published in the Proceedings of VISual 2000, Lyon, France, November 2000
available at <URL:http://www.cis.hut.fi/picsom/publications.html>

Evaluating the Performance of Content-Based Image Retrieval Systems[1]

Markus Koskela, Jorma Laaksonen, Sami Laakso, and Erkki Oja

Laboratory of Computer and Information Science,
Helsinki University of Technology,
P.O.BOX 5400, Fin-02015 HUT, Finland
{markus.koskela,jorma.laaksonen,sami.laakso,erkki.oja}@hut.fi

Abstract. Content-based image retrieval (CBIR) is a new but in recent years widely-adopted method for finding images from vast and unannotated image databases. CBIR is a technique for querying images on the basis of automatically-derived features such as color, texture, and shape directly from the visual content of images. For the development of effective image retrieval applications, one of the most urgent issues is to have widely-accepted performance assessment methods for different features and approaches. In this paper, we present methods for evaluating the retrieval performance of different features and existing CBIR systems. In addition, we present a set of retrieval performance experiments carried out with an experimental image retrieval system and a large database of images from a widely-available commercial image collection.

1 Introduction

The recent development of computing hardware has resulted in a rapid increase of visual information such as databases of images. To successfully utilize this increasing amount of data, we need effective ways to process it. Content-based image retrieval (CBIR) utilizes the visual content of images directly in the process of retrieving relevant images from a database. The task of developing effective products based on CBIR has, however, proven to be extremely difficult. Due to the limitations of computer vision, the current CBIR systems have to rely only on low-level features extracted from the images. Therefore, images are typically described by rather simple features characterizing the color content, different textures, and primitive shapes detected in them.

Unfortunately, quantitative measures for the retrieval performance of an image retrieval system, or any single feature used in the process, are problematic due to the subjectivity of human perception. As each user of a retrieval system has individual expectations, there does not exist a definite right answer to an image query. Also, there exist no widely accepted performance assessment methods.

[1] This work was supported by the Finnish Centre of Excellence Programme (2000-2005) of the Academy of Finland, project New information processing principles, 44886.

The discriminating powers of features also vary with different types of images. Therefore, we need to use a comprehensive set of diverse images to evaluate the performance of different features and the systems employing them. Due to the lack of standard methods in this application area, the Moving Picture Experts Group (MPEG) has also started to work on a content representation standard for multimedia information search, filtering, management and processing called MPEG-7 or formally "Multimedia Content Description Interface" [9], expected to be completed in 2001.

2 Methods for Evaluating Retrieval Performance

The standard evaluation methods in information retrieval are precision and recall, which have been used also in evaluating different CBIR approaches. Although being objective measures of retrieval effectiveness, they suffer from certain shortcomings when used to evaluate image retrieval applications. CBIR applications are generally based on ranked lists of retrieved images. Therefore, precision and recall are usually calculated using a prespecified cutoff number M. Unfortunately, these measures are very sensitive to the choice of M. The simple adaptation of these methods also neglects the provided rank information [5].

The need for applicable evaluation criteria still persists. Reliable evaluation methods for performance would enable us to rate and rank different approaches and methods, and to find the best ones for a given task. For this purpose, a number of measures for evaluating image retrieval performance are presented in this section. First, a method for assessing the ability of various visual features to reveal image similarity is discussed. Second, a method for evaluating performance of whole retrieval systems is described.

Consider a database $\mathcal{D}$ containing a total of N images. First, we gather subsets of images which can be regarded as portraying a selected topic from the database. Such a set of images is called an image class $\mathcal{C}$. Image classes can include, for example images of aircraft, buildings, nature, human faces, etc. The process of gathering these classes is naturally arbitrary, as there are no distinct and objective boundaries between different image classes. If the images in the database already have reliable textual information about the contents of the images, it can be used directly; otherwise, manual classification is needed.

Now we have a database $\mathcal{D}$ containing a total of N images, and a class $\mathcal{C} \subset \mathcal{D}$ with $N_\mathcal{C}$ relevant images. The *a priori* probability $\rho_\mathcal{C}$ of the class $\mathcal{C}$ is

$$\rho_\mathcal{C} = \frac{N_\mathcal{C}}{N} \ . \tag{1}$$

An ideal performance measure should be independent of the *a priori* probability and the type of images in the image class.

2.1 Observed Probability

Let the images of $\mathcal{D}$ be ordered so that each has a unique index. For each image $I \in \mathcal{C}$ with a feature vector $\mathbf{f}^I$, we calculate the Euclidean distance $d_{L_2}(I, J)$ of

$\mathbf{f}^I$ and the feature vectors $\mathbf{f}^J$ of the other images $J \in \mathcal{D} \setminus \{I\}$ in the database. Then, we sort the images based on their ascending distance from the image I and store the indices of the images in a $(N-1)$-sized vector $\mathbf{g}^I$. We now have a vector $\mathbf{g}^I$ for each $I \in \mathcal{C}$ containing a sorted permutation of the images in $\mathcal{D} \setminus \{I\}$ based on their increasing Euclidean distance to I. By g_i^I, we denote the ith component of $\mathbf{g}^I$.

Next, for all images $I \in \mathcal{C}$, we define a vector $\mathbf{h}^I$ as follows

$$\forall i \in \{0, \ldots, N-2\} : \; h_i^I = \begin{cases} 1\,, & \text{if } g_i^I \in \mathcal{C}\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{2}$$

The vector $\mathbf{h}^I$ thus has value one at location i, if the corresponding image belongs to the class $\mathcal{C}$. As $\mathcal{C}$ has $N_{\mathcal{C}}$ images, of which one is the image I itself, each vector $\mathbf{h}^I$ contains exactly $N_{\mathcal{C}} - 1$ ones. In order to perform well with the class $\mathcal{C}$, the feature extraction should cluster the images I belonging to $\mathcal{C}$ near each other. That is, the values $h_i^I = 1$ should be concentrated on the small values of i.

We can now define the *observed probability* p_i:

$$\forall i \in \{0, \ldots, , N-2\} : \; p_i = \frac{1}{N_{\mathcal{C}}} \sum_{K \in \mathcal{C}} h_i^K \,. \tag{3}$$

The observed probability p_i is a measure of the probability that an image in $\mathcal{C}$ has as the i:th nearest image, according to the feature extraction $\mathbf{f}$, another image belonging to the same class.

In the optimal case, $p_i = 1$ if $i \leq N_{\mathcal{C}} - 2$, and $p_i = 0$ if $i > N_{\mathcal{C}} - 2$. This is equivalent to the situation where all the images in class $\mathcal{C}$ are clustered together so that the longest distance from an image in $\mathcal{C}$ to another image in the same class is always smaller than the shortest distance to any image not in $\mathcal{C}$. On the other hand, the worst case happens when the feature $\mathbf{f}$ completely fails to discriminate the images in class $\mathcal{C}$ from the remaining images. The observed probability p_i is then close to the *a priori* $\rho_{\mathcal{C}}$ for every value of $i \in [0, N-2]$.

2.2 Forming Scalars from the Observed Probability

The observed probability p_i is a function of the index i, so it cannot easily be used to compare two different feature extractions. Therefore, it is necessary to derive scalar measures from p_i to enable us to do such comparisons. As large values of p_i with small values i and small values of p_i with large values i correspond to good discriminating power, the scalar measure should respectively reward large values of p_i when i is small and punish large values of p_i when i is large.

We chose to use three figures of merit to describe the performance of individual feature types. First, good features should have high observed probabilities for the very first indices. Therefore it is justifiable to use a local performance measure based only on the first indices. A simple and straightly derived measure can be calculated as the average of the observed probability p_i for the first n retrieved images, i.e.:

$$\eta_{\text{local}} = \frac{\sum_{i=0}^{n-1} p_i}{n} \tag{4}$$

The η_{local} measure obtains values between zero and one. If $n \leq N_{\mathcal{C}} - 2$, it yields the value one in the optimal case. For η_{local} to measure local performance, a suitable value for the parameter n could be approximately 1–5% of the size of the database. With the η_{local} measure, figures near one can be obtained even though the classes were globally split into many clusters if each of these clusters are separate from the clusters of the other classes. This measure can thus be regarded as an indicant for the local separability of a given image class with a certain feature extraction method. Note that η_{local} is dependent on the *a priori* probability $\rho_{\mathcal{C}}$ of the image class.

As mentioned above, the η_{local} measure may give high values even if the images are scattered into many small clusters in the feature space. This suggests using an appropriate weighting function, which would take global clustering into consideration. A general method to construct a scalar Φ from a function $f(x)$ is to use a weighting or kernel function $h(x)$ and integrate the product of $f(x)$ and $h(x)$ over the whole input space as in

$$\Phi = \int_{-\infty}^{\infty} f(x)\, h(x)\, dx \ . \tag{5}$$

By selecting a suitable weighting function $h(x)$ we can set the measure Φ to fit to our purposes.

In this case, the weighting function should reward large values of p_i in small indices and punish large values of p_i in large indices. One such weighting function is obtained by first defining a DFT-based complex-valued function $P(u)$ as follows:

$$P(u) = \sum_{i=0}^{N-2} p_i\, h(i) = \sum_{i=0}^{N-2} p_i\, e^{j\pi i/(N-1)} \ . \tag{6}$$

Now, the weighting function $h(i)$ rotates in $N-1$ steps $\varphi = 0$ to $\varphi = \frac{N-2}{N-1}\pi$, which equals the upper half of the unit circle.

Finally, a global figure of merit, η_{global}, is obtained by considering the real part of $P(u)$ and normalizing the result with $N_{\mathcal{C}} - 1$. Thus,

$$\eta_{\text{global}} = \frac{\text{Re}\left\{\sum_{i=0}^{N-2} p_i\, e^{j\pi i/(N-1)}\right\}}{N_{\mathcal{C}} - 1} \ . \tag{7}$$

Also η_{global} attains values between zero and one. It favors observed probabilities that are concentrated in small indices and punishes for large probabilities in large index values. To achieve high performance values, the features should cluster all the images belonging to $\mathcal{C}$ near each other, preferably into a single cluster.

The third value of merit, η_{half}, measures the total fraction of images belonging to $\mathcal{C}$ found when only the first half of the p_i sequence is considered,

$$\eta_{\text{half}} = \frac{\sum_{i=0}^{N'} p_i}{N_{\mathcal{C}} - 1} \ , \tag{8}$$

where $N' = \text{int}(N/2)$. The η_{half} measure obviously yields a value one in the optimal case and a value half with the *a priori* distribution of images.

Overall, for all the three figures of merit, η_{local}, η_{global}, and η_{half}, the larger the value the better the discrimination ability of the feature extraction is.

2.3 τ Measure

Measuring feature performance is essential in order to find the set of features for CBIR applications which on the average perform as well as possible. Still, even more important task is to measure performance of different CBIR applications and approaches. In this section, we present one quantitative figure, denoted as the τ measure, for performance of CBIR systems. The measure can be applied to systems utilizing the relevance feedback [11] approach in some form.

Content-based image searches can be divided at least into three categories [4]: target search, category search, and open-ended browsing. In target search, the goal is to find a specific image from the database. The user may or may not know if the image actually exists in the database. In category search, the user is interested in one or more images from a category. This is a harder problem, as images in semantic categories can be visually very dissimilar. In the third search type, open-ended browsing, the user is just browsing the database without a specific goal in mind. System performance with the browsing approach is hard to measure objectively.

With the τ measure, it is assumed that the user is facing a target search task from a database $\mathcal{D}$ for an image I belonging to class $\mathcal{C} \subset \mathcal{D}$. Before the correct image is found, the user guides the search by marking all shown images which belong to class $\mathcal{C}$ as relevant images. Then, the τ value measures the average number of images the system retrieves before the correct one is found. The τ measure resembles the "target testing" method presented in [4], but instead of relying on human test users, the τ measure is fully automatic.

The τ measure is obtained by implementing an "ideal screener", a computer program which simulates the human user by examining the output of the retrieval system and marking the images returned by the system either as relevant (positive) or non-relevant (negative) according to whether the images belong to $\mathcal{C}$. This process is continued until all images in $\mathcal{C}$ have been found. The queries can thus be simulated and performance data collected without any human intervention.

For each of the images in the class $\mathcal{C}$, we then record the total number of images presented by the system until that particular image is shown. From this data, we form a histogram and calculate the average number of shown images needed before a hit occurs. In the optimal case, the system first presents all images in $\mathcal{C}$. The optimal value for the average number of images presented before a particular image in $\mathcal{C}$ is thus $\frac{N_{\mathcal{C}}}{2}$.

The τ measure for class $\mathcal{C}$ is then obtained by dividing the average number of shown images by the size of the database, N. The τ measure yields a value

$$\tau \in [\frac{\rho_{\mathcal{C}}}{2}, 1 - \frac{\rho_{\mathcal{C}}}{2}] \tag{9}$$

where $\rho_{\mathcal{C}} = \frac{N_{\mathcal{C}}}{N}$ is the *a priori* probability of the class $\mathcal{C}$. For values $\tau < 0.5$, the performance of the system is thus better than random picking of images and, in general, the smaller the τ value the better the performance.

The number of new images the system presents each round, i.e., the cutoff number, is denoted as M. The selection of this parameter has also some effect on the resulting τ value.

3 PicSOM

The PicSOM image retrieval system is a framework for research on algorithms and methods for content-based image retrieval. The system is designed to be open and able to adapt to different kinds of image databases, ranging from small and domain-specific picture sets to large general-purpose image collections. The features may be chosen separately for each specific task and the system may also use keyword-type textual information for the images, if available. Image retrieval with PicSOM is based on querying by pictorial examples (QBPE) [2], which is a common retrieval paradigm in CBIR applications. With QBPE, the queries are based on reference images shown either from the database itself or some external location. The user classifies these example images as relevant or non-relevant to the current retrieval task and the system uses this information to select such images the user is most likely to be interested in. The accuracy of the queries is then improved by relevance feedback [11] which is a form of supervised learning adopted from traditional text-based information retrieval. In relevance feedback, the previous human-computer interaction is used to refine subsequent queries to better approximate the need of the user.

In PicSOM, the queries are performed through a WWW-based user interface and the queries are iteratively refined as the system exposes more images to the user. PicSOM supports multiple parallel features and with a technique introduced in the PicSOM system, the responses from the used features are combined automatically. This is useful, as the user is not required to enter weights for the used features. The goal is to autonomously adapt to the user's preferences regarding the similarity of images in the database.

In this section, a brief overview of the PicSOM approach is presented. A more detailed description of the system can be found in our previous papers, e.g. [8, 10]. The PicSOM home page including a working demonstration of the system is located at *http://www.cis.hut.fi/picsom*.

3.1 The Self-Organizing Map

The image indexing method used in PicSOM is based on the Self-Organizing Map (SOM) [6]. The SOM defines an elastic net of points that are fitted to the input space. It can thus be used to visualize multidimensional data, usually on a two-dimensional grid. The SOM consists of a regular grid of neurons where a model vector is associated with each map unit. The map attempts to represent all the available observations with optimal accuracy using a restricted set of models. At

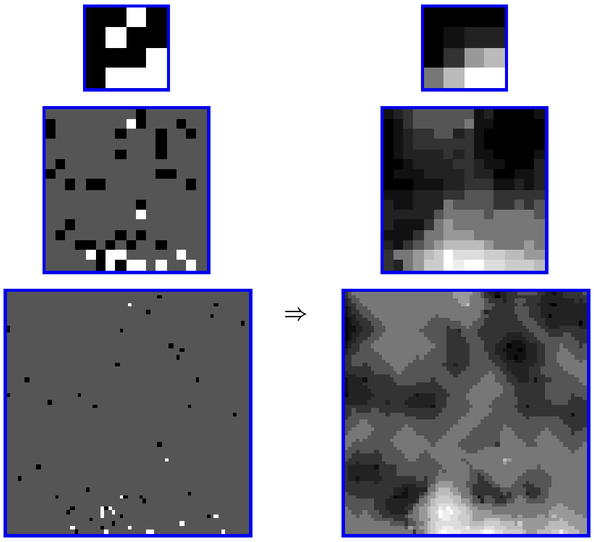

Fig. 1. An example of converting the positive and negative map units to convolved maps in a three-level TS-SOM. Map surfaces displaying the positive (white) and negative (black) map units are shown on the left. The resulting convolved maps are shown on the right.

the same time, the models become ordered on the grid so that similar models are close and dissimilar models far from each other. The PicSOM retrieval method can be described as a SOM-based implementation of relevance feedback.

In order to achieve a hierarchical representation of the image database and to alleviate the computational complexity of large SOMs, we use a special form of the SOM namely the Tree Structured Self-Organizing Map (TS-SOM) [7]. The TS-SOM is used to represent the database in several hierarchical two-dimensional lattices of neurons. Each feature is used separately to train a corresponding TS-SOM structure. As the SOM organizes similar feature vectors in nearby neurons, the resulting map contains a representation of the database with similar images according to the given feature located near each other. The tree structure of the TS-SOM, on the other hand, provides several map levels forming a set of SOMs with different resolutions.

3.2 Image Querying

In the beginning of a new query, the system presents the user the first set of reference images which are uniformly picked from the top levels of the TS-SOMs in use. The user then selects the subset of images which match her expectations best and to some degree of relevance fit to her purposes. Query improvement is achieved as the system learns the user's preferences from the selections made on the previous rounds.

The system marks the images selected by the user with a positive value and the non-selected images with a negative value in its internal data structure. These values are then summed up in their best-matching SOM units in each of the TS-SOM maps. Each SOM level is then treated as a two-dimensional matrix formed of values describing the user's responses to the contents of the map unit. Finally, the map matrices are low-pass filtered with symmetrical convolution masks in order to spread the user's responses to the neighboring units which, by presumption, contain images that are to some extent similar to the present ones. Starting from the SOM unit having the largest convolved response value, PicSOM retrieves from the database the image whose feature vector is nearest to the weight vector in that unit. If that image has not been shown to the user, it is marked to be shown on the next round. This process is continued with the second largest value and so on until a preset number of new images have been selected. This set is then presented to the user.

The conversion from the positive and negative marked images to the convolutions in a three-level TS-SOMs is visualized in Figure 1. First, a TS-SOM displaying the positive map units as white and negative as black is shown on the left. These maps are then low-pass filtered and the resulting map surfaces are shown on the right. It is seen that a cluster of positive images resides at the lower edge of the map.

A typical retrieval session with PicSOM consists of a number of subsequent queries during which the retrieval is focused more accurately on images resembling the positive example images. These queries form a list (or a tree of queries if the user is allowed to go back to previous query rounds and proceed with a different selection) in which all the queries contain useful information for the retrieval system.

3.3 User Interface

The PicSOM user interface used in our current WWW-based implementation in a midst of an ongoing query is displayed in Figure 2. First, the three parallel TS-SOM map structures represent three map levels of SOMs trained with RGB color, texture, and shape features, from left to right. The sizes of the SOM layers are 4×4, 16×16, and 64×64, from top to bottom. Below the convolved SOMs, the first set of images consists of images selected on the previous rounds of the retrieval process. These images may then be unselected on any subsequent round, thus changing their value from positive to neutral. In this example, a query with a set of images representing buildings selected as positive is displayed. The next images, separated by a horizontal line, are the 16 best-scoring new images in this round obtained from the convolved units in the TS-SOMs.

4 Experiments

We evaluated a set of features and the PicSOM approach with a set of experiments using an image collection from the Corel Gallery 1 000 000 product [3].

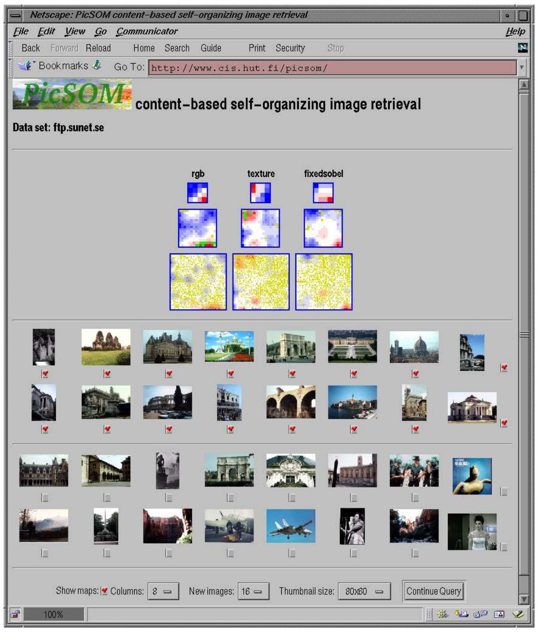

Fig. 2. The PicSOM user interface.

The collection contains 59 995 photographs and artificial images with a very wide variety of subjects. All the images are either of size 256×384 or 384×256 pixels. The majority of the images are in color, but there are also a small number of grayscale images.

4.1 Settings

Five different feature extraction methods were applied to the images and the corresponding TS-SOMs were created. The TS-SOMs for all features were sized 4×4, 16×16, 64×64, and 256×256, from top to bottom. The features used in this study included two different color and shape features and a simple texture feature. All except the FFT-based shape feature were calculated in five separate zones of the image. The zones are formed by first determining a circular area in the center of the image. The size of the circular zone is approximately one fifth of the area of the image. Then the remaining area is divided into four zones with two diagonal lines.

Average Color (*cavg* in Table 2) is obtained by calculating average R-, G- and B-values in five separate zones of the image. The resulting 15-dimensional feature vector thus describes the average color of the image and gives rough information on the spatial color composition.

Color Moments (*cmom*) were introduced in [12]. The color moment features are computed by treating the color values in different color channels in each

Table 1. Comparison of the performances of different feature extraction methods for different image classes. Each entry gives three performance figures ($\eta_{\text{local}}/\eta_{\text{global}}/\eta_{\text{half}}$).

features	classes		
	plane	face	car
cavg	0.06/0.16/0.59	0.05/0.10/0.56	0.03/0.21/0.63
cmom	0.06/0.16/0.59	0.05/0.10/0.56	0.04/0.21/0.63
texture	0.04/0.04/0.52	0.06/0.16/0.57	0.07/0.22/0.63
shist	0.11/0.62/0.84	0.10/0.54/0.82	0.13/0.34/0.68
sFFT	0.04/0.49/0.78	0.07/0.39/0.72	0.10/0.30/0.65

zone as separate probability distributions and then calculating the first three moments (mean, variance, and skewness) from each color channel. This results in a $3 \times 3 \times 5 = 45$ dimensional feature vector. Due to the varying dynamic ranges, the feature values are normalized to zero mean and unit variance.

Texture Neighborhood (*texture*) feature in PicSOM is also calculated in the same five zones. The Y-values (luminance) of the YIQ color representation of every pixel's 8-neighborhood are examined and the estimated probabilities for each neighbor being brighter than the center pixel are used as features. When combined, this results in one 40-dimensional feature vector.

Shape Histogram (*shist*) feature is based on the histogram of the eight quantized directions of edges in image. When the histogram is separately formed in the same five zones as before, a 40-dimensional feature vector is obtained. It describes the distribution of edge directions in various parts of the image and thus reveals the shape in a low-level statistical manner [1].

Shape FFT (*sFFT*) feature is based on the Fourier Transform of the binarized edge image. The image is normalized to 512×512 pixels before the FFT. Then the magnitude image of the Fourier spectrum is low-pass filtered and decimated by the factor of 32, resulting in a 128-dimensional feature vector [1].

To study the performance of the selected features with different types of images, three separate image classes were picked manually from the database. The selected classes were *planes*, *faces* and *cars*, of which the database consists of 292, 1115 and 864 images, respectively. The corresponding *a priori* probabilities are 0.5%, 1.9%, and 1.4%. In the retrieval experiments these classes were thus not competing against each other but mainly against the "background" of 57 724, i.e., 96.2% of other images. The used value for the parameter M was 20.

4.2 Results

Table 1 shows the results of forming the three scalar measures, η_{local}, η_{global}, and η_{half}, from the measured observed probabilities. The η_{local} measure was calculated for $n = 50$ first images. It can be seen that the η_{local} measure always is larger than the corresponding *a priori* probability. Also, the shape features *shist* and *sFFT* seem to outperform the other feature types for every image class and every performance measure. Otherwise, it is not yet clear which one of the

Table 2. The resulting τ values in the experiments.

features					classes		
cavg	*cmom*	*texture*	*shist*	*sFFT*	plane	face	car
×					0.30	0.35	0.39
	×				0.31	0.43	0.34
		×			0.26	0.26	0.34
			×		0.16	0.22	0.18
				×	0.19	0.22	0.18
×		×	×		0.16	0.21	0.18
×		×		×	0.17	0.23	0.18
×		×	×	×	0.14	0.21	0.16
	×	×	×		0.15	0.21	0.18
	×	×		×	0.18	0.22	0.19
	×	×	×	×	0.14	0.20	0.16
×	×	×	×	×	0.14	0.20	0.16

three performance measures would be the most suitable as a single measure of effectiveness.

The results of the experiments with the whole PicSOM system are shown in Table 2. First, each feature was used alone as the basis for the retrieval and then different combinations of features were tested. The two shape features again yield better results than the color and texture features, which can seen from the first five rows in Table 2. By examining the results with all tested classes, it can be seen that the general trend is that using a larger set of features yields better results than using a smaller set. Most notably, using all features gives better or equal results than using any single feature or subset of features. The implicit weighting of the relative importances of different features models the semantic similarity of the images selected by the user.

In the second and third sections of Table 2, the results of the experiment are presented when using first only one shape feature and then both features in the retrieval. It can be seen that the results are slightly better when using both shape features. These experiments thus also validate the overall trend that using more features generally improves the results. Therefore, it can be concluded that the PicSOM system is able to benefit from the existence of multiple feature types. As it is generally not known which feature combination would perform best for a certain image query, the PicSOM approach provides a robust method for using a set of different features and image maps formed thereof in parallel so that the result exceeds the performances of all the single features.

However, it also seems that if one feature vector type has clearly worse retrieval performance τ than the others, it may be more beneficial to exclude that particular TS-SOM from the retrieval process. For the proper operation of the PicSOM system, it is thus desirable that the used features are well balanced, i.e., they should on the average perform quite similarly by themselves.

5 Conclusions

In this paper, we have presented a set of methods for quantitative performance evaluations of different features and CBIR systems. The proposed τ measure is a general and automatic measure of a performance of retrieval systems based on the relevance feedback technique where the query is iteratively refined during multiple rounds of user-system interaction.

As a single visual feature cannot classify images into semantic classes, we need to gather the information provided by multiple features to achieve good retrieval performance. The results of our experiments show that the PicSOM system is able to effectively select from a set of parallel TS-SOMs a combination which outperforms single TS-SOMs in performance. The features used in the experiments are yet quite tentative and results suffer from the relative differences in performance between them. Therefore, we have started a series of experiments for selecting a proper and well-balanced set of features to be used in the PicSOM system in our future assessments.

References

[1] Sami Brandt, Jorma Laaksonen, and Erkki Oja. Statistical shape features in content-based image retrieval. In *Proceedings of 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 2000. To appear.

[2] N.-S. Chang and K.-S. Fu. Query by pictorial example. *IEEE Transactions on Software Engineering*, 6(6):519–524, November 1980.

[3] The Corel Corporation WWW home page, `http://www.corel.com`, 1999.

[4] Ingemar J. Cox, Matt L. Miller, Stephen M. Omohundro, and Peter N. Yianilos. Target testing and the PicHunter bayesian multimedia retrieval system. In *Advanced Digital Libraries ADL'96 Forum*, Washington, DC, May 1996.

[5] Alexander Dimai. Assessment of effectiveness of content based image retrieval systems. In *Third International Conference on Visual Information Systems*, pages 525–532, Amsterdam, The Netherlands, June 1999.

[6] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, Berlin, 1997. Second Extended Edition.

[7] P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. In *Proc. IJCNN-90, Int. Joint Conf. on Neural Networks, Washington, DC*, volume II, pages 279–285, Piscataway, NJ, 1990. IEEE Service Center.

[8] Jorma Laaksonen, Markus Koskela, and Erkki Oja. Content-based image retrieval using self-organizing maps. In *Third International Conference on Visual Information Systems*, pages 541–548, Amsterdam, The Netherlands, June 1999.

[9] MPEG-7: Overview (version 2.0). March 2000/Noordwijkerhout (The Netherlands) ISO/IEC JTC1/SC29/WG11 N3349.

[10] Erkki Oja, Jorma Laaksonen, Markus Koskela, and Sami Brandt. Self-organizing maps for content-based image retrieval. In Erkki Oja and Samuel Kaski, editors, *Kohonen Maps*, pages 349–362. Elsevier, 1999.

[11] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. Computer Science Series. McGraw-Hill, 1983.

[12] Markus Stricker and Markus Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases III (SPIE)*, volume 2420 of *SPIE Proceedings Series*, pages 381–392, San Jose, CA, USA, February 1995.