Annals of University of Craiova, Math. Comp. Sci. Ser.
Volume 33, 2006, Pages 87–93
ISSN: 1223-6934

Combining Color and Shape Features for Efficient Indexing and Image Retrieval

Maria Magdalena Ruxanda

Abstract. We present a method for image retrieval by color and shape similarity matching using metric indexing. We extract color features into the HSV color and for shape features extraction we use Fourier descriptors. We index the combined color and shape features vectors into a M-tree. We complement the use of the M-tree with a filtering technique that also permits partial-image matching. We use a similarity measure calculated as an affine linear combination of L1 and L2 metrics. In the refinement query step for color extraction we use a special histogram with a uniform color transition which enables a window-based smoothing during retrieval. Our experimental results show that our technique is both correct and efficient.

2000 Mathematics Subject Classification. Primary 68U10; Secondary 68P20.
Key words and phrases. image retrieval, similarity search, index structures, image feature computation.

1. Introduction

Visual information plays an important role in our society in fields like biomedicine, military, education, commerce, entertainment, crime prevention, and others. Such large image databases need efficient image retrieval methods based on the contents of the images. In modern CBIR systems, visual content is addressed by low-level features such as color, texture, shape or spatial relationship.

Image retrieval involves three primary issues: feature extraction, similarity measure and feature vectors indexing. While a large work has been done around each of them, limited attention has been placed on the combination of useful multi-dimensional data representations and similarity models with efficient index structures.

We are aiming at developing an image retrieval system using and combining efficient techniques for all three issues mentioned above.

Our similarity matching is based on combined color and shape features. For color extraction we have chosen to work in HSV color space as being closer to human visual perception. The visual properties of HSV color space and its usefulness in content based image retrieval applications were analyzed in-depth by Sural et al [8]. A three dimensional representation of the HSV color space is a hexacone, where the central vertical axis represents the Value or intensity of the color having values from 0 to 1, Hue is defined as an angle in the range $[0, 2\pi]$ and Saturation is the purity of the color and is measured as a radial distance from the central axis with value between 0 at the center and 1 at the outer surface.

Our shape feature extraction method is based on Fourier Descriptors because it achieves both well representation and well normalization [7].

Indexing the search space improves considerably the efficiency of the search [1][2]. Today, a great challenge in database area is to manage various nontraditional types of

Received: August 2005.

data such as video, audio, image, text and to develop efficient content-based retrieval techniques for such data. In particular, it is of vital importance to use indexing structures able to support execution of similarity queries. There have been proposed several such indexing structures, grouped in two main categories: SAMs (Spatial Access Methods) such as the R-tree and its variants, and the metric trees such as the M-tree and the MVP-tree. Metric trees represent a more general approach to the similarity indexing problem [2]. In order to organize the search space, metric trees consider relative distances of objects, rather than their absolute positions in a multi-dimensional space. Also, metric trees only require that the function used to measure the distance between objects is a metric, so that the triangle inequality property applies and can be used to prune the search space. We shall use the M-tree for indexing the feature vectors, as it has been proven to be the most flexible and efficient metric tree [2].

We complement the use of the M-tree by applying a filtering technique which permits both a partial image matching, as well as a refined feature extraction on smaller regions of the image. Thus, the database can be queried in two steps [12].

First, we extract color and shape features on entire image area. For color extraction we calculate mean values of Hue, Saturation and Value components on blocks of the image. For shape extraction we use a Fourier Descriptors method. We index the color and shape feature vectors thus extracted, into an M-tree. The metric that organizes the M-tree is also used as a similarity measure in both query steps, being calculated as an affine linear combination of Manhattan and Euclidian distances (L1 and L2 metrics).

During the second query step, only the candidate images returned in the first step are dynamically processed, extracting new local features. The second query step, introduces a refined similarity matching on a smaller region of the image, selectable by the user. The new local shape features are extracted with the same Fourier Descriptors method but only on the selected region. For new local color extraction we generate a special 101 bins histogram to which each pixel contributes either by its Hue or Value component, as it is described in [8]. Because this histogram provides a perceptual gradation of colors in the feature vector, the retrieval result in the refinement step can be further tuned by comparing two histograms through smoothing windows instead of comparing the feature vector components directly [8]. Applying this result, as our experimental tests will show, will considerably improve the partial-image similarity matching.

The paper is organized as follows. Sections 2 and 3 present our approach in features extraction on entire image area and indexing the feature vectors. Section 4 presents our idea of complementing the use of the M-tree with a filtering technique in order to refine the image retrieval process and to support sub-image matching. Section 5 describes the smoothing window improvement strategy applied in the refinement query step. In Section 6, some experimental results are shown. Finally, Section 7 concludes the paper.

2. General Features Extraction

In order to build the index structure and to perform similarity queries, we first need to extract color and shape features from images in the database on entire image area. Because we shall complement the use of the M-tree with a filtering technique, we can trade at this level some extra loss of precision in feature extraction for a smaller number of features. So, each image will be automatically processed as follows.

For color extraction, we simply partition the image into 16 blocks (4 x 4 grid), and for each block we calculate the mean values of Hue, Saturation and Value components of the HSV color space.

For shape extraction, the process goes through several steps. The image is first transformed into a gray level image, then it is transformed into a binary one. Further, the binary image suffers a denoise process in order to eliminate those isolated pixels or small isolated regions because very often the shape obtained from the binary image has noise around the shape boundary. Then, the boundary is finally traced using a 8-connectivity contour tracing technique [6].

We remind that the Fourier Descriptor C_k is defined as the k-th discrete Fourier transform coefficient [6]:

$$C_k = \sum_{n=0}^{N-1} z_n e^{-2\pi ikn/N}, \qquad k = 0, 1, ..., N-1$$

where $z_0, z_1, ..., z_{N-1}$ represents the boundary coordinates in a counter-clockwise order.

Next, we apply the following theoretical results proven in [6]. For rotational invariance and invariance with respect to the starting point we use only the absolute values of the descriptors C_k. For translational invariance we discard the Fourier Descriptor C_0 and for scale invariance we divide the Fourier Descriptors by $\mid C_1 \mid$. The number of coefficients generated from the transform is usually large. As we are interested in minimizing as much as possible the size of the feature vectors, but in the same time achieve a good representation of shape, we finally use only a limited number of 20 Fourier Descriptors, more precisely those that describe the lowest frequencies and that contain information about the general features of the shape.

3. Indexing and Filter Query Step

We index the 68 element feature vectors previously extracted (48 color features and 20 shape features) into a M-tree.

Among other metric trees, the M-tree stands out as the single investigation of a fully general external metric-space index structure [4]. It is a paged, balanced and dynamic radius-based structure providing performance optimization concerning both CPU (distance computations) and I/O costs [2]. We remind that the M-tree partitions objects on the basis of their relative distances, as measured by a specific distance function d, and stores these objects into fixed-size nodes. So, the M-tree is fully parametric on the distance function d.

We propose a distance function calculated as a linear combination of two L_p metrics. It can easily be proven that our distance function is also a metric. We calculate the Manhattan distance (L_1 metric) for color feature elements and the Euclidian distance (L_2 metric) for shape feature elements, finally summing the two values after applying to them different weights. In this way, we can stress the contribution of one set of features which may be considered more important relatively to the description of the images in the database. Moreover, combining the two distances, we will somehow reduce the negative effect of each in part. For instance, L_1 metric may cause too few of the images that should be returned to be actually retrieved, while the L_2 metric may cause the opposite effect. We have chosen to use Manhattan distance for color features and not for shape features for the same reason, as we represent color based on a very general approximation (by mean values on blocks of the image).

Let v_1 and v_2 be the feature vectors representing two objects that we want to compare:

$$v_1 = (c_1^1, ..., c_M^1, s_1^1, ..., s_N^1)$$
$$v_2 = (c_1^2, ..., c_M^2, s_1^2, ..., s_N^2)$$

where M is the number of color features and N is the number of shape features. Then,

$$Color\_Dist = \sum_{i=1}^{M} \mid c_i^1 - c_i^2 \mid \tag{1}$$

$$Shape\_Dist = \sqrt{\sum_{i=1}^{N} (s_i^1 - s_i^2)^2} \tag{2}$$

$$d = a * Color\_Dist + b * Shape\_Dist, \qquad a + b = 1 \tag{3}$$

At this stage, users can perform K-NN similarity queries searching for images that are similar to the query image on its entire area. We shall refer to this query step as being the "the filter step" (see Figure 2b). The similarity measure used in this filter step is the same distance function d of the index structure.

4. Local Features Extraction and Refinement Query Step

We propose to complement the use of the M-tree with a filtering technique in order to perform a more precise similarity matching and to support sub-image matching. Our basic idea that supports the filtering technique is very simple: "if two images are similar on their entire area, then they will be similar on a smaller region as well".

If the user is interested in a more detailed image matching, he/she can perform a second query step, which we shall call it "the refinement step". In order to do so, the user can further select a smaller region and perform a refined similarity search but only on that region. For image regions the user has a choice of 9 predefined regions: "top-left", "top-center", "top-right", "middle-left", "middle-center", "middle-right", "bottom-left", "bottom-center", "bottom-right" (see Figure 1). During the refinement step, only the candidate images returned in the filter step are dynamically processed, extracting new local color and shape features. The similarity measure is calculated with the same formula (3) as in filter step, but using the newly extracted features.

Working with smaller regions of the image and applying the calculations on a small number of candidate images, we can afford to perform a more refined and accurate feature extraction.

For local shape extraction we use the same Fourier Descriptors method presented in the previous sections, and we have considered for our tests a number of 10 Fourier Descriptors describing lowest frequencies.

For local color extraction we work again in the HSV color space, but this time taking full advantage of some of its properties. Analyzing the HSV color space properties, one can notice that for $Saturation = 0$, as the Value increases along the vertical axis, the color goes from black to white through various shades of gray. Also, for a given Value and Hue, if the Saturation is changed from 0 to 1, the color changes from a gray shade to the most pure form of the color represented by its Hue. So, we can conclude that by sufficiently lowering the Saturation, any color in the HSV color space will be a gray value whose particular gray shade is determined by the Value component, while for increasing the Saturation towards 1, the colors are perceived as true colors represented by their Hues. Sural et al [8] proved that a Saturation value of 0.2 is an appropriate threshold to differentiate between Hue and Value dominance.

We shall use this result to generate a special histogram where each pixel in the selected region will be represented by its Hue if its Saturation is above 0.2, or by its Value if the Saturation is less than 0.2. We normalize this values by transforming them into integer values in the range $[0, 100]$ and so we have obtained a 101 element vector that efficiently represents the color feature on the selected region.

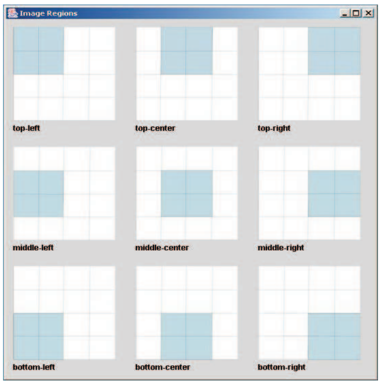

Figure 1. Image regions.

5. Further Improvement on Refinement Query Step

It has been observed that when color histograms are extracted from similar images, often two neighboring components have high values. But when a standard measure like the Euclidian distance is used to compare such feature vectors, it might result in a high distance value. To overcome this issue, Sural et al [8] proposed to compare the histograms through smoothing windows instead of compare the vector components directly. This idea can be applied here because the histogram we use provide the required perceptual gradation of colors in the feature vectors.

So, before applying the similarity measure in the refinement step, each element of the local color feature vector will be transformed as follows. Let $v = (v_0, v_1, ..., v_{100})$ be the local color feature vector, vt the transformed one and N the size of the smoothing window (for our experimental tests we have considered N=5). Then,

$$vt_j = \sum_{i=j-N}^{j+N} 2^{-|i-j|} v_i, j = 0, 1, ..., 100 \quad (4)$$

6. Experimental Results

We have tested our image retrieval method on a small database of 300 natural scene images. For the automatic extraction of general feature vectors we have used a Matlab program. Our M-tree implementation is based on the XXL v.1.0 Java library. The index structure is disk resident and implemented with the following parameters: the minimum capacity of nodes is 10, the maximum capacity of nodes is 15, and the type of split is hyperplane.

The generation of the 101 bins histogram used in the query refinement step is also implemented in Java, as well as the system user interface.

For our tests, we set the parameters a and b used in formula (3) at $a = 0.55, b = 0.45$ for the query filter step and at $a = 0.65, b = 0.35$ for the query refinement step. We have chosen to apply a bigger weight value for color features contribution in the similarity measure because the number of color features in the feature vector is much larger than the number of shape features. Also, because we extract color in a more precise manner in the refinement step, we have increased the color features weight from 0.55 to 0.65. Nevertheless, the similarity measure formula is flexible allowing any combination of values for these parameters for the best tuning of the retrieval process.

Some retrieval examples are shown in Figures 2. We have performed K-NN queries for k=6. Retrieved images are presented in decreasing order of similarity from left to right. As expected, the refinement query step introduces a more accurate similarity matching on sub-images. Figure 2c shows such retrieval results for a partial image matching on region "middle-center". The smoothing window technique further improves the retrieval results on sub-images (see Figure 2d).

Figure 2. Query results.
a. Query image. **b.** Retrieval results for filter query step. **c.** Retrieval results for refinement query step on region "middle-center". **d.** The same as **c.** but applying the smoothing window with size 5.

7. Conclusion

We have proposed a method for image retrieval based on color and shape similarity matching. We have combined efficient techniques for feature extraction, feature vectors indexing and similarity search. Our idea of complementing the use of the M-tree with a filtering technique allows the user to perform both a similarity search on entire image area and a more refined partial image matching only on those regions of the image that he/she is more interested in. Applying the smoothing window technique further improves the quality of similarity matching on sub-images. Even when testing on such a small database, the experimental results prove both the efficiency and the correctness of the proposed method.

References

[1] P. Zezula, P. Savino, G.Amato, F. Rabitti, Approximate Similarity Retrieval with M-trees, *The VLDB Journal*, Springer-Verlag, 1998
[2] P. Ciaccia, M. Patella, P. Zezula, M-Tree: An Efficient Acces Method for Similarity Search in Metric Spaces, *Proc. of the 23rd VLDB Conference*, 1997
[3] Anne H. H. Ngu, Q. Z. Sheng, Du Q. Huynh, Ron Lei, Combining Multi-visual Features for Efficient Indexing in a Large Image Database, *The VLDB Journal*, 2001
[4] Rui Mao, W. Liu, Q. Iqbal, D. P. Miranker, On Index Methods for an Image Database, *ACM-MMDB*, 2003
[5] S. Beretti, A. Del Bimbo, P. Pala, Retrieval by Shape Using Multidimensional Indexing Structures, *10th International Conference on Image Analysis and Processing*, 1999
[6] T. Petkovic, J. Krapac, Technical Report. Shape Description with Fourier Descriptors, 2002
[7] D. Zhang, G. Lu, A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures, *Proc. of ICIMADE*, 2001
[8] S. Sural, G. Qian, S. Pramanik, Segmentation and Histogram Generation using the HSV Color Space for Image Retrieval, *IEEE- ICIP*, 2002
[9] T. Seidl, H. P. Kriegel, Efficient User-Adaptable Similarity Search in Large Multimedia Databases, *Proc. of the 23rd VLDBConference*, 1997
[10] S. Lin, M. T. Ozsu, V. Oria, R. Ng, An Extendible Hash for Multi-Precision Similarity Querying of Image Databases, *Proc. of the 27th VLDBConference*, 2001
[11] I. Kompatsiaris, E. Triantafillou, M. G. Strintzis, Region-Based Color Image Indexing and Retrieval, *Proc. of IEEE International Conference on Image Processing*, 2001
[12] M. M. Ruxanda, A Framework for Image Retrieval by Color and Shape, *To Appear in Proc. of AIDC*, Craiova 2005

(Maria Magdalena Ruxanda) Computer Science Department, University of Craiova,
Al.I. Cuza Street, No. 13, Craiova RO-200585, Romania
E-mail address: magda_ruxanda@yahoo.com