U.P.B. Sci. Bull., Series C, Vol. 68, No. 1, 2006

AN INDEXING SCHEME FOR CONTENT-BASED RETRIEVAL OF IMAGES BY SHAPE

Irina MOCANU[*]

Apariția calculatoarelor personale şi a stațiilor de lucru performante, a standardelor de compresie audio/video precum şi existența diferitelor aplicații ce folosesc biblioteci digitale sau baze de date medicale au crescut rolul sistemelor de baze de date multimedia. Deoarece datele vizuale necesită o cantitate mare de memorie pentru stocare, precum şi o putere de calcul mare necesară procesării acestor informații au fost necesare structuri de indexare, precum şi algoritmi eficienți de regăsire a acestor informații. În cadrul regăsirii imaginilor bazate pe conținut, acestea sunt descrise pe baza caracteristicilor de nivel scăzut, cum ar fi culoare, forma, textură sau combinații ale acestora. Dintre aceste caracteristici forma reprezintă caracteristica cea mai importantă. Lucrarea prezintă o schemă de indexare pentru regăsirea imaginilor bazată pe forme.

Multimedia database systems are becoming increasingly important with the advent of high-powered PC's and workstations, audio/visual compression standards, and many applications such as digital libraries or medical databases. Because visual data require a large amount of memory and computing power for storage and processing, it is greatly desired to efficiently index and retrieve the visual information from image database systems. In content-based image retrieval, several low-level image features, such as color, texture, shape or the combination of these features, describe image. Shape is an important low-level image feature. This paper presents an indexing scheme for image retrieval by shape.

Keywords: software architecture, image retrieval, shape, distance histogram, content-based indexing, R*-tree.

Introduction

In the recent years, image database systems are increasingly important with the advent of broadband networks, high-powered PC's and workstations and many applications such as digital libraries, medical databases, trademark and copyright databases. Because images data require a large amount of memory and computing power for storage and processing, it is greatly desired to efficiently index and retrieve the visual information from image database systems.

[*] Assist., Dept. of Computer Science, University "Politehnica" of Bucharest, Romania

State of the art in content-based image indexing is based on feature extraction. The features are extracted (semi)automatically from the images objects [1] and mapped into points in n-dimensional vector space. An image object is represented by a set of feature vectors which represent color, shape or texture. After obtaining feature vectors, any multi-dimensional indexing method (like k-d-b trees, R-trees, R+-trees, R*-trees) can be used to organize, cluster and efficiently search the image objects as points in the n-dimensional vector space.

Among features used for image objects representation shape is an important image feature. A good shape representation and similarity measurement for recognition and retrieval purposes should have the following two important properties [2]:

(i)each shape should have a unique representation, invariant to translation, rotation, and scale;

(ii)similar shapes should have similar representations so that retrieval can be based on distances among shape representations.

There are generally two types of shape descriptors: contour-based shape descriptors and region-based shape descriptors. Contour-based shape descriptors exploit only boundary information, they cannot capture shape interior content and these methods cannot deal with disjoint shapes. In contrast, in region-based techniques all the pixels within a shape region are taken into account to obtain the shape representation. These techniques can describe disjoint shapes. This paper presents an indexing scheme used together with a contour shape based method for image retrieval by content. This paper describes two contour-based methods for image retrieval by shape: the distance histogram method and the method based on centroid radii and turning angle (CRTA) and proposes an indexing scheme used together with CRTA method. The experimental results are presented, too.

Distance Histgram Method

A method for shape representation and retrieval based on the shapes' contour – the distance histogram method - is presented in [3]. The distance histogram method computes a histogram based on the distances of the shape's radii measured from the shape's centroid to its boundary. The main steps of the method are:

1. calculate the centroid of the shape $c = (x_c, y_c)$:
2. for each edge a set of sample points are chosen: the sample points are evenly spread on each edge. The number of the sample points per edge is proportional with the length of the edge.
3. calculate the distances between the centroid of the shape and the sample points (radii).

4. normalize the distances calculate at the step 3: divide the value of all distances by the value of the maximum distance. After that, the value of all the distances will be in [0, 1]
5. calculate the distance histogram of the normalized distances obtained at step 4. The range of all distances [0, D_{max}] is separated into several ranges, i. e. R ranges. Then, the ranges of distances can be represented by: [0, D_{max}/R], (D_{max}/R, 2 D_{max}/R], (2 D_{max}/R, 3 D_{max}/R], …, ((R-1) D_{max}/R, D_{max}], and the distance histogram can be represented as: D: (d_0, d_1, d_2, …, d_{R-1}), where d_i, $i \in [0, R-1]$ is the number of distances belonging to this distance range.

This approach is invariant to translation because the distance set will not change after translating the shape. Because the sample points are chosen based on the length of the edge and are evenly spread on it, two similar shapes with different sizes will generate the same normalized distances. Therefore the method is invariant to scale after normalization.

The similarity between two shapes with the distance histograms D1: ($d1_0$, $d1_1$, $d1_2$, …, $d1_{R-1}$) and D2: ($d2_0$, $d2_1$, $d2_2$, …, $d2_{R-1}$) is given by the Euclidean distance:

$$d\ (D1,D2) = \sqrt{\sum_{i=0}^{R-1} (d1_i - d2_i\ )} \tag{1}$$

This method is robust to small variations on the boundary of a shape. A problem is that the distance histogram method discards the spatial information to achieve the robustness to rotation. Therefore there are situations when different shapes may appear similar when compared using their distance histograms.

A Method for Shape Representation Based on Centroid Radii and Turning Angle (CRTA)

Consider the two different shapes in Fig. 1. The histograms obtained by applying the distance histogram method are presented in Fig. 2. The distance between the two shapes from Fig. 1, computed by (1), will result under a predefined threshold. Therefore, the two shapes will be similar, although they look different. This is because the distance histogram method discards spatial information.

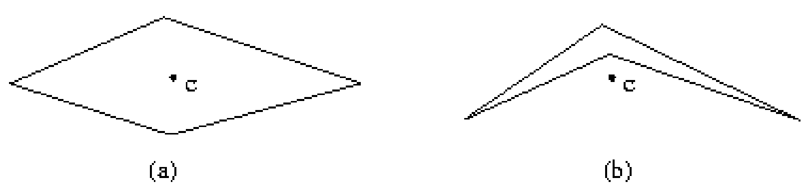


Fig. 1 Two different shapes with similar histograms

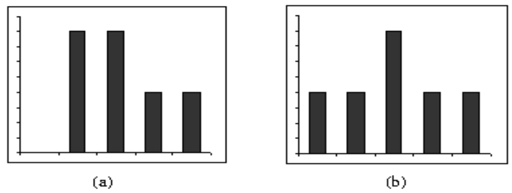


Fig. 2 The histograms of shapes from Fig. 1

To consider spatial representation, in [4] is presented a method which represents a shape using radii and directions of the edges. The directions of the edges are represented by the turning angles of each edge. Turning angle [5] is defined as the angle formed with a reference axis by the counter-clockwise tangent to the boundary of a shape which goes from a boundary point of a shape to the next one. In Fig. 3 the edge directions of the shape are represented by their turning angles, where the reference axis is considered to be the x axis.

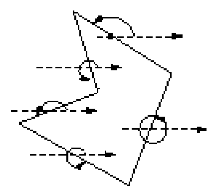


Fig. 3 A shape and turning angles of each edge

Therefore, a shape will be represented by its edge directions (the turning angle of the edge), each edge direction having associated a list of corresponding radii.

The main steps of the proposed method are:

1. determine the edges' directions of the shape;
2. calculate the centroid of the shape;
3. select a set of sample points on the boundary of the polygon and calculate the distances between the sample points and the centroid of the shape, like in distance histogram method;
4. construct a list of edges' directions such that each entry in this list will have associated all the radii corresponding to that direction.

This representation is invariant to translation, but it is not invariant to scale. For making it invariant to scale, like in distance histogram method, all the distances between the centroid and the chosen sample points will be normalized. The normalization of the radii consists in dividing the value of all distances by the value of the maximum distance. The representation is not invariant to rotation. For a shape rotated with an angle α, all the edges' directions will increase or decrease with the same angle α.

To see if two shapes are similar using this method, first the lists of directions must be compared, and if these directions are similar, the radii associated with them must also be compared.

In the above representation one shape can have more edge directions than the other. But it cannot be said that they are different because the shapes can be affected by noise. To make this supposition, it is necessary to smooth the contour of the shape. Therefore the shape representation is obtained based on a smoothing algorithm with linear complexity presented in [6].

The result of the algorithm is a shape without noise. If two shapes generate different numbers of edges' directions, it can be said that the shapes are not similar and this method does not compute the distance between them.

The similarity between two shapes will be calculated in two steps:

1. Compare the edges' directions. For two shapes A and B with lists of edges' directions (dA_1, dA_2, …, dA_n) and (dB_1, dB_2,…, dB_n) they may be similar if their directions correspond. This means that the directions are the same, or one list of directions is rotated with an angle α. For testing this, the differences between each pair of angles need to be computed. If all these n differences are approximately the same (less than a predefined threshold) the shapes may be similar. But when constructing the lists of edges' directions it is not known which edge is the first one. Therefore to see if the directions of the two shapes are the same, the differences between the angles of one list of directions and all the circular shifts of the other list are computed. Doing this, the method will be invariant to rotation and does not care about the starting point. If an equality of all these differences is found, then the shapes may be similar, otherwise the shapes are not similar. If an equality between the two lists of directions exists, suppose the order of these lists of directions is (dA'_1, dA'_2, …, dA'_n) and (dB'_1, dB'_2,…, dB'_n).
2. Compare the radii associated with the lists of edges' directions obtained in step 1. For example, the radii corresponding to direction dA'_1 must be compared with the radii corresponding to direction dB'_1, and so on. In fact the lists of radii must be compared. To compare these values it is sufficient to compare the standard deviation of the radii associated with each direction. The distance between the two shapes will be the Euclidean distance between the standard deviations of radii corresponding to each direction. Consider that the standard deviations of radii corresponding to each direction are ($stdA'_1$, $stdA'_2$, …, $stdA'_n$) and ($stdB'_1$, $stdB'_2$, …, $stdB'_n$). Then the distance between the two shapes will be:

$$d(A,B)=\sqrt{\sum_{i=1}^{n}(stdA'_i-stdB'_i)} \qquad (2)$$

An Indexing Scheme for CRTA Method

To reduce the sequential time for retrieving shapes based on the CRTA method it is necessary an indexing scheme. This section presents an indexing scheme based on the method presented in [7], F-index.

The Indexing Scheme F-index

In [7] is proposed the F-index scheme based on the following problems:

1. completeness of feature extraction: how to extract features and how to guarantee that it is not missed any qualifying object. To guarantee no ”false dismissal”, objects should be mapped to points in k-dimensional space such that the Euclidean distance in the k-dimensional space is less than or equal to the real distance between the two objects;
2. dimensionality ”curse”: most multidimensional indexing methods scale exponentially for high dimensionalities, eventually reducing the sequential scanning. The R-tree based methods are the most robust for higher dimensions. Experiments [8] indicate that R*-trees work well for up 20 dimensions. The feature extraction method should therefore be such that a few features are sufficient to differentiate between objects.

Based on the above problems, the F-index scheme consists of the following steps [7]:

1. obtain the coefficients of the Discrete Fourier Transforms of each sequence in the database based on which will be made the comparisons;
2. build a multidimensional index using the first f_c Fourier coefficients. Thus, each sequence becomes a point in 2 f_c – dimensional space. Based on the results from [8] the indexing structure is a R*-tree;
3. for searching in a database, first will be obtain the first f_c Fourier coefficients for the query sequence. Use the F-index to retrieve the set of matching sequences that are at most ε distance away from the query sequence;
4. the final answer will be obtained in a post-processing step in which the actual distance between two sequences is computed in the time domain and only those within ε distance are accepted.

As the number of Fourier coefficients (f_c) increases, the dimensionality of the R*-tree increases. The increase in dimensionality results in better index selectivity, which gives fewer false hits. This reduction in false hits is reflected in the post-processing time, which decreases with the f_c . However, the time to search the R*-tree increases with the dimensionality. For obtaining an optim between the tree-search time and the post-processing time, based on the experimental results from [7] the optimal value for f_c is 2.

It was choosen the Discrete Fourier Transform because it is the most well known, its code is readily available and it does a good job of concentrating the energy in the first few coefficients.

I_{HR*} indexing scheme

This paper presents an indexing method for CRTA method based on the indexing scheme F-index, presented above. In CRTA method shapes are compared in two steps:

1. compare the directions of the shapes;
2. compare the standard deviation for corresponding directions only for that shapes for which the directions are similar.

Therefore the shapes with similar number of directions with the query shape will be organized in a hash table. Among the shapes with the same number of directions as the query shape, only those with similar directions will be considered. The number of directions can be high and it is necessary to reduce the search space for obtaining the shapes with similar directions. Based on F-index, the proposed indexing scheme, I_{HR*}, consists of a hash table built on the number of directions of the shapes. Each entry in the hash table is a R*-tree of the first f_c Fourier coefficients of the list of edges' directions for corresponding shapes. The resulting indexing scheme is presented in Fig. 4.

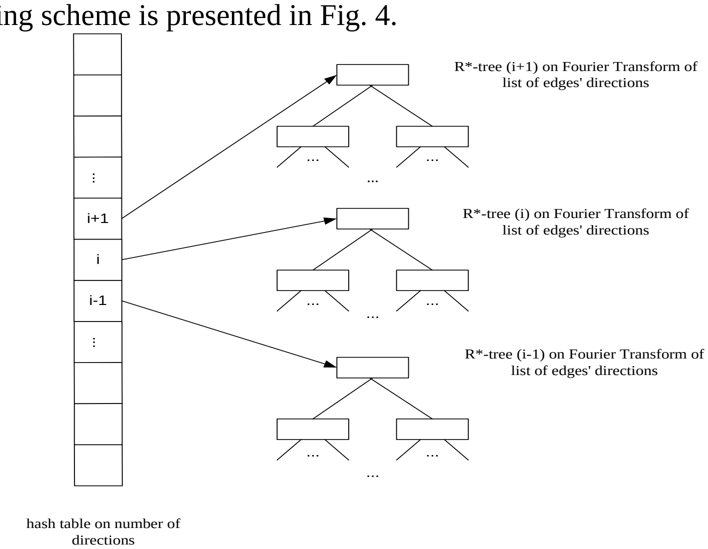


Fig. 4 The indexing scheme IHR*

The algorithm for retrieving similar shapes with a query shape q, based on the I_{HR*} indexing scheme, is described bellow:

Similarity (I_{HR*}, FV^q)
/* the query shape q is described by feature vector
$\mathbf{FV^q}$ = $(d^q_1, std^q_1), (d^q_2, std^q_2), \ldots, (d^q_n, std^q_n)$, where $(d^q_1, d^q_2, \ldots, d^q_n)$ represents edges' directions of the shape and $(std^q_1, std^q_2, \ldots, std^q_n)$ the standard deviations associated with each direction of the shape */

-From index I_{HR*}, choose the entry from hash table for shapes with n directions
-Let R_d the R^*-tree corresponding to the entry from hash table chosen above
-Calculate the Fourier coefficients: $(c^q_1, c^q_2, \ldots, c^q_n)$, for the list of edges of the query shape

```
for i=1 to n execute
/* search in Rd tree for points obtained from the first fc of the list of edge
directions for the query shape; the list of edge directions of the query
shape must be circulary shifted */
    -select the first fc coefficients from (cq1, cq2, ..., cqn ) starting with
    position i
    -let Cfc = (cqi, cqi+1, ..., cqfc-i+1 ) be the Fourier coefficients chosen
    above
    -update the phase for the Fourier coefficients for Cfc (obtained after
    circular shift with i positions in edges' list)
    -search in Rd tree for point formed by the first fc coefficients of Cfc
    if are similar points then
        -verify if lists of edges' directions are similar
        if similar directions then
            -calculate the distance between shapes based on
            their corresponding standard deviations (by
            equation (2) )
            -return distance
        endif
    endif
endfor
end
```

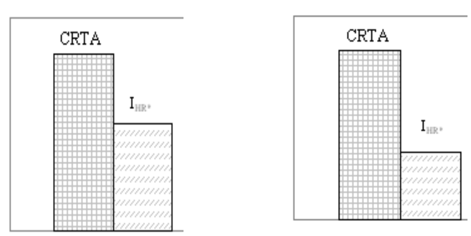
In this case it is not necessary to compare the list of edges' directions of the query shape with all lists of edges' directions for those shapes with the same number of directions as the query shape. [7] proved that:

- the distance in the frequency domain is the same as the distance in the time domain.
- F-index introduces no false dismissals

Based on these assumptions, the search in the R*-tree of the Fourier coefficients of the edges' directions of the shapes with the same number of directions with the query shapes will obtain only those shapes with similar lists of edges' directions. Therefore, the search space is reduced.

Experiments

It was implemented a retrieval framework, on a database with synthetic shapes for testing the indexing scheme I_{HR*}. Synthetic shapes were generated from a database composed of fish shapes used by the SQUID system. For each of these shapes, three affine distorted shapes, one scaled shape, one mirror shape and one flip shape were generated. The used database consists of approximately 3,000 shapes with single contours. Based on the results presented in [7] f_c was considered 2. For each query shape the results were obtained using the CRTA method and the I_{HR*} index. The execution time and the number of compared shapes were measured. The average execution time and average number of compared shapes are presented in Fig. 5.



a) average execution time b)average number of compared shapes

Fig. 5 Experimental results for CRTA method and indexing scheme I_{HR*}

It is visibile that using I_{HR*} index reduces both the average execution time and the average number of compared shapes.

Conclusions

The paper presents a new indexing scheme, I_{HR*}, used together with a method for image retrieval based on shapes' contours. Based on the experimental results the average execution time and number of compared shapes are smaller in case of using the indexing scheme than without index. This happens because the proposed indexing scheme is based on a decreasing dimensionality of the searching space. The indexing scheme uses the first f_c Fourier coefficients of the lists edges' directions and organizes the resulting search space in a R*-tree which works well up to 20 dimensions.

REFERENCES

1. *D. S. Zhang, G. Lu:* An Integrated Approach to Shape Based Image Retrieval. *The Fifth Asian Conference on Computer Vision*, 2002, pp. 652-657
2. *G. Lu: Multimedia Database Management Systems*, Artech House Publishers, Boston, 1999
3. *S. Fan:* Shape Representation and Retrieval Using Distance Histograms. *Technical Report TR*, No. 14, Vol. **1**, 2001, pp. 250-275
4. *I. Mocanu:* Shape Representation and Retrieval Using Centroid Radii and Turning Angle. *Scientific Bulletin of the Politehnica University of Timisoara. Transactions on Electronics and Communications*, No. 49, Vol. **2**, 2004, pp. 146-149
5. *E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, J. S. B. Mitchell:.* An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on PAMI*, No. 13, Vol. **3**, 1991, pp. 232-248
6. *L. P. Cordella, G. Dettori*: An O(N) algorithm for polygonal approximation. *Pattern Recognition Letters*, Vol. **3**, 1995, pp. 93-97
7. *R. Agrawal, C. Faloutsos, A. Swami*: Efficient Similarity Search in Sequence Databases. FODO, 1993, pp. 69-84
8. *R. Beckmann, H. P. Kriegel*: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1990, pp. 322-331