*published in the Proceedings of WSOM'99, Espoo, Finland, July 1999*
*available at <URL:http://www.cis.hut.fi/picsom/publications.html>*



# Self-Organizing Maps for Content-Based Image Database Retrieval

E. Oja, J. Laaksonen, M. Koskela, and S. Brandt

Laboratory of Computer and Information Science, Helsinki University of Technology
P.O. Box 5400, Fin-02015 HUT

We have developed a novel system for retrieving images similar to a given set of reference images in large image databases, based on Tree Structured Self-Organizing Maps (TS-SOMs). Our image retrieval system is called PicSOM. It has been designed with the purpose to provide a framework for generic research on algorithms and methods for content-based image retrieval. A new technique introduced in this paper facilitates automatic combination of the responses from multiple TS-SOMs and their hierarchical levels. Each TS-SOM is tuned with a different image feature representation like color, texture, or shape. This mechanism adapts to the user's preferences in selecting which images resemble each other, in the particular sense the user is interested in. The image queries are performed through the World Wide Web and the queries are iteratively refined as the system exposes more images to the user.

## 1. Introduction

Content-based image retrieval from unannotated image databases has been an object for ongoing research for a long period [1]. Digital image and video libraries are becoming more widely used as more visual information is produced at a rapidly growing rate. The technologies needed for retrieving and browsing this growing amount of information are still, however, quite immature and limited. This is an elusive scientific problem, still at the exploratory stage, with solid engineering solutions expected to appear in the future.

Many projects have been started in recent years to research and develop efficient systems for content-based image retrieval. The best-known system is Query By Image Content (QBIC) [2] developed at the IBM Almaden Research Center. Other notable systems include MIT's Photobook [3] and its more recent version, FourEyes [4], the search engine family of VisualSEEk [5], WebSEEk [6], and MetaSEEk [7], which all are developed at Columbia University, and Virage [8], a commercial content-based search engine developed at Virage Technologies Inc.

We introduce here a recently implemented image-retrieval system called PicSOM. It uses a World Wide Web browser as the user interface and the Tree Structured Self-Organizing Map (TS-SOM) [9,10] as the image similarity scoring method. The implementation of our PicSOM system is based on a general framework in which the interfaces of co-operating modules are defined. Therefore, the use of TS-SOMs is only one choice for the similarity measure. However, the results we have gained so far, are very promising on the potentials of the TS-SOM method.

As far as the current authors are aware, there has not been until now notable image retrieval applications based on the Self-Organizing Map (SOM) [11]. Some preliminary experiments with SOM have been made previously [12]. MIT's FourEyes image browser uses Self-Organizing Maps to cluster weights for different features [13].

## 2. Principle of PicSOM

Our method is named PicSOM due to its similarity to the well-known WEBSOM [14,15] document browsing and exploration tool that can be used in free-text mining. WEBSOM is a means for organizing miscellaneous text documents into meaningful maps for exploration and search. It is based on the Kohonen SOM [11] that automatically organizes documents into a two-dimensional grid so that related documents appear close to each other. Up to now, databases over one million documents have been organized for search using the WEBSOM system. In an analogous manner, we have aimed at developing a tool that utilizes the strong self-organizing power of the SOM in unsupervised statistical data analysis for digital images.

PicSOM is intended as a general framework for multi-purpose content-based image retrieval. The system is designed to be open and able to adapt to different kinds of image databases, ranging from small and domain-specific picture sets to large general purpose image collections. The features may be chosen separately for each specific task and the system may also use keyword-type textual information for the images, if available. In this paper, we describe the PicSOM system in its current form.

The basic operation of the PicSOM image retrieval is as follows: 1) An interested user connects to the WWW server providing the search engine with her web browser. 2) The system presents a list of databases available to that particular user. Later, there will also be a list of available search strategies; currently only the TS-SOM-based engine has been implemented. 3) After the user has selected the database, the system presents an initial set of tentative images scaled to a small "thumbnail" size. The user then 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. 4) The system marks the images selected by the user with a positive value and the non-selected images with a negative value in its internal data structure. Based on this data, the system then presents the user a new set of images along with the images selected this far. 5) The user again selects the relevant images, submits this information to the system and the iteration continues. Hopefully, the fraction of relevant images increases in each image set presented to the user and, finally, one of them is exactly what she was originally looking for.

### 2.1. Feature extraction

PicSOM may use one or several types of statistical features for image querying. Separate feature vectors can thus be formed for describing the colors, textures, and shapes in the images. A separate Tree Structured Self-Organizing Map is then constructed for each feature vector set and these maps are used in parallel to calculate the best-scoring similarity results. The feature selection is not restricted in any way and new features can be added to the system later on, as long as an equal number of features are calculated from each picture in the database.

*Color* is a natural and widely-used feature in content-based image retrieval. Common representations for color information in image retrieval include color histograms, color moments, color layouts and the recent color correlograms.

In PicSOM, average R-, G-, and B-values are calculated in five separate regions of the image, as seen in Figure 1. This division of the image area increases the discriminating power by providing a simple color layout scheme. The resulting 15-dimensional color feature vector thus not only describes the average color of the image but also gives information on the spatial color composition.

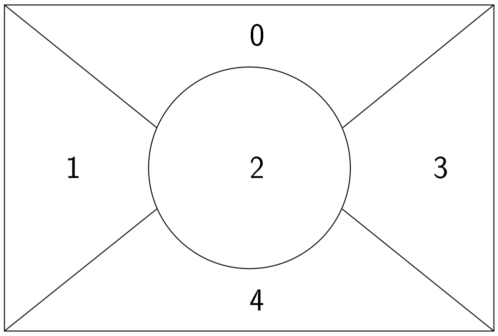


Figure 1. Image regions used calculating color and texture feature vectors.

*Texture* is an innate property of all surfaces and therefore a suitable feature for image retrieval. Texture features for pattern recognition and computer vision have been researched extensively over the past decades and the achievements in the field include co-occurrence matrices, multi-resolution simultaneous autoregressive (MRSAR) models, shift-invariant eigenvector (EV) models, the Wold decomposition, and wavelets, among others.

The texture feature vectors in PicSOM are calculated separately in the same five regions as the color features, shown in Figure 1. The Y-values of the YIQ color representation of every pixel's 8-neighborhood are examined and the estimated probabilities for each neighbor pixel being brighter than the center pixel are used as features. This results in five eight-dimensional vectors which are combined to one 40-dimensional texture feature vector.

*Shape* features can also be used in content-based image indexing. In PicSOM, various shape-describing features have been experimented with. They are all formed from a thresholded binary edge image obtained by convolving the image with Sobel masks of size $3 \times 3$. The edge filtration is performed on the saturation and intensity components of the HSI color presentation and the resulting two binarized edge images are then logically ored to form the edge image. An example of the images used in the experiments is seen in Figure 2. The corresponding edge image is displayed in Figure 3.

The first shape features are based on the histogram of the eight quantized directions of the edges in the image. When the histogram is separately formed in all the five regions seen in Figure 1, 40-dimensional feature vectors are obtained. They describe the distribution

Figure 2. An example of the images used in the experiments.

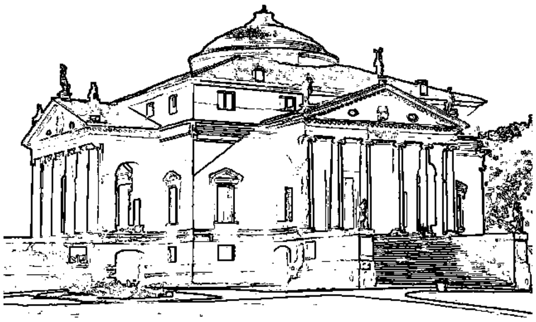

Figure 3. Edges extracted from Figure 2 and used while forming the shape features.

of edge directions in various parts of the image and thus reveal the shape in a low-level statistical manner. The second shape features are formed from the co-occurrence matrix of neighboring edge elements. As the number of quantized directions is again eight, 320-dimensional vectors are obtained.

The third and fourth shape-describing features are based on the Fourier Transform of the binarized edge image. The image sizes are normalized to $512 \times 512$ pixels before FFT. The 2-dimensional amplitude spectrum is then smoothed and down-sampled to form feature vectors of length 512 coefficients. The formation of the fourth set of shape features is otherwise similar but the edge image is transferred from the Cartesian coordinates to polar coordinates before FFT.

### 2.2. Tree Structured SOM (TS-SOM)

The Tree Structured Self-Organizing Map (TS-SOM) [9,10] is a tree-structured vector quantization algorithm that uses Self-Organizing Maps (SOMs) [11] at each of its hierarchical levels. In PicSOM, all TS-SOM maps are two-dimensional. The number of map units increases when moving downwards in the TS-SOM. The search space on the underlying SOM level is restricted to a predefined portion just below the best-matching unit on the above SOM. Therefore, the complexity of the searches in TS-SOM is remarkably lower than if the whole bottommost SOM level were accessed without the tree structure. The structure of TS-SOM is illustrated in Figure 4.

The computational lightness of TS-SOM facilitates the creation and use of huge SOMs which, in our PicSOM system, are used to hold the images stored in the image database. The feature vectors (color, texture, or shape) calculated from the images are used to train the levels of the TS-SOMs beginning from the top level. During the training, each feature vector is presented to the map multiple times and the model vectors stored in the map units are modified to match the distribution and topological ordering of the feature vector space. After the training phase, each unit of the TS-SOMs contains a model vector which may be regarded as the average of all feature vectors mapped to that particular unit. In PicSOM, we then search in the corresponding data set for the feature vector which best matches the stored model vector and associate the corresponding image to that map unit. Consequently, a tree-structured hierarchical representation of all the images in the

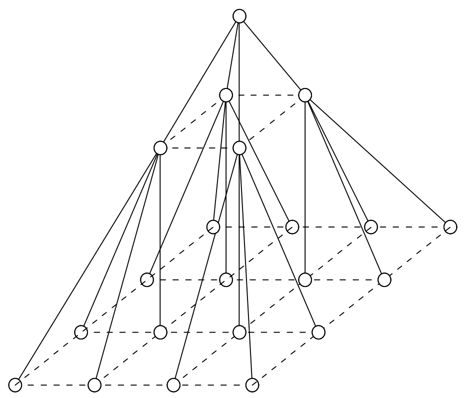

Figure 4. The structure of a three-layer two-dimensional TS-SOM.

database is formed. In an ideal situation, there should be one-to-one correspondence between the images and TS-SOM units in the bottom level of each map.

### 2.3. Using multiple TS-SOMs

Combining the results from several feature maps can be done in a number of ways. A simple method would be to ask the user to enter weights for different maps and then calculate a weighted average. This, however, requires the user to give information which she normally does not have. Generally, it is a difficult task to give low-level features such weights which would coincide with human perception of images at a more conceptual level. Therefore, a better solution is to combine the results of multiple maps automatically, using the implicit information from the user's responses during the query. The PicSOM system thus tries to learn the user's preferences from the interaction with her and sets its own responses accordingly.

The rationale behind our approach is as follows: If the images selected by the user map close to each other on a TS-SOM map, it seems that the corresponding feature performs well on the present query and the relative weight of its opinion should be increased. This can be implemented simply by marking on the maps the images shown to the user until now with positive and negative values depending whether she has selected or rejected them, respectively. The mutual relations of positively-marked units residing near to each other can then be enhanced by convolving the maps with a simple low-pass filtering mask. As a result, those areas which have many positively marked images spread the positive response to their neighboring map units. The images associated with these units are then good candidates for next images to be shown to the user, if they have not been shown already. The current PicSOM implementation uses convolution masks whose values decrease as the 4-neighbor or "city-block" distance from the mask center increases. The convolution mask size increases as the size of SOM layer increases.

Figure 5 shows a set of convolved feature maps during a query. The three images on the left represent three map levels on the Tree Structured SOM for the RGB color feature, whereas the convolutions on the right are calculated on the texture map. The sizes of the SOM layers are $4 \times 4$, $16 \times 16$, and $64 \times 64$, from top to bottom. The dark regions have positive and the light regions negative convolved values on the maps. Notice the dark regions in the lower-left corners of the three layers of the left TS-SOM. They indicate that there is a strong response and similarity between images selected by the user in that particular area of the color feature space.

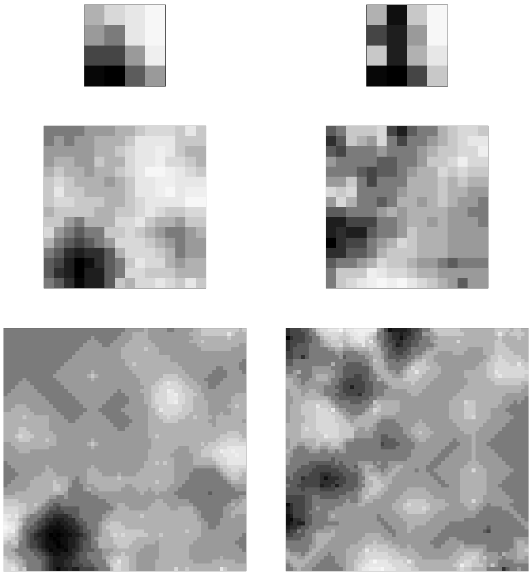

Figure 5. An example of convolved TS-SOMs for color (left) and texture (right) features. Black corresponds to positive and white to negative convolved values.

### 2.4. Refining queries

In our current PicSOM implementation, all positive values on all convolved TS-SOM layers are sorted in descending order in one list. Then, a preset number, e.g. 15, of the best candidate images which have not been shown to the user before are output as a new tentative image selection. Image retrieval with PicSOM is therefore an iterative process in which new images get selected or rejected by the user.

Initially, the query begins with a set of reference images picked from the top levels of the TS-SOMs in use. The SOM map units associated with the selected and rejected images get positive and the negative values, respectively. The positive and negative responses are normalized so that their sum equals to zero. Previously positive map units can also be changed to negative as the retrieval process iteration continues. In early stages of the image query, the system tends to present the user images from the upper TS-SOM levels. As soon as the convolutions begin to produce large positive values also on lower map levels, the images on these levels are shown to the user. The images are therefore gradually picked more and more from the lower map levels as the query is continued.

The inherent property of PicSOM to use more than one reference image as the input information for retrievals is important. This feature makes PicSOM different from other content-based image retrieval systems, such as QBIC, which uses only one reference image at a time.

## 3. Implementation of PicSOM

The issues of the implementation of the PicSOM image retrieval system can be divided in two categories. First, concerning the user interface, we have wanted to make our search engine, at least in principle, available and freely usable to anybody by implementing it in the World Wide Web. This also makes the queries on the databases machine independent, because the standard web browsers can be used. Second, the functional components in the server running the search engine have been implemented so that the parts responsible for separate tasks have been isolated to separate processes. The functional interfaces between these processes have then been designed to be open and easily extensible to allow the inclusions of new features in the system in future.

### 3.1. User interface

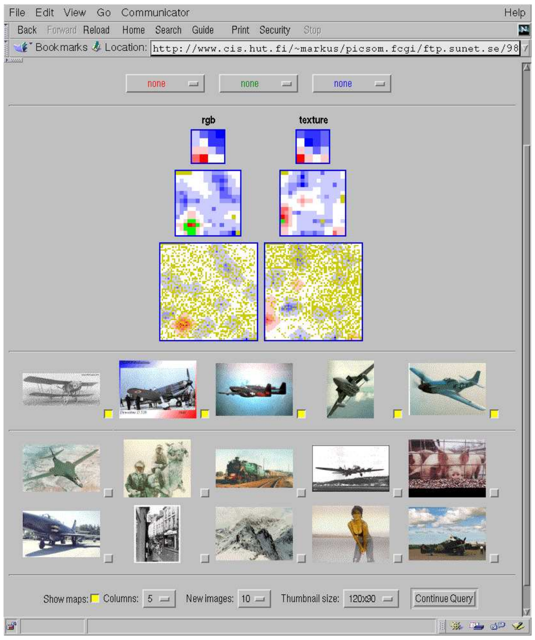


Figure 6. WWW-based user interface of PicSOM. The user has already selected five aircraft images in the previous rounds. The system is displaying the user ten new images to select of.

Figure 6 shows a screenshot of the current web-based PicSOM user interface, which can be found at *http://www.cis.hut.fi/picsom/*. On the top of the page, there are three pull-down menus for examining class information on the RGB color bands, if that information is available for the particular database. The convolved feature maps are shown next on the page. In this query, RGB color and texture maps have been used as seen on the labels above the maps. On color terminals, positive map points are seen as blue and negative as red. White represents zero values.

The first row displays images selected on previous rounds of the retrieval process. This example shows a query with five images of airplanes selected. The next images are the ten best-scoring new images obtained from the convolved units in the TS-SOMs. It seems that these ten images contain four airplanes. Finally, the page has some user-modifiable settings and a "Continue Query" button which submits the new selections back to the search engine.

The user can at any time switch from the iterative queries to examine the TS-SOM map surfaces simply by clicking the map images. Relevant images on the maps can then also be selected for continuing queries.

### 3.2. Parts of the PicSOM system

The current computer implementation of PicSOM has three separate modular components:

**picsom.cgi** is a CGI/FCGI script which handles the requests and responses from the user's web browser. This includes processing the HTML form, updating the information from previous queries and executing the other components as needed to complete the requests.

**picsomctrl** is the main program responsible for updating the TS-SOM maps with new positive and negative response values, calculating the convolutions, creating new map images for the next web page, and selecting the next best-scoring images to be shown to the user in the next round.

**picsomctrltohtml** creates the HTML contents of the new web pages based on the output of the *picsomctrl* program.

Figure 7 illustrates the components of the current PicSOM system and the operations needed in handling the queries. The numbers indicate the usual order of actions.

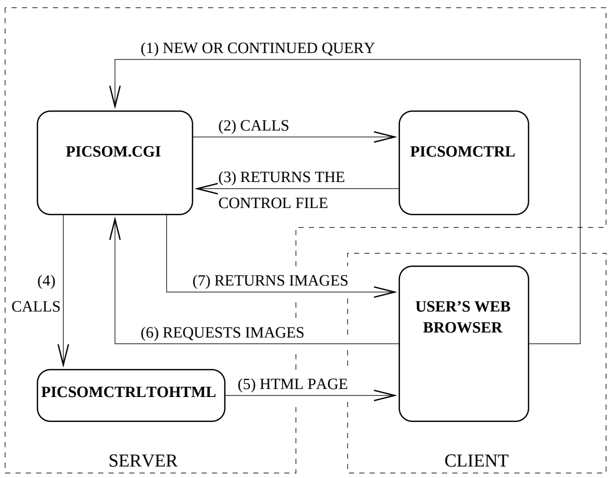


Figure 7. The components of the PicSOM system and the operations performed in handling the queries.

## 4. The experimental image database

Currently, we have made experiments with an image database of 4350 images. Most of them are color photographs in JPEG format. The images were downloaded from the image collection residing at the Swedish University Network FTP server, located at *ftp://ftp.sunet.se/pub/pictures/*.

PicSOM also supports the utilization of textual class information for the images, if that kind of information is available in the database. The original directory structure of the collection has been used to give the images rough textual content classes. Figure 8 shows a tree-form representation of a small subset of the used classes. The classes on child nodes are subclasses of the classes on their father nodes. For instance, {"cars"} ⊂ {"vehicles"}.

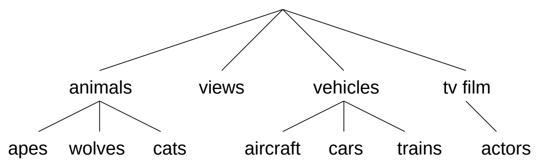


Figure 8. A subset of the image classes in the *ftp.sunet.se* database.

In the user interface, the convolved TS-SOM map views can be changed to maps colored with this external information of the image content. The three color bands in the RGB color space can be used to visualize the spreads of three individual classes on the maps.

## 5. Quantitative results

A number of measures to evaluate various visual features are presented in this section. Assume a database $\mathcal{D}$ containing a total of $N$ images, and an image class $\mathcal{C} \subset \mathcal{D}$ with $N_\mathcal{C}$ relevant images. Then, the *a priori* probability $\rho_\mathcal{C}$ of the class $\mathcal{C}$ is

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

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

### 5.1. Observed probability

For each image $I \in \mathcal{C}$ with a feature vector $\mathbf{f}^I$, we calculate the Euclidean distance $d_{L_2}(I, J)$ of $\mathbf{f}^I$ and the feature vectors $\mathbf{f}^J$ of the other images $J \in \mathcal{D} \setminus \{I\}$ in the database. Then, we sort the images based on their ascending distance to the image $I$ and store the indices of the images in a $(N-1)$-sized vector $\mathbf{g}^I$. We now have a vector $\mathbf{g}^I$ for each $I \in \mathcal{C}$ containing a sorted permutation of the images in $\mathcal{D} \setminus \{I\}$ based on their increasing Euclidean distance to $I$. By $g_i^I$, we denote the $i$th component of $\mathbf{g}^I$.

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

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

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

We can now define the *observed probability* $p_i$:

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

The observed probability $p_i$ is a measure between $[0, 1]$ of the probability that a given image $K \in \mathcal{C}$ has an image belonging to the class $\mathcal{C}$ as the $i$:th nearest image according to the feature vector $\mathbf{f}$. A good feature should cluster similar images (in this case, images belonging to $\mathcal{C}$) close to each other and thus the value of $p_i$ should be high for small values of $i$ and decrease monotonically as $i$ grows.

In the optimal case, $p_i^* = 1$ if $i \leq N_\mathcal{C} - 1$, and $p_i^* = 0$ if $i > N_\mathcal{C} - 1$. This is equivalent to the situation, where all the images in class $\mathcal{C}$ are clustered together. In this case, the shortest distance to the closest image not in $\mathcal{C}$ is always greater than any of the inter-class distances. This is obviously very rarely the case with the low-level features currently used in content-based image retrieval. Still, $p_i$ can be used as a comparable performance measure for different features. The worst case happens when the feature $\mathbf{f}$ completely fails to discriminate the images in class $\mathcal{C}$ from the images which do not belong to the class $\mathcal{C}$. The observed probability $p_i$ is then close to the *a priori* $\rho_\mathcal{C}$ for every value of $i \in [1, N-1]$.

### 5.2. Weighting the observed probability

The observed probability $p_i$ is a function of the distance $i$, so it cannot easily be used to compare two different features $\mathbf{f_1}$ and $\mathbf{f_2}$. Therefore, it is useful to derive a scalar metric from $p_i$ to enable us to do such comparisons directly.

As the large values of $p_i$ with small values $i$ and small values of $p_i$ with large values $i$ correspond to good discriminating power of the feature $\mathbf{f}$, the weighting function $h(u, x)$ should respectively reward the large values of $p_i$ when $i$ is small and punish the large values of $p_i$ when $i$ is large. The Discrete Fourier Transform (DFT) $P(u)$ of the observed probability function $p_i$ can be used in forming measures for the discriminating power. $P(u)$ is calculated as follows:

$$P(u) = \sum_{i=0}^{N-1} p_i \, h(u, i) = \sum_{i=0}^{N-1} p_i \, e^{j2\pi iu/N} \ . \tag{4}$$

The weighting function $h(u, x)$ equals now a point on the complex value unit circle with the angle $\varphi = 2\pi iu/N$. The DFT is defined only for integer values of $k$, but in our weighting purposes, there is no reason for this limit. For example, with the parameter value $u = \frac{1}{2}$, the weighting function $h(\frac{1}{2}, x)$ rotates in $N$ steps from $\varphi = 0$ to $\varphi = \pi$, which equals the upper half of the unit circle. The values of $h(1, x)$ are distributed on the whole unit circle, respectively. With the parameter value $u = 0$, the metric reduces to the sum of the probabilities $p_i$ and is useful in normalizing the values of $P(u)$.

Suitable values of $u$ could then include, for example, $u = \frac{1}{2}$ and $u = 1$ and performance metrics could include $\mathrm{Re}\{P(u)/P(0)\}$, $\mathrm{Im}\{P(u)/P(0)\}$, $\mathrm{Abs}\{P(u)/P(0)\}$ and $\mathrm{Arg}\{P(u)/P(0)\}$, representing the real part, imaginary part, absolute value, and the angle of $P(u)/P(0)$, respectively.

### 5.3. Comparison of feature extraction methods

In order to assess the indexing ability of the color, texture, and four shape features currently used in PicSOM, a series of experiments were performed. We chose to use two figures of merit to describe the performances. First, a local measure calculated as the average of the observed probability $p_i$ for the first 50 retrieved images, i.e.:

$$\eta_{\mathrm{local}} = \frac{\sum_{i=1}^{50} p_i}{50} \tag{5}$$

The $\eta_{\mathrm{local}}$ measure obtains values between zero and one. Figures near one can be obtained even though the classes were globally split into many clusters if each of these clusters are separate from the clusters of the other classes. On the other hand, for a global figure of merit we used the weighted sum of the observed probability $p_i$ calculated as:

$$\eta_{\mathrm{global}} = \mathrm{Re}\{P(1/2)/P(0)\} \tag{6}$$

This figure again attains values between zero and one. It favors observed probabilities that are concentrated in small indices and additionally punishes for large probabilities in large index values as described above.

Table 1
Comparison of the performances of different features for different image classes

| | Image classes | | | | | |
|---|---|---|---|---|---|---|
| | aircraft (0.08) | | buildings (0.11) | | faces (0.08) | |
| Feature types | $\eta_{\mathrm{local}}$ | $\eta_{\mathrm{global}}$ | $\eta_{\mathrm{local}}$ | $\eta_{\mathrm{global}}$ | $\eta_{\mathrm{local}}$ | $\eta_{\mathrm{global}}$ |
| RGB | 0.19 | 0.22 | 0.21 | 0.20 | 0.13 | 0.11 |
| Texture | 0.15 | 0.10 | 0.28 | 0.23 | 0.12 | 0.03 |
| Shape Histogram | 0.35 | 0.46 | 0.30 | 0.16 | 0.15 | 0.22 |
| Shape Co-occurrence | 0.36 | 0.48 | 0.35 | 0.13 | 0.15 | 0.25 |
| Shape FFT | 0.26 | 0.42 | 0.41 | 0.30 | 0.22 | 0.30 |
| Shape Polar FFT | 0.20 | 0.29 | 0.21 | 0.35 | 0.19 | 0.20 |

Three hand-picked image classes were used in the experiments. These were: aircraft, building, and human face views. The results are shown in Table 1. The figures in parenthesis after the class names are the *a priori* probabilities of the classes. As could be expected, no single feature extraction method performs well for all the three classes. For example, the shape FFT features are better than others for buildings and faces but show relatively worse performance for aircraft images. The shape co-occurrence features seem to suit well for the aircraft class whereas their global performance in the buildings class is poor. In general, the local and global merit figures seem to agree in most evaluations.

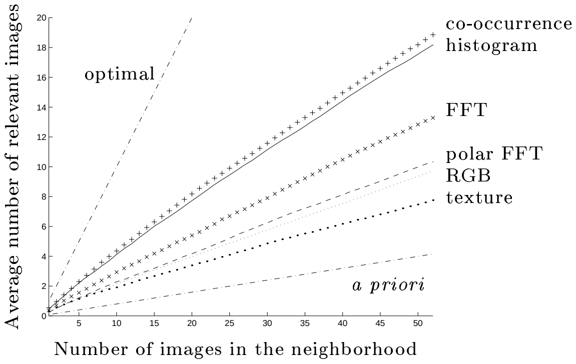


Figure 9. The slopes of the average cumulation of relevant images for different features and the aircraft images.

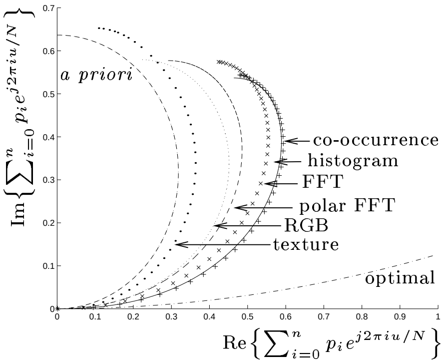


Figure 10. The accumulation of the global merit of indexing $\eta_{\text{global}}$ for the aircraft image class.

Figures 9 and 10 illustrate the calculation of the $\eta_{\text{local}}$ and $\eta_{\text{global}}$ values, respectively. The slopes in Figure 9 can be verified to agree with the average observed probabilities in the 'aircraft' column of Table 1. Accordingly, the real parts of the endpoints of the weighted probability curves in Figure 9 are those tabulated as the $\eta_{\text{global}}$ values in the 'aircraft' column.

### 5.4. Retrieval precision of the SOMs

Since we are using the Self-Organizing Map as the image indexing tool, we can use the map ordering directly to evaluate feature performance. The SOM algorithm organizes similar input vectors near to each other. Therefore, with a good feature, similar images should be attached either to the same or to a neighboring map unit on the SOMs. For each image $I \in \mathcal{C}$, the precision $\mathcal{P}(r)$, i.e. the fraction of relevant images in the total number of retrieved images, is calculated for different neighborhood distances $r$. The images belonging to $\mathcal{C}$ are the relevant images. Then, the average precision $\mathcal{P}_{avg}(r)$ is calculated and used to measure the retrieval performance of the used feature. Regardless of the used distance function, the precision $\mathcal{P}(r)$ should then be high when $r$ is small and decrease as the distance $r$ becomes greater, eventually dropping below the *a priori* probability $\rho$.

It is also possible to calculate the observed probability function $p_i$ for the SOMs. With the SOM the nearest feature vectors are located in the same or neighboring map units, and the complexity of the search is significantly smaller than in the full search of all the original data. On the other hand, the SOM does perform some averaging on the feature vectors and, as a result, its performance cannot match the direct use of the Euclidean distance in the $n$-dimensional feature space. First, several map units have equal distance to the current map unit. For example, the map units directly north, south, west, and east of the current map unit have all the same distance. So, the map units cannot be ordered and must be considered equal. Second, the distances between vectors mapped into a same SOM unit are not preserved. Therefore, it is not possible to sort these vectors unambiguously based on the distance function.

The observed probability $p_i$ for the Self-Organizing Maps is created otherwise similarly as above for the original data but the correctness values $h_i^I$ of the vectors with equal distances are averaged. The resulting averaged value is then used to replace all the original values in the probability function. This will undoubtedly somewhat worsen the resulting probability function. Figure 11 displays the observed probabilities for the original data and the 64 × 64-sized bottommost TS-SOM layer in the case of the RGB features and aircraft images. The calculation of the $\eta_{\text{global}}$ value of merit for the same observed probabilities is illustrated in Figure 12. It can be seen that the performance measures of the original data are somewhat superior to those of the SOM, i.e. $\eta_{\text{local}}(\text{SOM})=0.16<0.19$ and $\eta_{\text{global}}(\text{SOM})=0.17<0.22$.

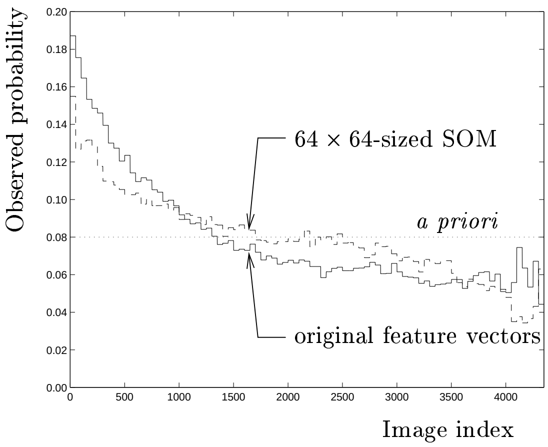


Figure 11. The observed probabilities of the original RGB feature vectors and the bottommost TS-SOM layer formed thereof for the aircraft image class.

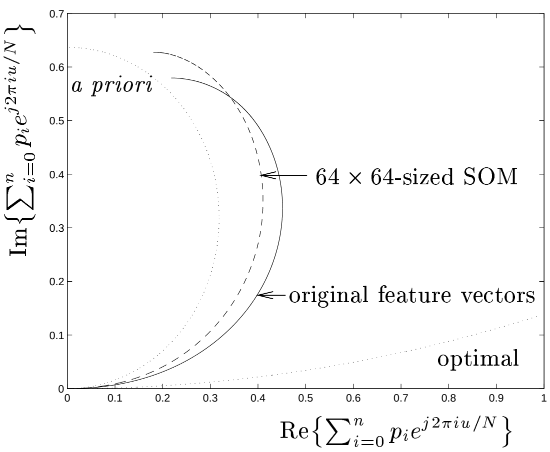


Figure 12. The accumulation of the global merit of indexing $\eta_{\text{global}}$ for the original RGB features and the bottommost TS-SOM layer in the case of aircraft images.

## 6. Conclusions and future plans

Following some of the core principles of the WEBSOM text document exploration tool [14], we have developed an image database retrieval and browsing system called PicSOM. It uses several feature representations for a digital image, currently the spatial color, texture, and shape compositions, and several feature maps structured according to the TS-SOM method [9,10]. In preliminary experiments, the PicSOM system does show potential and we are confident that it can evolve into a usable and fully functional tool for image retrieval. A generic problem in image database search methods is how to measure their performance. The same quantitative measurements that have been used for PicSOM could be applied to other content-based image retrieval systems to facilitate fair evaluations and comparisons. The MPEG organization [16] has started to work on a new standard, called MPEG-7, to develop a set of features for image content description. The organization also plans a standard testbed for image retrieval applications, that will later be used to assess the performance of the fully functional PicSOM system.

In order to study our method's applicability for larger image databases, we have started experimenting with the Corel Gallery [17]. Another vast collection of images is spread in the Internet, and we have plans to use PicSOM as an image search engine for the World Wide Web.

## REFERENCES


1. Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Past, present and future. *Journal of Visual Communication and Image Representation*, 1998. To appear.
2. M. Flickner, H. Sawhney, W. Niblack, et al. Query by image and video content: The QBIC system. *IEEE Computer*, pages 23–31, September 1995.
3. A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *Storage and Retrieval for Image and Video Databases II*, volume 2185 of *SPIE Proceedings Series*, San Jose, CA, USA, 1994.
4. T. P. Minka. An image database browser that learns from user interaction. Master's thesis, M.I.T, Cambridge, MA, 1996.
5. J. R. Smith and S.-F. Chang. VisualSEEk: A fully automated content-based image query system. In *Proceedings of the ACM Multimedia 1996*, Boston, MA, 1996.
6. J. R. Smith and S.-F. Chang. Searching for images and videos on the world-wide web. Technical Report #459-96-25, Columbia University, 1996.
7. A. B. Benitez, M. Beigi, and S.-F. Chang. Using relevance feedback in content-based image metasearch. *IEEE Internet Computing*, pages 59–69, July-August 1998.
8. A. Gupta. Visual information retrieval technology: A Virage perspective. Available online at `http://www.virage.com/wpaper/`, 1997.
9. P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. In *Proceedings of 1990 International Joint Conference on Neural Networks*, volume II, pages 279–284, San Diego, CA, 1990. IEEE, INNS.
10. P. Koikkalainen. Progress with the tree-structured self-organizing map. In A. G. Cohn, editor, *11th European Conference on Artificial Intelligence*. European Committee for Artificial Intelligence (ECCAI), John Wiley & Sons, Ltd., 1994.
11. T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, 1997. Second Extended Edition.
12. H. Zhang and D. Zhong. A scheme for visual feature based image indexing. In *Storage and Retrieval for Image and Video Databases III (SPIE)*, volume 2420 of *SPIE Proceedings Series*, San Jose, CA, February 1995.
13. T. P. Minka and R. W. Picard. Interactive learning using a 'society of models'. Technical Report #349, M.I.T Media Laboratory, 1995.
14. WEBSOM - self-organizing maps for internet exploration, `http://websom.hut.fi/websom/`.
15. T. Honkela, S. Kaski, K. Lagus, and T. Kohonen. WEBSOM—self-organizing maps of document collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 310–315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997.
16. The moving picture experts group MPEG home page, `http://www.cselt.it/mpeg/`.
17. The Corel Corporation Home Page, `http://www.corel.com/`.