Pattern Analysis & Applications (2001)4:140–152

Self-Organising Maps as a Relevance Feedback Technique in Content-Based Image Retrieval

Jorma Laaksonen, Markus Koskela, Sami Laakso and Erkki Oja

Laboratory of Computer and Information Science, Helsinki University of Technology, Fin-02015 HUT, Finland

Abstract: Self-Organising Maps (SOMs) can be used in implementing a powerful relevance feedback mechanism for Content-Based Image Retrieval (CBIR). This paper introduces the PicSOM CBIR system, and describes the use of SOMs as a relevance feedback technique in it. The technique is based on the SOM's inherent property of topology-preserving mapping from a high-dimensional feature space to a two-dimensional grid of artificial neurons. On this grid similar images are mapped in nearby locations. As image similarity must, in unannotated databases, be based on low-level visual features, the similarity of images is dependent on the feature extraction scheme used. Therefore, in PicSOM there exists a separate tree-structured SOM for each different feature type. The incorporation of the relevance feedback and the combination of the outputs from the SOMs are performed as two successive processing steps. The proposed relevance feedback technique is described, analysed qualitatively, and visualised in the paper. Also, its performance is compared with a reference method.

Keywords: Content-Based Image Retrieval (CBIR); Multi-dimensional indexing; Neural networks; Relevance feedback; Self-Organizing Map (SOM); Unannotated image databases

1. INTRODUCTION

In this paper we describe how relevance feedback has been implemented by using the Self-Organising Maps (SOMs) [1] in our Content-Based Image Retrieval (CBIR) system named PicSOM [2,3]. Relevance feedback is a technique originally proposed for text-based information retrieval to improve the performance of information access systems. The improvement is achieved by modifying the system's responses based on the user's reaction to the previously retrieved documents. The relevance feedback techniques in the CBIR domain are addressed in Section 2.

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 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 the trademark images [4,5]. The results of applying CBIR in such a setting have been rather good. 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. In most cases, the images are accompanied by some kind of textual information. As current content-based methods are not always sufficient to extract enough information for effective image retrieval, text-based information can be a useful addition to the system, and should be utilised. There are several good overall reviews of CBIR [6–8].

The Self-Organising Map is a neurally-motivated unsupervised learning technique which has been used in many data analysis tasks. A genuine feature of the Self-Organising Map

Received: 5 July 2000
Received in revised form: 9 February 2001
Accepted: 15 February 2001

is its ability to form a nonlinear mapping of a high-dimensional input space to a typically two-dimensional grid of artificial neural units. During the training phase of a SOM, the weight vectors in its neurons get values which form a topographic or topology-preserving mapping in which vectors that reside near each other in the input space are mapped in nearby map units in the output layer. Patterns that are mutually similar in respect to the given feature extraction scheme are thus located near each other on the SOM. The PicSOM system uses a special form of the SOM, namely Tree Structured Self-Organising Map (TS-SOM) [9,10], which incorporates a hierarchical view in the database. An introduction to the Self-Organising Map and its tree-structured version will be presented in Section 3.

The PicSOM system and its use for content-based retrieval of images are briefly described in Section 4. In Section 5, the application of the SOM technique in the implementation of relevance feedback in CBIR is presented. Section 6 presents a set of experiments performed with the PicSOM system. Concluding remarks are drawn and future directions addressed in Section 7.

2. RELEVANCE FEEDBACK IN CBIR

Query By Pictorial Example (QBPE) is a common retrieval paradigm in content-based image retrieval applications [11]. With QBPE, the 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. Techniques which have been used in traditional text database retrieval would be applicable to image searching, too, if only a textual description of the contents of the images could be automatically produced. Unfortunately, in the current state of machine vision techniques, this is out of our reach.

As image retrieval cannot be based on matching the user's query with the images in the database on an 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 to the user, the image query becomes an iterative and interactive process towards the desired image or images.

In this section we first introduce the reader to the basic principle of relevance feedback in Section 2.1. Next, we present our view of the general structure of CBIR systems in Section 2.2, and review some of the existing relevance feedback implementations in Section 2.3. Finally, we address some fundamental questions in the implementation of relevance feedback in Section 2.3.

2.1. Principle of Relevance Feedback

The iterative and automatic refinement of a query is known as *relevance feedback* in information retrieval literature [12]. In text-based retrieval, relevance feedback can be implemented 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, modifying the user profile, or using reinforcement learning [13]. 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 on the following rounds of the retrieval process to better approximate the present need of the user.

A system implementing relevance feedback in CBIR tries to learn the optimal correspondence between the high-level concepts people use and the low-level features obtained 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 [14]. 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. Secondly, 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 behaviour depending on which images are included in the positive and negative image sets. During the retrieval process, more and more images are accumulated in the two image sets, and the system has an 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. General Structure of CBIR Systems

As described above, a content-based image retrieval system must in general be based on low-level visual features. These representations can be either statistical or structural in nature. In the case of non-restricted content of images in the database, only statistical features can be called upon. Statistical data can be modelled with a wide variety of parametric and semiparametric methods, including regression techniques and neural networks. These methods, however, have very limited use in CBIR, as there generally does not exist an inverse transformation from the feature representation back to the image domain. Therefore, nonparametric prototype-based techniques are *de facto* the only feasible alternative.

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 prototype vectors in respective feature spaces. When the system tries to find images which are similar to the positive-marked seen images, it searches for 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 minimisation, 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 features spaces is also a good candidate for a point where relevance feedback can be implemented.

The CBIR process can be formalised 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

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

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

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 vectors for each image, they can be written as $\mathbf{f}^m(\mathcal{I}_n) = \mathbf{f}^m_n$, $m = 1, 2, \ldots$, M. The N^* images the system will display to the user next can be denoted with $\mathcal{D}^* = \{\mathcal{I}^*_1, \mathcal{I}^*_2, \ldots, \mathcal{I}^*_{N^*}\} \subset \mathcal{D}'$. Finding the images most similar to the positive seen images can then be formally written, for example, in a straightforward manner:

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

where the w_ms are the weights for individual features and $d_m(\cdot,\cdot)$ is the distance function suitable for being used with feature type $\mathbf{f}^m$. The outermost summation over the images in $\mathcal{D}^*$ is equivalent to the selection of the N^*-sized subset of $\mathcal{D}'$ that have the smallest total distance according to the inner two summations over the features and positive seen images. Though Eq. (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. Or, one could use, for example, maximum norm instead of summation over the M different features.

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. Therefore, some computational shortcuts need to be taken in order to circumvent this restriction. First, as much as possible of the calculations should be performed in advance in off-line mode and stored for use when the CBIR system is used. As this stored information needs to be accessed quickly, it may not be feasible to save it in 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 to some extent fights against the attempt to employ advance calculations.

The second computational shortcut is to divide and conquer the image selection process by making it in two stages. Relevance feedback can be implemented in either or both selection stages. Figure 1 illustrates this idea. Each feature representation can be used separately for finding a set of matching 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 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 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.

The third technique for answering the challenge of huge databases is to use quantisation. Two approaches exist, namely *scalar quantisation* and *vector quantisation*. With either technique, the feature vectors are divided in subsets in which the vectors resemble each other. In the case of scalar quantisation the resemblance is in respect to one component of the feature vector, whereas resemblance in vector quantisation means that the feature vectors are similar as whole. By forming an intersection of scalar-quantised subsets created for each individual vector component, one can obtain a subset of prototypes which lie within a fixed distance along each component direction of the feature vector component direction. This roughly corresponds to vector quantisation if the components of the feature vector are assumed independent. With either quantisation technique, the membership of each image in these quantisation bins can be calculated in advance and stored in sort of *inverse files*. Those unseen images which have fallen into the same quantisation bins as the positive-marked shown images are then good candidates for the next images to be displayed to the user. One may also want to calculate the exact distance between the prototypes. In that case, quantisation serves as an effective method for *pruning the database* before exhaustive search.

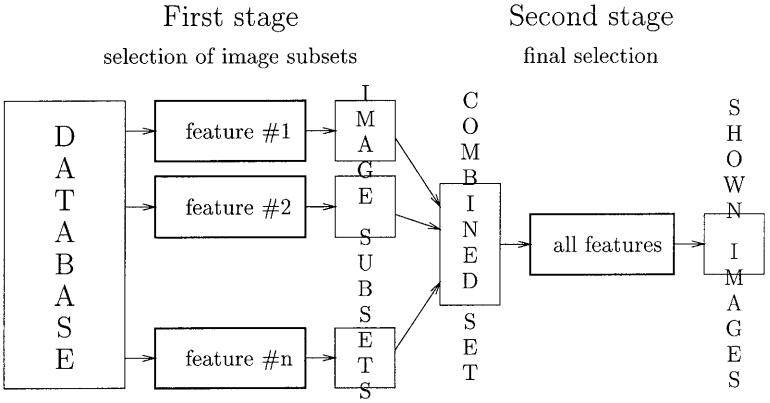


Fig. 1. The stages of image selection in CBIR.

If a CBIR system uses the two-stage approach depicted in Fig. 1 and quantisation in the first stage for selecting images for the subset, there exist then two alternative routes. Either one of them selects one or few quantisation bins which, according to some performance criterion, seems 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 one picks a few representative images from all those quantisation bins which, according to the criterion function, are performing well, the system will implement a sort of *breadth first search.*

2.3. Existing Relevance Feedback Systems

The relevance feedback approach has been applied to content-based image retrieval with a variety of techniques. If all features are binary, i.e. they reflect the presence or absence of specific combination of visual properties in the image, the retrieval and feedback techniques can be directly copied from the textual database methods [15].

In the simplest CBIR relevance feedback implementations, such as the PicHunter system [16], each image in the database is scored according to the distances from it to the positive images. Therefore, only the relative placement of vectors in the feature space is meaningful, and the distance metric is not affected in a global fashion.

Each component in the feature representation of the images can be given a weight which is used in calculating the distances between the images. These weights are then modified according to the user's responses so that feature components which have the smallest variances among the positive images get the largest weights [17,18]. The weighting can also be made dependent on the difference of the inverse variances of the positive and all shown images [19]. Another alternative is to adjust the distance metric to produce the smallest attainable relative distance between the positive images [20].

Relevance feedback can also be implemented by combining hierarchical clustering of the database and set-theoretic machine learning of new rules from the user-given examples [21]. If the feature space is low dimensional, it is possible to use relevance feedback to modify its multi-interval discretisation. Histogram-based matching will then turn to favour the user's view of similarity with respect to a specific feature, e.g. colour [22].

The relevance feedback systems are generally such that the accumulated relevance information is discarded between successive queries. So each retrieval session is started from the same initial situation, and preceding uses of the system have no influence on the present query. A totally opposite approach has been selected in the MetaSeek system, where all user interactions are stored and used in later queries in selecting between a set of independent image search engines [23].

2.4. Nonlinearity of Image Similarity

Most of the existing relevance feedback techniques described in the previous section treat the feature space globally rather than locally. This global attitude is manifested, for example, in linear weighting of the distances along individual feature directions. However, it should be clear that a distance measure or feature weighting which is advantageous in the vicinity of a set of images which are positive and therefore similar to each other, may not produce favourable results for the rest of the images. Also, rules which are applicable in one part of the feature space are not as such generalisable to handle the whole space. All these phenomena are direct consequences of the inherent nonlinear nature of image similarity [24].

On the contrary, the relevance feedback technique in our PicSOM system is local in the sense that it operates only in the local neighbourhoods of the images marked positive or negative by the user. Therefore, the method respects the nonlinear nature of image similarity. Simultaneously, it produces an implicit weighting of the different features so that those features which seem to perform better than the others in that particular task are weighted the most. This process will be elaborated in detail in Section 5.

3. SELF-ORGANISING MAPS

The Self-Organising Map (SOM) [1] is an unsupervised, self-organising neural algorithm widely used to visualise and interpret large high-dimensional data sets. The SOM defines an elastic net of points that are fitted to the distribution of the training data in the input space. It can thus be used to visualise multidimensional data, usually on a two-dimensional grid. Typical applications include visualisation of process states or financial results by representing the central dependencies within the data on the map [25].

The SOM consists of a two-dimensional lattice of units or artificial neurons. A model vector $\mathbf{m}_i$ is associated with each map unit i. The map attempts to represent all the available observations $\mathbf{x}$ with optimal accuracy by using the map units as a restricted set of models. During the training phase, the models become ordered on the grid so that similar models are close to and dissimilar models far from each other.

3.1. Training of a SOM

The fitting of the model vectors is usually carried out by a sequential regression process, where $t = 0, 1, \ldots, t_{max} - 1$ is the step index: For each input sample $\mathbf{x}(t)$, first the index $c(\mathbf{x})$ of the Best-Matching Unit (BMU) or the 'winner' model $\mathbf{m}_{c(\mathbf{x})}(t)$ is identified by the condition

$$\forall i: \|\mathbf{x}(t) - \mathbf{m}_{c(\mathbf{x})}(t)\| \leq \|\mathbf{x}(t) - \mathbf{m}_i(t)\| \qquad (4)$$

The usual distance metric used here is the Euclidean one. After finding the BMU, a subset of the model vectors constituting a neighbourhood centered around node $c(\mathbf{x})$ are updated as

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h(t; c(\mathbf{x}), i)\ (\mathbf{x}(t) - \mathbf{m}_i(t)) \qquad (5)$$

Here $h(t; c(\mathbf{x}), i)$ is the 'neighbourhood function', a decreasing function of the distance between the ith and $c(\mathbf{x})$th nodes on the map grid. This regression is reiterated over the available samples and the value of $h(t; c(\mathbf{x}), i)$ is allowed to decrease in time to guarantee the convergence of the prototype vectors $\mathbf{m}_i$. The large values of the neighbourhood function $h(t; c(\mathbf{x}), i)$ in the beginning of the training initialise the network and the small values on later iterations are needed in fine-tuning.

The search for the best-matching unit dominates the computing time of the SOM algorithm, and it can be computationally expensive in high input dimensionalities or large SOM networks. The basic algorithm uses full search, where all the units must be considered to find the BMU. This makes the complexity of the search $O(N)$, where N is the number of units.

3.2. Tree Structured Self-Organising Map, *TS-SOM*

To speed up the search of the best-matching unit, Koikkalainen and Oja introduced a variant of SOM called the Tree Structured Self-Organising Map (TS-SOM) [9,10]. TS-SOM is a tree-structured vector quantisation algorithm that uses normal SOMs at each of its hierarchical levels. The TS-SOM is loosely based on the traditional tree-search algorithm. Due to the tree structure, the number of map units increases when moving downwards the SOM levels of the TS-SOM. The search space for the best-matching vector of Eq. (4) on the underlying SOM level is restricted to a predefined portion just below the best-matching unit on the above SOM. Unlike most tree-structured algorithms, the search space is not limited to the children of the BMU on the upper level. As each level of the TS-SOM is a normal SOM, the search space can be set to include also neighbouring nodes having different parent nodes in the upper level. The structure of a TS-SOM in one-dimensional case with three SOM levels illustrated in Fig. 2.

The tree structure reduces the time complexity of the search from $O(N)$ to $O(\log N)$. The complexity of the searches using TS-SOM is thus remarkably lower than if the whole bottom-most SOM level had been accessed without the tree structure. The computational lightness of the TS-SOM facilitates the creation and use of huge SOMs, which, in the image retrieval context, can be used to hold the images stored in the image database.

3.3. Self-Organising Maps in Content-Based Image Retrieval

A hierarchical SOM has been utilised as an indexing tool with texture features in CBIR [26]. In another system a hierarchical SOM has been constructed for image database exploration and similarity search by using colour information [27]. In one study, SOM was used in classifying and retrieving similar subimages by their textural contents [28]. Objects of an image database have also been organised according to their boundary shapes in a two-dimensional browsing tree by using a SOM [29]. SOMs have additionally been used for feature extraction in image databases containing astronomical images [30]. The unsupervised clustering property of the SOM has been used also for image segmentation [31].

The Self-Organising Map has been used in the abovementioned studies mostly for visualisation purposes. As similar images are mapped near each other on the map, browsing of a database becomes easier when a set of representative images can be seen on a computer's screen in a two-

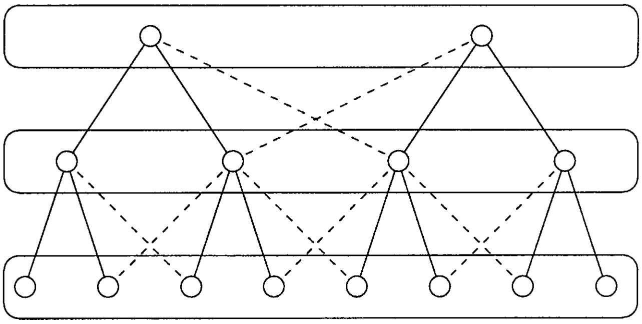


Fig. 2. The structure of a three-level one-dimensional TS-SOM. The solid lines represent parent-child relations and the dash lines represent neighbouring nodes included in the BMU search space.

dimensional grid. Clicking any of the images then either descends to a lower tree level, or displays all images mapped in that map unit. Another motivation mentioned in the original papers has been the savings in computation time when a huge database can be accessed with tree search. However, none of the systems described implements relevance feedback with the SOM, nor is capable of using more than one SOM simultaneously.

4. PicSOM CBIR SYSTEM

This section presents a short description of our PicSOM retrieval system. A more detailed description of the system and results of experiments performed with it can be found elsewhere [2,3]. The PicSOM image retrieval system is a framework for generic research on algorithms and methods for content-based image retrieval. Our method is named PicSOM due to its similarity to the well-known WEBSOM [32] document browsing and exploration tool that can be used in free-text mining. WEBSOM is based on a SOM that automatically organizes documents into a two-dimensional grid so that the related documents appear close to each other. In an analogous manner, we have aimed at developing a tool that utilises the strong self-organising power of the SOM in unsupervised statistical analysis for digital images.

PicSOM supports multiple parallel features and with a technique introduced in the PicSOM system, the responses from the parallel TS-SOMs are combined automatically. This question will be elaborated in detail in Section 5.2. The currently-implementeded features include simple colour and texture features as well as six different shape features [33]. The features used in our experiments are described briefly in Section 6.1.

4.1. Training the Image Maps

For the sake of computational effectiveness, Tree Structured SOMs are used in PicSOM instead of plain SOMs. In addition, the use of TS-SOMs incorporates a hierarchical view to the database through the TS-SOM levels. A separate TS-SOM is created for each feature type used.

Given a set of images in the form of feature vectors, the training of each TS-SOM starts from its top level. When the top-most level has been trained, it is frozen and the training of the second level is started. Every time one SOM level has finished learning, all image feature vectors are mapped to that SOM, each in the SOM unit which is nearest to it. Each map unit which has one or more images mapped in it is then given a visual label. This label is the image whose feature vector is nearest to the model vector. Feature vectors of images and weight vectors of map units are illustrated in an artificial two-dimensional example in Fig. 3.

The map units are thus given visual labels which can be used to represent all the images mapped in that particular map node. The image labels of a 16 × 16 SOM trained with average colour as the feature are shown in Fig. 4. From the SOM surface, the topological ordering of the label images based on their color content can be observed: reddish images are located in the upper left corner of the map, and the overall colour changes gradually to blue when moving diagonally towards the bottom right corner. On the other hand, light images are situated in the bottom left corner and dark images in the opposite position in the upper right corner of the map.

4.2. Basic Operation of PicSOM

The operation of PicSOM image retrieval is as follows: (1) An interested user connects to the WWW server providing

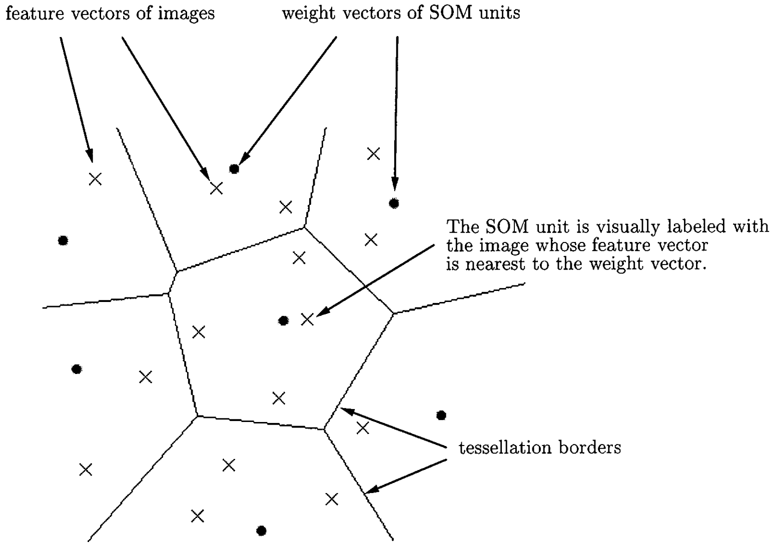

feature vectors of images
weight vectors of SOM units
The SOM unit is visually labeled with the image whose feature vector is nearest to the weight vector.
tessellation borders

Fig. 3. A visualisation of the distributions of the feature vectors of images (crosses) and the SOM weight vectors (circles) in a two-dimensional space.

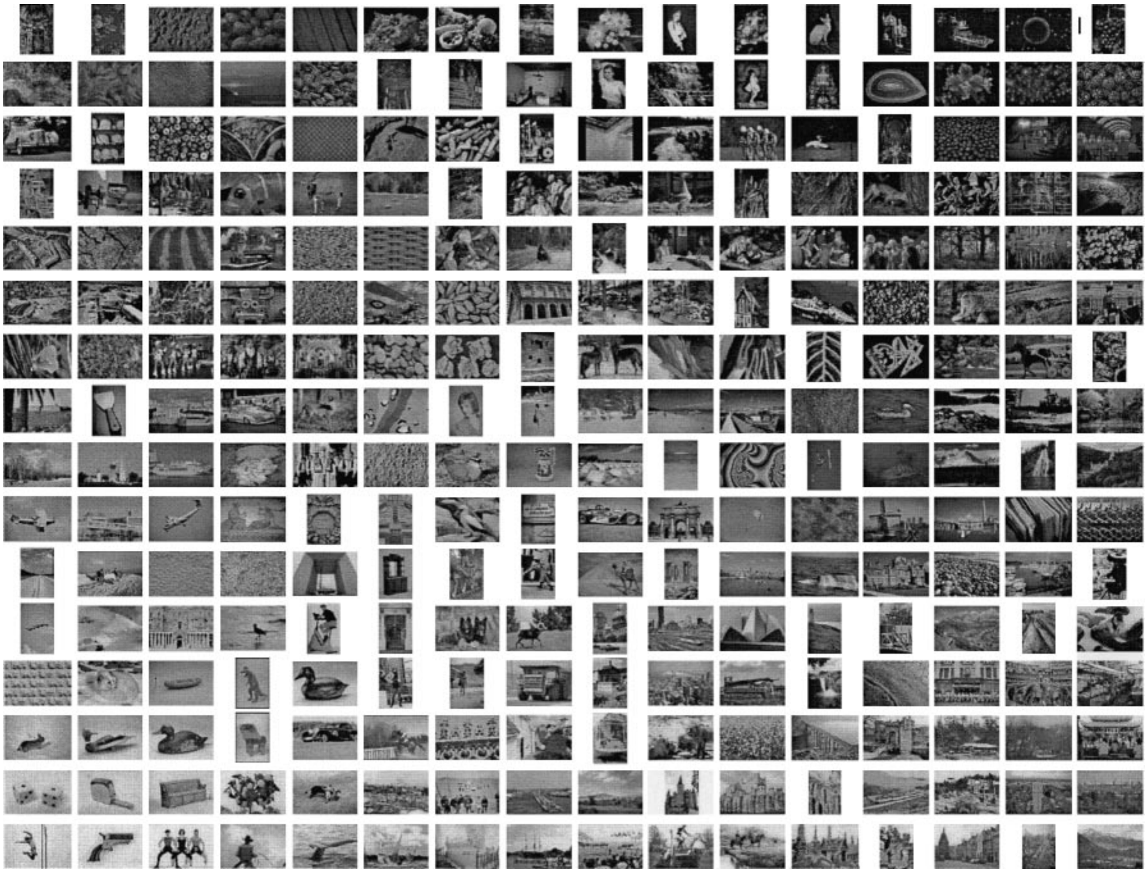


Fig. 4. The surface of the 16 × 16-sized SOM formed with the average RGB colour feature.

the search engine with her web browser; (2) the system presents a list of databases available to that particular user; (3) when the user has selected the database the system presents a list of available features in that database; (4) after the user has selected the features, the system presents an initial set of tentative images scaled to a small 'thumbnail' size. The user selects the subset of these images which best matches her expectations and, to some degree, of relevance fits to her purposes. Then she hits the 'Continue Query' button in her browser, which sends the information on the selected images back to the search engine; (5) based on this data, the system then presents the user with a new set of images along with the images selected so far and the query iteration is continued.

In the experiments presented in Section 6, the images shown on the first thumbnail page have been randomly picked from the database. Random picking can be argued to produce a sample which uniformly enough represent the whole database. Another possibility would be to use a subset of the visual labels of the topmost levels of the TS-SOMs corresponding to the features selected for the query by the user. Due to the nature of the SOMs, these images would together form a good uniform sample of the image database.

4.3. User Interface

In PicSOM, the queries are performed through a WWW-based user interface. The PicSOM home page including a working demonstration of the system for public access is located at *http://www.cis.hut.fi/picsom*.

The PicSOM user interface in the midst of an ongoing query is displayed in Fig. 5. On the top, the three parallel TS-SOM map structures represent three map levels of SOMs trained with RGB colour, texture, and shape features, from left to right. The sizes of the SOM levels are 4 × 4, 16 × 16 and 64 × 64, from top to bottom. Below the TS-SOM maps, the first set of images consists of relevant images selected by the user on the previous rounds of the retrieval process. In this example, all images representing buildings have been selected. The next images, separated with a horizontal line from the selected ones, are the current 16 new images of which the user should now select the relevant ones, and then hit the 'Continue Query' button.

5. SELF-ORGANISING MAPS AS A RELEVANCE FEEDBACK TECHNIQUE

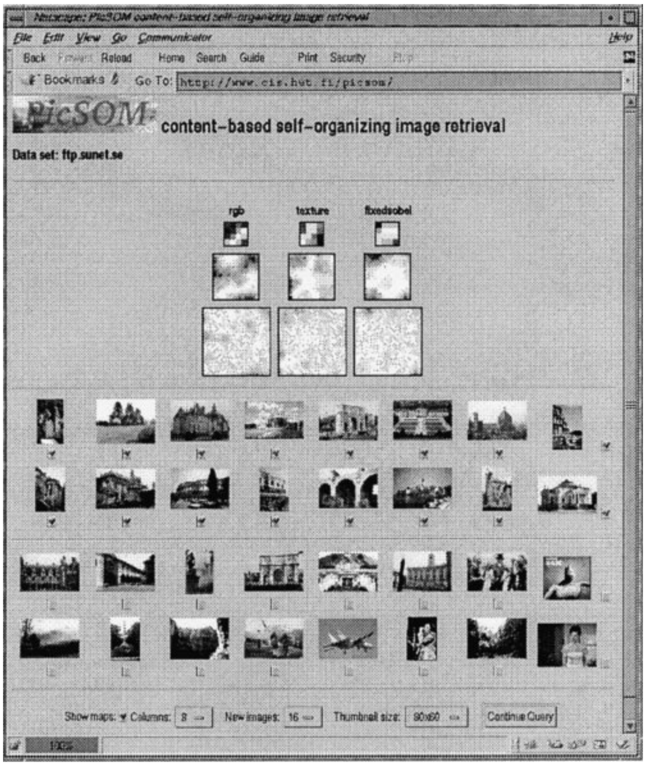
This section describes how Self-Organising Maps can be used to implement relevance feedback. The introduced technique is the backbone of our PicSOM CBIR system, and has been tested with numerous feature extraction methods and various databases.

5.1. Relevance Feedback in PicSOM

A novel technique introduced in the PicSOM system implements relevance feedback and simultaneously facilitates automatic combination of the responses from multiple Tree Structured SOMs and all their hierarchical levels. This mechanism aims at autonomous adaptation to the user's behaviour in selecting which images resemble each other in the particular sense the user seems to be interested in.

As described in Section 4.2, the PicSOM system presents the user on each round of the image query a set of images she has not seen before. She then marks the relevant (i.e. positive images), and the system implicitly interprets the unmarked images as negative ones. Because all the database images have been previously mapped in their best-matching SOM units at the time the SOMs were trained, it is now easy to locate both the positive and negative images on each level of every TS-SOM in use. The map units are scored with a fixed positive value for each positive image mapped in them. Likewise, negative images contribute negative values. These values are selected so that the sum of all positive values equals plus one, and the sum of all negative values equals minus one. The total sum of all values on each map is thus equal to zero.

The system remembers all image responses the user has given since the query was started. Information on all the images seen and the user's opinions on them thus becomes stored in every single SOM in the system. Everything until this point can be regarded as trivial and even redundant storing of data. However, this is the point where relevance feedback really enters the play. The basic idea is simple: the formation of a Self-Organising Map brings similar images in nearby map units – so let's exploit that property. If a particular SOM unit has been the best-matching one for many positive images and for none or only few negative ones, it can be deduced that its content coincides with the user's opinion well. By assumption, the neighbouring SOM units are similar to it, and the images mapped in them can likewise be supposed to be relevant for the user.


Netscape: PicSOM content-based self-organizing image retrieval
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go To:
PicSOM
rgb
texture
fixedsobel
Continue Query
100%

http://www.cis.hut.fi/picsom/

content-based self-organizing image retrieval

Data set: ftp.sunet.se

Show maps: ▾ Columns: 8 New images: 16 Thumbnail size: 90x60

Fig. 5. The PicSOM user interface.

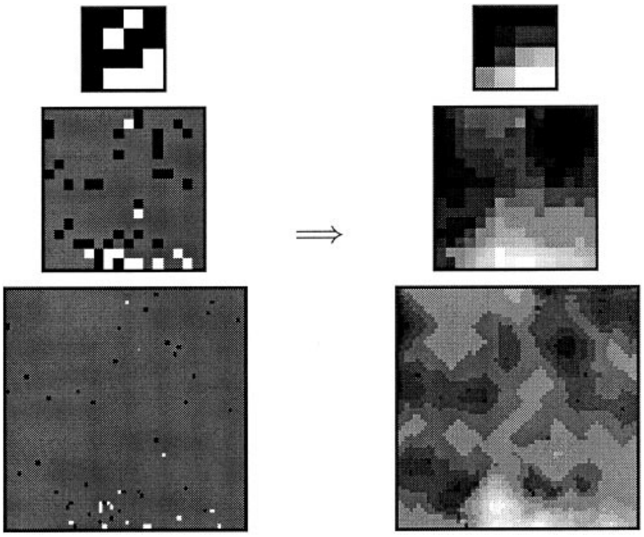


Fig. 6. An example showing how the levels of a TS-SOM, on which the images selected and rejected by the user are shown with white and black marks, respectively, are convolved with low-pass filters.

Each TS-SOM uses different feature extraction, and therefore the spreading of the positive and negative values is different in every SOM. While some feature extractions may spread the responses evenly all over the map surface, other features may cluster the positive, i.e. relevant responses densely in one area of the map. The latter situation can be interpreted as being an indication on the good performance of those particular features in the current query. The denser the positive responses are the better the feature coincides in that specific area of the feature space with the user's perception on images' relevance.

Now, all the three factors, namely (1) the degree of the separation of the positive and negative images on the SOM, (2) the relative denseness of the positive images, and (3) the similarity of images in neighbouring map units, can be accounted for in a single action. This joint action is low-pass filtering of response values on the two-dimensional map surfaces. Strong positive values from dense relevant responses get expanded to neighbouring SOM units, whereas weak positive and negative values in the map areas where the responses are sparse cancel each other out. What follows in the low-pass filtering is the polarisation of the entire map surface in areas of positive and negative cumulative relevance. In practice, the filtering has been implemented by convolving the map image with triangle-shaped horizontal and vertical masks whose size is approximately one fifth of the width of the corresponding TS-SOM level. Figure 6 illustrates how the positive and negative responses, displayed with white and black map units, respectively, are first mapped on three levels of a TS-SOM, and how the responses are expanded in the convolution.

The images used as labels for the SOM units which have the strongest positive relevance value after the low-pass filtering are then obvious candidates for the next images to be shown to the user. This leads to a breadth first selection of images on other TS-SOM levels but the bottom-most ones. On the bottom levels all images mapped in the particular SOM unit are given equal precedence in being displayed to user, not just the one image nearest to the weight vector. Depth first search is thus used on the lowest TS-SOM levels. By selecting the size of the bottom-most SOM level relative to the size of the image database, one can tune the balance between breadth first and depth first selections.

One may also be interested in how sets of images that are known to be similar to each other in some respect are mapped on the SOM surfaces. This kind of inspection reveals the feature extraction method's capability to map similar images near each other in the feature space and, further, the SOM training algorithm's ability to preserve the spatial ordering of the feature space. Figure 7 gives an example. There are in three columns one hand-picked image class, *cars, faces* or *planes*, in each. The rows correspond to three different feature extraction methods, *Average Colour, Shape Histogram* and *Shape FFT*. Both the classes and features will be described in more detail in Section 6.1. It can be seen that the *Average Colour* feature is able to cluster only the images in the *planes* class, whereas the *cars* and *faces* classes are widely distributed. On the other hand, the *Shape Histogram* feature clusters all three classes well, but the *cars* and *planes* classes are somewhat overlapping on the left side of the maps. Finally, *Shape FFT* feature does not make as tight clusters as *Shape Histogram* does, but the separation between the *cars* and *planes* classes is better.

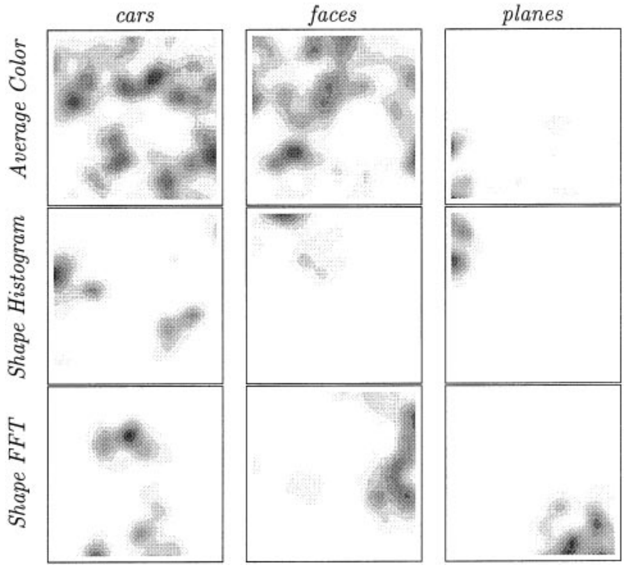


Fig. 7. Mappings of different image classes (shown in columns) on the lowest-level SOMs of different features (shown in rows). The distributions have been low-pass filtered for easier inspection.

5.2. Combination of Different Features

The mapping of the positive and negative responses and the succeeding low-pass filtering of the SOM surfaces is thus performed independently for every level of every TS-SOM the user has selected to be active in the search. The resulting values in the SOM units are, however, mutually comparable, and they can be globally ordered in order to find the best among the best candidate images. Thus, any kind of explicit weighting of different features is not needed, as the denseness of the positive responses is implicitly involved in the process.

There now exist two alternative options. Either we disregard the possibility that some image may simultaneously obtain a strong positive score on one TS-SOM level and a strong negative score on another, in which case duplicate images are simply removed and the maximum value of the score is used for the image, or we take that situation into account and implement a final stage of value combination for images which appear in the candidate sets of more than one TS-SOM level. The latter selection, corresponding to the second stage in the block diagram of Fig. 1, has been made in the current PicSOM implementation. In the first stage, there are always K or less images selected among the visual labels of the SOM units of every level of each of the M TS-SOMs. Score values of duplicate images are then summed and the best N^* images form the final selection. This inevitably somewhat increases the number of calculations needed in every query iteration, but on the other hand, reinforces the interplay of the different features.

The way the relevance feedback is implemented in PicSOM has one additional advantage to be noted. As the cumulative responses are calculated for each TS-SOM level separately, and the topologies of the feature spaces of all the TS-SOMs are different, the images which become selected due to the good performance of only one feature type are likely to be mapped in nonadjacent and sparsely distributed areas on the other TS-SOMs. If these images are then indicated as relevant by the user, new areas of relevance will be found on the other maps. The search will thus not be stuck in the local environments of the first relevant images found, but will eventually expand to all neighbourhoods of the different feature types of all positive seen images.

6. EXPERIMENTS

In order to evaluate the applicability of the relevance feedback technique employed in the PicSOM system we carried out a series of experiments. The database, manually-picked image classes and features used in the study are described together with the created TS-SOMs in Section 6.1. In order to compare the use of SOMs with another technique which, in other respects, is similar to our system, we devised a reference CBIR system. This reference technique will be described in Section 6.2. The performance measure used in the evaluation is explained in Section 6.3, and the results of the experiments finally in Section 6.4.

6.1. Database, Classes, Features and *TS-SOM*s

We evaluated the two CBIR approaches with a set of experiments using an image collection from the Corel Gallery 1,000,000 product [34]. 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 colour, but there are also a small number of greyscale images. The images were converted from the original WIF (Wavelet-Compressed Image) format to JPEG.

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, i.e. 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.

The features used in the experiments included two different colour and shape features and a 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 centre 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 use of the zoning is motivated as it incorporates some amount of information on the spatial distribution of the low-level visual characteristics in the images.

Average Colour 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 colour of the image, and gives rough information on the spatial colour composition.

Colour Moments were introduced by Stricker and Orengo [35]. The colour moment features are computed by treating the colour values in different colour channels in each zone as separate probability distributions, and then calculating the first three moments (mean, variance and skewness) from each colour channel. This results in a 3 × 3 × 5 = 45 dimensional feature vector. Due to the varying dynamic ranges, the feature values are normalised to zero mean and unit variance.

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

The Shape Histogram feature is based on the histogram of the eight quantised 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 [33].

Shape FFT feature is based on the Fourier Transform of the binarised edge image. The edges are sought by using eight Sobel masks in the intensity and saturation channels of the image. A binarised edge pixel is registered if the gradient value in the intensity channel exceeds 15% of the maximum intensity gradient in the image, or if the saturation gradient exceeds 35% of the respective maximum. The image is normalised without affecting its aspect ratio to the maximum of 512 × 512 pixels by bicubic interpolation 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 [33].

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 bottom-most TS-SOM levels, there were thus approximately the same number of SOM units (65,536) as the number of database images (59,995). During the SOM training, each feature vector was used 100 times in the adaptation.

6.2. Reference System

We wanted to perform an evaluation in which the PicSOM system could be compared to another CBIR system similar to it in all other respects, but the implementation of relevance feedback. For that purpose, we devised a competitive system which used the same TS-SOM maps as used in the PicSOM system. Now, the maps were only used for vector quantisation purposes. Of the four TS-SOM levels we chose to use the second from bottom, i.e. the one sized 64 × 64. On average, there were thus approximately 14 images mapped in each quantisation bin.

The quantisation bins were scored and sorted according to a function

$$S_i = S_i(N_i^+, N_i^-) = \begin{cases} \dfrac{N_i^+}{N_i^+ + N_i^-}, & \text{if } N_i^+ + N_i^- \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where N_i^+ and N_i^- are the numbers of positive and negative seen images, respectively, mapped in vector quantisation bin

i. From each 64 × 64-sized map an image subset of K images were selected in the order of descending S_i. If the largest S_i was greater than zero, images were picked from that bin until the limit of K images was reached. If there were not enough images in that bin the picking was continued in the bin of next largest S_i and so on. If the count K could not be filled from bins of positive S_i, the remainder were randomly picked from bins with zero S_i. This mode of operation is necessary in the end of an exhaustive query, when all images from positive-marked quantisation bins have already been used. Also, at the beginning of a query, randomly-picked images are in this system shown to the user as long as the first positive responses are received.

For the second selection stage (see Fig. 1) the reference system had two alternatives. Either the next shown images were selected based solely on their maximal first-stage scores, or the final selection was performed according to the sums of squared distances to the positive images seen, as in Eq. (3). In the latter case, the subsets of K images from each of M vector quantisers were combined so that only duplicate images were removed. In Eq. (3) we had set $w_m = 1$ for all features, and

$$d_m(\mathbf{f}^m(\mathcal{I}_l^*), \mathbf{f}^m(\mathcal{I}_n^+)) = \|(\mathbf{f}^m(\mathcal{I}_l^*) - \mathbf{f}^m(\mathcal{I}_n^+))\|^2 \tag{7}$$

i.e. the squared Euclidean distance.

6.3. Performance Measure

For measuring the retrieval performance, we have applied a quantitative figure denoted by us as the τ measure. For obtaining the τ value, it is assumed that the user is searching from a database $\mathcal{D}$ for an image I belonging to an image class $C \subset \mathcal{D}$. Before the correct image is found, the user guides the search by marking all shown images which belong to class C as relevant. This process is then repeated for each image in C. Now, the τ measure equals the average number of images the system retrieves before the correct one is found. The τ measure resembles the 'target testing' method presented by Cox et al [16], 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 or non-relevant, according to whether the images belong to class C determined in advance. This process is continued until all images in C have been found. For every image in the class, we thus obtain the number of images the system presented before that particular image was displayed. From this data, we then form a histogram and calculate the average number of shown images needed for the class. The τ measure for class C is obtained by dividing the average number of shown images by the size of the database, N.

In the optimal case, the system presents all images in class C before any other image. The minimum value for the average number of images presented before a particular image in C is found is thus $\frac{N_c}{2}$, where N_c is the cardinality of the class. Therefore

$$\tau \in \left[\frac{\rho_c}{2}, 1 - \frac{\rho_c}{2}\right] \tag{8}$$

where

$$\rho_c = \frac{N_c}{N}$$

is the *a priori* probability of class 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 size and contents of the initial image set shown by the system on the first query round has some effect on the resulting τ value. In our experiments, this set has been formed by random picking and the same set has been used in every test. Of course, the count of images seen before hit will be between zero and $N^* - 1$ for those images in the initial set that happen to belong to class C. Even though this brings some undeserved benefit for such particular images, it is not a serious problem, because the τ value is obtained as the average over the whole class. A more problematic situation takes place if the initial set does not contain any images from class C. In that case, it may take many iteration rounds before any relevant images will be seen, because the CBIR systems are always better off with positive than negative examples.

6.4. Results

In the experiments, we had two forms of both the PicSOM system and the reference system. Corresponding to Fig. 1, the PicSOM and reference systems differ in the way the image subsets are formed in the first selection stage. In the reference system it was possible to base the second-stage selection solely on the first-stage scores or to use in addition exhaustive distance calculations between the combined set images and the positive seen images. For the sake of completeness, we wanted to test these two alternatives also with the PicSOM system, where the default has been to employ the first-stage scores only.

We had the parameter values set as follows: K, the maximum number of images selected in the first stage from each SOM or vector quantiser, was set to 100. N^*, the number of images 'shown' to the 'ideal screener' on each round, was set to 20. This means that the iteration needed to be performed 3000 times before all images were retrieved.

The calculation of the τ measure was repeated twelve times in total, twice for both the PicSOM and reference systems and for the three image classes, *cars, faces* and *planes*. Table 1 shows the results of the experiments. The first result column shows the τ values for the baseline PicSOM system, and the next one for the modification, where additional distance calculations have been used in the second stage. The last two columns show the analogous results for the reference system.

It can be seen that the τ value for the baseline PicSOM system is in all cases better than the result with the additional distance calculations. On the contrary, the reference system without distance calculations always produces

Table 1. τ-measure results of the PicSOM and reference systems and their two variations for retrieving the three hand-picked image classes of the Corel database

τ	PicSOM	PicSOM+ distance	reference	reference+ distance
cars	0.177	0.193	0.212	0.187
faces	0.209	0.229	0.235	0.181
planes	0.137	0.147	0.203	0.185
average	0.174	0.190	0.217	0.184

results which are worse than those obtained when the distance calculations are being used. In two cases (*cars* and *planes*) the τ value of the baseline PicSOM system is better than that of the reference system. Also, on average, the PicSOM system is superior to the reference system. Very definitive conclusions and assessments cannot, of course, be drawn from a small-scale experiment like this.

7. CONCLUSIONS

We have shown that a powerful relevance feedback mechanism can be implemented by using Self-Organising Maps. The content-based image retrieval problem was first discussed in general terms, and then in connection with relevance feedback. We then introduced our PicSOM CBIR system, and elaborated on the way relevance feedback has been implemented in it with Tree Structured Self-Organising Maps. As the final part, the PicSOM system's performance was compared to that of a reference system which was, in other respects, similar to the original PicSOM system, but did not use Self-Organising Maps for implementing the relevance feedback. The results obtained with three hand-picked images classes showed that the PicSOM system outperformed the reference system on the average. This small study, however, only serves to show that the proposed relevance feedback technique is promising. A series of larger-scale experiments is needed to properly compare the PicSOM system with other existing CBIR systems.

Acknowledgements

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

References

1. Kohonen T. Self-Organizing Maps. Springer Series in Information Sciences, 30, Springer-Verlag, third edition, 2001
2. Oja E, Laaksonen J, Koskela M, Brandt S. Self-organizing maps for content-based image retrieval. In Oja E, Kaski S (eds), Kohonen Maps, Elsevier, 1999; 349–362
3. Laaksonen JT, Koskela JM, Laakso SP, Oja E. PicSOM – Content-based image retrieval with self-organizing maps. Pattern Recognition Letters 2000; 21(13,14):1199–1207
4. Eakins JP, Boardman JM, Graham ME. Similarity retrieval of trademark images. IEEE Multimedia 1998; 53–63
5. Jain AK, Vailaya A. Shape-based retrieval: A case study with trademark image databases. Pattern Recognition 1998; 31(9):1369–1390
6. Rui Y, Huang TS, Chang SF. Image retrieval: Current techniques, promising directions, and open issues. Journal of Visual Communication and Image Representation 1999; 10(1):39–62
7. Del Bimbo A. Visual Information Retrieval. Morgan Kaufmann, 1999
8. Gong Y. Intelligent Image Databases: Towards Advanced Image Retrieval. Kluwer Academic, 1998
9. Koikkalainen P, Oja E. Self-organizing hierarchical feature maps. Proceedings International Joint Conference on Neural Networks, San Diego, CA, 1990; II:279–284
10. Koikkalainen P. Progress with the tree-structured self-organizing map. 11th European Conference on Artificial Intelligence, August 1994
11. Chang N-S, Fu K-S. Query by pictorial example. IEEE Transactions on Software Engineering 1980; 6(6):519–524
12. Salton G, McGill MJ. Introduction to Modern Information Retrieval. McGraw-Hill, 1983
13. Leuski A. Relevance and reinforcement in interactive browsing. Proc. Ninth International Conference on Information and Knowledge Management (CIKM'00), Washington, DC, November 2000
14. Picard RW, Minka TP, Szummer M. Modeling user subjectivity in image libraries. Technical Report #382, MIT Media Laboratory, 1996
15. Squire D, Müller W, Müller H. Relevance feedback and term weighting schemes for content-based image retrieval. Third International Conference on Visual Information Systems, VISual'99, Amsterdam, The Netherlands, June 1999; 549–556
16. Cox IJ, Miller ML, Omohundro SM, Yianilos PN. Target testing and the PicHunter bayesian multimedia retrieval system. Advanced Digital Libraries ADL'96 Forum, Washington, DC, May 1996
17. Rui Y, Huang TS, Mehrotra S. Content-based image retrieval with relevance feedback in MARS. Proc. of IEEE Int. Conf. on Image Processing '97, Santa Barbara, CA, October 1997; 815–818
18. Rui Y, Huang TS, Ortega M, Mehrotra S. Relevance feedback: A power tool in interactive content-based image retrieval. IEEE Transactions on Circuits and Systems for Video Technology 1998; 8(5)
19. Schettini R, Ciocca G, Gagliardi I. Content-based color image retrieval with relevance feedback. Proc. Int. Conf. on Image Processing (ICIP'99) 1999; 3:75–79
20. Taycher L, La Cascia M, Sclaroff S. Image digestion and relevance feedback in the ImageRover WWW search engine. Proc. Visual 1997, San Diego, CA, December 1997
21. Minka TP. An image database browser that learns from user interaction. MS thesis, MIT, Cambridge, MA, 1996
22. Chua T-S, Low W-C, Chu C-X. Relevance feedback techniques for color-based image retrieval. Proc. Multimedia Modeling October 1998; 24–31
23. Benitez AB, Beigi M, Chang S-F. Using relevance feedback in content-based image metasearch. IEEE Internet Computing July–August 1998; 59–69
24. Santini S, Jain R. Similarity measures. IEEE Transactions on Pattern Analysis and Machine Intelligence 1999; 21(9):1–13
25. Kohonen T, Oja E, Simula O, Visa A, Kangas J. Engineering applications of the self-organizing map. Proceedings of the IEEE 1996; 84(10):1358–1384
26. Zhang H, Zhong D. A scheme for visual feature based image

indexing. Storage and Retrieval for Image and Video Databases III (SPIE), San Jose, CA, February 1995; 2420

27. Sethi IK, Coman I. Image retrieval using hierarchical self-organizing feature map. Pattern Recognition Letters 1999; 20:1337–1345
28. Ma WY, Manjunath BS. Texture features and learning similarity. Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, San Francisco, June 1996; 425–430
29. Han K-A, Myaeng S-H. Image organization and retrieval with automatically constructed feature vectors. SIGIR Forum (19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval), 1996; 157–165
30. Csillaghy A. Neural network-generated indexing features and retrieval effectiveness. Proceedings of the Converging Computing Methodologies in Astronomy (CCMA) Conference, Sonthofen, Bavaria, September 1997
31. Chen T, Chen L-H, Ma K-K. Colour image indexing using SOM for region-of-interest retrieval. Pattern Analysis & Applications 1999; 2:164–171
32. Kohonen T, Kaski S, Lagus K, Salojärvi J, Honkela J, Paatero V, Saarela A. Self organization of a massive text document collection. IEEE Transactions on Neural Networks 2000; 11(3):574–585
33. Brandt S, Laaksonen J, Oja E. Statistical shape features in content-based image retrieval. Proceedings of 15th International Conference on Pattern Recognition, Barcelona, Spain, September 2000; 2:1066–1069
34. The Corel Corporation World Wide Web home page, http://www.corel.com, 1999
35. Stricker M, Orengo M. Similarity of color images. Storage and Retrieval for Image and Video Databases III (SPIE), San Jose, CA, February 1995; 2420:381–392

Jorma Laaksonen received his Dr. of Science in Technology degree in 1997 from Helsinki University of Technology, Finland, where he is presently Senior Research Scientist at the Laboratory of Computer and Information Science. He is an author of several journal and conference papers on pattern recognition, statistical classification and neural networks. His research interests are in content-based image retrieval and recognition of handwriting. Dr Laaksonen is an IEEE member, a founding member of the SOM and LVQ Programming Teams and the PicSOM Development Group, and a member of the International Association of Pattern Recognition (IAPR) Technical Committee 3: Neural Networks and Machine Learning.

Markus Koskela received his MSc (Tech) degree in 1999 from Helsinki University of Technology, Finland, where he is presently a PhD student at the Laboratory of Computer and Information Science, PicSOM Development Group. His research interests are in neural networks, image processing and content-based image retrieval.

Sami Laakso received his MSc (Tech) degree in 2000 from Helsinki University of Technology, Finland, where he presently works as a Research Scientist at the Laboratory of Computer and Information Science. Being a member of the PicSOM Development Group, his research interests are in content-based image retrieval and Web structure utilisation.

Erkki Oja received his Dr. Sc. degree in 1977 from Helsinki University of Technology, Finland, where he is presently Professor of Computer Science and Director of the Neural Networks Research Centre. He has been Academy Professor of the Academy of Finland since August 2000. His research interests are in the study of principal components, independent components, self-organisation, statistical pattern recognition, and applying artificial neural networks to computer vision and signal processing. Dr Oja is an IEEE Fellow, IAPR Fellow, and President of the European Neural Network Society. He is member of the editorial boards of several journals, including *Neural Computation, IEEE Transactions on Neural Networks* and the *International Journal of Pattern Recognition and Artificial Intelligence.*

Correspondence and offprint requests to: J. Laaksonen, Laboratory of Computer and Information Science, Helsinki University of Technology, PO Box 5400, Fin-02015 HUT, Finland.