published in the Proceedings of SCIA 2001, Bergen, Norway, June 2001
available at <URL:http://www.cis.hut.fi/picsom/publications.html>

COMPARISON OF TECHNIQUES FOR CONTENT-BASED IMAGE RETRIEVAL

Markus Koskela, Jorma Laaksonen, and Erkki Oja

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

ABSTRACT

Content-based image retrieval (CBIR) is a new but widely-adopted method for finding images from vast and unannotated image databases. In CBIR images are indexed on the basis of low-level features, such as color, texture, and shape, that can automatically be derived from the visual content of the images. The operation of a CBIR system can be seen as a series of more or less independent processing stages. As there exists multiple choices for each of these stages, a multitude of CBIR systems can be implemented by combining a set of common building blocks. In this paper, we present a comparison of different techniques for three consecutive stages of a CBIR system. These include: (1) the initial per feature selection of considered images, (2) the combination of the lists of selected images, and (3) the final selection of images based on all available features simultaneously. The results of the performed experiments show that CBIR systems can be implemented using consecutive stages where, at each stage, a number of parallel techniques can be provided.

1. INTRODUCTION

Content-based retrieval from unannotated image databases is a wide and versatile field of research interests. Depending on the domain of interest, the database in question, and the amount of *a priori* information available on the images, the CBIR problem exhibits a varying degree of difficulty. A rather simple CBIR problem occurs when the database in question consists of images of a strongly restricted domain. One widely-studied application of this complexity is retrieval of trademark images, mainly based on different shape features as the lack of background enables automatic segmentation of such images. In the other extreme lies the problem of retrieving relevant images from large and dynamic collections of miscellaneous images. One massive example of such a challenging domain is indexing the images contained in the World Wide Web.

The basic problem in CBIR is the gap between the high-level semantic concepts used by humans to understand image content and the low-level visual features extracted from images and used by a computer to index the images in a database. Two most important research topics in CBIR are thus (1) the selection of the used features and the measure of similarity between them, and (2) the techniques for indexing the images, ie., how to select the images the system will display to the user next. In this paper, we will concentrate on the latter issue.

The operation of a CBIR system can be seen as a series of more or less independent processing stages. As there exists multiple choices for each of these stages, a multitude of CBIR systems can be implemented by combining a set of common building blocks. We have implemented a framework of a CBIR system in three consecutive stages including: (1) the initial per-feature selection of considered images, (2) the combination of the lists of images selected in the first stage, and (3) the final selection of images based on all the available features simultaneously. With four different techniques for the first selection stage, two choices for the combination stage, and three for the final selection stage, we have obtained a total of fourteen combinations the results for which will be presented.

2. CONTENT-BASED IMAGE RETRIEVAL

Query by pictorial example (QBPE) is a common retrieval paradigm in content-based image retrieval applications [2]. With QBPE, the image queries are based on example 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. In CBIR the user is thus an inseparable part of the query process. CBIR is in this sense different from most other applications in computer vision which are usually automatic and self-contained.

As the retrieval of unannotated images cannot be based on matching the user's query with the images in the database on abstract conceptual level, lower-level pictorial features need to be used. This changes the role of the human using the system from a requester to a mere selector who indicates the appropriateness of the offered images. As a retrieval system is usually not capable of giving the wanted images in its first response, the image query becomes an iterative and interactive process towards the desired image or images.

2.1. Principles of Relevance Feedback

The iterative and automatic refinement of a query is known as *relevance feedback* in information retrieval literature [9]. In text-based retrieval, relevance feedback can be implemented eg. by adjusting the weights of different textual terms when matching the query text with the documents of the database in a vectorial form. Other typical implementations of relevance feedback include adding new terms or removing irrelevant ones in the query phrase or modifying the user profile. Relevance feedback can be seen as a form of supervised learning to adjust the subsequent queries using the information gathered from the user's feedback. This helps the system in the following rounds of the retrieval process to better approximate the present need of the user.

An image retrieval system implementing relevance feedback tries to learn the optimal correspondence between the high-level concepts people use and the low-level features obtainable from the images. The user thus does not need to explicitly specify weights for different computational features because the weights are formed implicitly by the system. This is desirable, as it is generally a difficult task to give low-level features such weights which would coincide with human perception of images at a more conceptual level [8]. The correspondence between concepts and features is in addition temporal and case specific. This means that, in general, every image query is different from the others due to the hidden conceptions on the relevance of images and their mutual similarity.

In implementing relevance feedback in a CBIR system, three minimum requirements need to be fulfilled. First, the system must show the user a series of images, remember what images have already been shown, and not to display them again. Thus, the system will not end up in a loop and all images will eventually be displayed. Second, the user must somehow indicate which images are to some extent relevant to the present query and which are not. We call them here positive and negative seen images, respectively. It is thus not sufficient that the user picks just one of the shown images. Instead, a set of images must be indicated as positive ones while the remaining ones can implicitly be regarded as negative. As the third requirement, the system must change its behavior depending on which images are included in the positive and negative image sets. During the retrieval process more and more images are accumulated in these two image sets and the system has increasing amount of data to use in retrieving the succeeding image sets. The art of relevance feedback is finding the ways which use this information most efficiently.

2.2. A General CBIR System Structure

When a CBIR system is implemented with prototype-based statistical methods, each image in the database is transformed with a set of different feature extraction methods to a set of lower-dimensional prototypes in respective feature spaces. When the system tries to find images which are similar to the positive-marked seen images, it searches for those images whose distance to the positive images in some sense is minimal in any or all of the feature spaces. The distances between prototypes in the feature spaces can be defined in a multitude of ways, the Euclidean distance being the one used most. How the distances in various feature spaces are weighted and combined in order to form a scalar suitable for minimization, leaves a lot of room for different techniques. It can be stated that in general there does not and will not ever exist one single "correct" answer to this central question of CBIR. The stage of combining the distances calculated in different feature spaces is also a good candidate for a point where relevance feedback can be implemented.

The CBIR process can to some extent be formalized by denoting the set of images in the database as $\mathcal{D}$ and its non-intersecting subsets of positive and negative seen images as $\mathcal{D}^+$ and $\mathcal{D}^-$, respectively. The unseen images can then be marked as $\mathcal{D}'$, which leads to

$$\mathcal{D}' = \mathcal{D} \setminus (\mathcal{D}^+ \cup \mathcal{D}^-) \quad , \tag{1}$$

$$N' = N - (N^+ + N^-) \quad , \tag{2}$$

where the Ns denote the cardinalities of the respective sets. Let us denote the images as $\mathcal{I}_n, n = 1, 2, \ldots, N$. If we have M different feature representations for each image, they can be written as $\mathbf{f}^m(\mathcal{I}_n) = \mathbf{f}_n^m$, $m = 1, 2, \ldots, M$. The L images the system will display to the user next can be denoted with $\mathcal{D}^* = \{\mathcal{I}_1^*, \mathcal{I}_2^*, \ldots, \mathcal{I}_L^*\} \subset \mathcal{D}'$. Finding the images most similar to the positive seen images can then be formally written, for example, in a straightforward manner:

$$\min_{\mathcal{D}^*}! \; d = \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N^+} w_m \, d_m(\mathbf{f}^m(\mathcal{I}_l^*), \mathbf{f}^m(\mathcal{I}_n^+)) \quad , \tag{3}$$

where w_ms are the weights for individual features and $d_m(\cdot, \cdot)$ is the distance function suitable for being used with feature type $\mathbf{f}^m$. Though (3) is quite general in nature, it is still only one possibility among others. One might, for example, want to devise a discriminant function which includes also terms that depend on the negative-marked seen images.

An image database may contain millions of images. It is not possible to calculate accurately all distances between all the positive seen images and all the unseen images in the database. For this reason, some computational shortcuts need to be taken in order to circumvent this restriction. First, as much of the calculations as possible should be performed in advance in off-line mode and stored for being utilized when the CBIR system is used. As this stored information needs to be accessed fast, it may not be feasible to save it in a mass storage such as the computer's hard disk. If for efficiency reasons the data needs to be kept in the computer's random-access memory, the size of the available memory may become another bottleneck. Unfortunately, the dynamic nature of relevance feedback in CBIR

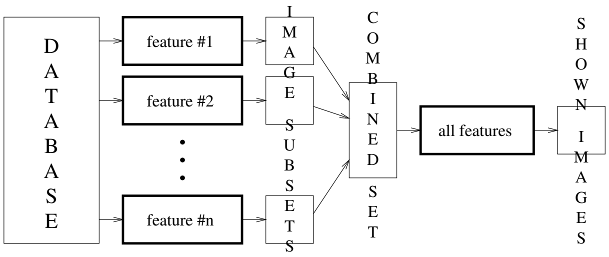

Fig. 1. The stages of image selection in CBIR.

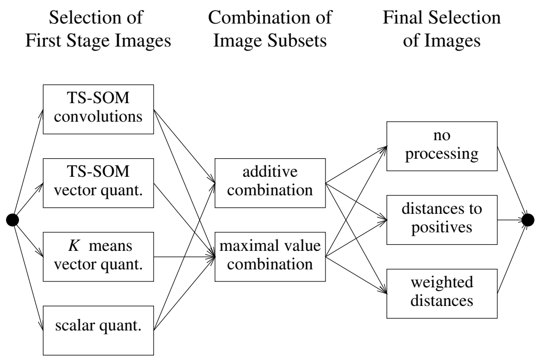

Fig. 2. The processing stages in the experiments.

to some extent fights against the aspiration to make use of advance calculations.

The second computational shortcut is to divide and conquer the image selection process by making it in two stages. Figure 1 illustrates this idea. Each feature representation can be used separately for finding a set of image candidates. This is especially advantageous if the distances calculated in the different feature spaces are weighted dynamically as in such a case it is not possible to order the images by their mutual distances in advance. The number of images in each subset may and should exceed the count of images to be finally shown to the user. These per-feature subsets with associated qualification values for each image included should then be combined in a larger set of images which will be processed in a more exhaustive manner. Depending on the sizes of the subsets and the combination algorithm, either all images in them or, for example, only those which are included in more than one of them, can be taken in the combined set. Nevertheless, in the final selection process there will be involved a substantially smaller number of images than the whole database. This enables to use computationally more demanding techniques for selecting among them.

2.3. CBIR System of the Experiments

We have extended our PicSOM CBIR system [5, 6] to accommodate alternative methods in some processing stages. For the current experiments, we have implemented for the first selection stage three new techniques that compete with the original method based on convolving the user's response on the Tree Structured Self-Organizing Maps (TS-SOMs). For the combination of the image subsets, we have added a new alternative treatment, namely the use of the maximum of the per-feature qualification values for each image instead of their sum. The original algorithm used in PicSOM does not contain any processing after the combination of qualification values from different features. Now we have implemented two choices for that purpose.

Figure 2 displays all the alternative functions for each of the stages. The following three sections will address every stage and the functions available for it in detail.

3. SELECTION OF FIRST STAGE IMAGES

According to the structure illustrated in Figure 1, the first stage of selection is performed for each feature in parallel. The result of the first stage will thus be a group of per-feature image subsets containing potentially relevant images according to the feature representation in question. In this study, three alternative approaches to this per-feature selection were applied. These three approaches are based on the TS-SOMs, vector quantization, and scalar quantization, respectively, and are described in this section.

3.1. Tree Structured SOMs

The first image indexing method used in the comparison is based on the Self-Organizing Map (SOM) [3]. 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 the same time, the models become ordered on the grid so that similar models are close and dissimilar models far from each other.

In order to achieve a hierarchical representation of the image database and to alleviate the computational complexity of training large SOMs, we use a special form of the SOM, namely the Tree Structured Self-Organizing Map (TS-SOM) [4]. 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 algorithm 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.

With the TS-SOM algorithm, the system marks the images selected by the user with a positive value and the other 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 Gaussian 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 response after the convolution, the algorithm retrieves 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. To assure that the whole database will be displayed, all images mapped to the particular SOM unit on the lowest levels are given equal precedence in being included in the returned image subset. Each image in the subset is given the convolved response value as its qualification value in the query.

3.2. Vector Quantization

The use of SOMs in indexing images according to their mutual similarity can be regarded as a special case of *vector quantization*. In the general case of vector quantization the topological ordering provided by the map lattice is lost and the similarity of two images is characterized only by they being or not being mapped to same vector quantization bin. The same sets of model vectors that were obtained when the TS-SOMs were trained can also be used as vector quantization codebooks. The number of model vectors in the lowest-level SOM is presumably too large for being used in vector quantization, but the second-lowest level will be suitable.

As the vector quantization produced by the TS-SOM is by no means optimal for representing the original data distribution, alternative quantization techniques can also be used. These include the well-known K-means or Linde-Buzo-Gray vector quantization [7].

With either choice, the SOM-based or K-means-based quantization, the feature vectors are divided in subsets in which the vectors resemble each other and the membership of each image in these quantization bins can be calculated in advance and stored in sort of *inverse files*. Those unseen images which have fallen into the same quantization bins as the positive-marked shown images are then good candidates for the next images to be displayed to the user.

The first stage image selection algorithm might select one or few quantization bins which according to some performance criterion seem to represent the positive images best. The image subset of that feature type will then be formed from the unseen images in that or those bins only. This leads to a sort of *depth first search* in the database. On the other hand, if the algorithm picks a few representative images from all those quantization bins which according to the criterion function are performing well, the system will implement a sort of *breadth first search*.

In the middle of an ongoing query, the quantization bins can be divided into the following four categories, in decreasing order of importance. First, bins containing both positive seen images and yet unseen images. These bins form a natural subset to concentrate on when searching for new images and they can be scored by the fraction of positive images in all seen images in them. In our implementation, the best-scoring bins of this category are processed using depth first search. The second category consists of bins containing only unseen images. All these bins are equally probable to contain relevant images so a natural way is to apply breadth first search. The third category is formed of bins whose all seen images have been negative. These bins are sorted by the ratio of unseen images to all images in them and breadth first search is used. The last category consists of bins having no unseen images. These are of no interest and can be discarded. Consequently, the quantization bins are scored and sorted according to the qualification value

$$S_i = \begin{cases} 1 + \frac{N_i^+}{N_i^+ + N_i^-} & \text{, if } N_i^+ > 0 \wedge N_i^\prime > 0 \\ 1 & \text{, if } N_i^+ = 0 \wedge N_i^- = 0 \wedge N_i^\prime > 0 \\ \frac{N_i^\prime}{N_i} & \text{, if } N_i^+ = 0 \wedge N_i^- > 0 \wedge N_i^\prime > 0 \\ 0 & \text{, otherwise} \end{cases} \tag{4}$$

where N_i^+, N_i^-, and $N_i^\prime$ are the numbers of positive and negative seen images and unseen images, respectively, mapped to vector quantization bin i. N_i is the total number of images in the bin.

From each vector quantization, an image subset of K images is selected in the order of descending S_i. As mentioned above, images from bins containing positive images, ie. the first category in (4), are picked until the limit of K images is reached. If there are not enough images in that bin the picking is continued in the bin with the next largest S_i and so on. If the count K could not be filled from bins with S_i larger than one, the remaining images are picked from the following categories using breadth first search, ie. picking only one image from each bin. This mode of operation is, however, only necessary in the end of an exhaustive query when all images from quantization bins containing positive-marked images have already been used. Each image that is included in the resulting image subset is assigned its quantization bin's qualification value S_i.

3.3. Scalar Quantization

In the case of scalar quantization the resemblance between images is in the first place found with respect to one component of a feature vector. Scalar quantization can be obtained by ordering the values in certain feature vector component. Ordered values can then be divided in a preset number of quantization bins each containing a few images. All images in the bin are then indexed with an inverse file. When the ordering and creation of the inverse is repeated for all components of the particular feature vector, one obtains different scalar quantization for each single feature component.

The similarity of two images can be expressed by the count of feature components for which they are quantized to same bin. With $\mathcal{D}^+$, the set of positive seen images, one can then rank all images which are mapped to the same scalar quantization bin as one or more images of $\mathcal{D}^+$, by counting the overall sum of bins shared by the image in question and the images of $\mathcal{D}^+$.

The complexity of such scalar quantization indexing is heavily dependent on the average count of images in the bins. The larger the average count is, the longer the inverse index lists will be and the greater fraction of all the images in the database will be considered on each iteration round. If, on the other hand, there exists an excessive number of scalar quantization bins, the average image count per bin will approach one. This results in lighter computational requirements but endangers the finding of all images when the iteration is continued.

In order to prevent longer feature vectors from dominating the creation of the combined image set in the following processing stage, we have divided the number of positive quantization bins shared by the image in question with the dimensionality of the feature vector to obtain the qualification value for the image.

4. COMBINATION OF IMAGE SUBSETS

The next step in the system of Figure 1 is combining the per-feature subsets of images provided by the first stage of processing. Mainly, this consists of simply taking an union of the image sets, but there are two issues requiring some consideration, namely dealing with duplicate images and limiting the size of the combined set. If the combined image set is to be processed in a computationally expensive manner, it may be practical to limit its size also at this stage. The size of the combined set naturally depends on the sizes of the image subsets but the number of duplicate images varies between queries and the size limit can be used to assure that the total number of images remains limited.

4.1. Additive Value Combination

If the image values provided by the first stage are additive, ie. they provide, in addition to ordinal scale, also information in the differences in size between values, a natural way to combine the image sets is to use value addition for duplicate images. This rewards images considered promising by multiple features, which, by assumption, is a desirable property. The convolved responses of TS-SOMs fulfill this requirement and therefore additive combination can be used to remove duplicates of the TS-SOM algorithm. Also, qualification values produced by scalar quantization are additive by nature.

4.2. Maximal Value Combination

Another combination approach, suitable for all image sets, is to use maximal values for duplicate images. This removes the effect of possible strong positive responses from multiple features. Maximal value combination can therefore be considered as a secondary option, needed for image sets with only ordinal values. The scoring function of vector quantization (4) can be regarded to provide only ordinal values and therefore maximal combination needs to be used with that algorithm.

5. FINAL SELECTION OF IMAGES

The last stage of processing, see Figure 1, is intended for a set of potentially relevant images. Here, the selection algorithm may be totally different from the one used in the first stage. In order to enable more demanding processing techniques, the set of remaining images should at this stage be substantially smaller than the whole database.

5.1. Selection by First Stage Information Only

The simplest processing at this stage is to do nothing. If the first selection stage and the set combination step already provide a justifiable set of images, it can be shown to the user as the system's response.

5.2. Selection by Distances to Positive Images

One possible method for selecting the final set of images is to rank the remaining images based on their cumulative distance to all already found positive-marked, ie., relevant images in the original feature space. With this method, the final selection is thus performed according to (3) with $w_m = 1$ for all features and

$$d_m(\mathbf{f}^m(\mathcal{I}_l^*), \mathbf{f}^m(\mathcal{I}_n^+)) = (\mathbf{f}^m(\mathcal{I}_l^*) - \mathbf{f}^m(\mathcal{I}_n^+))^T (\mathbf{f}^m(\mathcal{I}_l^*) - \mathbf{f}^m(\mathcal{I}_n^+)) \; , \quad (5)$$

ie. the squared Euclidean distance.

As calculating distance in a possibly very high-dimensional space is a computationally heavy operation, it may not be feasible to perform it for all images in a large database. Therefore, the first stage can be seen to act as a preprocessor which prunes the database as much as it is necessary before the actual image similarity assessment is carried out by using (5).

5.3. Selection by Weighted Distances

In the previous method, the weights w_m in (3) were all set to one. These feature weights, however, provide an opportunity for adaptation, as features which seem to work well in a given query could be given greater influence in determining the shown images. One approach to implementing this adaptation is to weight features based on the pairwise distances of the found positive images in the feature space. If the average distance of two positive images is relatively small, the feature can be considered as well-suited for the current image query. The absolute distances in different feature spaces vary and therefore the distances have to be normalized with the average pairwise distance of all images in the database. For a feature m, the weight w_m is thus given by

$$w_m = \frac{N(N-1)}{N^+(N^+-1)} \frac{\sum_{i=1}^{N^+} \sum_{j=i}^{N^+} d_m(\mathbf{f}^m(\mathcal{I}_i), \mathbf{f}^m(\mathcal{I}_j))}{\sum_{k=1}^{N} \sum_{l=k}^{N} d_m(\mathbf{f}^m(\mathcal{I}_k), \mathbf{f}^m(\mathcal{I}_l))} . \tag{6}$$

6. EXPERIMENT SETTINGS

In order to evaluate the applicability of the presented methods and their different combinations for CBIR, a series of experiments was performed. In this section, the experiment settings, including the used system framework, the image database, visual features, the generation of ground-truth information, and the performance measure, are described.

6.1. PicSOM

The PicSOM image retrieval system is a framework for research on algorithms and methods for content-based image retrieval. A more detailed description of the system and results of earlier experiments performed using the system can be found in [5, 6]. The PicSOM home page including a working demonstration of the system for public access is located at http://www.cis.hut.fi/picsom.

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 the responses from the used features are combined automatically by using the map surface convolutions and additive combination of the qualification values, as described above. The goal is to autonomously adapt to the user's preferences regarding the similarity of images in the database.

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.In the beginning of a new query, the system presents the user the first set of reference images which, in this study, are randomly selected from the database. Random images are displayed until one or more positive images are found. After that, the positive images are used as the starting point for the retrieval method in question.

6.2. The Image Database and Ground-Truth Classes

We evaluated the CBIR techniques with a set of experiments using an image collection from the Corel Gallery 1 000 000 product. 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. The images were converted from the original WIF (wavelet-compressed image) format to JPEG.

For the experiments, three separate image classes were picked manually from the database. The selected classes were *cars*, *faces* and *planes*, of which the database consists of 864, 1115 and 292 images, respectively. The corresponding *a priori* probabilities are 1.4%, 1.9%, and 0.5%. In the retrieval experiments these classes were thus not competing against each other but mainly against the "background" of 57 724, ie., 96.2% of other images.

The criterion for an image to belong to the *faces* class was that the main target of the image had to be a human head with both eyes visible and the head had to fill at least 1/9 of the image area. In the *cars* class, the main target of the image had to be a car, and at least one side of the car had to be completely shown in the image. Furthermore, the body of a car had to fill at least 1/9 of the image area. In *planes* class there were no restrictions, all images of aircraft or helicopters were accepted.

6.3. Features

The features used in the experiments 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 were formed by first determining in the center of the image a circular area whose size is one fifth of the area of the whole image. Then the remaining area was divided into four zones with two diagonal lines. The used features are briefly described below.

Average Color is obtained by calculating the average R-, G- and B-values in the 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 were introduced in [10]. The color moment features are computed by treating the color values in different color channels in each 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 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 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 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].

In our previous experiments with the PicSOM system, it was found out that using a larger set of features generally yields better results and that the used approach provides a robust method for using a set of different features in parallel so that the result exceeds the performances of all the single features [6]. Therefore, all five features were used in parallel in all experiments.

6.4. Parameters of the Methods

The TS-SOMs for all the five features were sized 4×4, 16×16, 64×64, and 256×256, from top to bottom. On the bottommost TS-SOM levels there were thus approximately the same number of SOM units (65 536) and database images (59 995) During the SOM training, each vector was used 100 times in the adaptation.

In vector quantization, one possibility is to use the same TS-SOMs but only for vector quantization purposes. Of the four TS-SOM levels we chose to use the second from bottom, ie. the one sized 64×64. On the average there were thus approximately 14 images mapped in each quantization bin. With the K-means vector quantization, the same number of quantization bins was used, which again results to the average of 14 images per bin. In the scalar quantization case, were used 15 000 bins which gives rise to approximately 4 images in each bin.

All the image subsets of Figure 1 contained 100 images before duplicate removal. The resulting combined image list was not shortened but all the images were involved in the final selection. After it, 20 best-scoring images were used as the system's response.

6.5. Performance Measure

Quantitative measures for the retrieval performance of image retrieval systems 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. In this section, one such figure, denoted as the τ measure, is presented. The measure can be applied to systems utilizing the relevance feedback approach in some form.

With the τ measure, it is assumed that the user is facing a target search task from $\mathcal{D}$ 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 $\mathcal{C}$ as relevant. This process is then repeated for each image in $\mathcal{C}$. Now, the τ measure equals the average number of images the system retrieves before the correct one is found.

The τ measure can be 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 $\frac{N_\mathcal{C}}{2}$, where $N_\mathcal{C}$ is the number of images in $\mathcal{C}$.

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

$$\tau \in [\frac{\rho_\mathcal{C}}{2}, 1 - \frac{\rho_\mathcal{C}}{2}] \quad (7)$$

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.

7. RESULTS

The results of the experiments are listed in Table 1. The rows of the table contain the first stage alternatives, Tree Structured SOMs, vector quantization using TS-SOM and K-means-based quantization, and scalar quantization. Similarly, the columns contain the methods for final stage processing. The first two columns differ only by the method for combining first stage image sets, addition (first column) and maximal value (second column). Results for using addition-based combining with vector quantization were not computed. In the last two columns, the used image set combination method is the standard one for each first stage method, ie. addition for TS-SOMs and scalar quantization, maximum value for vector quantization. Each entry in the table lists first the average value obtained from using the three image classes. The results for individual classes are shown in parentheses, in the following order: cars, faces, and planes.

First, considering the two first columns of Table 1, corresponding to the first stage, it can be seen that the TS-SOM algorithm yields best values for the τ measure. However, the overall best τ values are obtained using vector quantization with K-means clustering and second stage processing. The result when using TS-SOMs for quantization is worse

Table 1. Results of the retrieval experiments using the τ measure. Each entry in the table lists first the average value of using the three image classes and then results for the used classes (cars, faces, planes) in parentheses.

	no final stage, add comb.	no final stage, max comb.	distance to positives	weighted distance
SOM	0.174 (0.177 0.209 0.137)	0.175 (0.177 0.210 0.137)	0.190 (0.193 0.229 0.147)	0.179 (0.195 0.209 0.130)
VQ (SOM)		0.217 (0.212 0.235 0.203)	0.184 (0.187 0.181 0.185)	0.178 (0.191 0.164 0.178)
VQ (K-m)		0.185 (0.173 0.214 0.169)	0.155 (0.148 0.167 0.149)	0.154 (0.154 0.162 0.146)
SQ	0.251 (0.363 0.242 0.147)	0.302 (0.392 0.342 0.172)	0.300 (0.388 0.351 0.162)	0.267 (0.356 0.290 0.156)

as can be anticipated since the topological information of SOMs is discarded with vector quantization. Furthermore, the vector quantization algorithm can be seen to clearly require the final stage of processing. Of the two final stage techniques, the weighted distance seems to yield better results on the average.

The second-best results are obtained with the TS-SOM-based technique. The best results are located at the first column, corresponding to the standard version in which the final stage is not used. Also, it can be seen that adding this extra processing does not improve the results. More strikingly, using maximum value combination yields very similar results. Also, of the two algoritms applying SOMs, the TS-SOM-based technique performs better.

Of the studied first stage techniques, the performance of scalar quantization is clearly the worst. Another observation here is that the additive combination is notably better than using the maximum value. This may suggest that vector quantization might also benefit from using value addition for duplicates. This requires, however, a different scoring function than the one used in these experiments.

8. CONCLUSIONS

In this paper, a general multi-part structure of content-based image retrieval systems was presented. A variety of different CBIR techniques can be represented in terms of this system structure. Also, CBIR systems can be considered as being constructed of some basic building blocks. The operation of different CBIR systems can then be analyzed by studying the functionality of these blocks.

In the performed experiments, the K-means-based vector quantization used together with weighted distance to the positive examples was found to give the best results according to the used τ measure. The TS-SOM-based technique yielded the second-best results. It should be noted, however, that the topological ordering of the images on the TS-SOM map lattice is an additional, unique benefit which cannot be obtained with traditional vector quantization techniques.

ACKNOWLEDGMENTS

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

9. REFERENCES

[1] S. Brandt, J. Laaksonen, and E. Oja. Statistical shape features in content-based image retrieval. In *Proceedings of 15th International Conference on Pattern Recognition*, volume 2, pages 1066–1069, Barcelona, Spain, September 2000.

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

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

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

[5] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja. Self-organizing maps as a relevance feedback technique in content-based image retrieval. *Pattern Analysis & Applications*, June 2001. In press.

[6] J. T. Laaksonen, J. M. Koskela, S. P. Laakso, and E. Oja. PicSOM - Content-based image retrieval with self-organizing maps. *Pattern Recognition Letters*, 21(13-14):1199–1207, December 2000.

[7] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28(1):84–95, Jan. 1980.

[8] R. W. Picard, T. P. Minka, and M. Szummer. Modeling user subjectivity in image libraries. Technical Report #382, M.I.T Media Laboratory, 1996.

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

[10] M. Stricker and M. 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.