Multimedia Systems 7: 165–174 (1999)

Multimedia Systems

Region-based shape representation and similarity measure suitable for content-based image retrieval

Guojun Lu[1], **Atul Sajjanhar**[2]

[1] Gippsland School of Computing and Information Technology, Monash University, Churchill, Victoria, 3842, Australia
[2] Kent Ridge Digital Labs (KRDL), 11 Science Park Road, Science Park II, Singapore 117685

Abstract. A region-based approach to shape representation and similarity measure is presented. The shape representation is invariant to translation, scale and rotation. The similarity measure conforms to human similarity perception, i.e., perceptually similar shapes have high similarity measure. An experimental shape retrieval system has been developed and its performance has been studied. The shape retrieval performance of the proposed approach is better than that of the more established Fourier descriptor method.

Key words: Shape representation – Shape similarity measure – Pattern recognition – Image retrieval

1 Introduction

With advances in computing and communication technology, more and more images are being captured, stored and used in many areas such as medicine, the press, entertainment, education and manufacturing. In order to make effective and efficient use of information captured in these images, techniques for rapid image retrieval from a large image collection are required. Much research and development attention has been directed to these techniques in the past few years [1–7].

One important approach to image retrieval is based on contents of images such as color, shape and texture. This paper focuses on shape-based image retrieval techniques. We present a new approach to shape representation and similarity measure, called region-based approach, which is suitable for content-based image retrieval. Its retrieval performance is compared with that of the more established method based on Fourier descriptor (FD).

A suitable shape representation and similarity measure for content-based image retrieval should meet the following criteria.

– The representation of a shape should be invariant to scale, translation and rotation.
– The similarity measure between shape representations should conform to human perception, i.e., perceptually similar shapes should have high similarity measure.
– The shape representation should be compact and easy to derive, and the calculation of similarity measure should be efficient.

In the following section, we summarize work in content-based image retrieval related to this paper. In Sect. 3, we describe a region-based approach to shape representation and similarity measure which meets the above criteria. Section 4 presents our experiments and results. Section 5 concludes the paper with a discussion.

2 Related work

There are many papers in the area of content-based image retrieval. In this section, we only summarize the work in which shape feature is used for indexing and retrieval. A detailed comparison between the proposed region-based approach and closely related approaches will be presented in the final section of the paper.

Shape description or representation is an important issue both in image analysis for object recognition and classification and in image synthesis for graphics applications. Many techniques, including chain code, polygonal approximations, curvature, Fourier descriptors and moment descriptors, have been proposed and used in various applications [8, 9].

Recently, content-based image retrieval became important. As object shape is one of the important features of images, a number of shape representations have been used in content based image retrieval systems. Note that in almost all work, techniques integrating a number of features, such as colour, shape and texture, are used. But in this paper, we are only interested in shape representation and similarity measure. In QBIC [10], moment invariants and other simple features such as area are used for shape representation and similarity measure. Mohamad et al. also used moment invariants for trademark matching [11]. But it is found that similar moment invariants do not guarantee perceptually similar shapes. Cortelazzo et al. described a trademark shape description based on chain-coding and string-matching tech-

Correspondence to: G. Lu

nique [12]. Chain codes are not normalized and shape distance is measured using string matching, so it is not invariant to shape scale. In STAR [13, 14], both contour Fourier descriptors and moment invariants are used for shape representation and similarity measure. Jain and Vailaya proposed a shape representation based on a histogram of the edge directions [15]. But the scale normalization with respect to the number of edge points in the image is questionable, as the number of edge points is not directly proportional to scale. Also, the similarity measure is computationally expensive, as it requires to calculate all possible histogram shifts in order to achieve rotation normalization. Mehrotra and Gary used coordinates of significant points on the boundary as shape representation [16]. The representation is not compact and similarity measure is computationally expensive, as these coordinates must be rotated to achieve rotation normalization. In the retrieval techniques proposed by Jagadish [17], shapes are decomposed into a number of rectangles and two pairs of coordinates for each rectangle are used as the representation of the shape. The representation is not invariant to rotation. Recently, Kauppinen compared autoregressive and Fourier descriptors for 2D shape classification and found the latter is better [18]. Sajjanhar et al. compared moment invariants and Fourier descriptors for image retrieval and found their performance is not significantly different [19]. Scassellati et al. studied image retrieval by 2D shape representations, including algebraic moments, spline curve distances, cumulative turning angle, sign of curvature and Hausdorff distance [20]. Their study results are not conclusive and performance based on these measures is generally poor judged by human perception.

The above review indicates that there is a need for a better shape representation and similarity measure. This paper proposes an alternative shape representation and similarity measure, and compare its performance with one of the most popular methods, namely the FD-based method.

3 Region-based shape representation and similarity measure

3.1 Definitions of common terms

The following are some important terms associated with shape description which we will use in the following discussion.

Major axis: it is the straight line segment joining the two points on the boundary farthest away from each other.

Minor axis: it is perpendicular to the major axis and of such length that a rectangle with sides parallel to major and minor axes that just encloses the boundary can be formed using the lengths of the major and minor axes.

Basic rectangle: the above rectangle formed with major and minor axes as its two sides is called basic rectangle.

Eccentricity: the ratio of the major to the minor axis is called eccentricity of the boundary.

Centroid or Center of gravity: a single point of an object towards which other objects are gravitationally attracted. For two-dimensional shape, the coordinates (X_c, Y_c) of the centroid are defined as

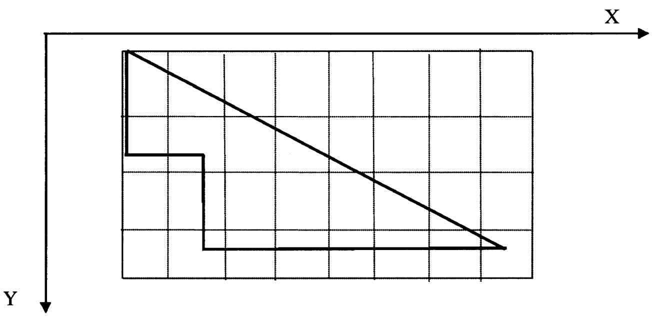


Fig. 1. Generation of binary sequence for a shape

$$X_c = \sum_x \sum_y f(x,y)x / \sum_x \sum_y f(x,y),$$

$$Y_c = \sum_x \sum_y f(x,y)y / \sum_x \sum_y f(x,y),$$

where (x, y) are pixel coordinates and $f(x, y)$ is set to 1 for points within or on the shape and set to 0 elsewhere.

3.2 Basic idea of region-based shape representation

Given a shape, we overlay a grid space over it (see Fig. 1). The grid space, which consists of fixed-size square cells, is just big enough to completely cover the shape. Some grid cells are fully or partially covered by the shape and some are not. We assign a 1 to the cell with at least 15% of pixels covered by the shape, and a 0 to each of the other cells. We then read these 1s and 0s from left to right and top to bottom to obtain a binary sequence for the shape. For example, the shape in Fig. 1 can be represented by a binary sequence 11100000 11111000 01111110 01111111. This binary sequence can be stored as a four-byte integer.

It can be seen that the smaller the cell size, the more accurate the shape representation and the more the storage and computation requirements. A good compromise of the cell size is around 10×10 to 20×20 pixels. Cell sizes of 12×12 and 24×24 pixels are used in our work and retrieval performances based on these two cell sizes are compared.

The above representation is compact, easy to obtain, and translation invariant, but it is not invariant to scale and rotation. Thus before deriving the binary sequence for a shape, we have to do scale and rotation normalization.

3.3 Rotation normalization

The purpose of rotation normalization is to place shapes in a unique common orientation. We decided to rotate the shape so that its major axis is parallel with the x-axis. There are still two possibilities for the shape placement: one of the farthest points can be on the left or on the right. This is caused by 180° rotation. For example, the shape in Fig. 1 can be placed in one of the two orientations as shown in Fig. 2.

Two different binary sequences are needed to represent these two orientations. As the binary sequence is used as

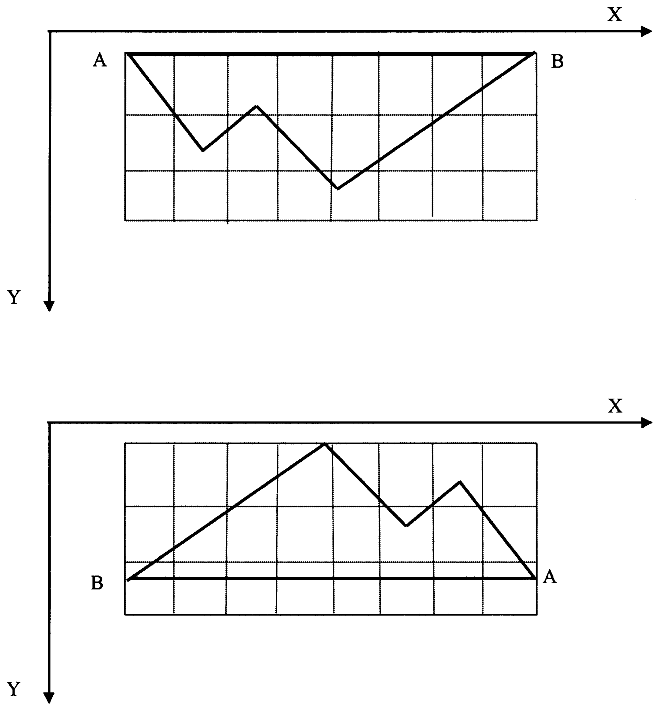


Fig. 2a,b. Two possible orientations with the major axis along the x-direction

index of the shape in our retrieval system, storing two for each shape needs twice the storage space. To save storage space, we obtain and store only one of the binary sequences. Which one to use is not important and is determined by implementation. The two orientations are accounted for during retrieval time by representing the query shape using two binary sequences which are compared to each shape index stored in the database.

3.4 Scale normalization

To achieve scale normalization, we proportionally scale all shapes so that their major axes have the same fixed length. In our study, the fixed length used is 192 pixels.

3.5 Unique shape representation – shape index

After rotation and scale normalization and selection of a grid cell size, we can obtain a unique binary sequence for each shape. This binary sequence is used as the representation or index of the shape. For example, the index of shape in Fig. 1 (normalized into shapes in Fig. 2) is either 111111110111111000011000 or 001111110111111111111 111.

As we use a grid just large enough to cover the normalized shape, when cell size is decided, the number of grid cells in x-direction is fixed. The number of cells in y-direction depends on the eccentricity of the shape, the maximum number being the same as that in x-direction. For example, when grid cell size is 24×24 pixels, the number of cells in x-direction is 8 and the number of cells in y-direction can range from 1 to 8, depending on shape eccentricity.

3.6 Similarity measure

The next issue is how to measure similarity between shapes based on their indexes. As the index indicates the cell positions covered by a shape, it is natural to define the distance between two shapes as the number of cell positions not commonly covered by these two shapes. Note that 180° rotation and other shape operations will be considered later. Based on the shape eccentricities, there are the following three cases for similarity calculation.

– If two normalized shapes have the same basic rectangle, we can bitwise compare the indexes of these two shapes, and the distance between them is equal to the number of positions having different values. For example, if shapes A and B have the same eccentricity of 4 and binary sequences 11111111 11100000 and 1111111111111100, respectively, then the distance between A and B is 3.
– If two normalized shapes have very different basic rectangles, i.e., they have very different minor-axis lengths, there is no need to calculate their similarity, as we can safely assume that these two shapes are very different. For example, if the eccentricities of shapes A and B are 8 and 2, respectively, i.e., the lengths of minor axes are 1 and 4 cells, then we can assume that these two shapes are quite different and there is no value to retrieve the shape. The difference threshold between minor axes depends on applications and cell size. Normally, if the lengths of minor axes of two shapes differ more than 3 cells, these two shapes should be considered quite different.
– If two normalized shapes have slightly different basic rectangles, it is still possible these two shapes are perceptually similar. We add 0s at the end of the index of the shape with shorter minor axis so that the extended index is of the same length as that of the other shape. The distance between these two shapes is calculated as in the first case. For example, if the length of the minor axis and binary sequence of shape A are 2 and 11111111 11110000 and the length of the minor axis and binary sequence of shape B are 3 and 11111111 111111000 11100000, respectively, then we should extend the binary number for shape A to 11111111 11110000 00000000. The distance between A and B is 4.

To facilitate the above similarity calculation during retrieval, shape eccentricity is stored together with the unique binary sequence. They together form the index of a shape.

3.7 Other shape operations

In addition to the 180° rotation of shapes, the other two operations which will result in perceptually similar shapes are horizontal and vertical flips. Figure 3 shows two shapes resulted from these two operations on the shape in Fig. 2a. These two shapes are perceptually similar to the shape in Fig. 1.

To take into account of these two operations and yet to save storage space, we still store one index for each shape, but we generate four binary sequences for each query shape during retrieval. In this case, perceptually similar shapes resulted from 180° rotation and horizontal and vertical flips can be retrieved.

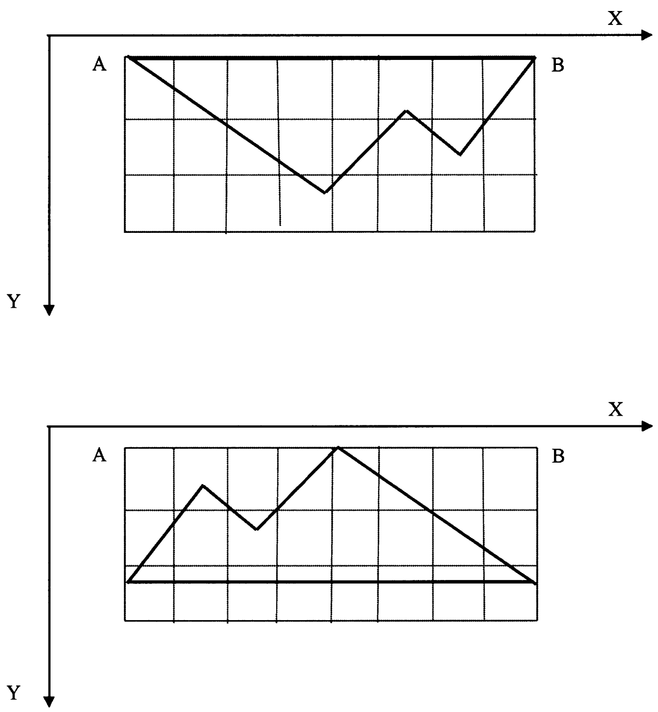


Fig. 3a,b. Horizontal and vertical flips

3.8 Handling multiple major axes

In the above discussion, we assumed that each shape has only one major axis. In practice, a shape may have multiple major axes of equal length. The same shape may result in different binary numbers depending on which major axis is used for rotation normalization.

To solve this problem, rotation normalization is done along each major axis and binary numbers for each normalization are used as shape index. The distance between two shapes is the minimum distance between each pair of binary numbers of these two shapes.

3.9 Summary of index and retrieval processes

In the above, we have described the region-based shape representation which is invariant to translation, scale, rotation and mirror operations, and similarity measure. In this subsection, we summarize the shape-indexing and retrieval process. In a retrieval system, all shapes in the database are indexed. During retrieval, the query shape is also indexed. Then the query index is compared with shape indexes in the database to retrieve similar shapes.

Each shape in the database is processed and indexed as follows (assuming each shape has only one major axis).

1. The major and minor axes and eccentricity of each shape are found.
2. The shape is rotated to place the major axis along the x-direction, and the shape is scaled so that the major axis is of a standard fixed length.
3. A grid space with fixed cell size is overlayed on top of the normalized shape.
4. 1s are assigned to cells covered by the shape and 0s to other cells. By reading these 1s and 0s from left to right and top to bottom, we obtain a binary sequence for the shape.
5. The binary sequence and the length of the minor axis are stored as the index of the shape.

During retrieval, the following steps are used to represent the query shape and carry out similarity comparison.

1. The query shape is represented by its minor-axis length and binary sequences using the same procedure as in the above index process. But note there are four binary sequences for each query to take into account 180° rotation and horizontal and vertical flip operations.
2. For efficiency reason, these four binary sequences are only compared with binary sequences of shapes in the database with the same or similar eccentricities.
3. The distance between the query and a shape in the database is calculated as the number of positions with different values in their binary sequences.
4. The similar shapes are displayed or retrieved in the increasing order of shape distance.

The above-outlined approach is simple and similar to the way we normally compare shapes. To compare two shapes, we prefer that they are of same or similar size (scale normalization). Then we will rotate one of the shapes over the other so that they are in the similar orientation (rotation normalization). Finally, we determine how much they differ, based on how much they do not overlap. The region-based approach incorporates all these steps.

4 Performance study

To study the retrieval performance of the above-described region-based shape representation and similarity measure, we implemented a prototype shape retrieval system. To compare its retrieval performance with the more established FD-based method, we also implemented an FD-based method [18]. In the following, we first briefly describe the implementation of the FD-based method. We then describe the experimental setup, the method of performance measurement, and various experimental results.

4.1 FD-based shape representation and similarity measure

In FD-based method, a shape is first represented by a feature function called shape signature. A discrete Fourier transform is applied to the signature to obtain FDs of the shape. These FDs are used as index of the shape and for shape similarity calculation.

The discrete Fourier transformation of a shape signature $f(k)$ is given by

$$F_u = 1/N \sum_{k=0}^{N-1} f(k) \exp[-j2\pi uk/N]$$

for $u = 0$ to $N - 1$, where N is the number of samples of $f(k)$.

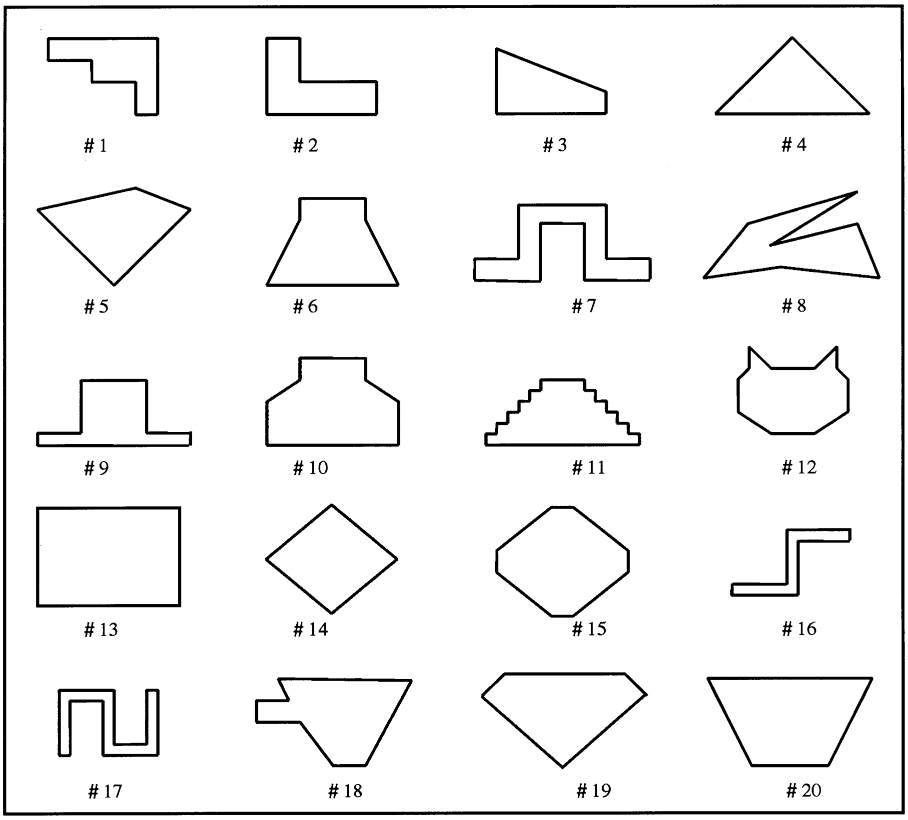


Fig. 4. Twenty sample shapes in the database

There are a number of types of shape signatures. The commonly used signatures are curvature based, radius based, and boundary coordinates based. It was found that shape classification performances based on these three signatures do not differ significantly [18]. In our experiment, we used radius-based signature, as it is simplest to implement.

Radius-based signature consists of a number of ordered distances from the shape centroid to boundary points (called radii). In our experiment, 64 uniformly sampled boundary points and thus 64 ordered radii are used as shape signature. The boundary points are sampled such that the number of pixels along the boundary between each two neighbouring points is the same. Shape radii, thus their transformations, are translation invariant. Note that shapes are not orientation normalized before the shape radii are used. The normalization is achieved by ignoring the phase values of FDs. Shape rotation is reflected in the phase information of F_u and the magnitude of F_u, i.e., $|Fu|$, is invariant to rotation. $|F_0|$ reflects the energy of the shape radii, thus $|F_u|/|F_0|$ will be scale invariant. So we use the following feature vector, which is invariant to translation, rotation and scale, to index the shape:

$$V = [|F_1|/|F_0||F_2|/|F_0|\ldots\ldots|F_N|/|F_0|]^T$$

The distance between shapes is calculated as the Euclidean distance between their feature vectors.

One may wonder why we should use FDs as shape index instead of radii directly. The main reason is that the direct representation is very sensitive to small changes and noise, leading to very poor retrieval performance. If 64 radius lengths are directly used as index, it would be very difficult to do scale and rotation normalization. It appears that we can achieve rotation normalization by identifying the shortest (or longest) radius and achieve scale normalization by fixing the length of the shortest radius. But this normalization is not stable, as small change on the boundary may affect the position of shortest radius and positions of sample points, leading to very different indexes and large distance between shapes due to the small change. The purpose of using FD is to convert the sensitive radius lengths into frequency domain, where the data is more robust to small changes and noise. This is because FDs capture the general feature and trend of the shape instead of each individual detail. We will experimentally compare the performance of using radii as index and FD-based method in Sect. 4.3.

4.2 Experiment setup and performance measurement

Our experimental image database have 160 two-dimensional planar shapes. Figure 4 shows some sample shapes in the database.

Each shape in the database is indexed. Four index files, corresponding to the region-based indexes with cell size of 12×12, and 24×24 pixels, index directly based on radii, and FD-based index, are created. For each query, four types of indexes are also calculated and they are used to compare with indexes in their corresponding index files to obtain shape distances. Note that, for each of the region-based methods, four indexes are obtained. Shapes are retrieved (shape name are returned) in increasing order of distances.

The retrieval performance is measured using recall and precision, as commonly used in information retrieval literature [21, 22]. Recall measures the ability of retrieving all

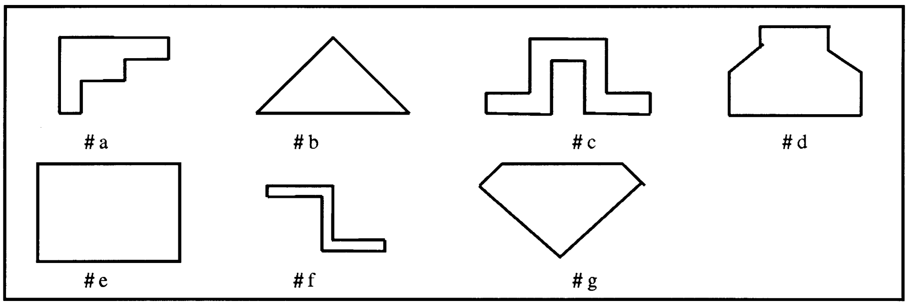


Fig. 5. Seven query shapes

relevant or similar information items in the database. It is defined as the ratio between the number of relevant or perceptually similar items retrieved and the total relevant items in the database. Precision measures the retrieval accuracy and is defined as the ratio between the number of relevant or perceptually similar items retrieved and the total number of items retrieved. For example, if there are 20 relevant items to a query in the database and the system returns 10 items, of which 8 are relevant, then the recall value is equal to 40% and the precision value is equal to 80%. Recall and precision are used together to measure the retrieval effectiveness, as precision varies depending on required recall. So we normally use a precision-recall curve to show a retrieval system's performance by varying the number of items returned.

For each query, the relevant items in the database are the shapes which are perceptually similar to the query. To calculate recall and precision, we have to know relevant items for each query.

Table 1. Distances between the seven queries in Fig. 5 and 20 shapes in Fig. 4

	a	b	c	d	e	f	g
1	0	104	46	76	37	44	65
2	33	65	69	80	38	61	56
3	17	81	33	96	48	37	60
4	104	0	87	76	71	100	37
5	72	62	75	41	53	78	61
6	98	55	104	50	110	103	77
7	46	87	0	86	79	44	60
8	54	88	32	117	89	40	69
9	32	84	34	101	55	52	65
10	76	76	86	0	72	93	76
11	40	76	50	89	47	64	59
12	105	68	121	65	69	90	80
13	37	71	79	72	0	71	66
14	108	55	116	64	74	91	69
15	123	71	132	62	88	137	62
16	44	100	44	93	71	0	63
17	54	92	71	109	49	50	91
18	54	51	52	54	62	69	36
19	65	37	60	76	66	63	0
20	91	38	109	65	81	86	70

4.3 Experimental results

We present four sets of results. The first set shows generally how the proposed method performs as judged by human perception. The second set compares the performance between the method of using radii as index and the FD-based method. The third set result compares the performance between the region-based method (with two different cell sizes) and the FD-based method. The fourth set result compares the performance between the region-based method and the FD-based method when Gaussian noise is added to all shape boundaries.

We summarize the results obtained by issuing seven queries. These seven shapes, shown in Fig. 5, are randomly chosen from the database (this is a query-by-example approach).

4.3.1 General performance

The aim of this experiment is to determine whether the distance measure of our proposed method conforms with human perception. Table 1 shows the distance among seven queries in Fig. 5 (labeled as a to g) and 20 shapes in Fig. 4 (numbered as 1 to 20). In general, the results obtained conform with human perception. Of course, human perception of shape similarity among some shapes is sometimes subjective and application dependent.

4.3.2 Performance comparison between the method of using radii as index and the FD-based method

When using the FD method, it is necessary to obtain radii for shapes and then derive the Fourier coefficients for the shape signature. It is interesting to compare the performance between the method of using radii directly as index and the FD-based method.

For the radii-based method, 64 equispaced points along the perimeter of the shape boundary were sampled. The centroid of the shape boundary was obtained and the centroidal distances of each of the sample points was computed. The centroidal distance at each sample point was scaled by a factor equal to the minimum centroidal distance. The sequence of the scale normalized centroidal distances at the sample points along the shape boundary are used to index the shapes. The distance between a query shape and shapes in the database is obtained as the global minimum of the sum of differences between the centroidal distances.

In the FD-based method, the feature vector is obtained from the above 64 radii before normalization. The results for the seven query shapes are averaged to obtain the graph shown in Fig. 6.

From Fig. 6, it is observed that the FD-based method performs better than the radii-based method. The superior performance of the FD method is explained by the fact that

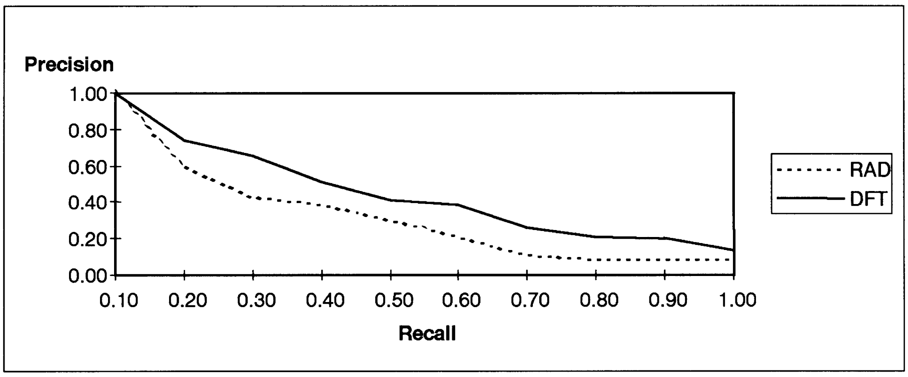


Fig. 6. Values of Recall and Precision for FD-based method (DFT) and radii-based method (RAD) averaged over seven queries

the Fourier coefficients which are used for deriving the feature vectors for the FD method represent the global features of shapes, reflecting relationships and variation patterns of these radii. On the other hand, the radii represent individual sample points along the shape boundary. Consider two shapes which are perceptually similar, one having an extra convex hull on the shape boundary. The two shapes would have a large difference when computing the difference between their radii. This is because the length of the perimeter of the two shapes is different and so the spacing between sample points on the two shape boundaries would be different. Hence, when the global best match method is used to compute the difference, only a few sample points on the shape boundaries would coincide and the others would be progressively further apart from the coinciding points. In FD-based method, there will be changes to FDs due to the minor changes to the boundary, but the change is not dramatic.

4.3.3 Performance comparison between the proposed method and the FD-based method

In this experiment, we compare the retrieval performance between the proposed method and the FD-based method. For the proposed method, two cell sizes (12×12 pixels and 24×24 pixels per cell) are used to determine the effects of different cell size on retrieval performance. Figure 7 shows the retrieval performances of the proposed method and FD-based method averaged over the seven queries. We can make the following observations. First, the smaller the cell size, the better the retrieval performance. This is expected, as the binary sequences based on smaller cell size are more accurate in representing shapes. Second, in general, the region-based approach has better retrieval performance than the FD-based approach. The region-based method (with both cell sizes) has higher precision value at each recall value than FD-based method.

4.3.4 Performance comparison between the proposed method and the FD-based method when Gaussian noise is added to shape boundaries

In order to evaluate the proposed method when noise is introduced on the shape boundaries, we indexed the shapes in the database for each method after adding Gaussian noise

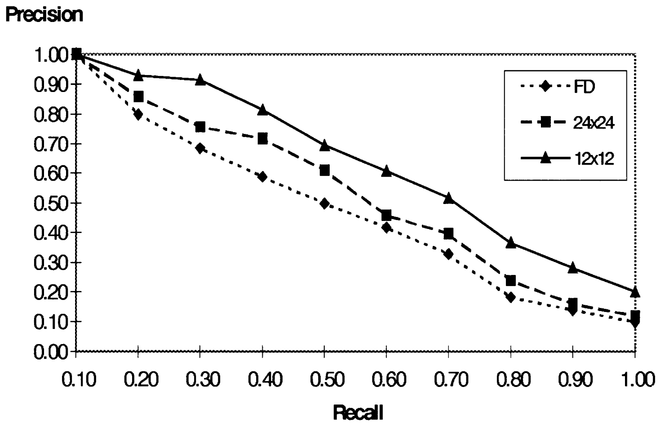


Fig. 7. Retrieval performance averaged over seven queries for FD and region-based method with cell sizes of 24×24 and 12×12 pixels

on the shape boundaries [18]. Noise was also added to the query shapes. The noisy coordinates (x'_i, y'_i) on the shape boundary are given by

$$x'_i = x_i + \mathrm{drc}\ \sin(t_i),$$
$$y'_i = y_i + \mathrm{drc}\ \cos(t_i)$$

where (x_i, y_i) are the original coordinates, d is the distance between the successive boundary points, r is a sample from the Gaussian distribution $N(0, 1)$, c is a parameter between 0.1 and 0.9 which controls the amount of noise (set to 0.5) and t_i is the tangent angle at the boundary point i.

The performance for the region-based (with cell size of 12×12 pixels) and FD-based methods for a noisy database and noisy queries is shown in Fig. 8.

Figure 8 shows that proposed region-based method still performs favorably compared with the FD-based method when noise is added. This result shows that the proposed method is not very sensitive to noise and perturbations on the shape boundary.

4.4 Comparison of computation cost

We have compared retrieval effectiveness based on recall and precision between the region-based and FD-based approaches. We now compare their efficiency in terms of storage requirement for indexes and computation requirement during retrieval.

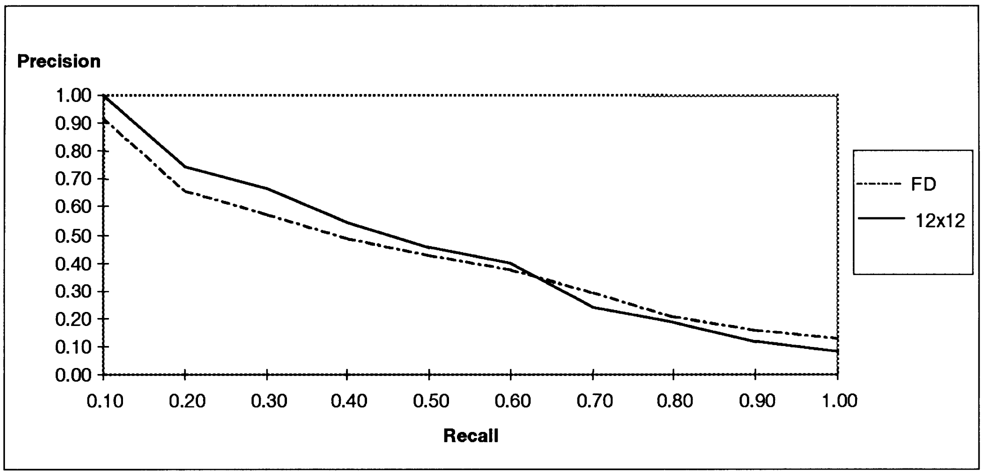


Fig. 8. Retrieval performance of the Fourier descriptors method (FD) and the proposed method (12×12) when noise is added

Table 2. Computation cost of indexing a shape for the region-base method, where N is the number of coordinates used (typically around 20) to obtain the polygonal approximation of the shape boundary and P is the number of pixels (192×192) of the grid used

Operation	Computation cost
Finding major axis	$O(N^2)$
Finding minor axis	O(N)
Rotating and scaling	O(N)
Deriving binary number	O(P)

Table 3. Computation cost of indexing a shape for the FD-based method, where R is the number of radii (normally 64) used

Operation	Computation cost
Finding centroid	O(R)
Finding all radii	O(R)
Finding all Fds	$O(R^2)$
Deriving feature vector	O(R)

For the region-based approach with cell size of 12×12 pixels, a maximum of 33 bytes (32 bytes for binary sequence and 1 byte for minor axis) is required for storing the index of each shape. With a cell size of 24×24, a maximum of 9 bytes (8 bytes for binary sequence and 1 byte for minor axis) is required for storing the index of each shape. In the FD-based approach, 64 real numbers must be stored as index (feature vector) for each shape if all 64 Fourier components are used. This number can be reduced to 16 or 32, with some performance degradation [18]. Since one real number needs at least 4 bytes, the FD-based approach needs much more storage space than the region-based approach.

During retrieval, both FD-based and region-based methods need to first derive the index for the query and then compare similarity between the query and shapes in the database. For the region-based method, the major operations and required computation cost for indexing each shape are summarized in Table 2.

For the FD-based method, the major operations and required computation cost for indexing each shape are summarized in Table 3.

Comparing Table 2 and Table 3 and considering that complex numbers are involved in FD-based method, the computation costs in indexing for the region-based and FD-based methods are not significantly different. In addition, we should note that only query indexing is done on-line, indexing of shapes in the database is done off-line.

To compare similarity between the query and shapes in the database, the region-based approach uses two steps. The first is to identify the shapes with similar eccentricities to the query and the second is to calculate the similarity or distance between the query and these shapes. Because of the first step, the number of similarity comparisons is reduced significantly. For each similarity calculation, all that is required is to carry out an exclusive OR between the query binary sequence and the stored-shape binary sequence and count the number of ones in the result.

In the FD-based approach, similarity must be calculated between the query and feature vectors of all stored shapes. For each similarity calculation, 64 real-number subtractions and 63 real-number additions are required. So the FD-based method requires slightly more computation power for similarity calculation than the region-based method.

Therefore, we can conclude that the region-based method has a similar computation cost as the FD-based method, but has lower storage requirements than the FD-based method.

5 Discussion

We have compared retrieval performance (including effectiveness and efficiency) between the region-based and FD-based methods. In this section, we look at the relationships between the region-based method and other closely related shape representations and similarity measure, and comment on which applications the proposed method is most suited for.

The region-based approach originated from the work to normalize chain code representation [23]. Using a similar normalization procedure as described in this paper, normalized chain code which is invariant to shape scale, translation and rotation can be obtained.

A closely related work is reported by Jagadish [17]. In his method, shapes are decomposed into a number of variable-size rectangles, and two pairs of coordinates for each rectangle are used as the representation of the shape. The major differences between his method and our method are as follows. First, our representation is invariant to shape scale, translation, rotation and mirror operations, whereas his method is not invariant to rotation and mirror operations. Second, we decompose a shape into a number of fixed-size squares (cells), whereas variable-size rectangles are used in Jagadish's work. Thus, it is difficult to do decomposition, and more data are required for representing these rectangles for most shapes in his technique. Third, it is easier to compute shape similarity in our approach.

Compared to methods based on curvature, significant edges and points [15, 16], our method has the following advantages. First, our normalization is more natural and accurate. Second, it is easier to calculate shape similarity in our approach.

Our proposed rotation normalization can be applied to other shape representation methods proposed by Jain and Vailaya [15], Mehrotra and Gary [17] and Jagadish [17] to reduce the computation cost during retrieval. For example, if the rotation normlization is applied to the method proposed by Mehrotra and Gary, the significant points coordinates may not need to be rotated during the similarity comparison.

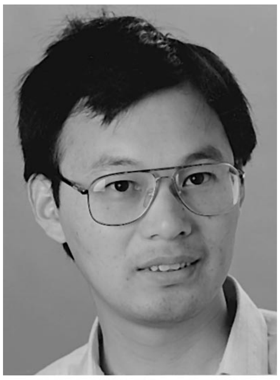
The reason why the region-based method performs well is that the binary representation is quite stable: minor changes or noise on the boundary will change very few bits in the binary number, provided that the major axis does not change. It initially appears that the retrieval performance of the region-based approach is sensitive to noise on shape boundaries, because noise may change the position of the major axis and thus change the binary sequence dramatically. But it may not be a serious problem, considering the following. It is likely that the chance that noise affects the major axis is very low. If the noise does not affect the major axis, our method may actually perform better than other methods. This is because slight noise may not change the binary sequence much, but may change the boundary properties such as curvature, significant edge and points. So overall, the retrieval performance of the proposed approach under noise may be still comparable or better than other methods. Our experimental results confirmed this observation.

The performance of the region-based method relies on the stability of the major axis. It is thus better suited for non-occluded shapes. For occluded shapes, it may work if the major axes remain unchanged. This is equivalent to the case where noise is added to the boundary. We have shown that the region-based method still performs well when Gaussian noise is added to shape boundaries. However, when the major axis changes due to occlusion or other operations, the region-based method will not perform well.

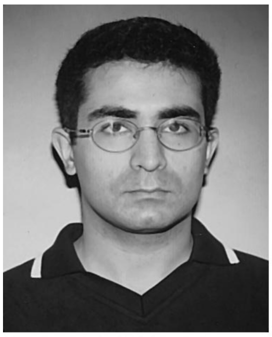
In conclusion, the proposed region-based method compares favorably with FD-based method in both retrieval effectiveness and efficiency. It is thus envisaged that this method can be integrated with other content-based image retrieval techniques to improve retrieval performance.

References

1. Jain R (1993) NSF Workshop on Visual Information Management Systems: Workshop Report. Proc SPIE 1908 (Proceedings of the Conference on Storage and Retrieval for image and Video Databases): 198–218
2. Jain R, Pentland AP, Petkovic D (1995) Workshop report: NSF-ARPA Workshop on Visual Information Management Systems. URL: http://vision.ucsd.edu/papers/VimsReport95. html
3. Niblack W (1993) Proceedings of the Conference on Storage and Retrieval for Image and Video Databases, San Jose, Calif., 2–3 February. Proc SPIE 1908
4. Niblack W, Ramesh J (1994) Proceedings of the Conference on Storage and Retrieval for Image and Video Databases II, San Jose, Calif., 7–8 February. Proc SPIE 2185
5. Niblack W, Ramesh J. (1995) Proceedings of the Conference on Storage and Retrieval for Image and Video Databases III, San Jose, Calif., 9–10 February. Proc SPIE 2420
6. Sethi IK, Ramesh J (1996) Proceedings of the Conference on Storage and Retrieval for Image and Video Databases IV, San Jose, Calif., 1–2 February. Proc SPIE 2670
7. Proceedings of the International Conference on Visual Information Systems, 5–6 February 1996, Melbourne, Australia
8. Gonzalez RC, Wintz P (1987) Digital Image Processing, 2nd edition. Addison-Wesley, Reading, Mass.
9. Pitas I (1993) Digital Image Processing Algorithms. Prentice Hall, Englewood Cliffs, N.J.
10. Niblack W, et al. (1993) QBIC Project: querying images by content using color, texture and shape. Proc SPIE 1908 (Conference of Storage and Retrieval for Image and Video Databases, San Jose, Calif., 2–3 February): 173–187
11. Mohamad D, Sulong G, Ipson SS (1995) Trademark matching using invariant moments. In: Li S, et al. (eds) Second Asian Conference on Computer Vision, 5–8 December, Singapore. Springer, Berlin, pp I-439–I-444
12. Cortelazzo G, et al. (1994) Trademark shapes description by string-matching techniques. Pattern Recognition 27(8): 1005–1018
13. Wu JK, et al. (1994) STAR – A multimedia database system for trademark registration. In: Litwin W, Risch T (eds) Proceedings of First International Conference, ADB, 21–23 June 1994, Vadstena, Sweden. Springer, Berlin, pp 109–122
14. Lam CP, Wu JK, Mehtre B (1995) STAR – a system for trademark archival and retrieval. In: Li S, et al. (eds) Second Asian Conference on Computer Vision, 5–8 December, Singapore. Springer, Berlin, pp III-214–III-217
15. Jain AK, Vailaya A (1995) Image retrieval using color and shape. In: Li S, et al. (eds) Second Asian Conference on Computer Vision, 5–8 December, Singapore. Springer, Berlin, pp II-529–II-533
16. Mehrotra R, Gary JE (1995) Similar-shape retrieval in shape data management. Computer 28(9): 57–62
17. Jagadish HV (1991) A retrieval technique for similar shapes. Proceedings of ACM SIGMOD, May 1991, Colorado. ACM, New York, pp 208–217
18. Kauppinen H, Seppanen T, Pietikainen M (1995) An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. IEEE Trans Pattern Anal Mach Intell 17(2): 201–207
19. Sajjanhar A, et al. (1996) An experimental study of moment invariants and Fourier descriptor for shape based image retrieval. Technical report. Gippsland School of Computing and Information Technology, Monash University, Australia; copies of the report can be obtained from Atul Sajjanhar on atuls@fcit.monash.edu.au
20. Scassellati B, et al. (1994) Retrieving images by 2D shape: a comparison of computation methods with human perceptual judgements. Proc SPIE (Storage and Retrieval for image and Video Databases II, 7–8 Feb. 1994, San Jose, Calif.) 2185: 2–14
21. Frakes WB, Baeza-Yates R (1992) Information Retrieval: Data Structures and Algorithms. Prentice Hall, Englewood Cliffs, N.J.
22. Salton G, McGill MJ (1983) Introduction to Modern Information Retrieval. McGraw-Hill, New York
23. Lu G (1996) Image retrieval based on shape. Proceedings of the International Conference on Visual Information Systems, 5–6 February 1996, Melbourne, Australia. Victoria University of Technology, Melbourne, Australia, pp 126–135



Dr. GUOJUN LU is currently a senior lecturer at Gippsland School of Computing and Informatin Technology, Monash University. He has held positions at Loughborough University, National University of Singapore, and Deakin University, after he obtained his PhD in 1990 from Loughborough University and BEng in 1984 from Nanjing Institute of Technology. Dr. Lu's main research interests are in multimedia communications and multimedia information indexing and retrieval. He has published over 50 technical papers in these areas and recently authored the book Communication and Computing for Distributed Multimedia Systems (Artech House 1996).



ATUL SAJJANHAR obtained Bachelor of Engineering in Electronics and Instrumentation from Indore University, India, in 1994. He obtained a Master of Computing by Research from Monash Uniersity, Australia in 1997 for his thesis titled "A technique for similarity retrieval of shapes". He is currently associated with Kent Ridge Digital Labs, Singapore (formerly, Information Technology Institute, National Computer Board of Singapore) and is working on information retrieval.