Color Co–Occurrence Descriptors for Querying–by–Example*

Vassili Kovalev
Belarus Academy of Sciences
Institute of Mathematics
Kirova St. 32–A
246652 Gomel
Belarus
goim@nauka.belpak.gomel.by

Stephan Volmer
Fraunhofer Institute for Computer Graphics
Coginitive Computing & Medical Imaging
Rundeturmstr. 6
64283 Darmstadt
Germany
volmer@igd.fhg.de

Abstract

Multimedia documents are different from traditional text documents, because they may contain encodings of raw sensorical data. This fact has severe consequences for the efficient indexing and retrieval of information from documents in large unstructured collections (e.g. WWW), because it is very difficult to automatically identify generic meanings from visual or audible objects.

A novel method for image retrieval from large collections is proposed in this paper. The method is based on color co–occurrence descriptors that utilize compact representations of essential information of the visual image content. The set of descriptor elements represents "elementary" color segments, their borders, and their mutual spatial distribution on the image frame. Such representation is flexible enough to describe image scenes ranging from simple combinations of color segments to high frequency color textures equally well.

At the retrieval stage the comparison between a given query descriptor and the database descriptors is performed by a similarity measure. Image descriptors are robust versus affine transformations and several other image distortions. The consideration of the descriptors as sets of elements allows the combination of several images or subimages into a single query.

Basic properties of the method are demonstrated experimentally on an image database containing 20000 images.

*The authors are grateful to the DAAD grant A/97/09313 and INTAS grant 96–785 that made this work possible.

1. Introduction

The emergence of the World Wide Web (WWW) has resulted in a huge collection of distributed hypertext documents. Although interconnecting links are used to organize and structure information locally, the search for an initial link associated with relevant information has become a central problem of information retrieval. In order to overcome this problem, web search engines like AltaVista, Excite, HotBot, Infoseek, Lycos, et al., have been employed to crawl distributed documents via their hypertext links in order to collect and store index information about the content and location of the documents they find.

New advances in multimedia authoring, graphics devices, and high–speed networking have sparked a tremendous increase in the amount of multimedia objects like images, videos, graphics, and sounds that are present in those documents. This trend is likely to continue in the future, since those documents enable a more efficient information transfer in terms of user perception ("a picture is worth a thousand words" — English aphorism), and forces us to rethink the way information is retrieved from those documents, because traditional web search is limited to text–based queries.

1.1. Background

The concept of matching of textual information on the level of semantics no longer applies to multimedia objects that have abstract data representations entailing sensorical information on the level of signals. This imposes new problems since sensorical data is encoded differently than the way it is perceived by humans. Information must be abstracted and translated into an encoding in some convenient form. Research in the past has focused on extracting pieces of semantical information for textual labeling from the raw

data — with only little success on objects derived from constrained environments. The basic objective of computer vision — image understanding — is still an unsolved problem. Therefore, the best strategy for the moment is to directly search for images based on their signal content. This approach is manifested by the fact that humans are much better than computers at extracting a meaning from a given picture or creating a picture for a given meaning, while computers are much better than humans at comparing measurable properties and retaining these in long–term memory.

The crux of the problem lies in the choice of an appropriate descriptor that efficiently and compactly represents the visual characteristics of an image that uniquely discriminates it among all others. Research has already applied feature primitives such as color [9], texture [13], and shape [18], as well as combinations [1] [16] of these with significant success. Given the number of unsolved problems in image understanding, current WWW image crawlers still rely on indexing text extracted from the HTML document containing the image [17] [5]. This method only suffices, if and only if, the surrounding text describes the image well enough. Furthermore it suffers from fundamental problems describing the content of an image consistently [19]. These considerations lead to a paradigm for an WWW image crawler that is based on visual similarities.

From the users point of view, there are two completely different scenarios for specification of an visual query. They are substantially differing by the style of user interaction with the system. The first is commonly referred to as query–by–example [4] [11] and searches an image on basis of existing images. The second as known as query–by–sketch [3] [12] [19] and requires the alternate specification of an query image by some means (the most natural way is sketching). The latter has the advantage that no initial image is required to be able to perform a query. However, the more important technique is querying–by–example, since the user usually turns to it, if one of his queries results in an image that is similar to the one he is looking for. The selection of the two querying techniques effects the choice of feature descriptors substantially, because querying–by–sketch puts more emphasis on rough correspondences.

The approach that is presented in this paper is essentially a query–by–example approach and is based on co–occurrence descriptors. Traditionally gray–level co–occurrence matrices [10] and gray–level run length matrices [6] are widely used for texture description, classification and segmentation. However, in recent years different extensions of their basic ideas have been successfully employed for a number of Computer Vision tasks such as the estimation of the optical flow [7], the extraction of salient (atypical) and background (typical) image features [8], the detection of structural defects [2], the quantification of 3D surfaces roughness [15], and the recognition and matching of objects in numerous applications [14].

In this paper we propose the use the frequency of spatial co–occurrences of pixel colors in order to describe the content of an image. Only a small amount of “representative” matrix elements are distilled into a very compact image descriptor for database storage and retrieval. The comparison between a given query descriptor and the database descriptors is performed by a similarity measure by accumulating the differences between corresponding values of the matrix elements. Elements that are not common for both compared descriptors are naturally assumed to be equal to zero. In terms of image content such a measure expresses differences in relative areas of corresponding “elementary” color segments and their boundaries quantitatively. This measure is independent of the number of those segments that are actually present in both of the compared images.

2. Method

The general scheme of the presented method consists of two stages. At the first stage, image descriptors are generated for storage in a remote database. For this purpose, color co–occurrence matrices are adapted. Representative matrix elements that describe the frequency of spatial co–occurrence of ”elementary” image structures are distilled into compact image descriptors.

At the image retrieval stage, a query image is compared with all potential images of the remote database. For this task the similarity between two images is calculated by a measure of dissimilarity between their corresponding descriptors. The most similar images for a given query image can be selected by simply sorting the resulting similarity measures according to their magnitude.

The consideration of image features as a set of corresponding matrix elements allows the combination of several query images into a single query by using logical operators. Such an option provides the user with an additional, simple and powerful way to achieve desirable retrieval results with no tiresome parameter tuning.

2.1. Image Storage

2.1.1 Color Co–occurrence Descriptors

Generally, any image can be considered as a composition of suitable ”elementary structures”. The elements of those structures (pixels) carry visual attributes (colors) and possess relations (distances between them). Consequently, the image content can be characterized by an appropriate M–dimensional co–occurrence matrix where attributes and relationships are represented by the different matrix axes [14]. A 2D color image can be sufficiently represented by a three-

dimensional matrix

$$\mathbf{W}\left(c_i, c_j, d_{ij}\right), \tag{1}$$

where c_i and c_j are indices of suitably quantized RGB color intensities of the pixels i and j, d_{ij} is the Euclidean distance between pixels i and j, and $\mathbf{W}$ is the frequency of the spatial occurrence of the elementary structures in the image. The matrix is invariant to translations and rotations of the image given that all possible d-neighbors are considered.

In the framework of this paper, the equivalence of an image and its co–occurrence representation is not discussed with further detail, because it ultimately converges to a combinatorial problem of image coding and can be formulated in terms of image transforms that do not affect the co–occurrence descriptors. In particular to prove this, it is good enough to investigate the rotation, reflection, and translation of a single image object on a sufficiently large homogeneous background ($d > d_{max}$).

2.1.2 Descriptor Calculation

Practically, the matrices are quite sparse and can be efficiently stored as a set of matrix elements

$$E_k \in \{\ (i_k, w_k) \ \mid \ \exists\, w_k = \mathbf{W}(c_i, c_j, d_{ij}) \neq 0 \ \wedge\, i_k = f(c_i, c_j, d_{ij})\ \} \tag{2}$$

where i_k is the element's index and w_k its value. The sparse matrix element's index is derived from the color indices c_i and c_j, and the distance index d_{ij} by an appropriate mapping function. The values w_k of the matrix elements are normalized with respect to the size of the individual image in order to avoid the dependence of retrieval results from the image sizes.

In terms of image content an elementary image structure E_k describes the relative area of a corresponding color segment if $c_i = c_j$, or the relative length of the borders between different segments if $c_i \neq c_j$. It is also important, that the set of elementary structures for any given image is not arbitrary and indirectly describes the mutual spatial relations between its segments, that is, the visual content of whole image scene.

The number of "representative" elements E_k depends on the image content and varies for different images. A trivial image with all pixels having the identical color is well enough described by only one matrix element, whereas an image containing uncorrelated color noise might need all matrix elements to be described sufficiently. Natural scenes with a large number of small color segments and a sophisticated mutual spatial distribution of those segments in the image frame require a considerably large number of elements. In any case, the image descriptor size is independent from the original image size, because the area information is stored in the magnitude of w_k.

A threshold A_{min} as a minimum value for w_k that expresses the minimum relative area of a considered elementary image structure E_k is used as the control parameter for the selection of the representative matrix elements. This control parameter is simple, intuitive, and defines the desirable level of image details to be considered during the retrieval stage. A second threshold A_{max} is introduced essentially to prevent that small — with regard to the image content possibly important — image elements can be numerically outgained and dominated by large — with regard to the image content possibly irrelevant — image elements such as the image background.

2.1.3 Descriptor Properties

The most important properties of the proposed image descriptors can be summarized as follows:

- invariance versus affine image transforms like rotation, reflection and translation
- independence from image size
- ability to retrieve images with a specified subimage from a query example.

These properties are demonstrated and quantitatively evaluated by appropriate experiments in section 3.

2.2. Image Retrieval

2.2.1 Dissimilarity Measure

The goal of image retrieval is to compare a given query image with all potential target images in order to obtain numerical measures of their similarity with the query image. These measures are commonly determined by some kind of distance between the corresponding descriptors in feature space. In the case of color co–occurrence descriptors the problem can be formulated in terms of determining their dissimilarity. The dissimilarity of the query image descriptor T_q and a potential target image descriptor T_t can be quantitatively captured by calculating the normalized difference between the sets of corresponding elements that are featured in both descriptors

$$D\left(T_q, T_t\right) = \frac{\sum\limits_{E_k \in T_q \cap T_t} \left| w_k^q - w_k^t \right|}{\sum\limits_{E_k \in T_q} w_k + \sum\limits_{E_k \in T_t} w_k} \tag{3}$$

The measure itself is symmetric ($D(T_q, T_t) = D(T_t, T_q)$) and is ranged within $[0, 1]$. The corresponding similarity measure can be simply derived from $D(T_q, T_t)$ as

$$S(T_q, T_t) = 1 - D(T_q, T_t) \tag{4}$$

2.2.2 Combination Queries

N queries can be combined by a logical OR operator to yield into a single query score. A virtual query descriptor V_q is constructed from the image descriptors T_q^1, T_q^2, ..., T_q^N and their elements:

$$V_q\left(\bigcup_i T_q^i\right) = \left\{E'_k | E'_k \in \bigcup_i T_q^i\right\} \tag{5}$$

with

$$w'_k = \begin{cases} \dfrac{\sum_i w_k^i}{N} & \text{if } E'_k \in \bigcap_i T_q^i \\ \dfrac{w_k}{N} & \text{otherwise} \end{cases} .$$

3. Results

All results were derived from a database of 20000 arbitrary images. For the descriptor calculation all images were normalized to 128x128 by tri–linear interpolation. In order to keep the complexity of the co–occurance matrix manageable, the color indices were crudely quantized to a maximum of 6 possible values per RGB color channel and only pixels within a city–block distance of 1 of each other were considered.

Figure 1 shows typical histograms of the image descriptor sizes for $A_{min} = 0.002$ and $A_{min} = 0.005$ (minimum area of the considered image structures with 0.2% and 0.5% of the image area respectively). For all subsequent experiments A_{min} was chosen to be equal to 0.002 resulting in an appropriate average size of 66 matrix elements per image descriptor. A_{max} was set empirically to 0.200.

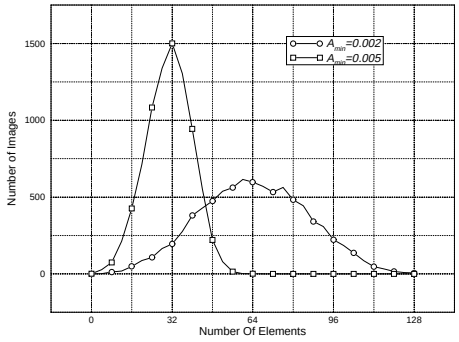

Figure 1. Histogram of descriptors for different minimum thresholds

3.1. Qualitative Estimation of Query Results

All results, no matter how objective they appear, depend on the content of the database that is used. This manifests one of the major problems of CBIR, since a query is usually submitted without any knowledge about the images contained in a database. Furthermore, query results are judged differently by individuals. Hence, the retrieval results presented only exemplarily show the retrieval performance of the proposed technique.

Figure 4 and 5 demonstrate examples of the querying results. The qualitative assessment is left to the subjective judgment of each individual viewer. However, these examples prove that the method provides good results for both color and gray scale images. In most cases the best query results includes some related images of the same topic. The rightmost column in figure 5 demonstrates the independence of the suggested method with regard to affine image transformations along with its robustness against aliasing effects and smaller changes in details.

As mentioned previously, the proposed method allows the combination of different query images into a single composite query. Figure 4 shows a simple example for two query images as well as their combination into a single composite query. As a result, images that combine most of the main features from both query images were retrieved (see the bottom row in figure 4). However, it can be clearly seen, that the results of querying by a combination of many image examples is a difficult task and is yet not fully understood by humans. This is mainly due to the fact that humans still tend to think in terms of meanings rather than in terms of image features. Considering this, it seems only a matter of human understanding in order to come up with useful results by combinatorial querying.

Finally, it should be mentioned that just because the "perfect" image is not found, does not necessarily mean the algorithm is not capable of finding it, but rather that there is simply not a similar image in the database.

3.2. Analytical Tests

In order to compare the retrieval abilities of these image descriptors more analytically, the following experiments have been performed: 1000 randomly chosen images from the database were subsequentially distorted by some means and used as an example to query the database:

Noise: white noise ranging from 0.0 to ± 0.5 added to each of the normalized RGB values of the images.

Subimages: arbitrary positioned subimages of relative size from 0.0 to 1.0 of the original images.

If the distorted image was able to find itself among the best 20 query results, the experiment was rated as a successful

retrieval otherwise as a failed. The results of these experiments are depicted in figures 2 and 3 correspondingly.

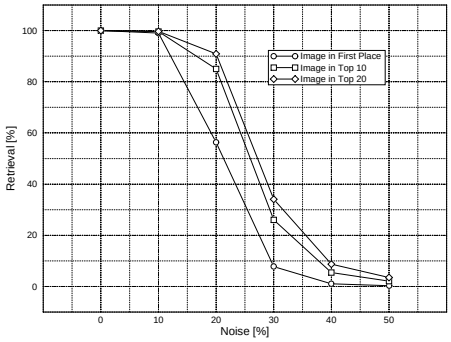

Figure 2. Retrieval performance with respect to added white noise

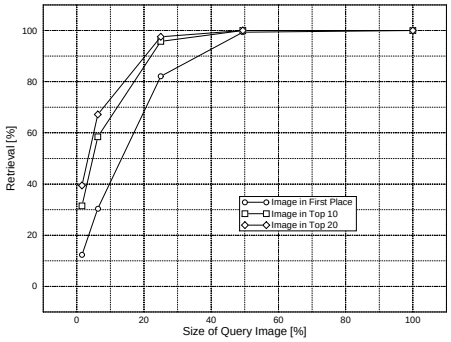

Figure 3. Retrieval performance with respect to the relative size of random subregions

3.3. Speed

The computation of the extraction of a single descriptor from an image takes about 160 msec on a Sun SPARCstation 20 MP (2 X 390Z50) running SunOS 5.5.1. The time needed for the evaluation of a query submitted to a database with 20000 images (including the sorting of the query result) accumulates to a total of 600 msec. This shows, that the method is well suitable as an interactive technique even for larger databases.

4. Conclusions

A new method, that is based on novel color co–occurrence descriptors utilizing a compact representation of the visual image content, has been proposed for image retrieval from large databases. The image information is represented by the descriptors on the basis of the frequency of spatial occurrence of certain “elementary structures” within the image area. The single elements of the descriptors essentially represent the color information of image segments and their borders (ranging from homogeneous color segments to high–frequency color textures), whereas the whole set of elements implicitely represents the mutual distribution of those structures in the image scene. The consideration of image descriptor as a set of elements allows the combination of several image examples by combinatorial operators into composites queries. The advantages of the method include the independence of results from affine transforms like image rotation, reflection, and translation.

4.1. Future Work

Future works are aimed towards a practical relevance of the proposed method for the retrieval of multimedia objects of any kind. This mainly targets the development of a concept for the sophisticated specification of a composite query. The authors strongly believe that the combination of different query specifications should affect the similarity measure directly instead of simply accumulating the single query results arithmetically. Furthermore, the ultimate image retrieval system has to “learn” the relevances of image features from the user interaction by “visually” combining positive and negative examples.

References

[1] J. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jian, and C. Shu. The Virage Image Search Engine: an Open Framework for Image Management. In *Storage and Retrieval for Image and Video Databases IV*, pages 76–87. SPIE, 1996. [http://www.virage.com/literature/spie.pdf].

[2] D. Chetverikov and K. Gede. Textures and Structural Defects. In *Proc. of 7th Int’l Conf. on Computer Analysis of Images and Patterns*, pages 167–174. Springer–Verlag, 1997.

[3] A. Del Bimbo and P. Pala. Image Retrieval by Elastic Matching of User Sketches. In *Proc. of 8th Int’l Conf. on Image Analysis and Processing*, pages 185–190. Springer–Verlag, 1995.

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image Content: the QBIC System. *IEEE Computer*, 28(9):23–32, September 1995.

[5] C. Frankel, M. Swain, and V. Athitsos. Webseer: an Image Search Engine for the World Wide Web. Technical Report 96–14, University of Chicago, August 1996.

[6] M. Galloway. Texture Analysis Using Gray Level Run Lengths. *Computer Graphics and Image Processing*, 4:172–179, 1975.

[7] J. Haddon and J. Boyce. A Relaxation Computation of Optic Flow from Spatial and Temporal Co–occurrence Matrices. In *Proc. of 11th Int'l Conf. on Pattern Recognition*, pages 594–597, 1992.

[8] J. Haddon and J. Boyce. Co–occurrence Matrices for Image Analysis. *Electronics & Communication Engineering Journal*, 4:71–83, 1993.

[9] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient Color Histogram Indexing for Quadratic Form Distance Functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

[10] R. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Trans. on Systems, Man and Cybernetics*, SMC–3(6):610–621, November 1973.

[11] K. Hirata and T. Kato. Query by Visual Example — Content Based Image Retrieval. In *Advances in Database Technology*, pages 56–71. Springer–Verlag, 1992.

[12] C. Jacobs, A. Finkelstein, and D. Salesin. Fast Multiresolution Image Querying. In *Proc. of SIGGRAPH '95*, pages 277–286. ACM, 1995. [ftp://ftp.cs.washington.edu/tr/1995/01/UW-CSE-95-01-06/UW-CSE-95-01-06-color.ps.gz].

[13] A. Kankanhalli, H. Zhang, and C. Low. Using Texture for Image Retrieval. In *Proc. of the Int'l Conf. on Automation, Robotics, and Computer Vision*. Nanyang Technological University, Singapore, 1994.

[14] V. Kovalev and M. Petrou. Multidimensional Co–occurrence Matrices for Object Recognition and Matching. *Graphical Modells and Image Processing*, 58(3):187–197, May 1996.

[15] V. Kovalev, M. Petrou, and Y. Bondar. 3D Surface Roughness Quantification. In *Proc. of 8th British Machine Vision Conf.*, pages 450–458, 1997.

[16] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC Project: Querying Images by Content Using Color, Texture, and Shape. In *Storage and Retrieval for Image and Video Databases*, pages 173–187. SPIE, 1993.

[17] J. Smith and S. Chang. VisualSEEK: a Fully Automated Content–Based Image Query System. In *Proc. of ACM Multimedia*, 1996.

[18] D. Tegolo. Shape Analysis for Image Retrieval. In *Storage and Retrieval for Image and Video Databases II*, pages 59–69. SPIE, 1996.

[19] S. Volmer. Tracing Images in Large Databases by Comparison of Wavelet Fingerprints. In *Proc. of the 2nd Int'l Conf. on Visual Information Systems*, pages 163–172, December 1997. [http://www.igd.fhg.de/~volmer/visual97.ps.gz].

[20] D. White and R. Jain. Similarity Indexing: Algorithms and Performance. In *Storage and Retrieval for Image and Video Databases IV*, pages 65–72. SPIE, 1996. [ftp://vision.ucsd.edu/pub/dwhite/spie.ps.gz].

Color Figures

Color prints of the following figures can be obtained from `http://www.igd.fhg.de/~volmer`.

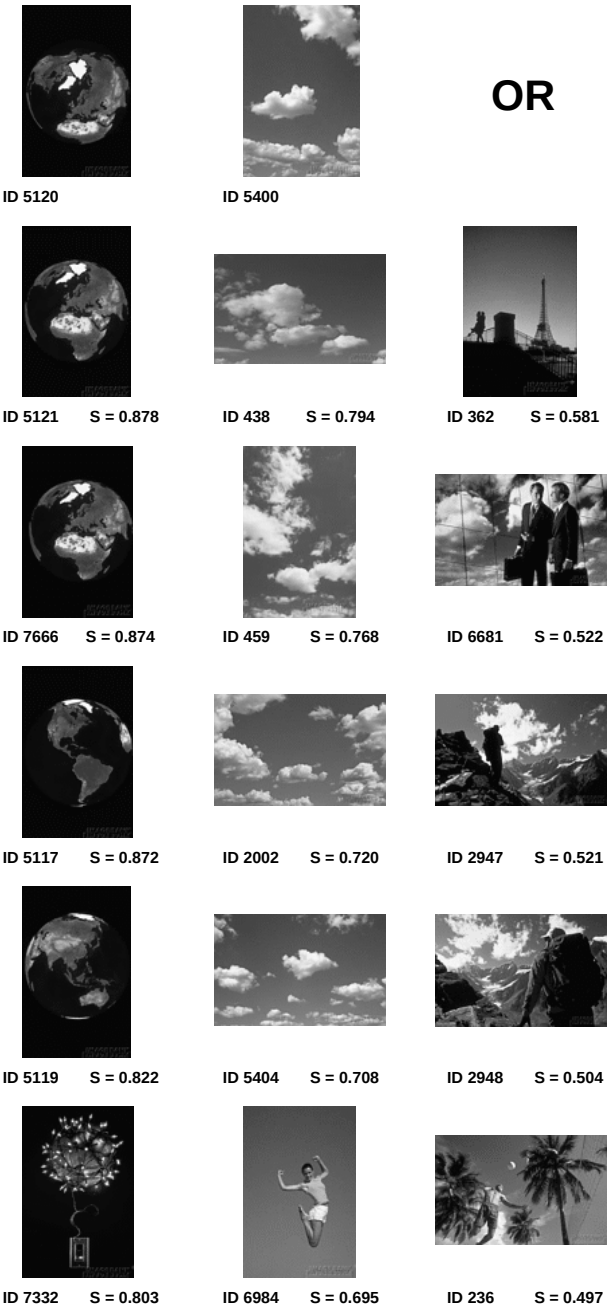

Figure 4. Top query results (from top to bottom) for different query examples (left and middle row) and their combination (right row)

Figure 5. Query results (from top to bottom) for different query examples (top row)