one may think of a criterion such as

$$x_i(I) = 1 - \frac{d^2(\phi_i(I), \phi_i(I^+))}{d^2(\phi_i(I), \phi_i(I^+) + d^2(\phi_i(I), \phi_i(I^-))} \quad (1)$$

Despite its simplicity, this choice is still the best we have found in terms of performance of mean average precision. In particular, early experiments showed that it could overpass the performances yielded by a support vector machines (SVM), albeit the use of an arbitrary high number of slack variables, and regardless of the form assumed by the kernel.

The Euclidean distance in Eq. 1 can also be replaced by a weighted square distance: putting $X = \phi_i(I) - \phi_i(I^+)$, $Y = \phi_i(I) - \phi_i(I^-)$, this yields:

$$x_i(I) = 1 - \frac{\mathbf{X.D.X^t}}{\mathbf{X.D.X^t + Y.D.Y^t}} \quad (2)$$

where $\mathbf{D} = diag(a_1, ..., a_{d_i})$ is a diagonal matrix that will enable further optimization, and d_i is the dimension of feature i. If we consider the set of all positive (resp. negative) answers over all queries, we will want each of the corresponding x_i to be set as close as possible to 1 (resp. 0). This leads to minimize

$$\begin{aligned} & \sum_{I \in \cup \mathcal{I}^+} 1 - x_i(I) + \sum_{I \in \cup \mathcal{I}^-} x_i(I) \\ \text{subject to} \quad & a_i \geq 0, i = 1, ..., d_i \\ & \sum_{i=1}^{d_i} a_i = 1 \end{aligned} \quad (3)$$

Substituting Eq. 1 in Eq. 3, we can see that the problem may be rewritten as

$$\begin{aligned} \text{minimize} \quad & \sum_i \frac{\mathbf{Z_i.D.Z_i^t}}{\mathbf{T_i.D.T_i^t}} \\ \text{subject to} \quad & a_i \geq 0, i = 1, ..., d_i \\ & \sum_{i=1}^{d_i} a_i = 1 \end{aligned} \quad (4)$$

where the coefficients in $\mathbf{Z_i}$ and $\mathbf{T_i}$ are defined as functions of the images, and do not depend on any a_i. Equation 4 is a linear fractional program that can be solved using the techniques presented in [11] and [4], and we refer the reader to these articles for further information.

2.3. Step 2: Non-linear expansion

Without putting the objective of the previous section into question, one may however wonder whether the choice of the x_i's in Eq. 1 is the most profitable one in term of separability. Indeed, Eq. 4 optimizes the x_i's *individually*; it does not attend to determine whether the use of a combination of them would reduce the misclassification error. So, we may consider using afterwards a product of x_i's rather than x_i's directly.

A first effort in this direction was made by Ishikawa et al. with Mindreader [6], in which feature similarity is expressed as a generalised Euclidean distance – this merely implies that shall it had to use our x_i's as features, Mindreader would consider a product of them of up to order 2. However, as pointed out by Rui in [10], a major drawback with their approach is that it requires the learning of $O(n^2)$ parameters for n features, which is often unrealistic given the small number of samples usually available. This is the well-known data overfitting problem.

To circumvent this without significant loss of performance, we suggest to resort to the following reduction method.

Step 1. Let $X = (x_1, ..., x_n)^t$ be a realization of the scores x_i of the features, as obtained in Eq. 1. X being given, we define $Y(X) = (y_1, ..., y_{2^n-1})^t$ as

$$y_k = \left(\prod_{i \in b(k)} x_i \right)^{\frac{1}{|b(k)|}}, \quad k = 1, ..., 2^n - 1 \quad (5)$$

where $b(k) = \{i \in \mathbb{N} : k \mod 2^{i-1} \neq 0\}$ is the set of indices of bits of k which are 1's. For instance, if

$$X = (x_1, x_2, x_3, x_4)^t \quad (6)$$

then

$$\begin{aligned} Y(X) = & (x_1, ..., x_4, \sqrt{x_1x_2}, ..., \sqrt{x_1x_4}, ..., \sqrt{x_3x_4}, \\ & \sqrt[3]{x_1x_2x_3}, ..., \sqrt[3]{x_2x_3x_4})^t \end{aligned}$$

Put simply, $Y(X)$ represents an ordered version of the "normalized" power set of X.

Step 2. Build $B = \{1, ..., 2^n - 1\}$, set V to the empty list, and build L as the list of the indices of the y_k's variables ranked by decreasing integral square error (ISE):

$$ISE_k = \sum_{I \in S^-} y_k^2(I) + \sum_{I \in S^+} (1 - y_k(I))^2$$

Step 3. Let y_h be the head of L. Add h to V, and update B as $B = B \setminus b(h)$.

Step 4. Let y_r be the head of L. If $b(r) \cap b(h) \neq b(r)$ then remove y_r from L and repeat this step.

Step 5. If V has less than N elements and B is not empty, then repeat to step 3, else stop.

To illustrate how the procedure works, let us suppose that X is defined as in Eq. 6. At step 2, we have $B = \{1, 2, ..., 15\}$, and the we may obtain $L = \{13, 9, 4, 3, 6, ...\}$, meaning that $h_{13} = \sqrt[3]{x_1x_3x_4}$, $h_9 = \sqrt{x_1x_4}$, $h_4 = x_3$, $h_3 = \sqrt{x_1x_2}$, $h_6 = \sqrt{x_2x_3}$ are the first top performing variables. Step 3 sets $V = \{13\}$, and reduces B to $\{2\}$. Step 4 will reduce L to $\{3, 6, ...\}$. A second iteration to step 3 will set V to $\{13, 3\}$ and steps 4 and 5 will leave it unchanged. So in this case, the procedure exits with only two variables: $y_1 = h_{13}$ and $y_2 = h_3$.

There are two underlying assumptions in this procedure: (i) the variable h_r which appears in the head of L is always the best possible one amongst all in term of ISE; (ii) h_r is a product of variables defined on a set $b(r)$, so all possible products defined on a subset of $b(r)$ are suboptimal and will not bring anything better for the subset $b(r)$.

This procedure determines $q \leq N$ new variables $y_i, ..., y_q$ from the x_i's. The choice of N as an upper bound has, for

instance, no known theoretical foundation; it is only made to avoid the problem of the curse of dimensionality.

2.4. Step three: linear discriminant analysis

The last step of our approach consists in combining the y_i's in a way that best explains the output (relevant or irrelevant document) given the input data (positive or negative sample).

An obvious method one may think of in this case is to resort to linear discriminant analysis: we write the final decision function f as a linear combination of the y_i's:

$$f = \sum_{i=1}^{q} b_i y_i \tag{7}$$

and seek for the b_i coefficients which maximize the Fisher criterion for the two classes of positive and negative sample images. Let $H = \{h_{ij}\}$ be the $m \times q$ matrix representing the value obtained by image I_i of the collection for the function y_j: $h_{ij} = y_j(I_i)$. Without loss of generality, let us suppose that the p first lines of H represent the scores of relevant images to the current query.

It can be shown (see, e.g, chap. 3 of [8]) that the problem is equivalent to minimize

$$f(b) = \frac{\mathbf{bCC^t b^t}}{\mathbf{bTb^t}} \tag{8}$$

where $\mathbf{b} = (b_1, ..., b_q)^t$, and $\mathbf{C} = (c_1, ..., c_q)$ is a column-vector such as

$$c_j = \frac{\sqrt{m(m-p)}}{m}(h_j^+ - h_j^-)$$

where $h_j^+ = \frac{1}{p}\sum_{i=1}^{p} h_{ij}$ and $h_j^- = \frac{1}{m-p}\sum_{i=p+1}^{m} h_{ij}$ are the respective averages of scores obtained by the relevant and irrelevant images for feature j. $\mathbf{T} = (\mathbf{t_{ij}})$ is the general covariance matrix of H, such as

$$t_{ij} = \frac{1}{q}\sum_{k=1}^{q}(h_{ki} - \bar{h_i})(h_{kj} - \bar{h_j})$$

where $\bar{h}_k = \frac{1}{m}\sum_{i=1}^{m} h_{ik}$. Since the value of f does not depend on the b_i's but on their ratio, the problem can be reformulated as

$$\begin{array}{ll} \text{maximize} & \mathbf{bCC^t b^t} \\ \text{subject to} & \mathbf{bTb^t} = 1 \end{array} \tag{9}$$

Introducing a Lagrange multiplier λ, we obtain from Eq. 9

$$\mathbf{CC^t b} = \lambda \mathbf{Tb}$$

and since $\mathbf{T}$ may be assumed non singular in the general case:

$$\mathbf{T^{-1}CC^t b} = \lambda \mathbf{b} \tag{10}$$

meaning that λ is an eigenvalue. If we now premultiply the last equation by $\mathbf{C^t}$, we get

$$(\mathbf{C^t T^{-1} C})\mathbf{C^t b} = \lambda(\mathbf{C^t b}) \tag{11}$$

and identifying member to member, we can see that $\lambda = \mathbf{C^t T^{-1} C}$ is indeed the only possible eigenvalue to Eq. 10, so the solution $\mathbf{b}$ to Eq. 10 is unique. The corresponding eigenvector readily follows:

$$\mathbf{b} = \mathbf{T^{-1}C} \tag{12}$$

3. EXPERIMENTAL EVALUATION

3.1. Setup

We conducted experiments on the COREL image collection [2], which contains altogether about 23800 images, distributed on 7 CDROMs. Among those available to us, we found that two were of particular interest for the variety their contents: CD1 and CD4.

Each CDROM contains a number of directories, each representing a specific topic with exactly 100 example images. Results presented in the literature very often refer to this collection; however, most of authors do not adopt COREL's classification to favour their own, and it is often difficult, if not impossible, to reconstruct the set of images they used from the information they provide.

To give a fair idea of the performances of our system, *we did adopted COREL's classification*. We therefore assumed one query per directory, and evaluated all of them. The minor discrepancies we observed w.r.t the official collection are:

- official queries 119 (Los Angeles), 121 (Denmark), and 125 (Coins and Currency) missing on our CD
- queries 129 (Germany) and 114 (Mountains) found on our CD, but not found on the official list of queries[1]

Table 1 presents the topics for CD4. CD1 contains more than 50 topics, which we did not feel useful to detail as they are very similar in nature to those of CD4.

3.1.1. Our method

The method described in this paper has been implemented in C++ on an G5 Apple PowerMac. We resorted to a number of image descriptors which come with MPEG-7's experimental code (XM), so as to dispose of at least one descriptor per visual feature class (color, shape, and texture,

[1] See http://elib.cs.berkeley.edu/corel/disknum_diskname_list.txt

qid	Description	qid	Description
103	Wildlife of the Galapagos	115	Orchids of the World
104	North American Deer	116	France
105	Lions	117	Pacific Coasts
106	Wildlife of Antarctica	118	Greek Isles
107	Elephants	120	Hong Kong
108	Tigers	122	Israel
109	Foxes & Coyotes	123	Backyard Wildlife
110	Wolves	124	Flowering Potted Plants
111	Cities of Italy	126	Austria
112	Rhinos & Hippos	127	North American Wild flowers
113	Arabian Horses	128	Russia, Georgia & Armenia
114	Mountains	129	Germany

Tab. 1. Table of topics for the COREL CD4 collection

qid	CL	CS	DC	EH	HS	MT	TX	MAP100	MAP100/M
103	33.2	10.2	4.7	30.4	9.5	15.7	21.0	17.3	33.4
104	5.9	7.8	2.6	19.6	3.2	15.6	4.2	9.1	16.7
105	22.7	1.5	25.5	25.3	4.5	27.6	4.7	17.8	36.9
106	25.0	11.6	27.1	23.6	49.1	27.2	2.4	27.3	36.2
107	40.4	5.1	21.1	15.5	15.0	40.7	25.1	23.0	42.2
108	18.1	10.8	17.9	22.3	7.0	14.8	14.3	15.2	36.7
109	16.4	8.1	11.3	12.1	3.3	16.1	4.8	11.2	17.3
110	26.3	3.3	19.2	6.2	10.2	10.5	11.6	12.6	20.4
111	12.6	17.6	5.3	5.3	17.9	6.0	7.8	10.8	8.1
112	23.5	3.1	9.1	17.8	10.1	27.6	10.7	15.2	34.1
113	63.7	0.0	30.4	32.4	46.8	52.0	27.1	37.6	73.6
114	37.6	4.0	34.0	49.2	16.2	19.4	26.7	26.7	53.5
115	29.1	1.4	19.1	30.2	19.4	41.3	24.5	23.4	46.7
116	10.1	4.3	5.6	3.6	0.4	10.1	9.5	5.7	9.2
117	9.6	14.1	12.5	31.0	27.1	10.5	19.1	17.5	30.9
118	8.2	7.4	6.9	5.3	5.7	19.1	4.3	8.8	25.0
120	6.1	15.3	4.7	4.5	5.0	15.0	5.9	8.4	9.4
122	3.9	21.1	7.5	3.9	1.2	13.0	7.8	8.4	9.8
123	20.4	5.6	21.9	15.1	8.4	24.8	19.1	16.0	22.8
124	28.5	34.4	24.5	30.8	6.0	47.7	11.9	28.6	61.0
126	14.1	17.0	4.7	6.2	2.6	24.9	14.4	11.6	26.6
127	35.9	1.9	18.0	7.6	18.4	39.5	15.6	20.2	54.5
128	5.4	2.5	10.1	5.1	6.4	21.8	12.2	8.6	15.2
129	1.6	2.1	1.7	3.5	2.7	5.4	4.9	2.8	9.8
MAP100	20.8	8.8	14.4	16.9	12.3	22.8	12.9	16.0	30.4

Tab. 2. Individual and overall mean average precisions (AP) in percent obtained on the COREL CD4 collection with 20 training samples. CL = color layout, CS = contour shape, DC = dominant color, EH = edge histogram, HS = histogram, MT = color moments, TX = texture. MAP100 = AP with no optimization; MAP100/M = AP obtained with our optimization method. See text for detailed explanations.

as described in [9], for example). These are: the color layout (CL), the contour shape (CS), the dominant color (DC), the edge histogram (EH), and the texture (TX) descriptors. Default parameters from MPEG-7 XM were kept for all of them.

We also resorted to a color histogram (HS) in La*b* space with the number of bins advocated in [3], and raw color moments up to order 9 in the same color space. These two were added for their known performances [12]. For each query, 20 images where used at learning stage, and in the case of our method, 10 for solving Eq. 4, and 10 others to estimate $\mathbf{T}$ and optimize the Fisher criterion, as stated in Eq. 12.

3.1.2. SVM Light

To give a comparison of our results to those obtained with a reference method, we ran experiments using the SVM$_{light}$, provided by Joachims [7]. The setup was identical to that used for our method – 20 images at learning stage for each query, which both method accessed through the same files.

3.2. Results and analysis

3.2.1. Used metric

We evaluate performances using the mean average precision at N documents, defined as follows:

$$MAP(N) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{i}\sum_{j=1}^{i}\delta(j) \qquad (13)$$

where $\delta(i) = 1$ if document i is relevant, and 0 otherwise. Note that this definition is slightly different from that used by NIST's trec_eval program, which, keeping the above notation, would be:

$$MAP_{trec}(N) = \frac{1}{\sum_{i=1}^{N}\delta(i)}\sum_{i=1}^{N}\frac{\delta(i)}{i}\sum_{j=1}^{i}\delta(j) \qquad (14)$$

For a definition of average precision over all documents according to TREC, see, e.g., part 2.1-II-B of [1]. This means, for example, that if a system retrieves 4 documents at ranks 1, 2, 4, and 7, our measure would raise

$$MAP(7) = \frac{1/1 + 2/2 + 2/3 + 3/4 + 3/5 + 3/6 + 4/7}{7} \approx 0.727$$

whereas NIST's trec_eval tool would raise

$$MAP_{trec}(7) = \frac{1/1 + 2/2 + 3/4 + 4/7}{4} \approx 0.83$$

We wished to use such a measure because it integrates the precision over *all* documents, not only on those found relevant. It is generally more penalizing than MAP_{trec}, but, in our opinion, it is also more accurate to reflect the performances of a given system. In particular, at the end of an arbitrary "cut" at N documents (say, N=100), the last retrieved documents are very often *not* relevant; in this case, the MAP_{trec} measure makes no difference on the final result, regardless of the number of irrelevant documents after the last relevant one.

3.2.2. Results

Our method

Table 2 gives the average precision in percents obtained at 100 documents for each topic of the CD4 collection. The $MAP100/M$ column is $MAP(100)$ as defined in Eq. 13, and is obtained with our optimization method. $MAP100$ follows the same definition, except that it has obtained by taking an equal weight on the features. $MAP100/M$ is obtained with our optimization method.

Three essential observations can be made from this table. Firstly, performances largely outstand those obtained with equal weights on features, which is indeed no surprise given the variability of performance for each visual feature.

Secondly, performances per query are very uneven: for example, topics pertaining on animals or flowers generally obtain an $MAP100/M$ over 30%, whereas those related to towns or countries rarely exceeds 10%. Indeed, this difference of average precision is directly linked to the difference of homogeneity in contents: pictures showing a tiger or an elephant will be very similar in visual contents; not those showing landscape from Germany (Fig. 4).

Thirdly, the MAP obtained with our method generally overpasses the top performing visual feature for each query, but it may not always be the case (for example, query 104 for EH, query 106 for HS, query 120 for MT, etc.). Indeed, it may happen that a product of variables behaves better (in terms of ISE) than a single one when the number of available samples is too small, the trend reversal being observed when the number of samples increases. We may therefore still encounter data overfitting in a few cases.

Fig. 5 shows the sensitivity of our method w.r.t the number of training samples, but the last step (linear discriminant analysis) had to be skipped. Indeed, the MAP reported here has been obtained with taking equal weights on features in step 3, as to avoid singularities in computing the inverse of $\mathbf{T}$ in Eq. 12 as the number of samples becomes very low – we preferred this to resorting to a pseudo-inverse in that case. The behaviour is nevertheless quite acceptable as the number of samples decreases, although it remains unstable below 6.

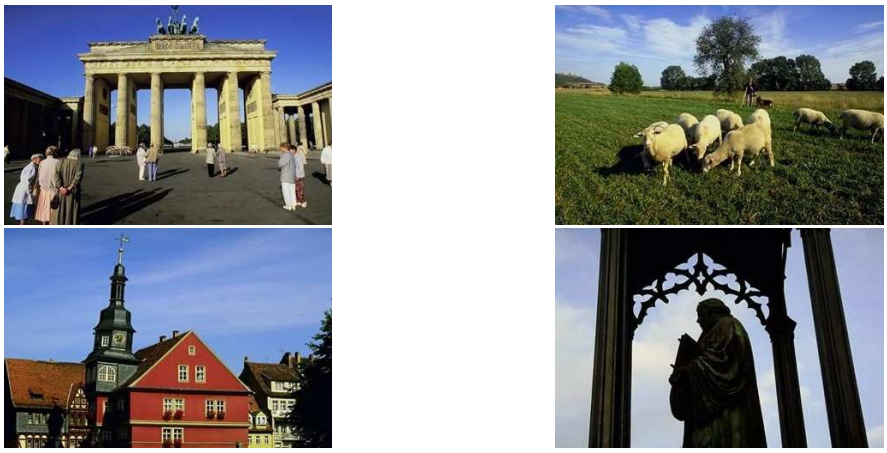

Fig. 4. A few result images for a same query (129, COREL CD4, "Germany").

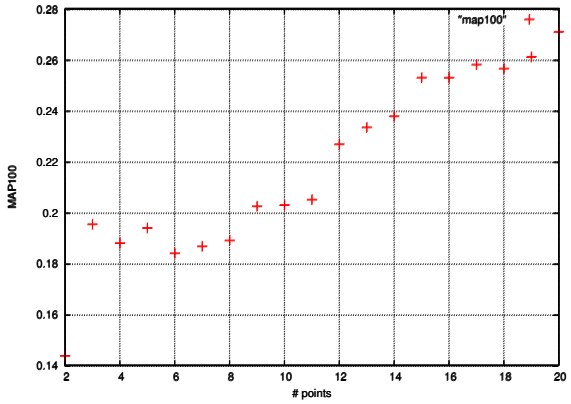

Fig. 5. Mean average precisions at 100 documents obtained on the COREL-CD4 collection with different number of samples (2–20).

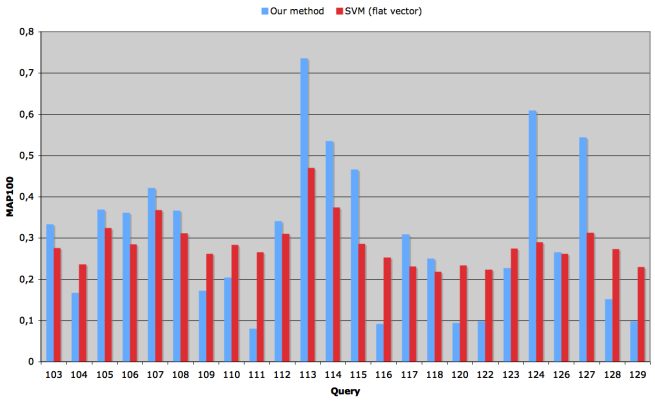

Fig. 7. Comparison of overall performances of both methods.

SVM$_{light}$

Results and behaviour of SVM$_{light}$ on our data set were a bit unexpected. First, we observed a great sensitivity of the results to both the type of kernel and its parameterization. We found that the best overall results were obtained with a radial basis function kernel:

$$K(x_i, x_j) = \exp - \frac{||x_i - x_j||^2}{2\sigma^2}$$

in which the σ parameter was properly estimated from the training data set *for each feature*. To provide a final result combining all the features' vectors, we resorted to the well-known "flat" vector strategy, which consists in stacking all the features' vectors in a single one for each sample. Again, the results were rather good, as the estimated σ parameter from the training set turned out to be close enough to the one estimated over the ground truth. However, *we did notice huge variations of the results as σ moves away from the optimal value*, which confirms that SVM would not be applicable if the number of training samples becomes too small.

On the other side, when provided enough samples, SVM behaved in a more stable way than our method, which suggests a better insensitivity to the difficulty of the queries – although the variations are still discernible. Figure 6 reports the MAP100 obtained for individual features with SVM$_{light}$, whereas Fig. 7 compares the overall performance of both methods. One may notice the difference of behaviour.

4. CONCLUSION

We have proposed an improved CBIR method based on several optimization steps (on similarity in feature space and on the linear combination of similarity metrics) and on a procedure which substitutes certain metrics by some more promising ones. So far, the method is used in a passive

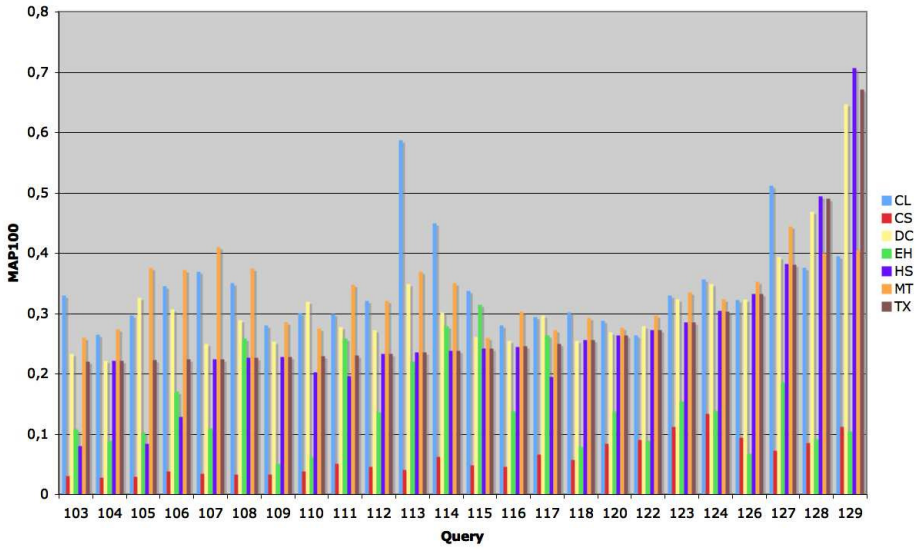

Fig. 6. Individual performances of SVM per feature.

learning framework, and its discriminatory power has been verified in this case on a database built from COREL images.

We believe that further work includes two important steps: the determination of the lower bound below which to use equal weights, and the adaptation of the method to the framework of active learning.

Acknowledgments

This research was supported by the European Commission under contract FP6-027122-SALERO.

5. REFERENCES

[1] Appendix: Common evaluation measures. In *Proceedings of TREC 2006*, http://trec.nist.gov/pubs/trec15/appendices.

[2] Corel image database, http://elib.cs.berkeley.edu/corel.

[3] R. Brunelli and O. Mich. Histogram analysis for image retrieval. *Pattern Recognition*, 34:1625–1637, 2001.

[4] A. Cambini. On optimizing a sum of ratios. Technical report, University of Pisa, Department of Statistics and Applied Mathematics, 1987.

[5] R. Datta, J. Li, and J. Wang. Content-based image retrieval: approaches and trends of the new age. In *7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 253–262, 2005.

[6] Y. Ishikawa, R. Subramanya, and C. Falousos. Mindreader: Query databases though multiple examples. In *Proceedings of the 24th VLDB Conference, New York, USA*, 1998.

[7] T. Joachims. Making large-Scale SVM Learning Practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[8] L. Lebart, A. Morineau, and K. M. Warwick. *Multivariate descriptive statistical analysis*. John Wiley and Sons, 1984.

[9] M. Lew. *Principles of Visual Information Retrieval*. Springer-Verlag, 2001.

[10] Y. Rui. Optimizing learning in image retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina (USA)*, volume 1, pages 236–243, 2000.

[11] S. Schaible. Fractional programming. Handbook of Global Optimization, pages 495–608. Kluwer Academic Publishers, Boston-London, 2005.

[12] M. Stricker and M. Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases III*, volume SPIE–2420, pages 381–392, San Diego/La Jolla, USA, 1995.

[13] R.C. Veltkamp and M. Tanase. A survey of content-based image retrieval systems. In *Content-Based Image and Video Retrieval*, pages 47–101. Kluwer Academic Publishers, 2002.