published in the Proceedings of CIR2000, Brighton, UK, May 2000
available at <URL:http://www.cis.hut.fi/picsom/publications.html>

The PicSOM Retrieval System: Description and Evaluations

Markus Koskela, Jorma Laaksonen, Sami Laakso, and Erkki Oja
Laboratory of Computer and Information Science, Helsinki University of Technology
P.O.BOX 5400, Fin-02015 HUT, Finland

Abstract

We have developed an experimental system called PicSOM for retrieving images similar to a given set of reference images in large unannotated image databases. The technique is based on a hierarchical variant of the Self-Organizing Map (SOM) called the Tree Structured Self-Organizing Map (TS-SOM). Given a set of reference images, PicSOM is able to retrieve another set of images which are most similar to the given ones. Each TS-SOM is formed using a different image feature representation like color, texture, or shape. A new technique introduced in PicSOM facilitates automatic combination of the responses from multiple TS-SOMs and their hierarchical levels. This mechanism adapts to the user's preferences in selecting which images resemble each other. In this paper, a brief description of the system and a set of methods applicable to evaluating retrieval performance of image retrieval applications are presented.

1 Introduction

Content-based image retrieval (CBIR) has been a subject for active research since the initial releases of the first notable CBIR systems such as QBIC [3] and Photobook [10] in the mid 90's. The task of developing effective products based on CBIR has, however, proven to be extremely difficult. Due to the limitations of computer vision, the current CBIR systems have to rely only on rather low-level features extracted from the images. Therefore, images are typically described by rather simple features characterizing the color content, different textures, and primitive shapes detected in them. On the other hand, humans can easily detect distinct objects and use high-level semantic concepts in image recognition. As a result, humans routinely group together images which can be visually very different and, for a computer, mimicking this behavior is a very challenging task.

For the development of effective image retrieval applications, one of the most urgent issues is to have widely-accepted performance assessment methods for different features and approaches. However, quantitative measures for the performance of an image retrieval system are problematic due to the subjectivity of human perception. As each user of a retrieval system has individual expectations, there does not exist a definite right answer to an image query. Also, there does not exist any widely accepted performance assessment methods. As a result, objective and quantitative comparisons between different algorithms or image retrieval systems based on different approaches are difficult to perform. Due to the lack of standard methods in this application area, the Moving Picture Experts Group (MPEG) has also started to work on a content representation standard for multimedia information search, filtering, management and processing called MPEG-7 [8].

2 PicSOM

The PicSOM image retrieval system is designed as a framework for generic research on algorithms and methods for content-based image retrieval. The system is based on querying by pictorial example (QBPE), which is a common retrieval paradigm in current CBIR applications. With QBPE, the queries are based on example images shown either from the database itself or some external location. The user identifies these example images as relevant or non-relevant to the current retrieval task and the system uses this information to select such images the user is most likely to be interested in. The accuracy of the queries is then improved by relevance feedback [11] which is a form of supervised learning adopted from traditional text-based information retrieval. In relevance feedback, the previous human-computer interaction is used to refine subsequent queries to better approximate the need of the user.

In the current PicSOM implementation, queries are performed through a WWW-based user interface. PicSOM supports multiple parallel features and with a technique introduced in the PicSOM system, the responses from different features are combined automatically. This is useful, as the user is not required to enter weights for the used features. The goal is to autonomously adapt to the user's preferences regarding the similarity of images in the database by iteratively refining the queries as the system exposes more images to the user. A more detailed description of the system than presented here can be found in [7, 9]. The PicSOM home page including a working demonstration of the system is located at *http://www.cis.hut.fi/picsom*.

2.1 The Self-Organizing Map

The image indexing method used in PicSOM is based on the Self-Organizing Map (SOM) [4]. The SOM defines an elastic net of points that are fitted to the input space. It can thus be used to visualize multidimensional data, usually on a two-dimensional grid. The SOM consists of a regular grid of neurons where a model vector m_i is associated with each map unit i. The map attempts to represent all the available observations with optimal accuracy using a restricted set of models. At the same time, the models become ordered on the grid so that similar models are close to each other and dissimilar models far from each other.

Fitting of the model vectors is usually carried out by a sequential regression process, where $t = 1, 2, \ldots$ is the step index: For each sample $x(t)$, first the index $c = c(x)$ of the best-matching unit (BMU) is identified by the condition

$$\forall i: \ \ \|x(t) - m_c(t)\| \leq \|x(t) - m_i(t)\| \ . \tag{1}$$

After that, all model vectors or a subset of them that belong to nodes centered around node $c(x)$ are updated as

$$m_i(t+1) = m_i(t) + h(t)_{c(x),i}(x(t) - m_i(t)) \ . \tag{2}$$

Here $h(t)_{c(x),i}$ is the "neighborhood function", a decreasing function of the distance between the ith and cth nodes on the map grid. This regression is then reiterated over the available samples and, to guarantee the convergence of the unit vectors, the value of $h(t)_{c(x),i}$ is let to decrease in time.

2.2 The Tree Structured SOM

In order to achieve a hierarchical representation of the image database and to alleviate the computational complexity of large SOMs, we use a variant of the SOM called the Tree Structured Self-Organizing Map (TS-SOM) [5, 6]. The TS-SOM is used to represent the database in several hierarchical two-dimensional grids of neurons where each grid is a standard SOM. The tree structure reduces the time complexity of the BMU search from $O(N)$ to $O(\log N)$. The complexity of the search is thus remarkably lower than if the whole large bottommost SOM level had been accessed without the tree structure. The structure of a two-dimensional TS-SOM with three SOM levels is illustrated in Figure 1.

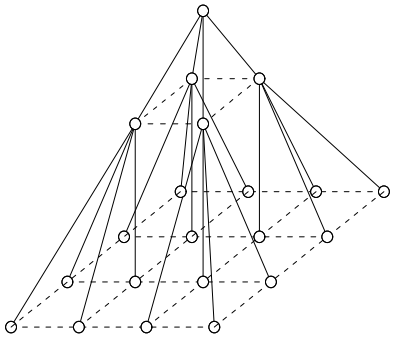

Figure 1: The structure of a three-level two-dimensional TS-SOM.

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 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, the images are distributed to the TS-SOMs according to their best-matching map units. The model vector of each map unit may thus be regarded as the average of all feature vectors mapped to that particular unit. Consequently, a tree-structured hierarchical representation of all the images in the database is formed.

For each map unit, we then search in the corresponding data set for the feature vector which best matches the model vector and associate the corresponding image as the reference image of that map unit. 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 Image Querying

Image retrieval with PicSOM is an iterative process utilizing the relevance feedback approach. The query begins with a fixed selection of representative images uniformly picked from the top levels of the TS-SOMs. On subsequent rounds, the query is focused more accurately to the user's present need as she selects the subset of images which best match her expectations and to some degree of relevance fit to her purposes. Query improvement is achieved as the system learns the user's preferences from the selections made on the previous rounds.

The system marks the images selected by the user with a positive value and the non-selected images with a negative value in its internal data structure. Based on this information, the system presents the user a new set of images aside with the images selected so far. The rationale behind the PicSOM approach is as follows: If the selected images map close to each other on a TS-SOM map, it seems that the corresponding feature performs well in the present query and the relative weight of its opinion should be increased. This can be implemented in practice by marking the locations of the previously shown images on the maps either with positive or negative values, depending on whether the user has selected or rejected the corresponding image. The responses are normalized so that their sum equals to zero.

Each SOM level is then treated as a two-dimensional matrix formed of values describing the user's responses to the contents of the seen map units. Then, the map matrices are low-pass filtered with symmetrical convolution masks in order to spread the responses to the neighboring map units which, by presumption, contain images that are to some extent similar to the present ones. Starting from the SOM unit having the largest positive convolved response, PicSOM

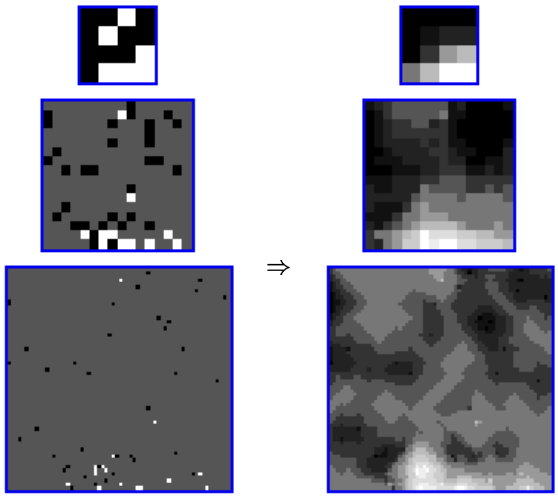

Figure 2: An example of converting the positive and negative map units to convolved maps in a three-level TS-SOM. Map surfaces displaying the positive (white) and negative (black) map units are shown on the left. The resulting convolved maps are shown on the right.

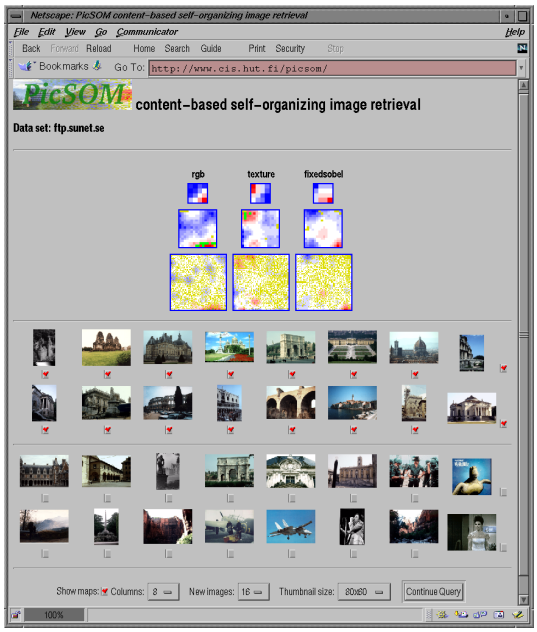

Figure 3: The PicSOM user interface.

retrieves the reference image associated with that map unit. If that image has not already been shown to the user, it is presented on the next round. This process is continued with the map unit having the second largest response and so on until a preset number of new images have been gathered. This set of images is then presented to the user.

The conversion from positive and negative marked images to the convolutions in a three-level TS-SOM is visualized in Figure 2. First, a TS-SOM displaying the positive map units as white and negative as black is shown on the left. These maps are then convolved and the resulting map surfaces are shown on the right. The white and black regions in the convolved TS-SOMs illustrate positive and negative values after the convolution, respectively. On the next round, the image query is, therefore, targeted to the white parts of the maps.

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

2.4 User Interface

The current PicSOM user interface in a midst of an ongoing query is displayed in Figure 3. First, the three parallel TS-SOM map structures represent three map levels of SOMs trained with RGB color, texture, and shape features, from left to right. The sizes of the SOM levels are 4×4, 16×16, and 64×64, from top to bottom. Below the convolved SOMs, the first set of images consists of images selected as positive on the previous rounds of the retrieval process. These images may be unselected on any subsequent round, thus changing their contribution from positive to neutral. In this example, a query with a set of images representing buildings selected as positive is displayed. The next images, separated with a horizontal line, are the current 16 best-scoring new images obtained from the convolved map units in the TS-SOMs.

3 Evaluating Retrieval Performance

A number of measures for evaluating the ability of various visual features to reveal image similarity 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 $N_\mathcal{C}$ somehow similar images. Then, the *a priori* probability $\rho_\mathcal{C}$ of the class $\mathcal{C}$ is $\rho_\mathcal{C} = \frac{N_\mathcal{C}}{N}$. An ideal performance measure should be independent of the *a priori* probability and the type of images in the used image class.

3.1 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 from the image I and store the indices of the images in a $(N-1)$-sized vector $\mathbf{g}^I$. By g_i^I, we denote the ith component of $\mathbf{g}^I$. Next, for all images $I \in \mathcal{C}$, we define a vector $\mathbf{h}^I$ as follows

$$\forall i \in [1, N-1] : \; h_i^I = \begin{cases} 1 & , \quad \text{if } g_i^I \in \mathcal{C} \; , \\ 0 & , \quad \text{otherwise} \; . \end{cases} \tag{3}$$

The vector $\mathbf{h}^I$ thus has value one at location i, if the corresponding image belongs to the class $\mathcal{C}$. As $\mathcal{C}$ has $N_\mathcal{C}$ images, of which one is the image I itself, each vector $\mathbf{h}^I$ contains exactly $N_\mathcal{C} - 1$ ones.

In order to perform well with the class $\mathcal{C}$, the feature extraction should cluster the images I belonging to $\mathcal{C}$ near each other. That is, the values $h_i^I = 1$ should be concentrated on the small values of i.

We can now define the *observed probability* p_i:

$$\forall i \in [1, N-1] : \; p_i = \frac{1}{N_\mathcal{C}} \sum_{K \in \mathcal{C}} h_i^K \; . \tag{4}$$

The observed probability $p_i \in [0, 1]$ is a measure 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 extraction $\mathbf{f}$.

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 so that the longest distance from an image in $\mathcal{C}$ to another image in the same class is always smaller than the shortest distance to any image not in $\mathcal{C}$. On the other hand, The worst case happens when the feature $\mathbf{f}$ completely fails to discriminate the images in class $\mathcal{C}$ from the remaining images. The observed probability p_i is then close to the *a priori* $\rho_\mathcal{C}$ for every value of $i \in [1, N-1]$.

3.2 Forming Scalars from the Observed Probability

The observed probability p_i is a function of the index i, so it cannot easily be used to compare two different feature extractions. Therefore, it is necessary to derive scalar measures from p_i to enable us to do such comparisons. As large values of p_i with small values i and small values of p_i with large values i correspond to good discriminating power, the scalar measure should respectively reward large values of p_i when i is small and punish large values of p_i when i is large. We chose to use three figures of merit to describe the performance of individual feature types. First, a local measure calculated as the average of the observed probability p_i for the first 50 retrieved images, i.e.:

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

The η_{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.

For a global figure of merit we used the weighted sum of the observed probability p_i calculated as:

$$\eta_{\text{global}} = \text{Re} \left\{ \frac{\sum_{i=0}^{N-1} p_i \, e^{j\pi i/N}}{\sum_{i=0}^{N-1} p_i} \right\} \tag{6}$$

Also η_{global} attains values between zero and one. It favors observed probabilities that are concentrated in small indices and punishes for large probabilities in large index values.

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

$$\eta_{\text{half}} = \frac{\sum_{i=0}^{N/2} p_i}{N_{\mathcal{C}} - 1} \tag{7}$$

η_{half} obviously yields a value one in the optimal case and a value half with the *a priori* distribution of images.

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

3.3 τ Measure

We have applied one quantitative figure, denoted as the τ measure, which describes the performance of the whole CBIR system instead of a single feature type. It is based on measuring the average number of images the system retrieves before the correct one is found. The τ measure resembles the "target testing" method presented in [2], but instead of using human test users, the τ measure is fully automatic.

For obtaining the τ measure we use the same subset $\mathcal{C}_i$ of images as before for the single features. We then implemented an "ideal screener", a computer program which simulates the human user by examining the output of the retrieval system and marking the images returned by the system either as relevant (positive) or non-relevant (negative) according to whether the images belong to $\mathcal{C}$. The query processing can thus be simulated and performance data collected without any human intervention.

For each of the images in the class $\mathcal{C}$, we then record the total number of images presented by the system until that particular image is shown. From this data, we form a histogram and calculate the average number of shown images needed before a hit occurs. After division by N, this figure yields a value

$$\tau \in [\frac{\rho_{\mathcal{C}}}{2}, 1 - \frac{\rho_{\mathcal{C}}}{2}] \tag{8}$$

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

4 Experiment Settings

We evaluated the PicSOM approach with a set of experiments using an image collection from the Corel Gallery 1 000 000 product [1]. The collection contains 59 995 photographs and artificial images with a very wide variety of subjects. All the images are either of size 256×384 or 384×256 pixels. The majority of the images are in color, but there are also a small number of grayscale images. The images were converted from the original WIF (wavelet-compressed image) format to JPEG.

Five different feature extraction methods were applied to the images and the corresponding TS-SOMs were created. The TS-SOMs for all features were sized 4×4, 16×16, 64×64, and 256×256, from top to bottom. During the training, each vector was used 100 times in the adaptation. The features used in this study included two different color and shape features and a simple texture feature [9]. The color and texture features were calculated in five separate zones of the image. The zones were formed by first determining a circular area in the center of the image. The size of the circular zone was approximately one fifth of the area of the image. Then the remaining area was divided into four zones with two diagonal lines.

Average Color feature (*cavg* in Tables 1 and 2) was obtained by calculating average R-, G- and B-values in five separate regions of the image. The resulting 15-dimensional feature vector thus not only describes the average color of the image but also gives information on the spatial color composition.

Color Moment feature (*cmom*) were introduced in [12]. The color moment features were computed by treating the color values in different color channels as separate probability distributions and then calculating the first three

features	classes		
	airplanes ($\rho_{\mathcal{C}} = 0.019$)	*faces* ($\rho_{\mathcal{C}} = 0.014$)	*cars* ($\rho_{\mathcal{C}} = 0.005$)
cavg	0.05/0.10/0.56	0.03/0.21/0.63	0.06/0.16/0.59
cmom	0.05/0.10/0.56	0.04/0.21/0.63	0.06/0.16/0.59
texture	0.06/0.16/0.57	0.07/0.22/0.63	0.04/0.04/0.52
shist	0.10/0.54/0.82	0.13/0.34/0.68	0.11/0.62/0.84
sFFT	0.07/0.39/0.72	0.10/0.30/0.65	0.04/0.49/0.78

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

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

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

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

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

In order to evaluate the performance of the single features and the whole PicSOM system with different types of images, three separate image classes were picked manually from the 59 995-image Corel database. The selected classes are *faces*, *cars*, and *airplanes*, of which the database consists of 1115, 864, and 292 images, respectively. The corresponding *a priori* probabilities are 0.019, 0.014, and 0.005. The criteria 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 *airplanes* class there were no restrictions, all images of aircraft or helicopters were accepted.

The number of new images the system presents each round has also some effect on the resulting τ value. In the experiments, the system was set to return 20 best-scoring images each round.

5 Results

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

The results of the experiments with the whole PicSOM system are shown in Table 2. First, each feature was used alone as the basis for the retrieval and then different combinations of the features were tested. The two shape features again yield better results than the color and texture features, which can be seen from the first five rows in Table 2. By examining the results with all the tested classes, it can be seen that the general trend is that using a larger set of features yields better results than using a smaller set. Most notably, using all features gives better results than using any one feature alone. The results in the second and third sections of the table also validate the overall trend that using more features generally improves the results. Therefore, it can be concluded that the PicSOM system is able to benefit from

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

Table 2: The resulting τ values in the experiments.

the existence of multiple feature types. As it is generally not beforehand known which feature type would perform best for a certain image query, the PicSOM approach provides a robust method for using a set of different features and image maps formed thereof in parallel. However, it also seems that if one feature type has clearly worse retrieval performance than the others, it may be more beneficial to exclude that particular TS-SOM from the retrieval process. Therefore, it is necessary for the proper operation of the PicSOM system that the used features are well balanced, i.e., they should on the average perform quite similarly by themselves.

6 Conclusions and Future Plans

We have in this paper introduced the PicSOM approach to content-based image retrieval and methods for quantitative evaluation of its performance. The results of our experiments show that the PicSOM system is able to effectively select from a set of parallel TS-SOMs a combination which yields the best retrieval performance. One obvious direction to increase PicSOM's retrieval performance is to do an extensive study of different feature representations to find a set of well-balanced features which on the average perform as well as possible. As a vast collection of unclassified images is available on the Internet, we have also made preparations to use PicSOM as an image search engine for the World Wide Web.

References

[1] The Corel Corporation World Wide Web home page, `http://www.corel.com`.

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

[3] Myron Flickner, Harpreet Sawhney, Wayne Niblack, et al. Query by image and video content: The QBIC system. *IEEE Computer*, pages 23–31, September 1995.

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

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

[6] Pasi 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., August 1994.

[7] Jorma Laaksonen, Markus Koskela, and Erkki Oja. PicSOM – a framework for content-based image database retrieval using self-organizing maps. In *11th Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland, June 1999.

[8] MPEG-7: Context and objectives (version 10, Atlantic City), October 1998. MPEG 98, ISO/IEC JTC1/SC29/WG11 N2460.

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

[10] Alex Pentland, Rosalind W. Picard, and Stan 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.

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

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