ELSEVIER

Pattern Recognition Letters 21 (2000) 1199–1207

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

PicSOM – content-based image retrieval with self-organizing maps

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

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

Abstract

We have developed a novel system for content-based image retrieval in large, unannotated databases. The system is called PicSOM, and it is based on tree structured self-organizing maps (TS-SOMs). Given a set of reference images, PicSOM is able to retrieve another set of images which are similar to the given ones. Each TS-SOM is formed with a different image feature representation like color, texture, or shape. A new technique introduced in PicSOM facilitates automatic combination of responses from multiple TS-SOMs and their hierarchical levels. This mechanism adapts to the user's preferences in selecting which images resemble each other. Thus, the mechanism implements a relevance feedback technique on content-based image retrieval. 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. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Content-based image retrieval; Image databases; Neural networks; Self-organizing map

1. Introduction

Content-based image retrieval (CBIR) from unannotated image databases has been an object for ongoing research for a long period (Rui et al., 1999). 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.

Many projects have been started in recent years to research and develop efficient systems for content-based image retrieval. The best-known CBIR system is probably Query by Image Content (QBIC) (Flickner et al., 1995) developed at the IBM Almaden Research Center. Other notable systems include MIT's Photobook (Pentland et al., 1994) and its more recent version, FourEyes (Minka, 1996), the search engine family of VisualSEEk, WebSEEk, and MetaSEEk (Smith and Chang, 1996; Beigi et al., 1998), which all are developed at Columbia University, and Virage (Bach et al., 1996), a commercial content-based search engine developed at Virage Technologies.

We have implemented an image-retrieval system that uses a World Wide Web browser as the user interface and the tree structured self-organizing map (TS-SOM) (Koikkalainen and Oja,

* Corresponding author. Tel.: +358-9-4513269; fax: +358-9-4513277.

E-mail addresses: jorma.laaksonen@hut.fi (J. Laaksonen), markus.koskela@hut.fi (M. Koskela), sami.laakso@hut.fi (S. Laakso), erkki.oja@hut.fi (E. Oja).

0167-8655/00/$ - see front matter
PII: S 0 1 6 7 - 8 6 5 5 (0 0) 0 0 0 8 2 - 9

1990; Koikkalainen, 1994) as the image similarity scoring method. The retrieval method is based on the relevance feedback approach (Salton and McGill, 1983) 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.

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) (Kohonen, 1997).

2. Basic operation of PicSOM

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.

From a user's point of view, the basic operation of the PicSOM image retrieval is as follows. (1) The user connects to the World Wide Web server providing the search engine with her web browser. (2) The system presents a list of databases and feature extraction methods available to that particular user. (3) After the user has made the selections, the system presents an initial set of tentative images scaled to a small "thumbnail" size. The user then indicates 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 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 together with the images selected this far. (5) The user again selects the relevant images 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 be formed for describing the color, texture, shape, and structure of the images. A separate TS-SOM (see Section 2.2) is then constructed for each feature. These maps are used in parallel to find from the database those images which are most similar to the images selected by the user. The feature selection is not restricted in any way, and new features can be included as long as the feature vectors are of fixed dimensionality and the Euclidean metric can be used to measure distances between them.

In the first stage, five different feature extraction methods were applied to the images, and the corresponding TS-SOMs were created. The feature types used in this study included two different color and shape features and a simple texture feature. All except one feature type were calculated in five separate zones of the image. The zones are formed by first extracting from the center of the image a circular zone whose size is approximately one-fifth of the area of the image. Then the remaining area is divided into four zones with two diagonal lines.

Average color (*cavg* in Tables 1 and 2) is obtained by calculating average R-, G- and B-values in the five zones of the image. The resulting $3 \times 5 = 15$-dimensional feature vector thus describes the average color of the image and gives rough information on the spatial color composition.

Color moments (*cmom*) were introduced by Stricker and Orengo (1995). The color moment features are computed by treating the color values in different color channels in each zone as separate probability distributions and then calculating the first three moments (mean, variance, and skewness) of each color channel. This results in a $3 \times 3 \times 5 = 45$-dimensional feature vector. Due to the varying dynamic ranges, the feature values are normalized to zero mean and unit variance.

Texture neighborhood (*texture*) feature in PicSOM is also calculated in the same five zones. The Y-values of the YIQ color representation of every pixel's 8-neighborhood are examined, and the estimated probabilities for each neighbor being brighter than the center pixel are used as features. When combined, this results in an $8 \times 5 = 40$-dimensional feature vector.

Shape histogram (*shist*) feature is based on the histogram of the eight quantized directions of edges in the image. When the histogram is separately formed in the same five zones as before, an $8 \times 5 = 40$-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 (Brandt et al., 2000).

Shape FFT (*sFFT*) feature is based on the Fourier transform of the binarized edge image. The image is normalized to 512×512 pixels before the FFT. No zone division is made. The magnitude image of the spectrum is low-pass filtered and decimated by a factor of 32, resulting in a 128-dimensional feature vector (Brandt et al., 2000).

2.2. Tree structured SOM (TS-SOM)

The five separate feature vectors obtained from each image in the database have to be indexed to facilitate similarity-based queries. For the indexing, similarity-preserving clustering using SOMs (Kohonen, 1997) is used. Due to the high dimensionality of the feature vectors and the large size of the image database, computational complexity is a major issue.

The TS-SOM (Koikkalainen and Oja, 1990; Koikkalainen, 1994) is a tree-structured vector quantization algorithm that uses SOMs 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 lower than if the whole bottommost SOM level had been accessed without the tree structure.

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 TS-SOMs for all features are sized 4×4, 16×16, 64×64, and 256×256, from top to bottom. 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, say 100, times. In the process, 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 dataset 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 database is formed. In an ideal situation, there should be one-to-one correspondence between the images and TS-SOM units at the bottom level of each map.

2.3. Using multiple TS-SOMs

In an image-based query, the five feature vectors computed from the images presented to the user are passed to the respective TS-SOMs and the best-matching units are found at each level of the maps. Combining the results from several 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 a human's perception of images at a more conceptual level. Therefore, a better solution is to use the relevance feedback approach in which the results of multiple maps are combined automatically by using the implicit information from the user's responses during the query session. The PicSOM system thus tries to learn the user's preferences over the five features from the interaction with her and then sets its own responses accordingly.

The rationale behind our approach is as follows: if the images selected by the user are located close to each other on one of the five TS-SOM maps, 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 the shown images on the maps with positive and negative values depending on whether the user has selected or rejected them, respectively. The positive and negative responses are then normalized so that their sum equals zero.

The combined effect of positively marked units residing near 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 close together spread the positive response to their neighboring map units. The images associated with the units having largest positive values after the convolution are then good candidates for the next images to be shown to the user.

The conversion from the positive and negative marked images to the convolutions in a three-level TS-SOM is visualized in Fig. 1. The sizes of the SOM levels are 4×4, 16×16, and 64×64, from top to bottom. First, SOMs displaying the positive map units as white and negative as black are shown on the left. These maps are then low-pass filtered, and the resulting map surfaces are shown on the right. It is seen that a cluster of positive images resides at the lower edge of the map.

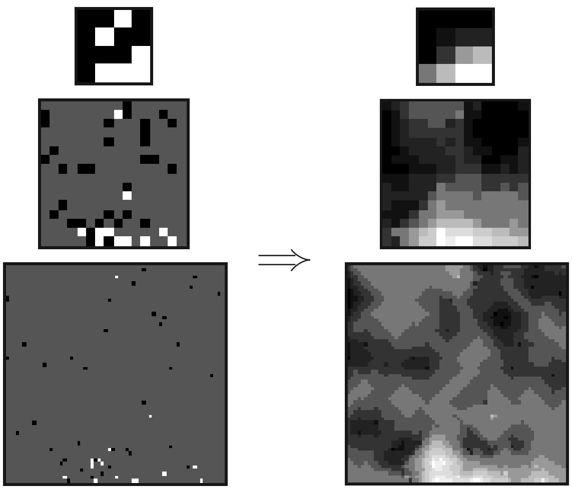

Fig. 1. An example of converting the positive and negative map units to convolved maps in a three-level TS-SOM. Map surfaces displaying the positive (white) and negative (black) map units are shown on the left. The resulting convolved maps are shown on the right.

2.4. Refining queries

A typical retrieval session with PicSOM consists of a number of subsequent queries during which the retrieval is focused more accurately on images resembling those shown images which the user has accepted. In the current implementation, all positive values on all convolved TS-SOM levels are sorted in descending order in one list. Then, a preset number, e.g., 16, of the best candidate images which have not been shown to the user before are output as the new image selection.

Initially, the query begins with a set of reference images picked from the top levels of the TS-SOMs. In the 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 from 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 and makes PicSOM differ from other content-based image retrieval systems, such as QBIC (Flickner et al., 1995), which use only one reference image at a time.

3. Implementation

We have wanted to make our PicSOM search engine available to the public by implementing it on the World Wide Web. This also makes the queries on the databases machine independent, because standard web browsers can be used. A demonstration of the system and further information on it is available at http://www.cis.hut.fi/picsom.

The PicSOM user interface in the midst of an ongoing query is displayed in Fig. 2. First, the three parallel TS-SOM map structures represent three map levels of SOMs trained with RGB color, texture, and shape features, from left to right. The

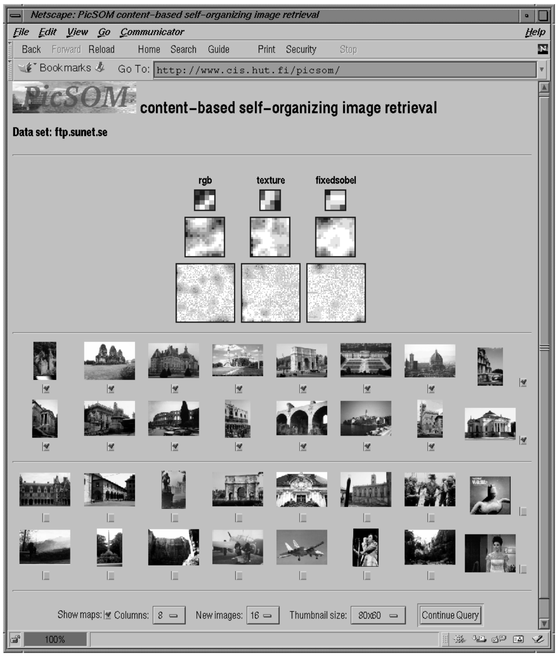

Fig. 2. The web-based user interface of PicSOM.

sizes of the SOM levels are 4×4, 16×16, and 64×64, from top to bottom. On color terminals, positive map points are seen as red and negative as blue. More saturated shades represent stronger responses and white points are zero valued.

Below the convolved SOMs, the first set of 16 images consists of images selected by the user on the previous rounds of the retrieval process. These images may then be unselected on any subsequent round. In this example, images representing buildings have been selected as positive. The next images, separated by a horizontal line, are the 16 best-scoring new images in this round obtained from the convolved units in the TS-SOMs. The relevant images are marked by checking the appropriate checkboxes. 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 examining of the TS-SOM surfaces simply by clicking on the map images. A portion of the map around the chosen viewpoint is then shown before the other images. This allows direct browsing in the image space. Relevant images on the map surface can then also be selected for continuing queries.

4. Databases

Currently, we have made our experiments with two image databases of different sizes. The first database contains 4350 miscellaneous images downloaded from the image collection residing at ftp://ftp.sunet.se/pub/pictures/. Most of the images are color photographs in JPEG format. Figs. 1 and 2 were produced using this database.

The second image database we have used is the image collection from the Corel Gallery 1 000 000 product (Corel, 1999). It contains 59 995 photographs and artificial images with a very wide variety of subjects. When this collection is mapped on a 256×256 SOM, each map unit approximately corresponds to one image. 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 experiments described in the next sections were performed using this dataset.

5. Performance measures

A number of measures for evaluating the ability of various visual features to reveal image similarity are presented in this section. Assume a database $\mathscr{D}$ containing a total of N images and an image class $\mathscr{C} \subset \mathscr{D}$ with $N_{\mathscr{C}}$ images. Such a class can contain, for example, images of aircraft, buildings, nature, human faces, and so on. The process of gathering a class is naturally arbitrary, as there are no distinct and objective boundaries between classes. If the images in the database already have reliable textual information about the contents of the images, it can be used directly; otherwise, manual classification is needed. Then, the a priori probability $\rho_{\mathscr{C}}$ of the class $\mathscr{C}$ is $\rho_{\mathscr{C}} = N_{\mathscr{C}}/N$. An ideal performance measure should be independent of the a priori probability and the type of images in the image class.

5.1. Observed probability

For each image $I \in \mathscr{C}$ with a feature vector $\boldsymbol{f}^I$, we calculate the Euclidean distance $d_{L_2}(I,J)$ of $\boldsymbol{f}^I$ and the feature vectors $\boldsymbol{f}^J$ of the other images $J \in \mathscr{D} \setminus \{I\}$ 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 an $(N-1)$-sized vector $\boldsymbol{g}^I$. By g_i^I, we denote the ith component of $\boldsymbol{g}^I$.

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

$$\forall i \in \{0,\dots,N-2\} : h_i^I = \begin{cases} 1, & \text{if } g_i^I \in \mathscr{C}, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The vector $\boldsymbol{h}^I$ thus has value one at location i, if the corresponding image belongs to the class $\mathscr{C}$. As $\mathscr{C}$ has $N_{\mathscr{C}}$ images, of which one is the image I itself, each vector $\boldsymbol{h}^I$ contains exactly $N_{\mathscr{C}} - 1$ ones. In the optimal case, the closest $N_{\mathscr{C}} - 1$ images to image I are also in the same class $\mathscr{C}$; thus $h_i^I = 1$ for $i = 0,\dots,N_{\mathscr{C}} - 2$, and 0 otherwise.

We can now define the *observed probability* p_i:

$$\forall i \in \{0,\dots,N-2\} : p_i = \frac{1}{N_{\mathscr{C}}} \sum_{K \in \mathscr{C}} h_i^K. \quad (2)$$

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

5.2. 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 perform such comparisons.

We chose to use three figures of merit to describe the performance of individual feature types. First, a local measure was calculated as the average of the observed probability p_i for the first 50 retrieved images, i.e.:

$$\eta_{\text{local}} = \frac{\sum_{i=0}^{49} p_i}{50}. \quad (3)$$

The η_{local} measure obtains values between zero and one.

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

$$\eta_{\text{global}} = \mathrm{Re}\left\{ \frac{\sum_{i=0}^{N-2} p_i \, \mathrm{e}^{j\pi i/(N-1)}}{N_{\mathscr{C}} - 1} \right\}. \quad (4)$$

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

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

$$\eta_{\text{half}} = \frac{\sum_{i=0}^{N/2} p_i}{N_{\mathscr{C}} - 1}. \quad (5)$$

η_{half} obviously yields a value of one in the optimal case and a value of 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.

5.3. τ measure

We have applied another 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 (Cox et al., 1996), but instead of using human test users, the τ measure is fully automatic.

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

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

$$\tau \in \left[\frac{\rho_{\mathscr{C}}}{2}, 1 - \frac{\rho_{\mathscr{C}}}{2}\right]. \quad (6)$$

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.

6. Results

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 were planes, faces, and cars, of which the database contains 292, 1115, and 864 images, respectively. The corresponding a priori probabilities are 0.5%, 1.9%, and 1.4%. In the retrieval experiments, these classes were thus not competing against each other but mainly against the "background" of 57 724, i.e., 96.2% of other images.

Table 1 shows the results of 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 and then different combinations of them were tested. The two shape features again yield better results than the color and texture features, which can seen from the first five rows in Table 2.

By examining the results with all 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 implicit weighting of the relative importances of different features models the semantic similarity of the images selected by the user. This kind of automatic adaptation is desirable as it is generally not known which feature combination would perform best for a certain image query. To alleviate the situation, the PicSOM approach provides a robust method for using a set of different features and image maps formed thereof in parallel so that the result exceeds the performances of all the single features.

However, it also seems that if one feature vector type has clearly worse retrieval performance 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.

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

Features	Classes		
	Planes	Faces	Cars
cavg	0.06/0.16/0.59	0.05/0.10/0.56	0.03/0.21/0.63
cmom	0.06/0.16/0.59	0.05/0.10/0.56	0.04/0.21/0.63
texture	0.04/0.04/0.52	0.06/0.16/0.57	0.07/0.22/0.63
shist	0.11/0.62/0.84	0.10/0.54/0.82	0.13/0.34/0.68
sFFT	0.04/0.49/0.78	0.07/0.39/0.72	0.10/0.30/0.65

Table 2
The resulting τ values in the experiments

Features					Classes		
cavg	cmom	texture	shist	sFFT	Planes	Faces	Cars
×					0.30	0.35	0.39
	×				0.31	0.43	0.34
		×			0.26	0.26	0.34
			×		0.16	0.22	0.18
				×	0.19	0.22	0.18
×		×	×		0.16	0.21	0.18
×		×		×	0.17	0.23	0.18
×		×	×	×	0.14	0.21	0.16
	×	×	×		0.15	0.21	0.18
	×	×		×	0.18	0.22	0.19
	×	×	×	×	0.14	0.20	0.16
×	×	×	×	×	0.14	0.20	0.16

7. 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. As a single visual feature cannot classify images into semantic classes, we need to gather the information provided by multiple features to achieve good retrieval performance. The results of our experiments show that the PicSOM system is able to effectively select from a set of parallel TS-SOMs a combination which coincides with the user's conceptual view of image similarity.

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 for using PicSOM as an image search engine for the World Wide Web.

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

Bach, J.R., Fuller, C., Gupta, A., et al., 1996. The Virage image search engine: an open framework for image management. In: Sethi, I.K., Jain, R.J. (Eds.), Storage and Retrieval for Image and Video Databases IV. In: SPIE Proceedings Series, Vol. 2670. San Jose, CA, USA.

Beigi, M., Benitez, A., Chang, S.-F., 1998. MetaSEEk: a content-based metasearch engine for images. In: Storage and Retrieval for Image and Video Databases VI (SPIE). In: SPIE Proceedings Series, Vol. 3312. San Jose, CA, USA.

Brandt, S., Laaksonen, J., Oja, E., 2000. Statistical shape features in content-based image retrieval. In: Proceedings of 15th International Conference on Pattern Recognition, Barcelona, Spain, Vol. 2, pp. 1066–1069.

Corel, 1999. The Corel Corporation world wide web home page, http://www.corel.com.

Cox, I.J., Miller, M.L., Omohundro, S.M., Yianilos, P.N., 1996. Target testing and the PicHunter Bayesian multimedia retrieval system. In: Advanced Digital Libraries ADL'96 Forum. Washington, DC.

Flickner, M., Sawhney, H., Niblack, W., et al., 1995. Query by image and video content: the QBIC system. IEEE Computer September, 23–31.

Kohonen, T., 1997. Self-Organizing Maps, second extended ed. In: Springer Series in Information Sciences, Vol. 30. Springer, Berlin.

Koikkalainen, P., 1994. Progress with the tree-structured self-organizing map. In: Cohn, A.G. (Ed.), 11th European Conference on Artificial Intelligence. European Committee for Artificial Intelligence (ECCAI). Wiley, New York.

Koikkalainen, P., Oja, E., 1990. Self-organizing hierarchical feature maps. In: Proceedings of 1990 International Joint Conference on Neural Networks, Vol. II. IEEE, INNS, San Diego, CA.

Minka, T.P., 1996. An image database browser that learns from user interaction. Master's thesis, M.I.T., Cambridge, MA.

Pentland, A., Picard, R.W., Sclaroff, S., 1994. Photobook: tools for content-based manipulation of image databases. In: Storage and Retrieval for Image and Video Databases II. In: SPIE Proceedings Series, Vol. 2185. San Jose, CA, USA.

Rui, Y., Huang, T.S., Chang, S.-F., 1999. Image retrieval: current techniques, promising directions, and open issues. Journal of Visual Communication and Image Representation 10 (1), 39–62.

Salton, G., McGill, M.J., 1983. Introduction to Modern Information Retrieval. In: Computer Science Series. McGraw–Hill, New York.

Smith, J.R., Chang, S.-F., 1996. VisualSEEk: a fully automated content-based image query system. In: Proceedings of the ACM Multimedia 1996. Boston, MA.

Stricker, M., Orengo, M., 1995. Similarity of color images. In: Storage and Retrieval for Image and Video Databases III (SPIE). In: SPIE Proceedings Series, Vol. 2420. San Jose, CA, USA.