published in the Proceedings of ICONIP 2000, Taejon, Korea, November 2000
available at <URL:http://www.cis.hut.fi/picsom/publications.html>

Analyzing Low-Level Visual Features Using Content-Based Image Retrieval

Jorma Laaksonen †, Erkki Oja †, Markus Koskela † and Sami Brandt ‡

{*jorma.laaksonen,erkki.oja,markus.koskela,sami.brandt*}*@hut.fi*

† Laboratory of Computer and Information Science
Helsinki University of Technology
P.O.BOX 5500, 02015 HUT, Finland

‡ Laboratory of Computational Engineering
Helsinki University of Technology
P.O.BOX 9400, 02015 HUT, Finland

Abstract

This paper describes how low-level statistical visual features can be analyzed in our content-based image retrieval system named PicSOM. The low-level visual features used in the system are all statistical by nature. They include average color, color moments, contrast-type textural feature, and edge histogram and Fourier transform based shape features. Other features can be added easily. A genuine characteristic of the PicSOM system is to use relevance feedback from the human user's actions to direct the system in scoring the relevance of particular features in the present query. While the link from features to semantic concepts remains an open problem, it is possible to relate low-level features to subjective image similarity, as perceived instantaneously by human users. The efficient implementation of PicSOM allows tests using statistically sufficiently large and representative databases of natural images.

Acknowledgement

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

1 Introduction

The structure and statistics of natural scenes has an essential influence on visual system design. First, it helps to understand the biological visual systems, that by necessity have adapted over evolutionary time scales to the real visual environment. For instance, it was shown by Atick and Redlich [1] how the statistics of images helps in predicting the properties of ganglion cell receptive fields. Second, natural image statistics have to be taken into account when designing imaging systems, like visual displays and compression codes that optimize subjective human evaluation criteria [2].

Yet a third field of research that is centrally affected by natural image statistics is computer vision, especially semi-automatic or interactive applications in which the results of computerized processing are used by humans. An emerging research topic in interactive computer vision is content-based image retrieval (CBIR) from image databases that are unannotated, i.e. no textual explanations are provided for the images. This is a wide and versatile field of research whose popularity is largely due to increasing computation power and the availability of huge image databases in the World Wide Web.

Depending on the domain of interest, the database in question, and the amount of *a priori* information available on the images, the CBIR problem exhibits a varying degree of difficulty. A rather simple CBIR problem occurs when the database in question consists of images of a strongly restricted domain. For example, a widely-studied application of this complexity is retrieval of trademark images, mainly based on different shape features as the lack of background enables automatic segmentation of the trademark images. The results of applying CBIR in such a setting have been rather good.

In the other extreme lies the problem of retrieving relevant images from large and dynamic collections of miscellaneous images. One massive example of such a challenging domain is indexing the images contained in the World Wide Web. The basic problem in CBIR is the gap between the high-level semantic concepts used by humans to understand image content and the low-level visual features extracted from images and used by a computer to index the images in a database. Good overall reviews of CBIR include [3, 4, 5].

This paper describes how low-level visual features are being used in our content-based image retrieval system named PicSOM [6]. Low-level vi-

sual features used in the system are all statistical by nature. They include average color, color moments, contrast-type textural feature, and edge histogram and Fourier transform based shape features. A genuine characteristic of the PicSOM system is to use relevance feedback from the user's actions to direct the system in scoring the relevance of particular features in the present query. While the link from features to semantic concepts remains an open problem, it is possible to relate low-level features to subjective image similarity, as perceived instantaneously by human users.

In the sequel, Section 2 addresses what are low-level visual features and gives some examples how they can be extracted from images. The relevance feedback techniques in the CBIR domain are addressed in Section 3, while Section 4 discusses feature-based image comparisons. The PicSOM system and its use for content-based retrieval of images is shortly described in Section 5. Concluding remarks are drawn and future directions addressed in Section 6.

2 Low-Level Visual Features

Feature extraction in databases that contain miscellaneous images, i.e. images that do not portray any specific topic but come from various sources and are without any common theme, is very difficult. Segmentation of an object out from the background is not possible as there generally is no particular object in the image. Therefore, segmentation is in such a case of very limited use as a stage preceding feature extraction.

The images thus need to be described as a whole and one should devise feature extraction schemes that do not require segmentation. This restriction excludes a vast number of well-known feature extraction techniques: all boundary-based methods and many area-based methods.

What is left are basic pixel-value-based statistics, possibly combined with edge detection techniques, that reflect the properties of the human visual system in discriminating between image patches. Such features are usable even when the images are not segmented beforehand. As the images are stationary, no dynamic features can be used.

The basic static features can be categorized in at least three groups, namely color features (two of which will be addressed in Sections 2.1 and 2.2), texture features (Section 2.3), and shape features (Sections 2.4-2.8). Experiments performed with the above-described features in the PicSOM system have been described in detail in [7, 8].

Invariance to specific transforms is an issue of interest in feature extraction. Feature extraction methods that are global in their nature or perform averaging over the whole image area are often inherently translation invariant. If the averaging is performed after the image area is first divided in separate zones, the degree of translation invariance can be controlled by the number and layout of the zones. Other types of invariance, e.g. invariance to scaling, rotation, and occlusion, can be obtained with some feature extraction schemes by using proper transformations. Whether these forms of invariance are at all beneficial is task-dependent and in the case of a general image database, they should be exploited with care [8].

2.1 Average Color

Average color feature is in PicSOM system obtained by calculating average R-, G- and B-values in five separate zones of the image. The resulting 15-dimensional feature vector thus describes the average color of the image and gives rough information on the spatial color composition. The image zones are depicted in Figure 1.

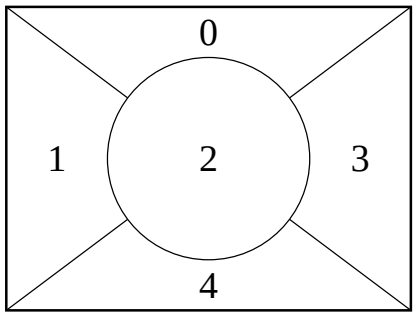

Figure 1: Five image zones used for low-level visual feature extraction in the PicSOM system.

2.2 Color Moments

Color moments were introduced in [9]. The color moment features are computed by treating the color values in different color channels in each of the five zones of Figure 1 as separate probability distributions. The first three moments (mean, variance, and skewness) are then calculated 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 components are normalized to zero mean and unit variance.

2.3 Texture Neighborhood

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

2.4 Histogram of Edge Directions

Low-level shape-based features can be formed from the edges in the image. A histogram of edge directions is translation invariant and it captures

the general shape information in the image. Because the feature is local, it is robust to partial occlusion and local disturbance in the image.

The edge image in PicSOM is formed by convolving the intensity and saturation channels of the image with the eight Sobel operators. The resulting gradient images are next thresholded to binary images by a proper value for each channel. The threshold values are manually fixed to certain levels which are the same for all images. The binarized intensity and saturation gradient images are combined by the logical OR operation in which the direction of the larger gradient value is chosen. Finally the 8-dimensional edge histograms are calculated by counting the edge pixels in each direction and normalizing with the total number of pixels.

2.5 Co-occurrence of Edge Directions

The edge histogram can yet be generalized. By taking every 8-neighboring edge pixel pair and enumerating them based on their directions a two-dimensional histogram or co-occurrence matrix is obtained. The resulting 64-dimensional histogram is normalized by the number of pixels in the image. Hence, the resulting values indicate the proportion of neighboring edge pixel pairs oriented in the specified directions.

2.6 Fourier Features

The edge image contains the most relevant shape information and the discrete Fourier transform can be used to describe it. Before forming the edge image, the image area is normalized to a maximum size of 512×512 so that the aspect ratio is maintained. After edge detection, the Fourier transform is computed for the normalized image using the FFT algorithm. The magnitude image of the Fourier spectrum is first low-pass filtered and thereafter decimated so that the resulting number of dimensions in the feature vectors is 128.

2.7 Polar Fourier Features

The Fourier features described above are translation invariant but not rotation invariant. Our method, which is named as polar Fourier features, is rotation invariant with respect to the center of the image but not invariant to translation and scale.

At first the image is normalized and the edge image is obtained similarly as with the Fourier features. The binary edge image is then transformed to the polar coordinates by using a procedure that prevents the formation of gaps between the edge pixels in the polar coordinate system.

For the polar image the Fourier transform and decimation are performed similarly as with the Fourier features and a 128-dimensional feature vector is obtained. The method is invariant to translation in the polar plane, and therefore rotation invariant with respect to the center of the image and translation invariant along the radius from the center.

2.8 Log–Polar Fourier Features

Even more invariances can be obtained by a slight modification to the feature. Log–polar Fourier features are invariant to affine transformations, i.e. to translation, rotation and scaling. Translation invariance is obtained by setting the centroid to the center of mass of the binary edge image. Rotation invariance is obtained by using the magnitude spectrum of the log–polar transform. Accordingly, the invariance for scale is obtained by taking logarithm of the radius in the polar coordinate plane.

All the Fourier-based features presented here are sensitive to occlusion: the direct use of the Fourier transform may lead to very different magnitude spectra for occluded images. In addition, if some parts of an image are missing, the calculation of the centroid will go wrong and significantly differing log–polar images will result.

3 Relevance Feedback in CBIR

Query by pictorial example (QBPE) is a common retrieval paradigm in content-based image retrieval applications [10]. With QBPE, the queries are based on example images shown either from the database itself or some external location. The user classifies these example images as relevant or non-relevant to the current retrieval task and the system uses this information to select such images the user is most likely to be interested in.

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

The iterative and automatic refinement of a query is known as *relevance feedback* in information retrieval literature [11]. Relevance feedback can be seen as a form of supervised or reinforcement learning to adjust the subsequent queries using the information gathered from the user's feedback. This helps the system in the following rounds of the retrieval process to better approximate the present need of the user.

4 Comparing Images in CBIR

A CBIR system is typically implemented with prototype-based statistical methods. This means that each image in the database is transformed with a set of different feature extraction methods to a set

of lower-dimensional feature vectors, or prototypes, in respective feature spaces. When the system tries to find images which are similar to the positive-marked images shown previously, it searches for images whose distance to the positive images in some sense is minimal in any or all of the feature spaces. The distances between prototypes in the feature spaces can be defined in a multitude of ways, the Euclidean distance being the one used most.

Figure 2 illustrates this idea. Each feature representation can be used separately for finding a set of image candidates. These per-feature subsets should then be combined in a larger set of images which will be processed in a more exhaustive manner. Depending on the sizes of the subsets either all images in them or, for example, only those which are included in more than one of them, can be taken in the combined set. Nevertheless, in the final selection process there will be a substantially smaller number of images than the whole database. This enables to use computationally more demanding techniques for selecting among them.

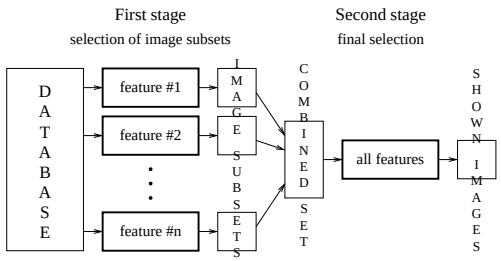

Figure 2: The stages of image selection in CBIR.

In selecting the image subsets in huge databases, *vector quantization* is a highly useful technique. The feature vectors representing the images are divided in subsets or quantization bins in which the vectors and thus the corresponding images resemble each other. Those unseen images which have fallen into the same quantization bins as the positive-marked shown images are then good candidates for the next images to be displayed to the user. One may also want to calculate the exact distance between the prototypes. In that case quantization serves as an effective method for *pruning the database* before exhaustive search.

5 The PicSOM System

This section presents a short description of our PicSOM retrieval system. A more detailed description of the system and results of experiments performed with it can be found in [6]. The PicSOM image retrieval system is a framework for generic research on feature extraction, algorithms, and methods for content-based image retrieval.

The Self-Organizing Map (SOM) [12] is a neurally motivated unsupervised learning technique which has been used in many data analysis tasks. A unique feature of the Self-Organizing Map is its ability to form a nonlinear mapping of a high-dimensional input space to a typically two-dimensional grid of artificial neural units. During the training phase of a SOM, the weight vectors of its neurons get values which form a topographic or topology-preserving mapping. As a result, feature vectors that reside near each other in the input space are mapped to nearby map units in the map layer. Images that are mutually similar in respect to the given feature extraction scheme are thus located near each other on the SOM.

PicSOM supports multiple parallel features, which in the present implementation are color, texture, and shape. With a technique introduced in the PicSOM system, the responses from the parallel SOMs are combined automatically. This question will be elaborated in detail in Section 5.3.

5.1 Forming the Image Maps

The PicSOM system uses a special form of the SOM, namely Tree Structured Self-Organizing Map (TS-SOM) [13, 14], which incorporates a hierarchical view in the database. The training of each TS-SOM starts from its top level. When the top-most level has been trained, it is frozen and the training of the second level is started. Once a SOM level has finished learning, all the data vectors in the training set are mapped to that SOM, each into the SOM unit which is nearest to it. Every map unit is in turn associated with that image among those mapped into it which is nearest to it.

The map units are thus given visual labels which can be used to represent all the images mapped in that particular map node. The image labels of a 16×16 SOM trained with average color as the feature are shown in Figure 3. In this experiment, the size of the image database was about 60.000. From the SOM surface, the topological ordering of the label images based on their color content can be observed: reddish images are located in the upper left corner of the map and the overall color changes gradually to blue when moving diagonally towards the bottom right corner. On the other hand, light images are situated in the bottom left corner and dark images in the opposite position in the upper right corner of the map.

5.2 Operation of PicSOM

The operation of 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. 3) When the user has selected the database the system presents a list of available features in that database. 4) After the user has selected the features, the system presents an initial set of tentative images scaled to a small "thumbnail" size. The user selects the subset of these images which best matches her ex-

Figure 3: The surface of the 16 × 16-sized SOM formed with the average RGB color feature.

pectations and to some degree of relevance fits to her purposes. Then she hits the "Continue Query" button in her browser, which sends the information on the selected images back to the search engine. 5) Based on this data, the system then presents the user a new set of images along with the images selected so far and the query iteration is continued.

5.3 Combining the Maps

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

Both the positive and negative images, i.e. images selected and not selected, respectively, by the user, are located on each level of every TS-SOM in use. The map units are scored with a fixed positive value for each positive image mapped in them. Likewise, negative images contribute negative values. These values are then normalized so that the sum of all the positive and negative terms on the map equals zero. What thus results is a set of map surface images whose sizes match the number of map units in the two-dimensional SOM grid of the particular TS-SOM level, see Figure 4.

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

Now, all the three factors, namely 1) the degree of the separation of the positive and negative images on the SOM, 2) the relative denseness of the positive images, and 3) the similarity of images in neighboring map units, can be accounted for in a single action. This joint action is low-pass filtering of response values on the two-dimensional map surfaces. Strong positive values from dense relevant responses get expanded to neighboring SOM units, whereas weak positive and negative values in the map areas where the responses are sparse cancel each other out. What follows in the low-pass filtering is the polarization of the entire map surface in areas of positive and negative cumulative relevance. In practice the filtering has been implemented by convolving the map image with a Gaussian-shaped mask whose size is approximately one fifth of the width of the corresponding TS-SOM level. The images used as labels for the SOM units which have the strongest positive relevance value after the low-pass filtering are then obvious candidates for the next images to be shown to the user. Figure 4 illustrates how the positive and negative responses, displayed with white and black colors, respectively, are first mapped on three levels of a TS-SOM and how the responses are expanded in the convolution.

5.4 Separation of Image Classes

One may also be interested in how sets of images that are known to be similar to each other in some respect are mapped on the SOM surfaces. This kind of inspection reveals the feature extraction method's capability to map similar images near each other in the feature space and, further, the SOM training algorithm's ability to preserve the spatial ordering of the feature space. Figure 5 gives

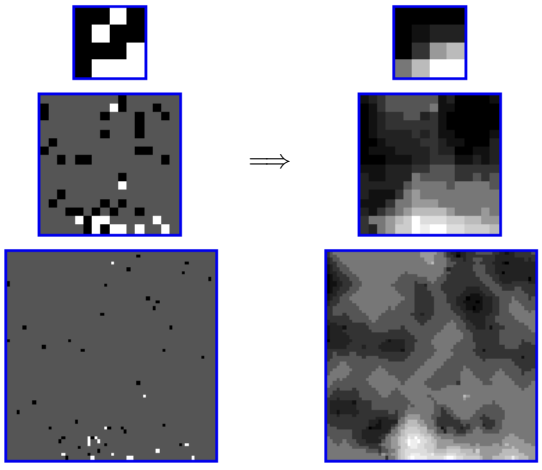

Figure 4: An example showing how the levels of a TS-SOM, on which the images selected and rejected by the user are shown with white and black marks, respectively, are convolved with low-pass filters.

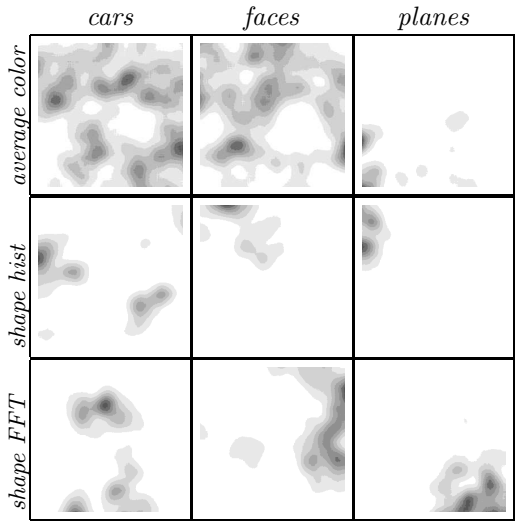

Figure 5: Mappings of different image classes (shown in columns) on the lowest-level SOMs of different features (show in rows). The distributions have been low-pass filtered to ease inspection.

an example. Each of the three columns represents one of three hand-picked image classes, *cars*, *faces*, or *planes*, respectively. The rows correspond to three different feature extraction methods, *average color*, *shape histogram*, and *shape FFT*. It can be seen that the *average color* feature is able to cluster only the images in the *planes* class whereas the *cars* and *faces* classes are widely distributed. On the other hand, *shape histogram* feature clusters all three classes well, but the *cars* and *planes* classes are somewhat overlapping on the left side of the maps. Finally, *shape FFT* feature does not make as tight clusters as *shape histogram* does, but separates the *cars* and *planes* classes better.

6 Discussion

We have shown in this paper that it is possible to extract powerful and representative low-level visual features from natural images, for which a preceding segmentation stage is not applicable. When such feature representations are clustered using Self-Organizing Maps, the topological ordering reflects well the mutual similarity of the images as given by subjective human judgement.

In general, it cannot be known beforehand what features are meaningful in a particular content-based image query. To answer this problem, the PicSOM system implements a novel automatic technique for incorporating a large number of parallel features and for selecting new images by relevance feedback from the user. In this way, the PicSOM system can be used for extensive testing of the relevance of statistical features with very large image databases.

References

[1] J. Atick and A. Redlich. Towards a theory of early visual processing. *Neural Computation*, 2:308–320, 1990.

[2] S. Sherr. *Fundamentals of Display System Design*. Wiley, 1970.

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

[4] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, Inc., 1999.

[5] Y. Gong. *Intelligent Image Databases: Towards Advanced Image Retrieval*. Kluwer Academic Publishers, 1998.

[6] E. Oja, J. Laaksonen, M. Koskela, and S. Brandt. Self-organizing maps for content-based image retrieval. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 349–362. Elsevier, 1999.

[7] M. Koskela, J. Laaksonen, S. Laakso, and E. Oja. The PicSOM retrieval system: Description and evaluations. In J. P. Eakins and P. G. B. Enser, editors, *Proc. Challenge of Image Retrieval 2000*, Brighton, UK, May 2000.

[8] S. Brandt, J. Laaksonen, and E. Oja. Statistical shape features in content-based image retrieval. In *Proc. of 15th ICPR*, Barcelona, 2000.

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

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

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

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

[13] P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. In *Proc. IJCNN-90*, pages 279–284, San Diego, 1990.

[14] P. Koikkalainen. Progress with the tree-structured self-organizing map. In *11th European Conference on Artificial Intelligence*. European Committee for Artificial Intelligence (ECCAI), August 1994.