University of Alberta

Shape Representation and Retrieval Using Distance Histograms

by

Shuang Fan

Technical Report TR 01-14
October 2001

DEPARTMENT OF COMPUTING SCIENCE
University of Alberta
Edmonton, Alberta, Canada

Shape Representation and Retrieval Using Distance Histograms

Shuang Fan

October 17, 2001

Abstract

Among all the issues related to Content Based Image Retrieval systems, retrieving images based on their shapes is an important one. Many approaches exist utilizing shape representation and comparison, e.g., the methods based on Fourier descriptors. In this thesis, we propose a novel method for shape representation. In our method, we calculate the centroid of a shape and choose a set of sample points around this shape's boundary. From those, we obtain a set of radii. We then use these radii lengths to construct a Distance Histogram as the representation of shape. The natural characteristic of our method makes itself invariant to rotation and translation. Furthermore, it can be made invariant to scale by a simple normalization. To evaluate our approach, we perform a large set of experiments on a database of shapes. Using either a database of synthetic shapes or a database of real shapes, we compare our method to that based on Fourier descriptors, which is a well-known and effective approach. The results of the experiments show that our method is an effective, economical, and flexible approach for shape representation.

1 Introduction

Image retrieval - how to retrieve images similar to a query - is becoming a more important problem with the rapid increase of media information on the web. Generally, users want to provide query images and use some kind of system which can present them with a set of similar images. Therefore, how to describe and model an image, how to compare different images and judge whether they are similar, how to construct an index of image databases, and how to conduct searches efficiently are some of the key problems in any so-called image retrieval system.

An image can be described by several low level features. These features, which include shape, color, texture, and spatial relationship are called the *content of image*. By using these features, we can not only describe and model an image, but also make comparisons among different images. Therefore, an image retrieval system which is based on these low level features is called content-based image retrieval (CBIR) system.

In this chapter we discuss these systems, in the context of the motivation they provided for our research.

1.1 Low level features of image

Low level features of an image denote several of its perceived characteristics, including shape, color, texture, and spatial relationship among different objects. These features are very important for describing and representing an image, and nowadays, many of them are used by CBIR systems. For example, IBM's QBIC (Query By Image Content) system [Qbi01] [FFN+93] [NBE93] [SC95] supports color and shape, as well as texture similarity measures.

Among all features, color is probably the most straightforward one which enables human to recognize different images. As a result, color similarity measure becomes a very important aspect in the implementation of the CBIR system. The color Histogram is a well known technique used to represent the color of an image [Qbi01] [NBE93] [SC95].

The shape of objects contained in an image is another important feature of the content of the image. In many situations, people can recognize an object only by its shape. The shape of an object can be obtained by tracking its boundary. Generally, there are two types of approaches which are used in shape analysis: one is based on the entire shape region, e.g., Moment Invariants [Hu62] [YA94]; the other is based on the boundary of objects, e.g., Fourier descriptors [KSP95] [PF77] [RSH96] [RM99]. Figure 1.1 illustrates some examples of shapes.

Figure 1: Three shapes of marine creatures

Texture is also an important feature in pattern recognition and image retrieval. It refers to the visual pattern of regions of an image. There are several properties of texture, which include contrast, coarseness, directionality, regularity, periodicity, and randomness. The texture of an image can be represented by Wave-let transform [MM96] [HMK00] - a technique which is always used in image processing. Just like Fourier transform, it can transform the signal from time domain into frequency domain. In addition to this, it has a coherence time proportional to the period. Figure 1.2 illustrates a few examples of texture.

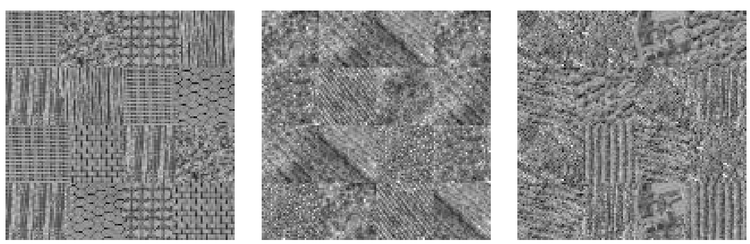


Figure 2: Three samples of texture

Spatial relationship represents some kind of relationship between different objects in an image. It, too, is an important feature of images. Generally there are two classes of representations for spatial relationships: one based on object and the other based on relation [Exp97] [NHS84] [CJ90]. An example is illustrated in Figure 1.3.

A large amount of research work on low level features of images has been done in past years. As a result, many techniques have been developed based on these features. However, there is no best technique; all have their own strengths and weaknesses. For example, in some situations a retrieval technique based on color may provide a good answer; it may not, however, get the expected answer in some other situations. Therefore, combining techniques based on different low level features is found to be helpful for efficient and accurate retrieval. In fact, most current CBIR systems support using more than one feature for similarity measuring.

1.2 CBIR system

It would appear that Kato was the first person to use the term *content-based image retrieval* [EG99]. He used it to describe his experiments about automatic retrieval of images from a large image database, by color and shape feature. Since then, the term has been widely used to represent the process of retrieving expected images from a

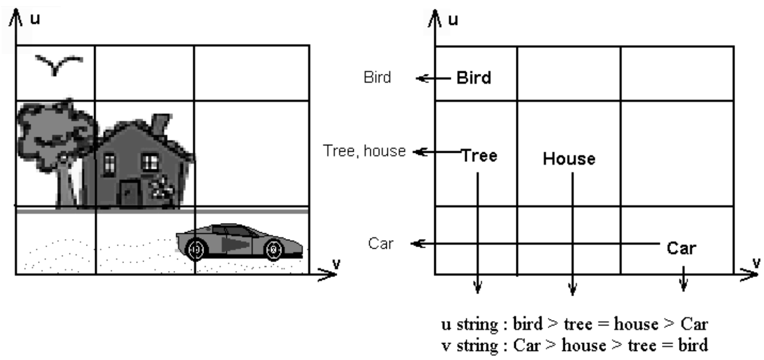


Figure 3: A sample of spatial relationship [Wa01]

large collection of images on the basis of low level features which can be automatically extracted from the images themselves. Nowadays, a large amount of related work is conducted on CBIR systems. There are several basic issues to consider with CBIR systems [EG99]:

- understanding image users' needs and information-seeking behavior
- identifying suitable ways of describing image content
- extracting features from raw images
- providing compact storage for large image databases
- matching query and stored images in a way that reflects human similarity judgments
- efficiently accessing stored images by content
- providing usable human interfaces to CBIR systems

In this thesis, we are concerned only with the following: how to extract the shape from raw images, how to describe the shape, how to provide compact storage for shapes of images, and how to compare query and stored images by their shapes.

There exist a lot of CBIR systems. For example:

- IBM's QBIC system [Qbi01] [FFN+93] [NBE93] [SC95] was developed by IBM's *Visual Media Management research group*. It is the earliest commercial CBIR system. Currently, the QBIC system supports several primitive image similarity measures, such as average color, color histogram, and texture. The QBIC system's technologies include two major parts - indexing and searching. Furthermore, the QBIC system provides several query approaches, including simple feature, multi-features, and multi-pass.

- VisualSEEK was created at the Image and ATV lab of Columbia University [VSE01] [SC96]. This system allows the user to pose queries using low level features of an image, such as color, spatial layout, and texture. These features are represented by using Color Set and Wave-let Transform based texture [Wa01] [SC95].

- Photobook [Pho01] [PPS94] was developed at MIT. It is a tool for performing queries on image databases, based on image content. Basically, just like the other CBIR systems, it compares the features of images, not the images themselves. These features include color, shape, and texture. The largest difference between

Photobook and other CBIR systems (e.g., QBIC) is that the Photobook system includes an interactive learning agent, called FourEyes [MP96]. This agent selects and combines models based on instances from users.

- SQUID [SQU01] is the first CBIR system on the Internet which allows users to submit shapes as query.

1.3 Content summary of thesis

In this thesis we will only discuss techniques related to shape-based image retrieval. After reviewing some existing approaches for shape representation, we present a novel approach for shape representation and comparison, which we call the method based on Distance Histograms. Furthermore, we organize a set of experiments to compare our approach with the method based on Fourier descriptors. Based on the results of these experiments, we then analyze and discuss the advantages and disadvantages of our method for shape representation. Finally, we draw some conclusions about our method.

The thesis is divided into five chapters, including this one. In Chapter 2, we review several existing approaches for shape representation and comparison. Of these we pay more attention to the method based on Fourier descriptors, since we will compare our approach to it. In Chapter 3, we present our approach for shape representation. In Chapter 4, we design and perform a set of experiments to compare our approach with a method based on Fourier descriptors. We first apply our method, and a method based on Fourier descriptors, to the retrieval of simple synthetic shapes - polygons. After that we apply both methods using shapes of real images. Finally, we present the advantages and limitations of our method. In the last chapter, we summarize the work described in the thesis and present our conclusions. We then discuss the future work that can be conducted following our contribution.

2 Related work

As stated earlier, there has been a rapidly increasing interest in image retrieval systems. The main purpose of such systems is to retrieve a set of answers from an image database which satisfies some specified criteria, e.g., visual similarity with a query.

Generally, low level features, e.g., color, shape, and texture are used for representation and comparison between images. This was discussed in the previous chapter.

One of the many varieties of techniques used by CBIR systems, is the shape-based technique which allows users to retrieve images similar, in shape, to a query image. Therefore, how to describe and represent shape is an important issue in designing and implementing these types of systems. In this chapter, we will review some approaches which we consider to be typical for shape representation in CBIR.

2.1 Fourier descriptors-based approach

In 1977, Persoon and Fu [PF77] first proposed the technique of using Fourier descriptor as the representation of shapes. A large amount of research had been done following their idea.

Rafiei and Mendelzon [RM99] proposed their method for shape representation using Fourier descriptors to describe the shape of an object.

If we consider a shape in the complex plane, we can obtain a one-dimensional complex function of time, b_t, by tracing the boundary. For example, in Figure 2.1, a point will move along the boundary and we can get a complex function $b_t = x_t + jy_t, t = 0, ..., N-1$. Here, the x axis of the figure is treated as the real axis, and the

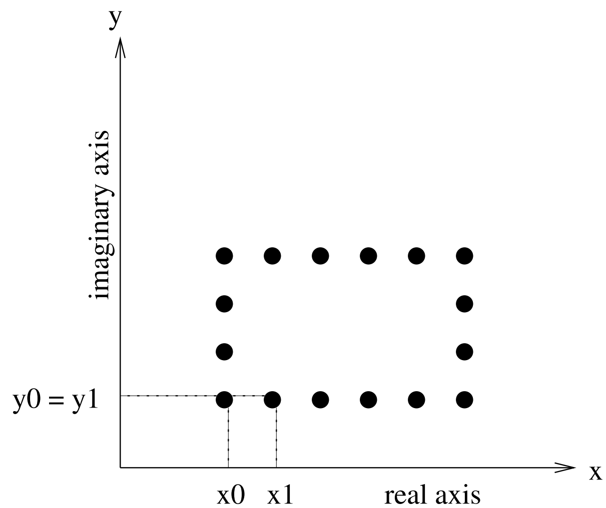


Figure 4: A boundary and its representation as a complex sequence

y axis is treated as the imaginary axis. Now the Fourier descriptors [RM98] for b_t can be computed by:

$$B_f = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} b_t e^{\frac{-j2\pi tf}{N}} f = 0, 1, ..., N-1.$$

They are used to describe the shape of the object in the frequency domain. Based on Parseval's theorem [You99] it is known that the energy in the frequency domain is the same as the energy in the spatial domain. Therefore, this transformation is lossless.

Given two shapes with boundaries described by functions $b_{1t} = x_{1t} + jy_{1t}$ and $b_{2t} = x_{2t} + jy_{2t}, t = 0, 1, ..., N-1$. Then the similarity between these two shapes can be measured by Euclidean distance:

$$D(\vec{b_1}, \vec{b_2}) = \sqrt{\sum_{t=0}^{N-1} (b_{1t} - b_{2t})^2}$$

Sometimes the two boundaries have different numbers of sample points. In this situation, we can simply calculate the Fourier descriptors $\vec{B}1$ and $\vec{B}2$ of $\vec{b}1$ and $\vec{b}2$, respectively, using a fixed number of lower frequency Fourier descriptors to obtain the Euclidean distance:

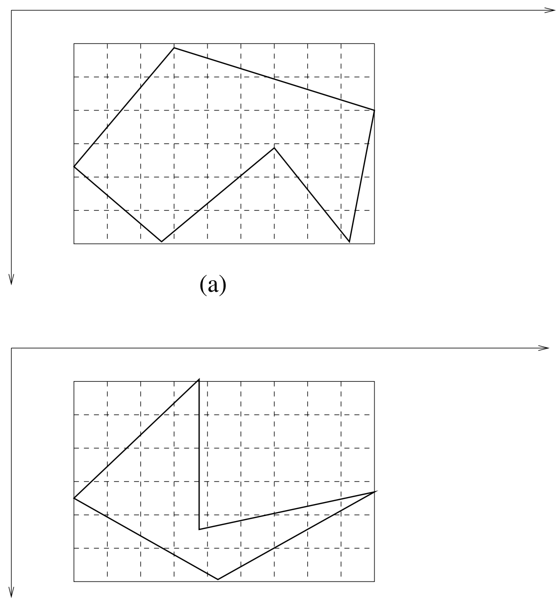
$$D(\vec{B_1}, \vec{B_2}) = \sqrt{\sum_{f=0}^{N-1} (B_{1f} - B_{2f})^2}$$

Of all the Fourier descriptors, we need to pay more attention to B_0 and B_1. Basically B_0 is the only descriptor which represents the information, location of shape. Therefore if we ignore B_0 or, say, set its value to 0, this representation will be invariant to translation. B_1 is another special descriptor. In this representation, $|B_1|$ is the largest amplitude in all of $|B_f|$; dividing every coefficient B_f by $|B_1|$ will make it invariant to scaling. However, this representation is still sensitive to the rotation. To solve this problem, we can rewrite the previous equation:

$$D(\vec{B_1}, \vec{B_2}) = \sqrt{\sum_{f=0}^{N-1} (B_{1f} - e^{j\theta} B_{2f})^2}$$

Here θ is the rotation angle. If we know the range of rotation angle θ, the minimum value of the equation when θ changes from its range will be the real difference between two shapes.

2.2 Grid based-method



(a)

(b)

Figure 5: Mapping a shape onto a grid to generate an index

Sajjanhar and Lu [SL97] propose another method which can be used for shape representation and similarity measure. It is called the grid-based method.

Let us use an example to illustrate this method. In Figure 2.2, the shape is mapped onto a grid of fixed cell size, and is justified to the top left corner. Then, we scan this grid from left to right, and also from top to bottom. We assign 1 to the cells of the grid which are partially or wholly covered by the shape, and 0 to the cells which are not covered by the shape, obtaining a sequence of 0's or 1's. This sequence can be used for shape representation. For instance, the binary numbers we get from Figure 2.2(a) are 001111000 011111111 111111111 111111111 111110111 011100000011. The binary numbers we get from Figure 2.2(b) are 001100000 011100000 111100000 111101111 111111110 001111000. Naturally, the difference between two different shapes can be computed as the number of cells in the grids which are covered by one shape, but not by the other. This number of cells can be represented by the sum of 1's in the result of the exclusive-or of the two sequence numbers. In our example, the difference is 21. Hence, we can represent any of the shapes and measure the similarity between them.

Obviously, this representation is invariant to translation but sensitive to rotation and scaling. To solve this problem, the term "major axis" of a shape is defined as follows: *The major axis of a shape is the straight line segment joining the two points on the shape boundary farthest away from each other* [SL97]. An example is given in Figure 2.3.

To make the representation invariant to rotation, every shape needs a kind of normalization, i.e, first obtain the major axis, then rotate the shape to make its major axis parallel with a specified line, e.g., the x-axis.

To make the representation invariant to scale, a fixed length of major axis is defined, which can be called

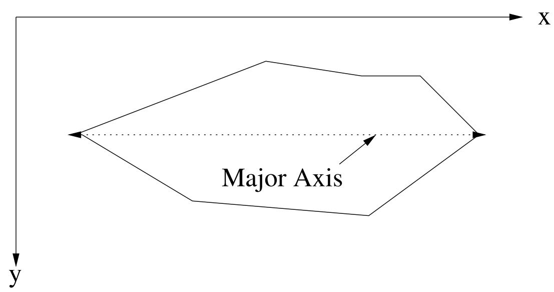


Figure 6: The major axis of a shape

"standard major axis", and every shape must be scaled in a manner such that its major axis equals the standard one.

2.3 Turning angle

Arkin and colleagues proposed an efficiently computable method to represent the polygonal shapes [ACH+91], called the *turning angle method*. Generally, a method of representing a simple polygon A is to describe its boundary by giving a list of vertices, where each vertex can be expressed by its x and y coordinate. However, Arkin presents an alternative representation of the boundary of a polygon. In his method, the boundary can be described by the *turning function* $\theta_A(s)$. This function $\theta_A(s)$ measures the angle of the counter-clock-wise tangent as a function of the arc-length s, which is measured from a given reference start point O on the polygon's boundary. Now $\theta_A(O)$ is the angle v which the tangent at the reference point O makes with the reference orientation, e.g., the x-axis. In this way, the turning function $\theta_A(s)$ will keep tracking around the boundary, and will increase with left-hand turns and decrease with right-hand turns.

Without loss of generality, Arkin assumes that all polygons are rescaled or normalized so that their perimeter length is 1. Then $\theta_A(s)$ becomes a function defined on the interval [0,1]. Figure 2.4 illustrates an example of a turning angle representation of a polygon. In Figure 2.4(a), O is the starting point and its corresponding turning angle is $v1$. All turning angles showed in Figure 2.4(b) can be calculated by tracking counter-clock-wise the boundary of the polygon. Note that for a convex polygon A, its turning function $\theta_A(s)$ will increase monotonically from v to $v + 2\pi$.

Given two polygons, A and B whose $turning\ functions$ are $\theta_A(s)$ and $\theta_B(s)$, the difference between them can be defined as:

$$D(A,B) = min_{r,\theta}(\sqrt{\sum_i (\theta_A(i) - \theta_B(i))^2})$$

where $min_{r,\theta}$ represents the minimum value of all possible shifting and rotation of polygon B.

This type of representation of shape is invariant to translation and scale. Furthermore, to make it invariant to rotation is straightforward. For example, $\theta_A(s)$ is the turning angle representation of polygon A; $\theta'_A(s)$ is the representation of this polygon after rotation by an angle α. Then $\theta'_A(s)$ is only a simple vertical shift α in y-axis of $\theta_A(s)$.

However, this representation is still sensitive to small variations of shape, despite the advantages discussed above. This is illustrated in Figure 2.5.

In Figure 2.5, polygon Q is obtained by only a small variation (noise) to polygon P. However, when we use their turning angle representation to make the comparison, the difference represented by the shadow part in the graph may be not small enough to flag the shapes as very similar.

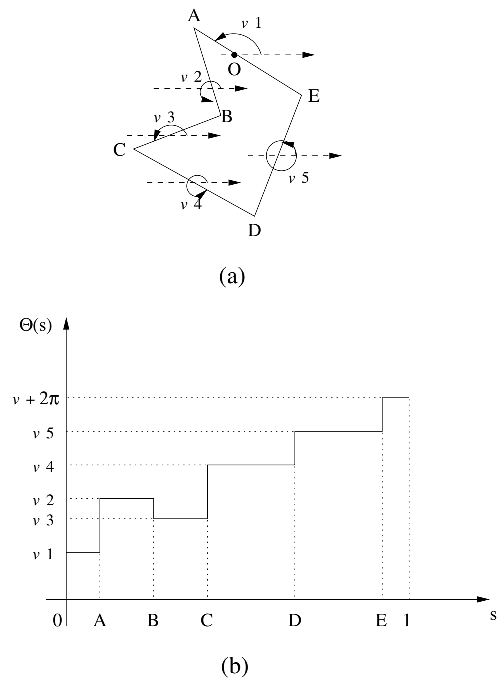


Figure 7: Turning angle representation

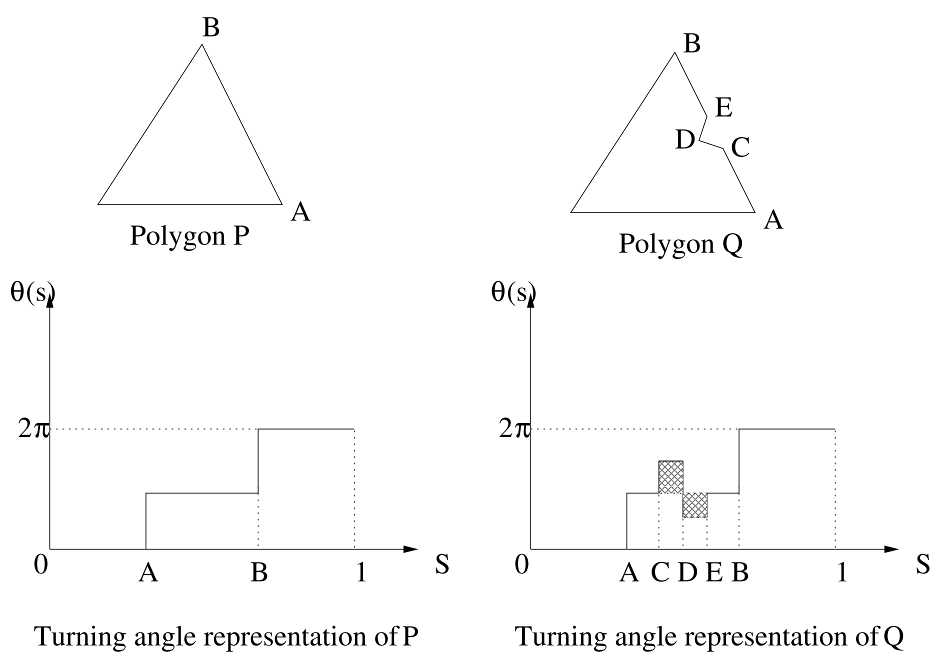


Figure 8: Turning angle representation under small variation

2.4 Rectangular covers

In Jagadish's paper [Jag91], another approach for representation of shapes, called rectangular covers, is proposed. This method restricts itself to rectilinear shapes in two dimensions, i.e., a polygon, not necessarily convex, all of whose angles are right angle and whose edges are all vertical and/or horizontal. This kind of restriction to rectilinear shapes is not too limiting because any general shape can be approximated by a rectilinear staircase.

Basically, there are two types of rectangular cover which are illustrated in Figure 2.6:

- *General rectangular covers*: the given rectilinear shape can be obtained with both addition and subtraction of rectangles.
- *Additive rectangular covers*: given a rectilinear shape, it can be represented by the union of several rectangles.

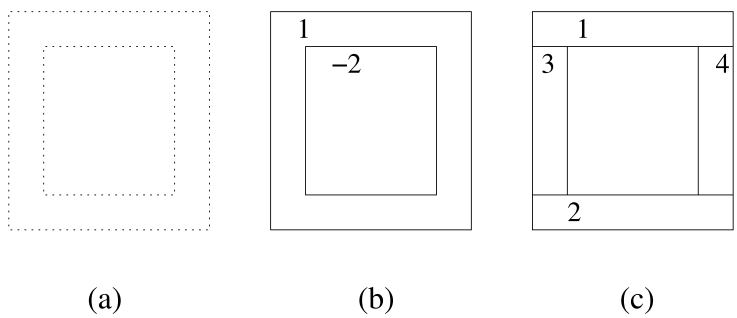


Figure 9: (a) An annular shape; (b) A general rectangular cover for (a); (c) An additive rectangular cover for (a)

There are no additive nor general rectangular covers for a shape which are unique. In Figure 2.7 we can see some different representations of an L-shaped object by using additive rectangular covers. It is easy to say that Figure 2.7(a) is not a good cover because it uses too many unnecessary rectangles to build the representation. However, there are still many choices, even if we place the restriction that exactly two rectangles can be used in representing this L-shape, as in Figure 2.7(b)-(e). Therefore, Jagadish proposed a "not larger than necessary" rule here. This means any rectangle in a rectangular cover that is "larger than necessary" is not permitted. As a result, Figure 2.7(d) and (e) are not permitted any more. However, we still have two choices left - Figure 2.7(b) and Figure 2.7(c). In this situation, Jagadish defines that the one to be used depends on some other requirements that may determine the order in which the two arms of L are to be added. For example, if the vertical arm is added first, then we will select Figure 2.7(c).

Jagadish did not give more details in his paper about whether his approach is invariant under translation, rotation, and scale.

2.5 Partition token

Berretti, Bimbo and Pala proposed an interesting representation for generic shapes [BBP00], using local features. Each shape is partitioned into several tokens corresponding to a set of perceptually salient attributes.

For example, in Figure 2.8, a shape of a horse is partitioned into 10 tokens. The information about each token can be represented by its orientation in the 2-D space. For example, given a token t_k, its orientation θ_k is defined as the orientation (in polar coordinates) of the vector which is linking the median point of the segment $p_k - p_{k+1}$, calculated with respect to an absolute reference system. Figure 2.9 illustrates the idea.

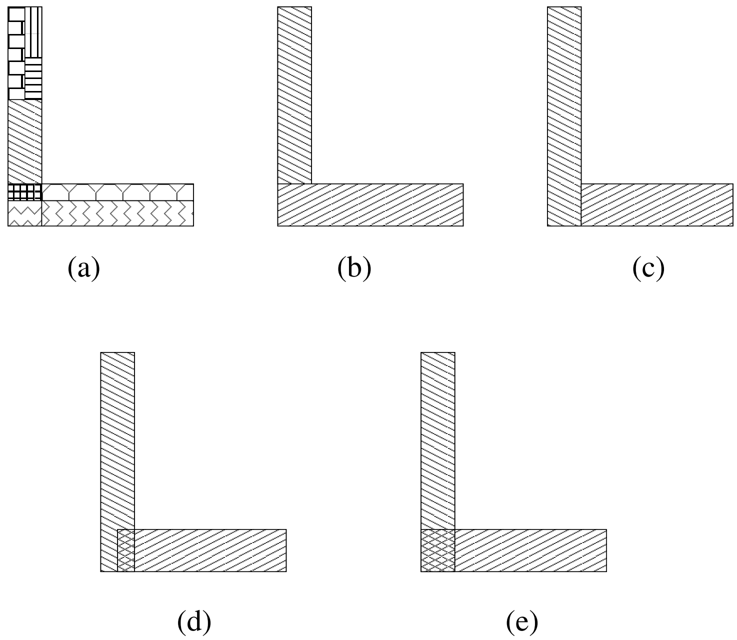


Figure 10: Some potential additive rectangular covers for an L shape

According to this, a token can be represented by two features, which are (m_k, θ_k), with m_k and θ_k in [0,360). m_k is the maximum value of token curvature which is used to represent the token width (token width is the feature that significantly affects the perception of a shape token [BBP00]). In this way, a generic closed curve, or, say, a shape can be represented by a set of tokens:

$$T(c) = \{(m_k, \theta_k)\}_{k=1}^{N}$$

2.6 Centroid-Radii Model

Tan and Thiang proposed a novel method for shape representation in their paper [TT01]. The basic idea of their proposal is to use the centroid-radii model to represent shapes. Figure 2.10 illustrates an example.

In this method, lengths of the shape's radii from centroid to boundary are used to represent the shape. The interval between radii, measured in degrees θ, is fixed. Therefore, the number of intervals can be calculated by $k = \lceil 360/\theta \rceil$. Without loss of generality, the intervals are taken counter-clockwise, starting from the x-axis direction. The shape can then be represented by the following vector:

$$(L_0, L_\theta, L_{2\theta}, ..., L_{(k-1)\theta})$$

where $L_{i\theta}$, $0 \leq i \leq (k-1)$, is the $(i+1)$th radius from the centroid to the boundary of shape. To make this approach invariant under scale, all radii lengths can be normalized by dividing with the longest radius among them:

$$(l_0, l_\theta, l_{2\theta}, ..., l_{(k-1)\theta})$$

In this way, the shape can be represented in detail with sufficient radii used.

In Tan and Thiang's method, two shapes are considered similar to each other if and only if the lengths of their radii at the respective angles differ by a trivial value [TT01]. For example, assuming two shapes are represented

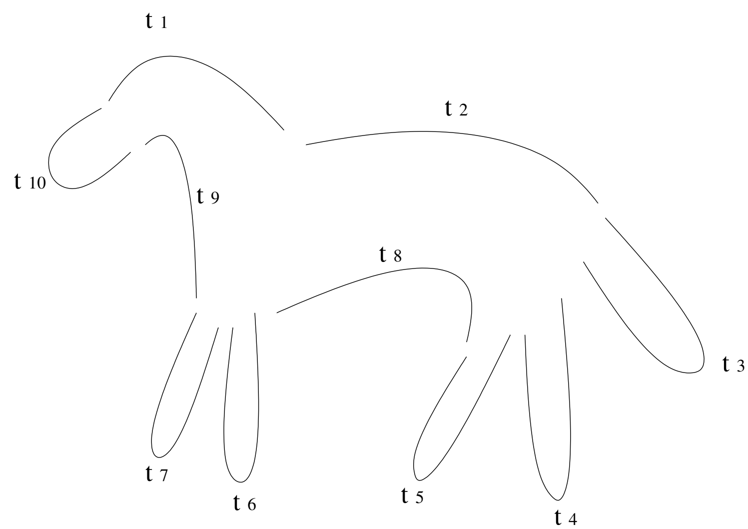


Figure 11: Shape of a horse is partitioned into tokens$(t_1, t_2, ..., t_{10})$

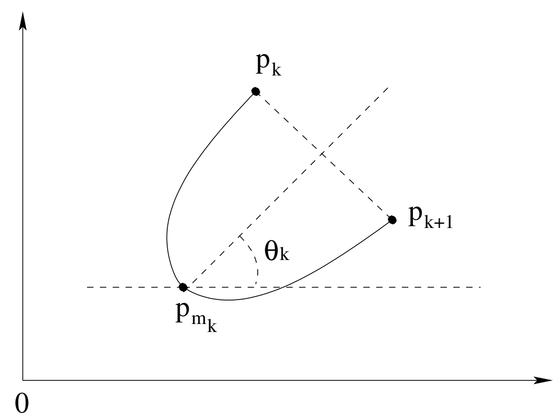


Figure 12: Information representation of a token in 2-D space

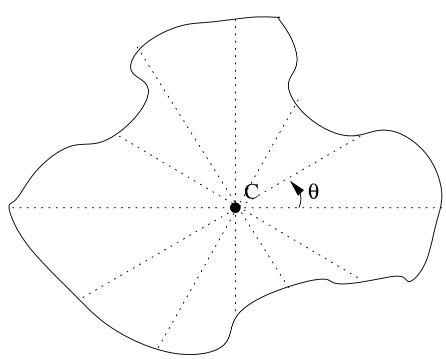


Figure 13: The centroid-radii modeling of shape

by two vectors:

$$(l1_0, l1_\theta, l1_{2\theta}, ..., l1_{(k-1)\theta})$$

and

$$(l2_0, l2_\theta, l2_{2\theta}, ..., l2_{(k-1)\theta})$$

then these two shapes are considered similar if:

$$|l1_{i\theta} - l2_{i\theta}| < MRDT \; \forall i \in [0, k-1]$$

where MRDT is the *maximum radius difference tolerance* [TT01].

Tan and Thiang's paper is very recent (yet unpublished at the time of this writing). Their approach to shape representation, which uses a centroid-radii model, is similar to some extent to our approach which will be presented in section 3.7. There are, however, some salient differences, which we will discuss after describing our method.

2.7 Other representations for shapes

Grosky and Mehrotra's proposal [GM90][GNM92] also presents a method for shape representation. This representation is based on local structural features, in which a local structural feature is defined by considering the internal angle at a vertex, the distance from the adjacent vertex, and the vertex coordinates. In each shape, a fixed number of these local features can be extracted.

Mehrotra and Gary develop a technique which is quite similar to Grosky and Mehrotra's [MG95], but is more robust. In their approach, they still use local structural features to represent the shape, and each structural feature is represented as a point in a multidimensional space.

Sclaroff and Pentland, in their proposal for the Photobook system [PPS94], represent the shapes as modal deformations of a prototype object. As a result, shape similarity can be computed by calculating the distances between mode vectors.

In Del Bimbo and Pala's approach to shape representation and comparison [BP97], shape similarity is considered by minimizing the cost which accounts for the amount of deformation of a sketch and the degree of matching achieved between the database shape and the deformed sketch.

The QBIC system [FFN+93][NBE93] has a function for similar shape matching. This system use a set of 22 global shape attributes to describe the shape. These attributes include area, circularity, major axis orientation, and a set of algebraic moments. The vector formed by these 22-dimensional features is mapped into a lower dimensional feature space through a distance-preserving transform.

Smith and Jain studied the shape matching problem [SJ82]. They proposed a method which consists of computing the distribution of random chords over the edges of the objects to be matched and comparing them by means of the Kolmogorov-Smirnov test [GSF77].

2.8 Conclusion

Among the various approaches to shape representation, the method based on Fourier descriptors attracts more of our attention, because:

- we think that using the information from the boundary to describe a shape is the most straightforward idea.
- using the Fourier descriptors to represent those sample points has the advantage that when data is transformed from the spatial domain into the frequency domain, most information will focus on the lower frequency section. This allows us to use less data when compared to that in the spatial domain.
- this method is effective. Also, the processing procedures for solving the problem of invariant under scale, translation, and rotation are reasonably simple.

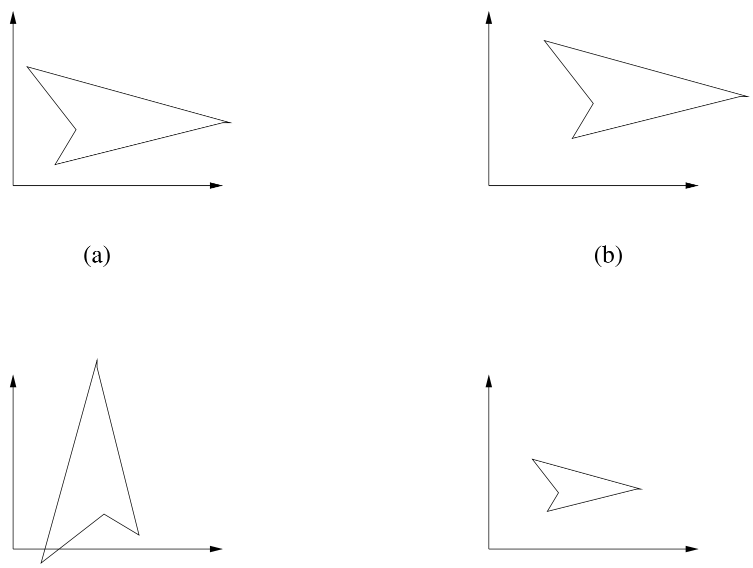
In the next chapter, we will propose our approach to shape representation which is similar to that of the method based on Fourier descriptors.

3 Our proposal - A method based on Distance Histograms

3.1 Motivation

In the previous chapter, we reviewed several approaches to shape representation. In our opinion, for good shape representation, there are three important criteria:

- The representation should be invariant to translation, rotation, and scale. Figure 3.1 illustrates an example of translating, rotating, and scaling a polygon. In this case, we assume the polygons in Figures 3.1(a) to 3.1(d) should be considered similar by a shape representation.
- It should be reasonably easy to implement.
- It should match the human intuitive notion of shape similarity.



(a) (b)

(c) (d)

Figure 14: (a) Polygon P (b) P after translating (c) P after rotating (d) P after scaling

When we think about the geometric feature of shapes, we find that just as with the center of a circle, the centroid of a shape is quite important (See Figure 3.2). Obviously, given a shape, the position of its centroid corresponding to its boundary will not change under translation and rotation.

We know that any point in a circle's boundary has the same distance, called the radius, to its centroid. We can apply this method to an ordinary shape. However, in other shapes, only one radius may not be enough because the distances from points in the boundary to the centroid may be different. Therefore, we can use a set of radii which begin at the centroid and end at the boundary, to represent a shape. It means we can describe the shape as long as we use enough radii. Fortunately, this representation is invariant to translation and rotation.

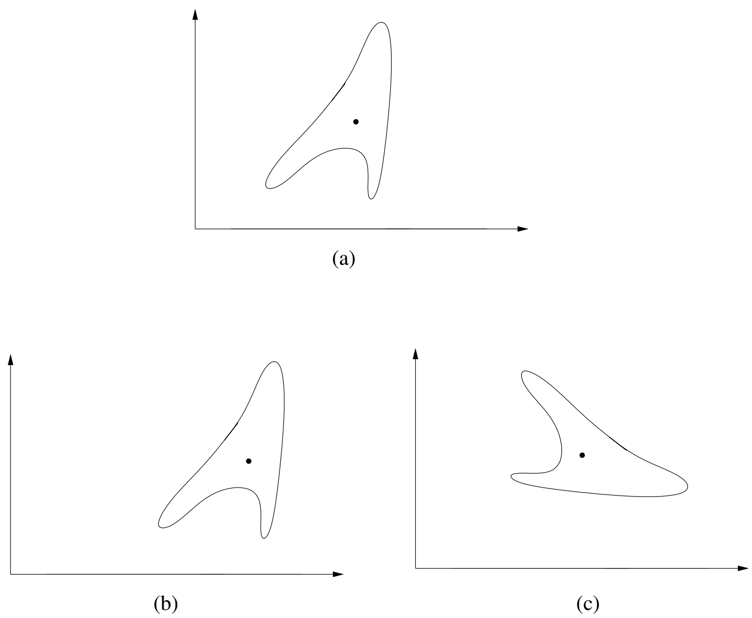


Figure 15: (a) A shape and its centroid (b) A shape and its centroid after moving (c) A shape and its centroid after rotation

Based on the discussion above, we propose a new methodology to describe and represent shapes. Further, we will also discuss how to use this methodology to compare and search for shapes.

Given an image which contains a single object, we assume the boundary of the object, i.e., its shape, has already been extracted. There are several types of algorithms which can be used to extract the boundary of objects. These are, however, beyond the scope of this thesis. The problem we are concerned with is how to describe the object's shape when its boundary is already available. Basically, the approach we will propose contains four parts:

1. Given a polygon which approximates the boundary of a shape, we calculate its centroid.
2. A set of sample points in the boundary of the polygon is selected and the distances between them and the centroid are computed.
3. A histogram based on the distances computed in the previous step is constructed.
4. Finally, we normalize the Distance Histogram.

3.2 Centroid of polygon

Given a polygon such as the one in Figure 3.3, we need first to find its centroid. In order to compute the x and y coordinate of the centroid, we first need to calculate the area of the polygon. The method we will use is based on Green's Theorem in a plane. Given a polygon and its vertices (x_i, y_i), i=0,...,n, with x_0=x_n and y_0=y_n, the following formula can be used to calculate the area of a polygon in a plane [Efg01]:

$$A = \frac{1}{2} \sum_{i=0}^{n-1} \alpha_i$$

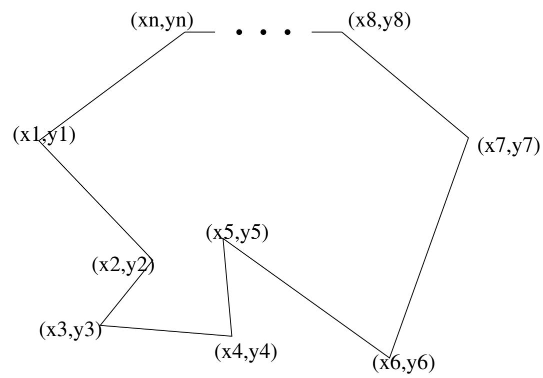


Figure 16: A polygon which has n vertices

where

$$\alpha_i = x_i y_{i+1} - x_{i+1} y_i$$

The area computed in this way is a signed value, where a negative sign indicates the vertices are in clockwise order, and a positive sign indicates the vertices are in a counter-clockwise order. Now we can compute the centroid based on the area. The coordinates for the centroid are:

$$\overline{x} = \frac{\mu_x}{A}$$

$$\overline{y} = \frac{\mu_y}{A}$$

where

$$\mu_x = \frac{1}{6}\sum_{i=0}^{n-1}(x_{i+1} + x_i)\alpha_i$$

$$\mu_y = \frac{1}{6}\sum_{i=0}^{n-1}(x_{i+1} + x_i)\alpha_i$$

The centroid of any polygon in a plane can be calculated by this method.

3.3 Sample points selection and distance calculation

In the previous section, we presented the method of computing the centroid of a polygon. Naturally, for describing the entire shape, the centroid alone is not enough. In our approach to shape representation, we need a set of sample points selected from the boundary to describe the shape. Therefore, in this section, we will discuss how to select the sample points around the boundary of a polygon, and how to calculate the distance from the centroid to the sample points.

Firstly, we should decide how many sample points will be selected around the boundary. We consider this number to be a variable, therefore we can change it for different situations. Secondly, we should decide how to select these sample points. Here, we have two problems. One is how many sample points should be in each edge; the other is how to select them within the edges.

To deal with the first problem, we can assign the number of sample points randomly to each edge. This method presents a difficulty though: sometimes a very short edge is assigned more sample points than a much longer one. Furthermore, two polygons with the same shape may have different sample points. Therefore we choose to assign the number of sample points in each edge proportionally to its length. For example, if the length of an edge is L_i, the sum of the length of all edges is L_{sum}, and the entire number of sample points is N, then the number of sample points in this edge N_i will be:

$$N_i = \frac{L_i}{L_{sum}} N$$

To deal with the second problem, we decide the location of sample points so that they are evenly spread in the edges.

Finally, we can use sample points and the centroid to calculate the distances. For example, given a sample point $s_i = (x_i, y_i)$ and the centroid $c = (x_c, y_c)$, the distance between them is:

$$d(s_i, c) = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}$$

In this way, we can calculate the distances from the centroid to all sample points around the boundary, and we are able to use them to construct a Distance Histogram to describe the shape. Using the centroid and sample points in the boundary to describe the shape is one of the central ideas of our proposal. This is discussed in the next section.

Our approach based on distances has some apparent advantages. The distance set we use to represent the shape will not change after translating the shape. This is because the distances only have a relationship with the centroid and sample points in boundary. If we just move the shape, the location of the centroid will not change, and neither will the locations of the sample points. Therefore, the distance set will not change. If we rotate the shape, the distance set will still not change, the reason being that we assigned the sample points in each edge proportionally to its length, and the sample points are spread evenly in each edge. Therefore, the location of the sample points will not change after rotating the entire shape. As for the problem of invariance under scale, we can make a simple normalization of the distances. We will discuss this in Section 3.5.

3.4 Histogram

A histogram is a very useful tool to represent some property of data, and is used in many fields and applications. In this section, we will present the idea of how to construct the Distance Histogram.

For shape representation, generally it is not trivial to be invariant to rotation. For example, in the methods based on Fourier descriptors [RM99] and turning angle [ACH+91], some kind of complicated process is required in order to solve the problem of invariant to rotation. Fortunately, in our method, solving this problem is quite easy and straightforward.

In the methods based on Fourier descriptors and turning angle, it is not only necessary to select the sample points around the boundary, but also to pick a starting point (e.g., the top-left corner) which will be used in the next computation. This starting point will be changed when the polygon is rotated, which will affect the next computation, and result in generating a different representation of a polygon before and after rotation. To avoid this problem, we use the Distance Histogram in our approach. It works as follows:

Firstly, we separate the range of all distances, which is [0, D_{max}], into several ranges, e.g., R ranges. Then, the ranges of distances can be represented by: [0, D_{max}/R], (D_{max}/R, $2\times D_{max}/R$], ($2\times D_{max}/R$,

$3\times D_{max}/R$],...,($(R-1)\times D_{max}/R$, D_{max}]. Secondly, we compute the number of distances in each range, e.g., if we have a sample point and its distance from the centroid is $1.5\times D_{max}/R$, then we will increase the number of ranges ($1\times D_{max}/R$, $2\times D_{max}/R$] by 1.

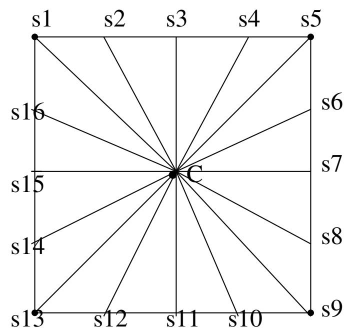


Figure 17: A polygon and its sample points and radii

Figure 3.4 illustrates an example. The polygon in Figure 3.4 is a square, and we evenly select 16 sample points around its boundary. We define D_n is the original distance from centroid C to sample point s_n, which $n \in (1, 2, 3, ..., 16)$. Given that the lengths of each edge of this rectangle are the same, which is 4, then we have:

$$D_1 = D_5 = D_9 = D_{13} = 2.828$$

$$D_2 = D_4 = D_6 = D_8 = D_{10} = D_{12} = D_{14} = D_{16} = 2.236$$

$$D_3 = D_7 = D_{11} = D_{15} = 2.000$$

The maximum value is 2.828. We separate the range of value [0,maximum] into 4 ranges, which are [0,0.707], (0.707,1.414], (1.414,2.121], (2.121,2.828]. We then know that: D_3, D_7, D_{11}, D_{15} belong to range (1.414, 2.121] and the other distances belong to range (2.121, 2.828]. Figure 3.5 illustrates the histogram:

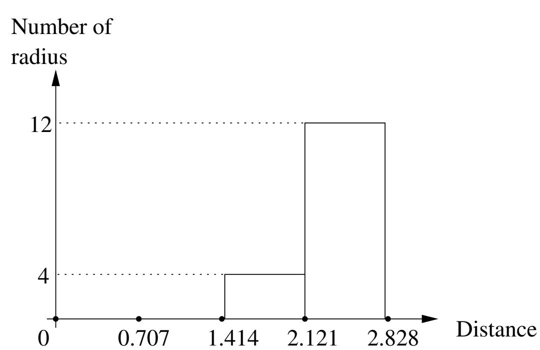


Figure 18: Distance Histogram of polygon in Figure 3.4

This histogram can be used to represent the shape of polygons. In this sense, our approach can be referred to as a method based on Distance Histograms.

In the procedure above, the selection of sample points will not change with the rotation, nor will the radii. Therefore, our shape representation by Distance Histograms is independent of the starting point in the boundary; what matters is only the number of radii in each range. This number will not change under the rotation. Therefore, our shape representation is invariant to rotation. However, our method has a problem: two similar shapes with different sizes may have different Distance Histograms as representations. This means that our approach is still sensitive to scale. To solve this problem, we need to make some kind of normalization to the distances. This is discussed next.

3.5 Normalization

Given two polygons with different sizes, but the same shape, they should be considered similar. However, if we only use original distances and a Distance Histogram constructed from them to represent and compare them, they will be considered different.

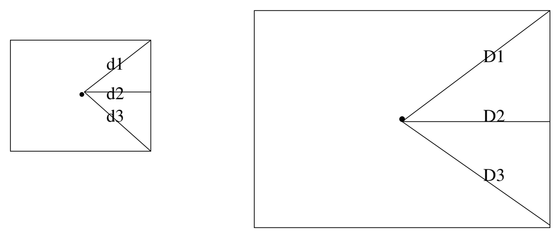


Figure 19: Two polygons with different sizes but similar shapes

For example, in Figure 3.6, two rectangles have different sizes but the same shape. If we only use (d1,d2,d3) and (D1,D2,D3) to represent the shapes and make the comparison, these two similar shapes will get different representations and the result of the comparison will be not similar. Therefore, we need to perform some kind of process on the distances which were obtained in Section 3.3.

Normalization is a very useful process for solving this problem. Firstly, we need to find the maximum distance among all the distances. Then, we divide all of them by this maximum one and get the normalized distances. After this process, the value of all normalized distances will be in the range $[0, 1]$. And because we assign the sample points based on the length of the edges, and evenly spread them in each edge, two polygons which have different sizes but the same shape will generate the same normalized distances. Therefore, our method is invariant to scale after normalization.

After normalization, the histogram shown in Figure 3.5 will become the one illustrated in Figure 3.7.

3.6 Similarity Metric

Given a shape, we can use the method discussed in the previous section to construct a Distance Histogram as its shape representation. The Distance Histogram can be represented as:

$$D : (d_0, d_1, d_2, ..., d_n)$$

where n is the number of ranges in the Histogram and d_i $i \in [0, n-1]$ is the number of distances belong to this distance range. In this way, given two shapes:

$$D_1 : (d_{10}, d_{11}, d_{12}, ..., d_{1(n-1)})$$

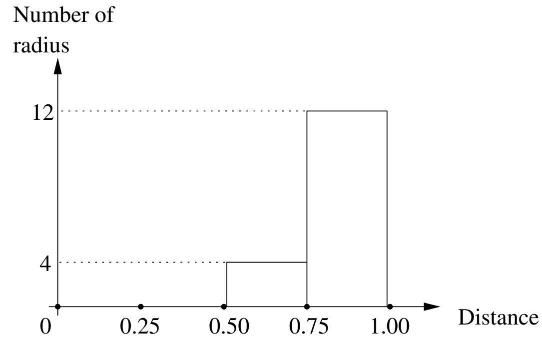


Figure 20: Normalized Distance Histogram of polygon in Figure 3.4

and

$$D_2 : (d_{20}, d_{21}, d_{22}, ..., d_{2(n-1)})$$

their similarity can be measured by Euclidean distance:

$$SIM(D_1, D_2) = \sqrt{\sum_{i=0}^{n-1} (d_{1i} - d_{2i})^2}$$

3.7 Summary of the Distance Histogram Technique

1. Given an object, extract its boundary and use a polygon P to approximate it.
2. Calculate the centroid of polygon P.
3. Select the sample points around the boundary of P, then compute the set of distances which are between the sample points and the centroid.
4. Normalize the set of distances.
5. Based on normalized distances, construct the Distance Histogram.
6. Use this histogram for shape representation and comparison.

In Section 2.6, we reviewed Tan and Thiang 's approach to shape representation. Their method, which uses a centroid-radii model, is similar, to some extent, to our proposal for shape representation. However, there are several important differences between them:

- Our method uses the centroid and radii of a shape to construct a Distance Histogram as the representation for it. Meanwhile, their method uses the centroid and radii to generate a vector, in which each element is the normalized length of the radius, for shape representation. In fact, the vector depends on the order in which the radii are measured. As a result, their method is not invariant to rotation, whereas ours is.
- We choose sample points for every edge proportionally to its length, and spread sample points evenly in each edge. Meanwhile, their method chooses sample points by radii which are at a regular angle interval from the centroid.

- In our method, we use the Euclidean distance of the Distance Histograms to measure the similarity of different shapes. Meanwhile, their method uses the equation in Section 2.6 to justify whether two shapes are similar.

In the next chapter, we will conduct a series of experiments to evaluate the effectiveness and efficiency of our approach.

4 Experiments

4.1 Motivation

In the last chapter we presented our approach, which is a method based on a Distance Histogram, for describing and comparing the shapes. As a shape representation, it needs the quantitative results to assure whether it is good or not. Therefore, we need to design a set of experiments to test our method compared with others. In Chapter 2, we reviewed several approaches for shape representation. Among them we decided to pay more attention to the method presented by Rafiei and Mendelzon [RM99], which is a method based on Fourier descriptors. This is because their approach is somewhat similar to ours. Both their method and ours need to collect data, which are sample points in the boundary of the shape. Both also use Euclidean distance for calculating the difference between two shapes. The main difference between them is that after picking the sample points, their method uses them to compute the Fourier descriptors, while ours uses them to construct a Distance Histogram. We therefore selected this approach to use as a comparison with ours.

4.2 Experiment 1: Polygons

For the purpose of our experiments, we constructed a framework which we could use to retrieve similar shapes from a shape database. This system uses our method, which is based on Distance Histograms, and also the method based on Fourier descriptors, as the basic techniques for shape representation and comparison. Given a query, this system can retrieve two answer sets of similar shapes using the approaches for shape representation.

In this experiment, we use polygons as the data. In other words, the query shape and all the shapes in the database are polygons. This is because we consider the polygon to be the simplest shape, and any shape can be approximated by a polygon.

The experiment contains several steps:

1. Generate a large number of non-selfcrossing polygons.

2. Pick a subset of polygons and generate controlled variations of them. These variations and the original polygons compose the primary database for experiment.

3. Use our method to process these polygons. For each of them, find the centroid, calculate the radii and ranges, construct the Distance Histogram and, finally, store these histograms in a data file.

4. Use the method based on Fourier descriptors to process these polygons. For each of them, calculate the sample points around the boundary, then compute Fourier descriptors from these sample points and store them in another data file.

5. Pick a polygon which has variations as the query. Calculate the histogram and Fourier descriptors of this query. Compare them with those histograms and descriptors in the data file and compute the differences. The smaller the difference is, the more similar the polygon is to the query. We can compute the differences of all polygons in the database from the query and rank them according to their differences from the query from the least (most similar) to the most (most dissimilar).

6. All the variations of a query are considered relevant. Hence, we can calculate the recall and precision of each query.

4.2.1 Generating polygons

We need to perform the following steps to generate the random non-self-crossing polygons: [AN00]

1. Choose a rectangle in the space which specifies the location and size of the polygon. The rectangle we use is a square S defined by 4 points: (0,0) (0,200) (200,200) (200,0).
2. Randomly choose a number of points inside S. These points will be the vertices of the polygon.
3. Randomly choose a horizontal line L which cuts through S; L divides the points into two sets: points above L and points below L.
4. Connect the points in each set according to their x (horizontal) coordinate, two "chains" are created.
5. The leftmost and rightmost points in the two chains are then moved vertically to the splitting line L. This is in order to avoiding self-crossing.
6. Connect two chains of points by their end-points, so that a polygon P is created.
7. Rotate P by a random angle.

Figure 4.1 illustrates the procedure of generating a polygon.

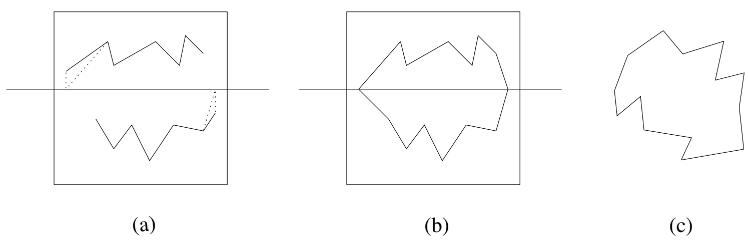


Figure 21: Generating a polygon [AN00]

4.2.2 Generating variations

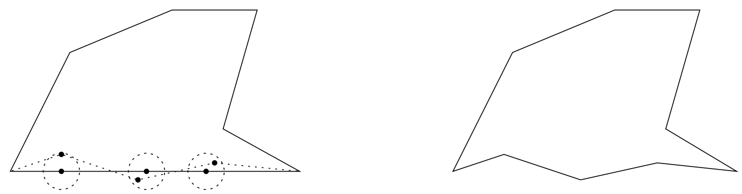
To generate variations of the original polygon, we proceed as follows:

First, we decide how many vertices will be added into the original polygon, and we consider that the number of added vertices should be in a specified range. For example, if the original polygon has n vertices, then we could add $\frac{n}{2} - n$ vertices into it.

We then randomly choose the number of vertices which will be added into a specified edge. For example, suppose we decide to add 3 vertices in edge (v_i, v_{i+1}). We choose 3 points p_m, p_n, p_l randomly in this edge. For each point in the edge, e.g., p_m, we change its x and y coordinate to create a new point p_M outside this edge. We specify that the distance between the new point p_M and the original one p_m should be less or equal to a pre-specified value $radius$. In this way, we can get 3 new points, p_M, p_N, p_L, which are outside the edge. These 3 points will be the new vertices added into the polygon. After reconnecting, the edge (v_i, v_{i+1}) is changed into 4 edges:

$$(v_i, p_M)(p_M, p_N)(p_N, p_L)(p_L, v_{i+1})$$

Figure 4.2 illustrates the procedure of adding new vertices on an edge. We perform the same process on all the edges in the original polygon. In this way, we create a new polygon.



(a) (b)

Figure 22: Adding new vertices on the edge

We can say that the new polygon is very similar to the original one as long as we make the $radius$ quite small. Therefore, this new polygon is called a variation of the original one. Figure 4.3 illustrates a polygon and its variation.

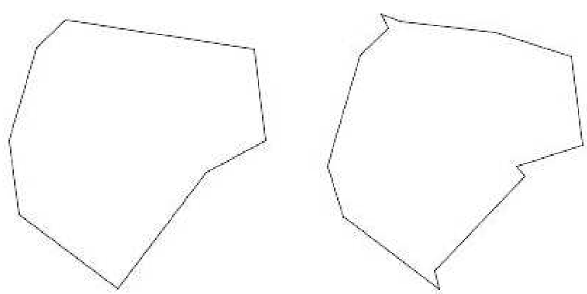


Figure 23: A polygon and its variation

4.2.3 Precision & recall

Basically, precision and recall are standard measures for Information Retrieval systems [Iss96]. Given a query Q:

- RET is the whole answer set the system has retrieved for Q;
- REL is the set of relevant answers for Q;
- RETREL is the set of relevant answers which is retrieved by the system for Q.

Furthermore, precision and recall measures can be defined as follows:

- precision = RETREL/RET
- recall = RETREL/REL

After calculating precision measures at pre-specified recall levels, i.e., (10%, 20%, ..., 100%), we can draw Precision-Recall curves in order to evaluate a system's retrieval effectiveness. Ideally, we want the curve to be as "high" as possible.

In the experiments, we define several variables:

- $Num\_Original$: the number of original polygons. The original polygon is the polygon that has variations. Some original polygons will be used as queries.
- $Num\_Variation$: the number of variations of an original polygon.
- $Range\_Add\_Vertex$: The range of how many vertices will be added into an original polygon to obtain the variations.
- $Range\_Radius$: the range of how large the modification of x and y coordinates will be in order to obtain the vertices which will be added into an original polygon.
- $Num\_Polygon$: total number of polygons in the database. It is equal to $Num\_Original \times (Num\_Variation + 1)$+(number of polygons which have no variation).
- $Num\_Samplepoint$: How many sample points will be selected around the border in order to obtain the radii and Fourier descriptors.
- Num_FD: How many Fourier descriptors will be used.
- Num_His_Ranges: How many ranges will be in the final histogram to describe a polygon.

In these experiments, we generate 10 polygon databases. The number of polygons in the databases is defined by variable $Num\_Polygon$. For each database, we pick 10 original polygons as queries and compute their precision measures at pre-specified recall levels, respectively. Therefore in total we obtain 100 groups of precision measures at pre-specified recall levels. Finally, we use the average values of these groups of precision measures to draw the Precision-Recall curves.

4.2.4 Experiment 1.1

First of all, we set default values for the variables listed above. In subsequent sections we change the values of some of these variables and compute the Precision-Recall curves. Default values are:

- $Num\_Original$: 50
- $Num\_Variation$: 9
- $Range\_Add\_Vertex$: 1 (i.e., the number of vertices is at most doubled)
- $Range\_Radius$: Medium (i.e., the modification of the x and y coordinate is at most 2.5% of the size of square we used to generate polygons)
- $Num\_Polygon$: 1000 (i.e., we generate another 500 different polygons, therefore 500+50×(9+1)=1000)
- $Num\_Samplepoint$: 128
- Num_FD: 64
- Num_His_Ranges: 30

Using the default setup, the Precision-Recall curve is as depicted in Figure 4.4:

In Figure 4.4, FD denotes the method based on Fourier descriptors, while DH denotes our method based on Distance Histograms. In Figure 4.4, we find that two curves are quite similar, which means that our method and the method based on Fourier descriptors offer similar retrieval effectiveness. In subsequent experiments, we will use these Precision-Recall curves under the default setup as the reference.

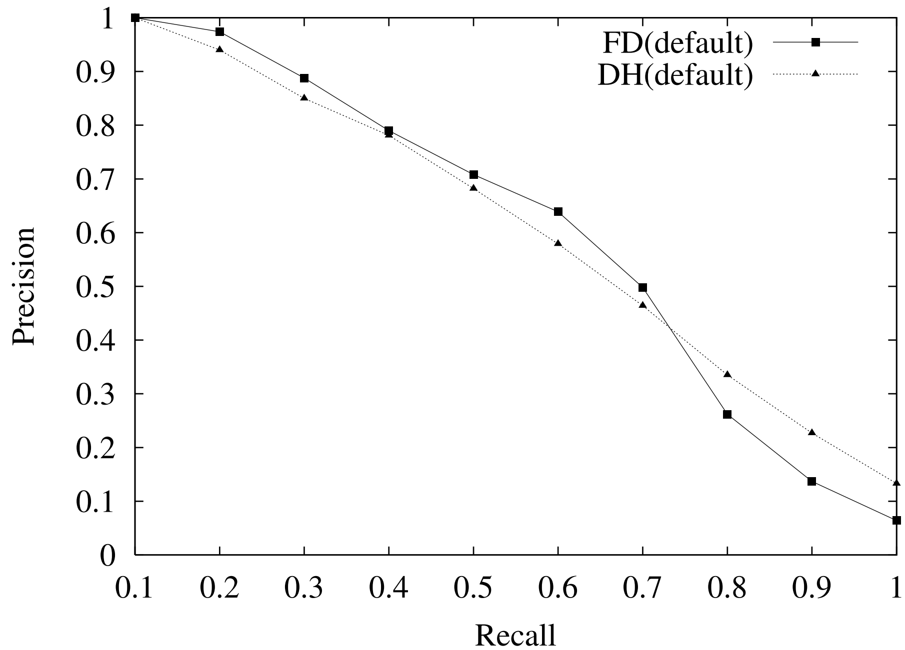


Figure 24: Precision-Recall curve: default setup

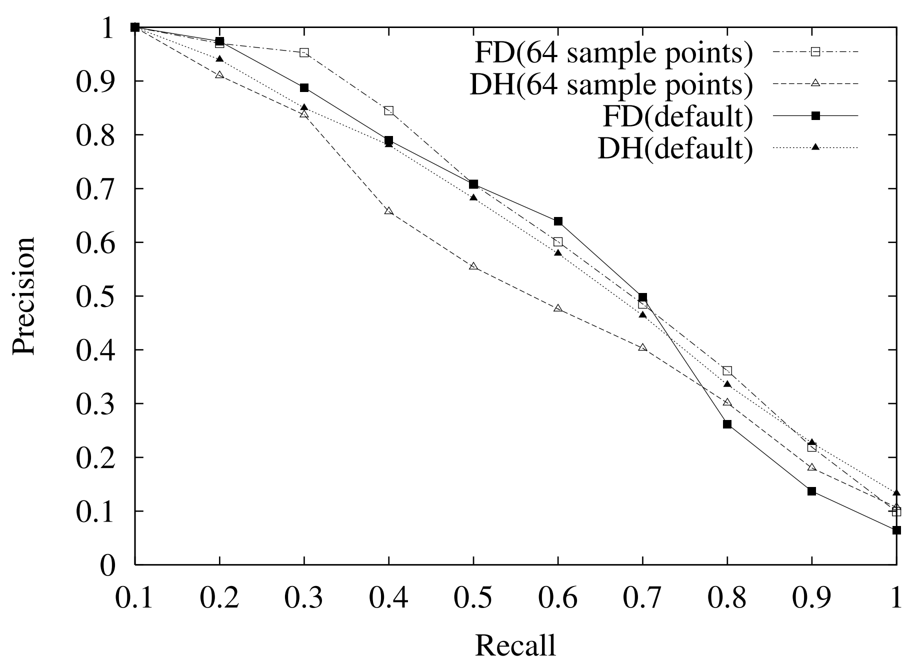


Figure 25: Precision-Recall curves: using 64 sample points (default of 128)

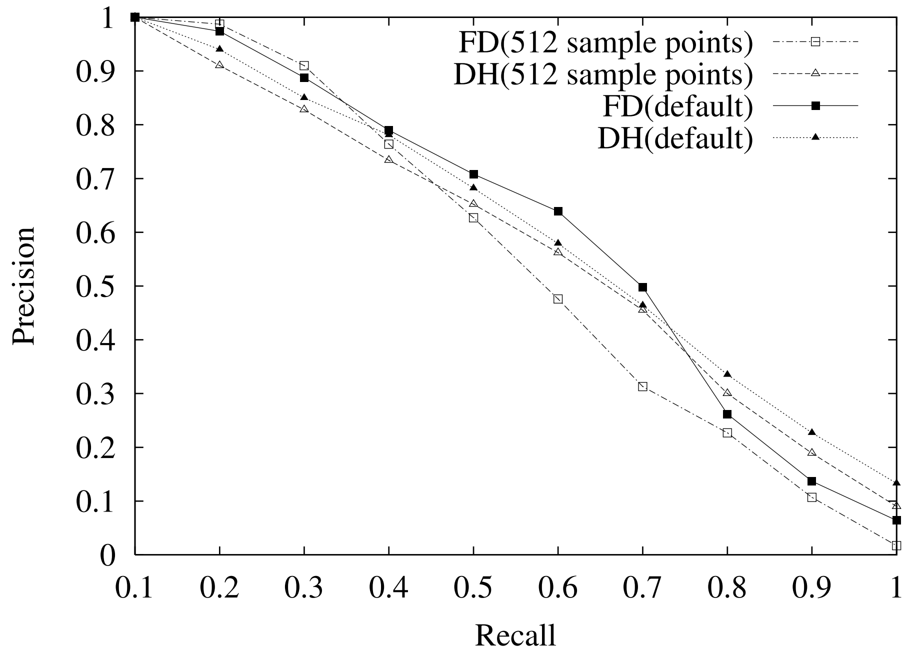


Figure 26: Precision-Recall curves: using 512 sample points (default of 128)

4.2.5 Experiment 1.2

In this experiment, we change the value of $Num\_SamplePoint$ and keep all other variables as default. From Figures 4.5 and 4.6, we can see that decreasing the sample points in the boundary, which means using less information on the shape's boundary, decreases the effectiveness of our method. On the contrary, increasing the sample points will make the effectiveness of our method slightly better. However, in Figure 4.6, we can see that there is no apparent difference between the performance of the method using 512 sample points and that using 128 sample points (default). Even the performance using 128 sample points is slightly better than the one using 512 sample points. This is due to the effect of noise in the experiment. Meanwhile, the performance of the method based on Fourier descriptors does not change a great deal. This is because we have increased only the number of sample points, not the number of Fourier descriptors. A larger number of sample points means we can obtain more detailed information and additional higher frequency information in the frequency domain. If we do not increase the number of Fourier descriptors accordingly, the higher frequency information will be lost in the final Fourier descriptors representation.

4.2.6 Experiment 1.3

In this experiment, we change the value of $Range\_Add\_Vertex$ which involves changing the number of vertices added into the original polygon in order to obtain variations. From Figures 4.7, 4.8, and 4.9, we can see that our method is more sensitive to small variations than the method based on Fourier descriptors. The variable $Range\_add\_vertex$ represents the number of vertices added into the original polygon. The larger it is, the more vertices will be added. Generally the most modification happens in the original polygon. As a result, there is a greater variation from the original one. Both our method and the method based on Fourier descriptors, demonstrate lower effectiveness with the addition of more vertices into the original polygon. However, the loss with our method (e.g., about 8% in average precision when $Range\_Add\_Vertex$=2.0) is larger than the loss using

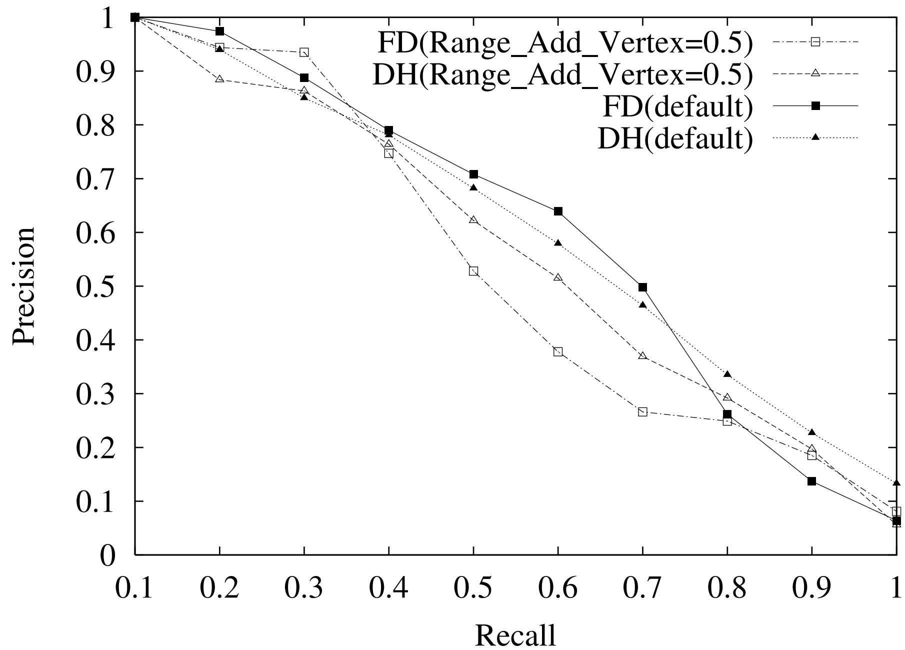


Figure 27: Precision-Recall curves: assuming the number of vertices of the original polygon is n, adding at most $\frac{n}{2}$ new vertices into it

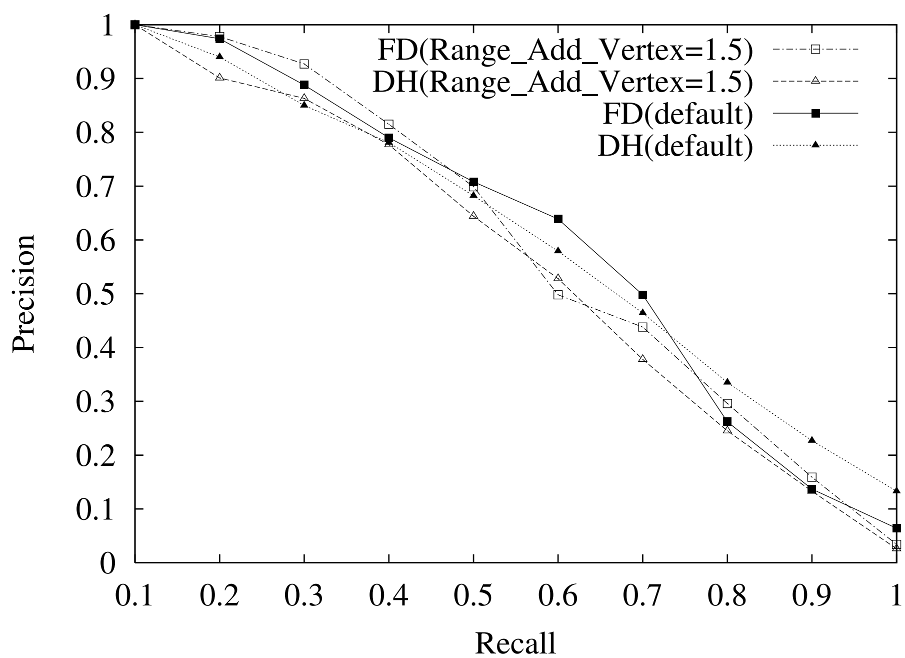


Figure 28: Precision-Recall curves: adding at most $1.5n$ new vertices into original polygon

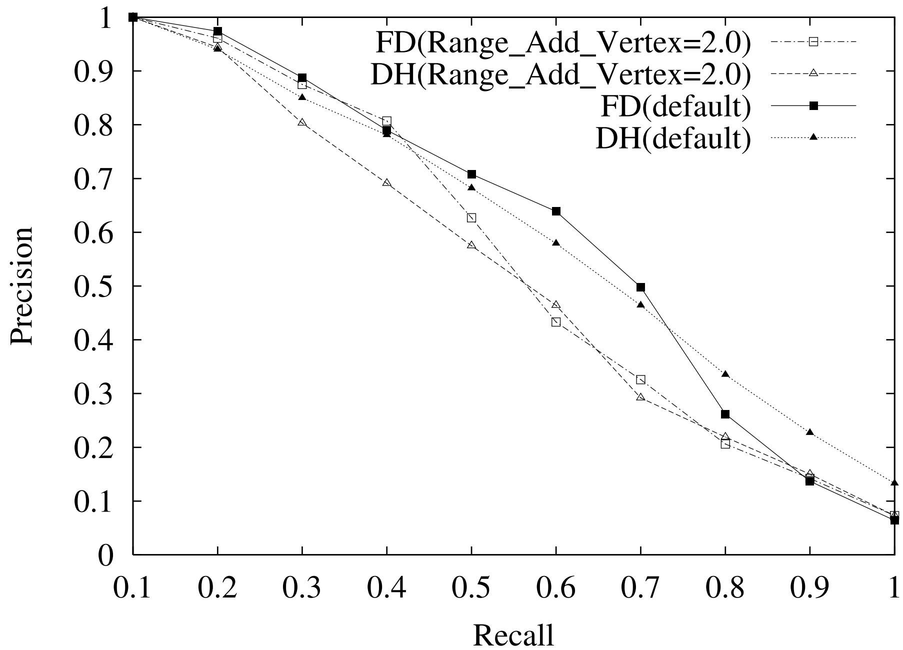


Figure 29: Precision-Recall curves: adding at most $2n$ new vertices into original polygon

the method based on Fourier descriptors (e.g., about 5% in average precision when $Range\_Add\_Vertex$=2.0). This means that our method is more sensitive to small variations than the method based on Fourier descriptors.

4.2.7 Experiment 1.4

In this experiment, we change the value of $Range\_Radius$. Figures 4.10 and 4.11 illustrate the Precision-Recall curves with $Range\_Radius$ changed. Similar to Experiment 1.3, increasing $Range\_Radius$ also means that the variation from the original polygon is greater. Changing $Range\_Radius$ from small to large, both our method and the method based on Fourier descriptors demonstrate an apparent lower effectiveness. However, the loss with our method (e.g., in average precision, about 19% lower than default when $Range\_Raduis$ is large) is apparently larger than the loss using the method based on Fourier descriptors (e.g., about 10%). This difference suggests that our method is more sensitive to the variations of variable $Range\_Radius$.

4.2.8 Experiment 1.5

In this experiment, we change the size of the polygon database. Here we need to modify 2 variables which are $Num\_Original$ and $Num\_Polygon$. From Figures 4.12, 4.13, and 4.14, we see that the effectiveness of both our method and the method based on Fourier descriptors decreases with an increase in the size of the database. Under the default setup, when there are 1000 polygons in the database for retrieval, the effectiveness of our method and the method based on Fourier descriptors is almost the same. When the database has 500 polygons, the effectiveness of the method based on Fourier descriptors is about 4% better than default, while the effectiveness of our method is almost the same as default. When the database has 5000 or 10000 polygons, the effectiveness of our method is about 21% worse than default, while the effectiveness of the method based on Fourier descriptors is about 8% worse than default. From these data, we can see that effectiveness of our method decreases faster than that of the method based on Fourier descriptors, with an increase in the size of the database.

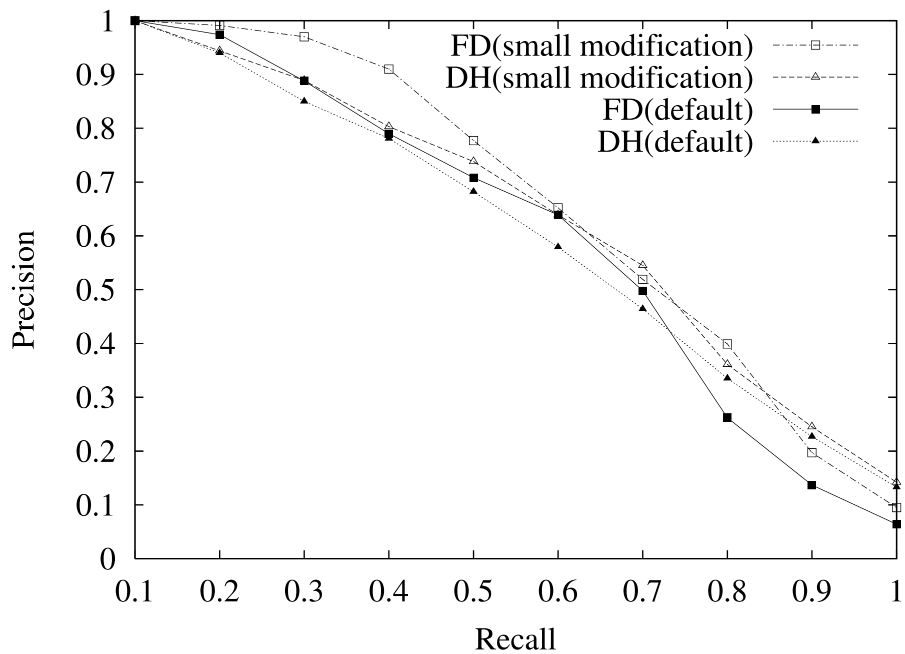


Figure 30: Precision-Recall curves: $Range\_Radius$=small, which means the modification of x and y coordinates is at most 1.5% of the size of square used to generate polygons.

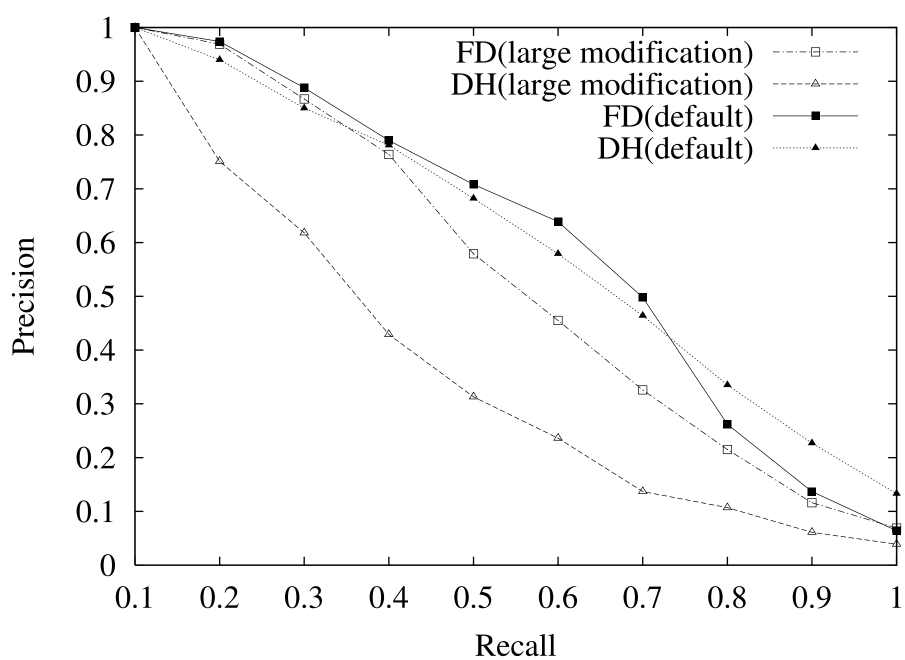


Figure 31: Precision-Recall curves: $Range\_Radius$=large, which means the modification of x and y coordinates is at most 5%.

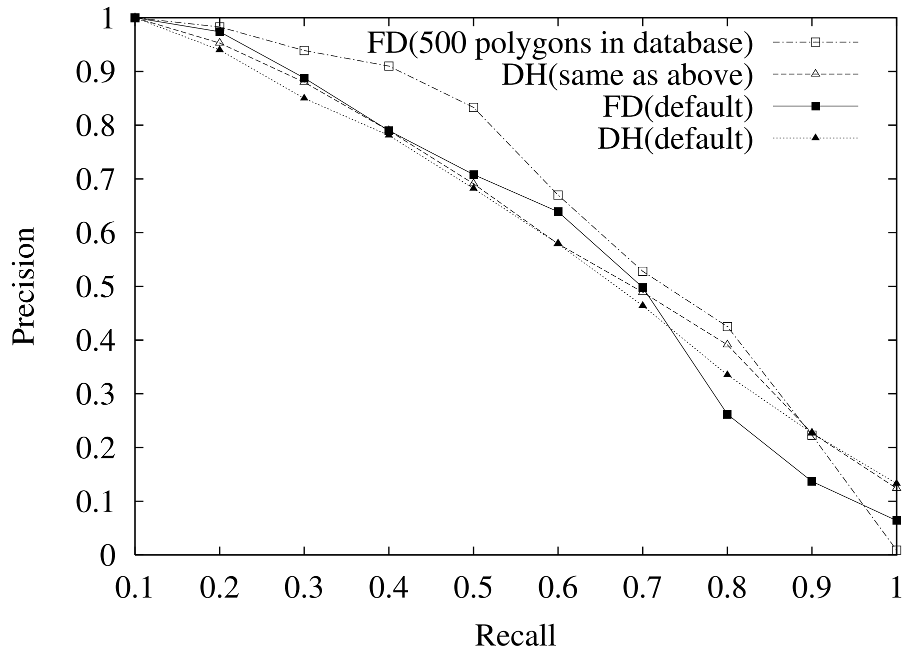


Figure 32: Precision-Recall curves: 500 polygons in database

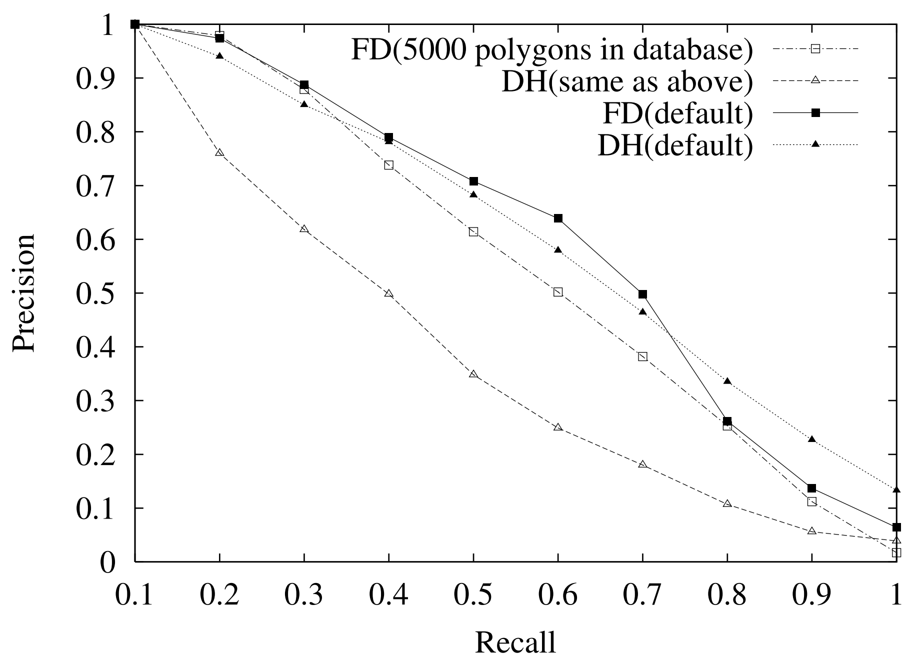


Figure 33: Precision-Recall curves: 5000 polygons in database

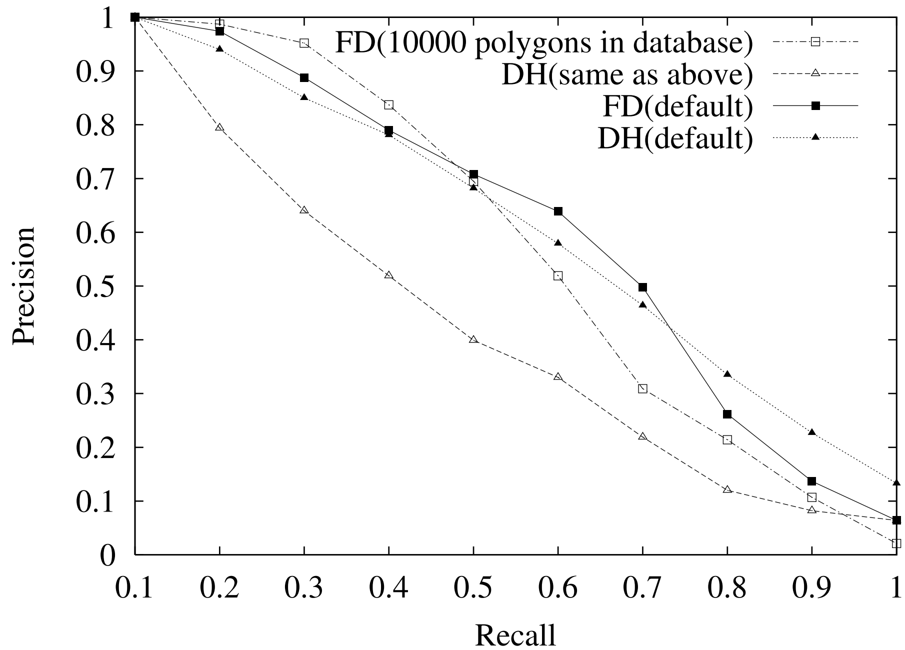


Figure 34: Precision-Recall curves: 10000 polygons in database

4.2.9 Experiment 1.6

In this experiment, we change the ranges of the Distance Histogram used in our method. Therefore, the Precision-Recall curve for FD is not included in Figure 4.15. From Figure 4.15, we can see that increasing the number of the ranges used in the Distance Histogram improves the effectiveness of our method slightly. This is because using more ranges means describing the shape in more detail.

4.2.10 Processing time and storage overhead

There is another criterion to evaluate the effectiveness of our method when compared with the Fourier Descriptors. It is called $average\_precison$. Basically, it means that we can compute the average precision value at some recall points. We use the precision at recall levels equal to 0.2, 0.5, and 0.8 to obtain the average value. However, to evaluate the entire performance of a technique, we have to consider the time and space cost, in addition to the effectiveness, which is represented by precision. In our experiments, the time cost is the time needed to conduct the search and obtain the answer. And the space cost refers to the storage in hard disk used for retrieval.

Tables 4.1 to 4.5 display the $average\_precision$, processing time, and storage overhead of our method and the method based on Fourier descriptors, under different conditions. As we have seen, under some conditions, e.g., a large database, the effectiveness of our method is somewhat less than the effectiveness of the method based on Fourier descriptors. However, under all conditions, the processing time and storage overhead of our method is much less than the method based on Fourier descriptors. For example, in Table 4.4, the number of polygons in the database is changed from 500, to 1000, 5000, and 10000. The processing time of our method is about 3-7 times less than that of the method based on Fourier descriptors. As for the storage overhead, the method based on Fourier descriptors is about 20 times larger than ours.

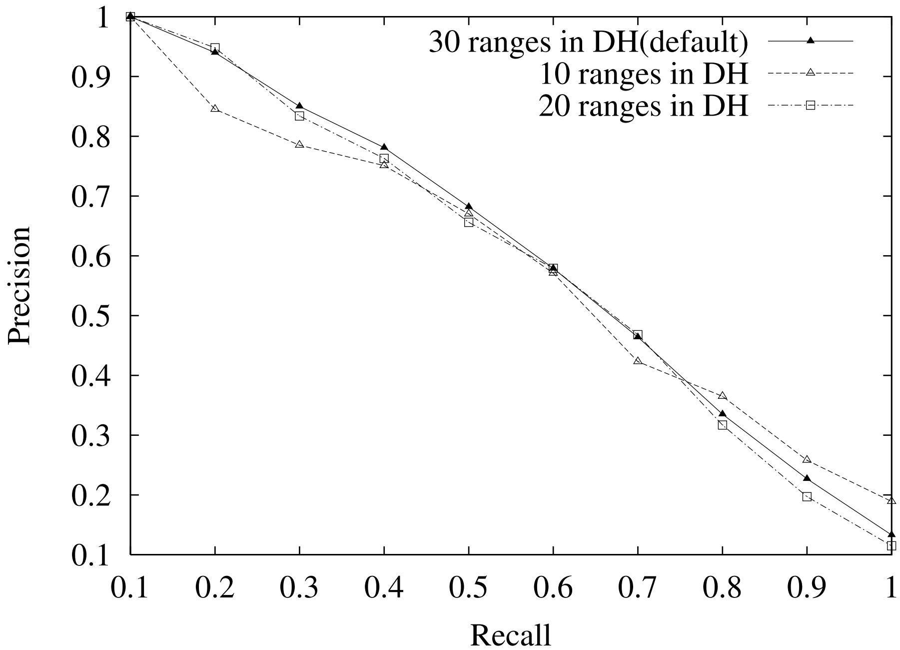


Figure 35: Precision-Recall curves: changing the ranges of the Distance Histogram for shape representation

$Num\_Samplepoint$	$avg.\_precision$		Size(Byte)		Time(Sec.)	
	FD	DH	FD	DH	FD	DH
64	0.678851	0.585980	1,435,213	136,650	0.453	0.083
128	0.651034	0.651445	1,429,935	136,650	0.889	0.168
512	0.616688	0.619291	1,433,878	136,650	3.628	0.312

Table 1: $average\_precision$, processing time and storage overhead when changing variable $Num\_Samplepoint$

$Range\_Add\_Vertex$	$avg.\_precision$		Size(Byte)		Time(Sec.)	
	FD	DH	FD	DH	FD	DH
0.5	0.744947	0.760853	2,856,345	136,650	0.912	0.157
1.0	0.651034	0.651445	2,854,300	136,650	0.889	0.168
1.5	0.692054	0.637826	2,855,113	136,650	0.874	0.172
2.0	0.599211	0.578389	2,853,433	136,650	0.881	0.158

Table 2: $average\_precision$, processing time and storage overhead when changing variable $Range\_Add\_Vertex$

$Range\_Radius$	$avg.\_precision$		Size(Byte)		Time(Sec.)	
	FD	DH	FD	DH	FD	DH
Small	0.723274	0.681366	2,856,609	136,650	0.888	0.182
Medium	0.651034	0.651445	2,854,300	136,650	0.889	0.168
Large	0.588128	0.390506	2,855,562	136,650	0.933	0.171

Table 3: $average\_precision$, processing time and storage overhead when changing variable $Range\_Radius$

$Num\_Polygon$	$avg.\_precision$		Size(Byte)		Time(Sec.)	
	FD	DH	FD	DH	FD	DH
500	0.773429	0.711567	1,496,032	71,525	0.443	0.06
1000	0.651034	0.651445	2,854,300	136,650	0.889	0.167
5000	0.617671	0.434086	13,734,810	657,650	4.835	1.508
10000	0.634121	0.439025	27,326,676	1,308,900	11.08	3.704

Table 4: $average\_precision$, processing time and storage overhead when changing variable $Num\_Polygon$

Num_His_Ranges	$avg.\_precision$		Size(Byte)		Time(Sec.)	
	FD	DH	FD	DH	FD	DH
10	0.665450	0.630268	2,856,108	52,650	0.877	0.094
20	0.727183	0.644462	2,855,559	94,650	0.879	0.141
30	0.651034	0.651445	2,853,300	136,650	0.889	0.167

Table 5: $average\_precision$, processing time and storage overhead when changing variable Num_His_Ranges

4.2.11 Experiment 1.7

From Experiment 1.6, we know that increasing the number of ranges for the Distance Histogram can improve the effectiveness of our method. Another way, which was discussed in Experiment 1.2, is to increase the sample points obtained from the boundary. Figure 4.16 illustrates the influence on the effectiveness of our method when increasing sample points used. From Figure 4.16, we can see that increasing sample points from 64 to 128 (default) improves the effectiveness of our method. However, increasing sample points from 128 (default) to 512 has little influence on effectiveness. This is because 128 sample points is already enough to describe the polygon of current size (the square used to generate the polygon is 200×200).

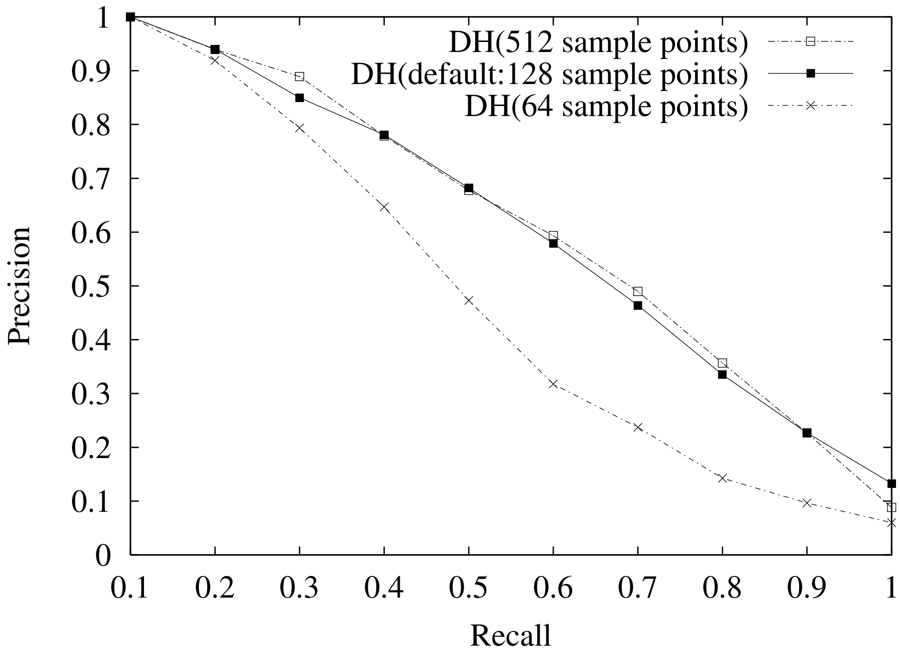


Figure 36: Precision-Recall curves: changing sample points used

Using more information, e.g., more sample points or ranges, in our method will increase its time and storage cost. However, from Table 4.1, we can see that if we use 64 sample points in the method based on Fourier descriptors, but use even 512 sample points in our method, the time cost of our method (0.312 sec.)is still smaller than the method based on Fourier descriptors (0.453 sec.). And from Table 4.5, we can see that compared with the storage used by the method based on Fourier descriptors, the storage used by our method is much smaller - even doubling or tripling the number of ranges in the Distance Histogram. Furthermore, from Figures 4.13 and 4.14, we observed that, using a large database (5000 or 10000 polygons), our method does not perform as well as the method based on Fourier descriptors. To improve our method's effectiveness, we need to use more sample points and more ranges in the Distance Histogram. Figure 4.17 illustrates the improvement in the effectiveness of our method by applying more sample points and ranges, where the database has 5000 polygons.

In Figure 4.17, we use 128 sample points for both our method and the method based on Fourier descriptors, in default conditions. The default number of ranges of Distance Histogram is 30. Under this condition, the effectiveness of our method is less than that of the method based on Fourier descriptors. However, the effectiveness of our method is improved by increasing the sample points and ranges used. Our method can offer similar effectiveness by using 512 sample points and 30 ranges in the Distance Histogram. In this experiment, the polygons in the database have, at most, 30 vertices. Hence, 128 sample points (the number we used as default) may be

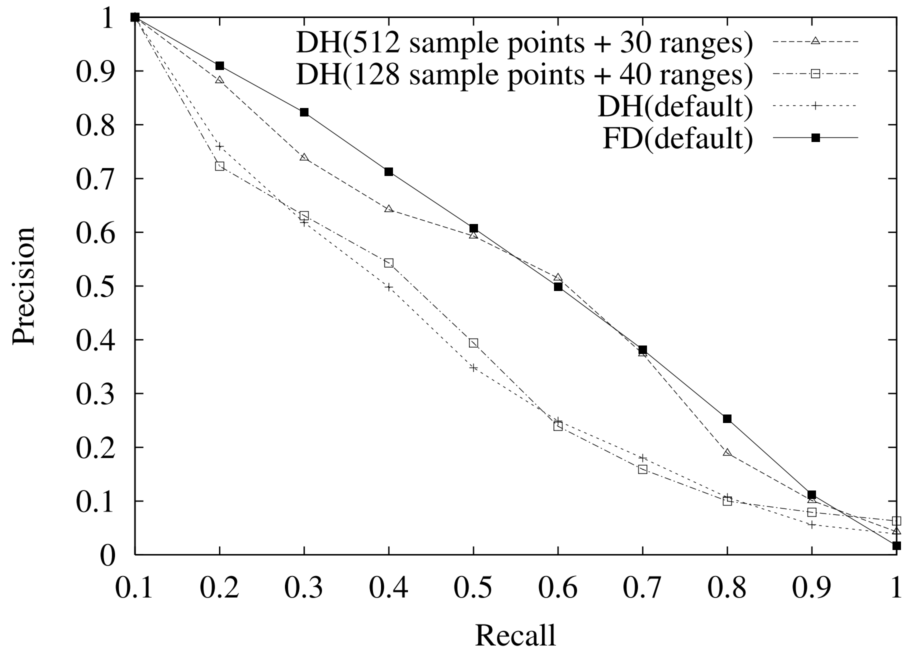


Figure 37: Precision-Recall curves: influence on effectiveness by changing sample points and ranges used in our method, with a database of 5000 polygons

enough for the method based on Fourier descriptors to describe a polygon of this size.

Now we increase the number of vertices of the polygon in the database and make it, at most, 60. We use 512 sample points in our method and still use 128 sample points in the method based on Fourier descriptors. The number of ranges in the Distance Histogram is 50; meanwhile the number of Fourier descriptors used in retrieval is still 64 (default). The number of polygons in the database is still 5000. Figure 4.18 illustrates the Precision-Recall curves of our method and the method based on Fourier descriptors.

In Figure 4.18, we can see that under these conditions, our method produces a better performance than the method based on Fourier descriptors. The processing time for our method, which is about 2.92 seconds, is still smaller (43.1%) than the time for the method based on Fourier descriptors, which is about 5.13 seconds. The storage overhead for both methods is 1105782 bytes (DH) and 13720954 bytes (FD), respectively, where the storage overhead for the method based on Fourier descriptors is approximately 11 times that of ours. We conclude, therefore, that our approach is not only more economical than the method based on Fourier descriptors, but also very flexible. In fact, we could make it as effective as the method based on Fourier descriptors. Further investigation would be needed to determine an optimal configuration for our method on an automatic basis.

4.3 Experiment 2: real images

In previous experiments, we compared our method with the one based on Fourier descriptors using the polygons which we randomly generated. In this experiment, we will apply our method to a more realistic shape database. For consistency, we still compare it with the method based on Fourier descriptors. In the SQUID database [SQU01] there are about 1100 images of marine creatures, described by their shapes. In the SQUID, each shape is represented by from 500-800 integer pairs. These integer pairs are the x and y coordinates of points around the boundary of the shape. They can be treated like the vertices of a polygon (shape of a fish). To facilitate representing and comparing shapes, we define the parameters in our method and the method based on Fourier

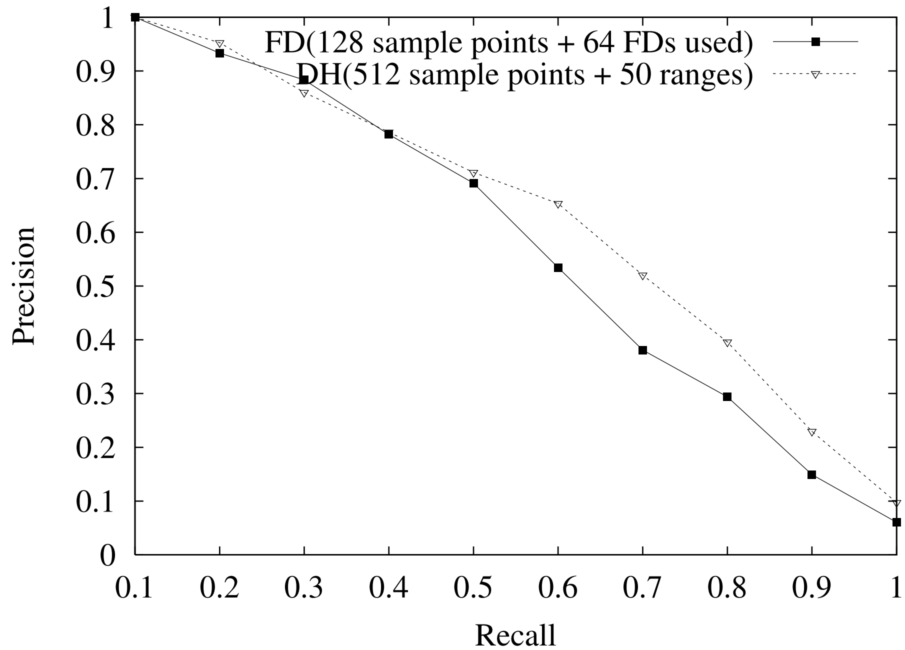


Figure 38: Precision-Recall curves: our method offers better performance than the method based on Fourier descriptors

descriptors as follows:

- Number of sample points selected from the boundary: 2048 (each shape has 500-800 vertices)
- Number of Fourier descriptors used: 64
- Number of ranges in the Distance Histogram: 30

The storage overhead for the next 4 experiments does not change. The storage for our method is about 174900 bytes, while the storage overhead for the method based on Fourier descriptors is about 1482418 bytes, or approximately 8 times ours.

4.3.1 Query 1

First, we choose an ordinary looking fish (illustrated in Figure 4.19) as the query.

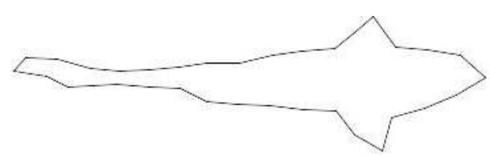


Figure 39: Query 1: an ordinary fish

In Table 4.6, we show the top 10 shapes which are retrieved from the database using our method. Some shapes are scaled for display purposes. The time needed for retrieval is about 0.03 seconds.

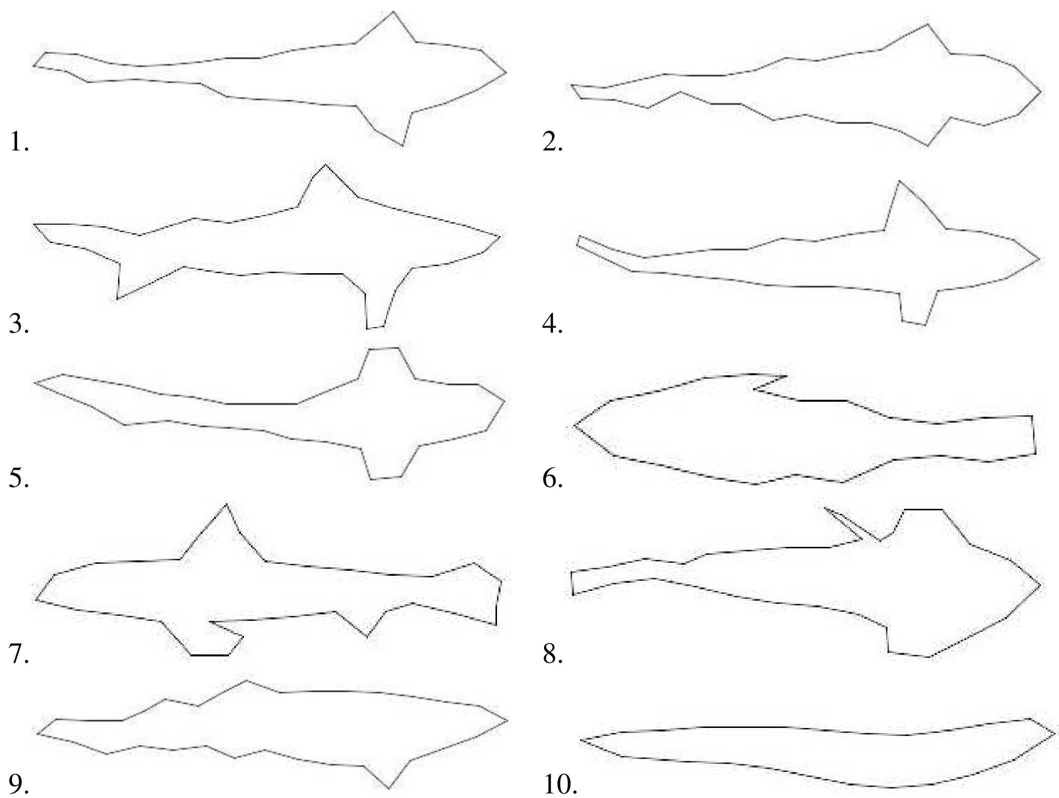


Table 6: Similar images to Figure 4.19 retrieved by the method based on Distance Histogram

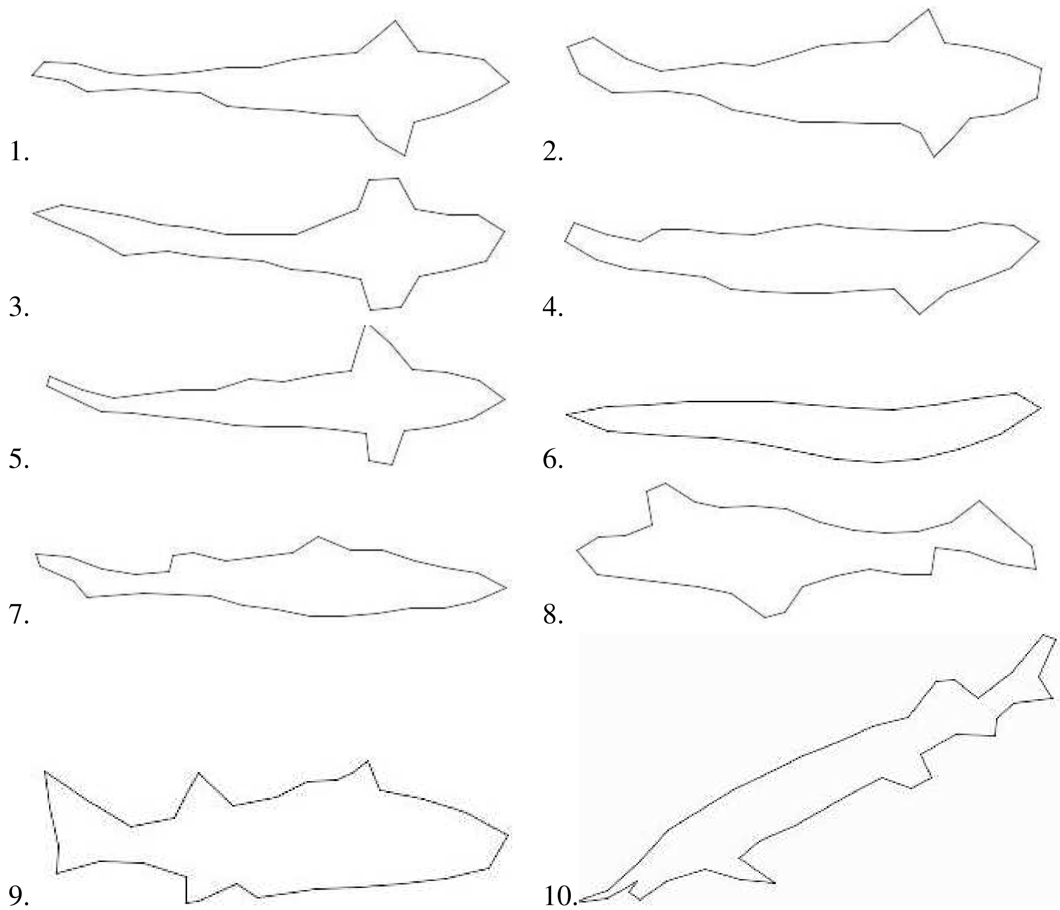


Table 7: Similar images to Figure 4.19 retrieved by the method based on Fourier descriptors

Table 4.7 contains the top 10 shapes retrieved by the method based on Fourier descriptors. The time used in retrieval is about 0.53 seconds (i.e., it is about 17 times more than our approach). From Tables 4.6 and 4.7, we can see that both our method and the method based on Fourier descriptors have retrieved similar shapes to the query when the query is one of an ordinary looking fish. However, the time cost of our method is much less than the method based on Fourier descriptors.

4.3.2 Query 2

In this experiment, we use the shape of the fish in Figure 4.20 as the query, and search for similar shapes in the database.

Figure 40: Query 2: a slim fish

Table 4.8 shows the 10 most similar shapes which were retrieved from the database using our method. The time cost for retrieval is about 0.03 seconds.

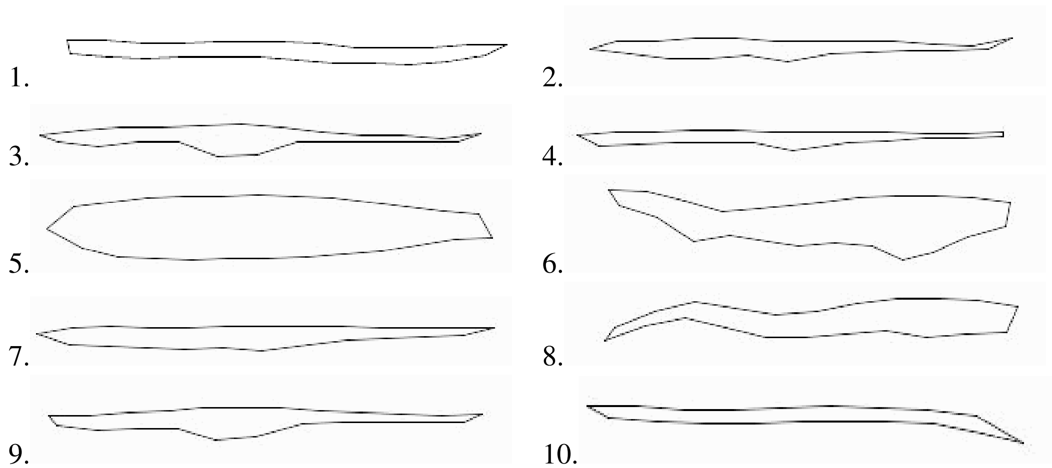


Table 8: Similar images to Figure 4.20 retrieved by the method based on Distance Histogram

In Table 4.9, we have the 10 shapes which were retrieved from the database using the method based on Fourier descriptors. The time used in retrieval is about 0.49 seconds. From Tables 4.8 and 4.9, we conclude that our method performs better in its initial retrieved shapes than the method based on Fourier descriptors.

4.3.3 Query 3

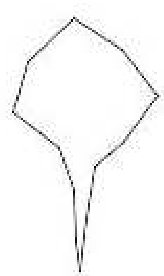


Figure 41: Query 3: a fan-like fish

Figure 4.21 illustrates a shape of a fan-like fish which is used as query 3. Table 4.10 illustrates the top 10 answers in answer set, retrieved by our method from the shape database. The time cost is about 0.04 seconds. Table 4.11 gives the top 10 answers in answer set retrieved by the method based on Fourier descriptors. The time cost is about 0.51 seconds. This time, the answers achieved by our method, and those obtained by the method based on Fourier descriptors (illustrated in Tables 4.10 and 4.11), show no apparent difference.

4.3.4 Query 4

The shape in Figure 4.22 is the shape of a circle-like fish. We use this shape as the query in this section.

In Table 4.12, we show 10 answers retrieved from the shape database. The time cost is about 0.03 seconds. In this retrieval, we obtained a clear "false hit" indexed by the number 7. Even though the shapes are quite different, in our method they may generate similar Distance Histograms. For example in Figure 4.23, the 2 shapes are apparently different; however, their Distance Histograms are quite similar.

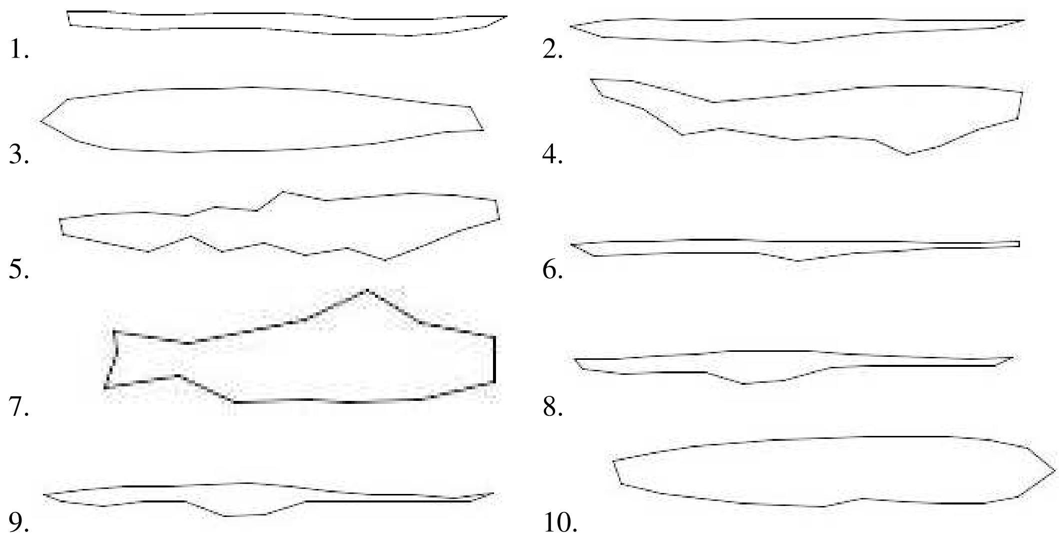


Table 9: Similar images to Figure 4.20 retrieved by the method based on Fourier descriptors

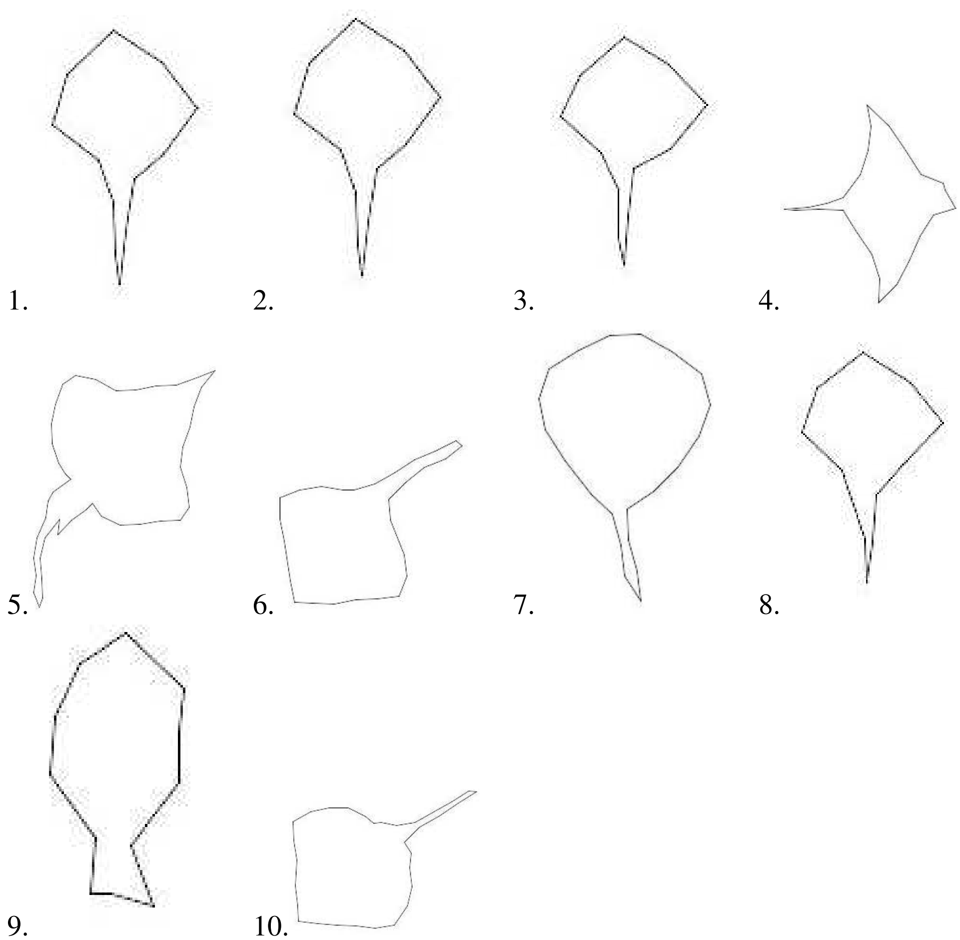


Table 10: Similar images to Figure 4.21 retrieved by the method based on Distance Histogram

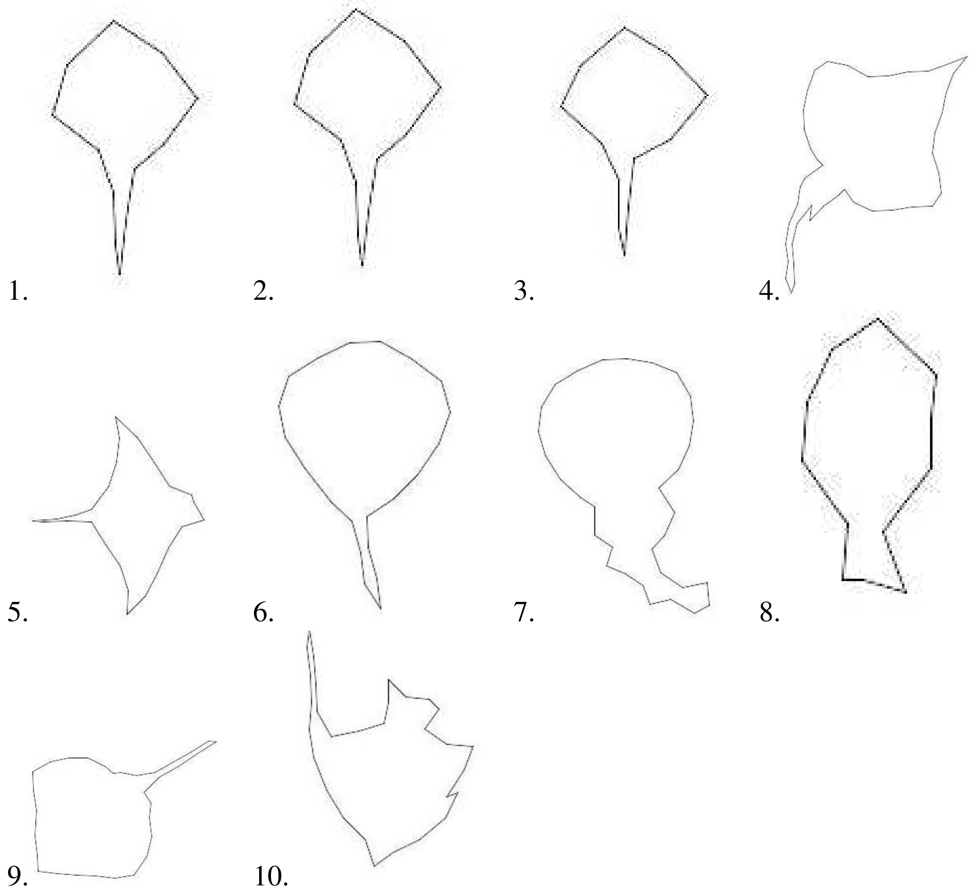


Table 11: Similar images to Figure 4.21 retrieved by the method based on Fourier descriptors

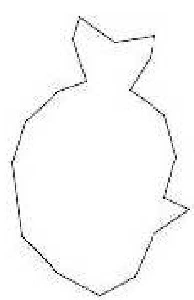


Figure 42: Query 4: a circle-like fish

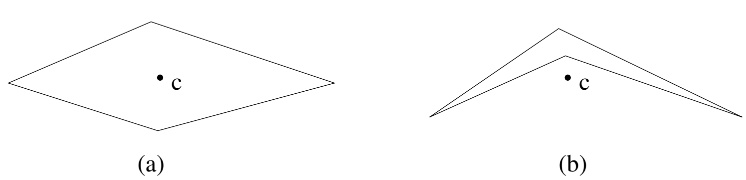


Figure 43: Two different shapes with similar Distance Histograms

To avoid this, we could consider introducing negative distance in constructing the Distance Histogram. If the centroid is inside the polygon, the value of the distance is positive, just as before. On the other hand, if the centroid is outside the polygon, the value of the distance will be defined as negative. The abstract value is equal to the value of real distance. After this modification, our method may be better at representing concave shapes than it was. This enhancement to our method will be dealt with in future work.

Table 4.13 illustrates the answers retrieved by the method based on Fourier descriptors. The time cost is about 0.52 seconds.

4.4 Conclusion

In the previous several sections, we have presented the details of our experiments. Based on the results, we can draw the following conclusions about our method for shape representation and retrieval:

- Compared with the method based on Fourier descriptors, under some situations our method has an inferior retrieval performance when we use the same amount of information.
- Our method can save considerable time and storage cost.
- Based on the advantages of our method, if our method is able to utilize more storage and time to conduct the retrieval - which means using more information (e.g., sample points and ranges in histogram) - the performance of our method can be superior, and still consume less time and space than the method based on Fourier descriptors.
- When using a database of real shapes, our proposed technique seems to offer reasonable performance and relatively good effectiveness.

5 Conclusion & Future work

5.1 Conclusion

The goal of finding a simple and straightforward method for shape representation, with relatively good efficiency, was the motivation for our research. As we have discussed, the existing approaches to shape representation, (the method based on Fourier descriptors, grid-based method, etc.), have their own special advantages; however, they have limitations as well. Therefore, we proposed a novel idea for shape representation: a method based on the Distance Histograms. In our approach, we represent shapes by using their Distance Histograms. A Distance Histogram represents the distribution of radii lengths from sample boundary points to the shape's centroid, and provides the main data for us to represent, store, compare, and index shapes. Compared with a classic approach to shape representation - a method based on Fourier descriptors - our approach has these features:

- It is much more economical in its use of storage overhead and query processing time during the retrieval procedure.
- It is highly flexible and may be adapted to be as efficient as the method based on Fourier descriptors.

The experiments we carried out to evaluate our approach were divided into two parts: first we used a synthetic polygon database. In this part we designated the Precision-Recall curve, average precision, time, and storage cost as the criteria for evaluation. In the second part, we applied our approach to a database with 1,100 real shapes of marine creatures. Again, our approach seemed to give good results at a lower cost (when compared to the method based on Fourier descriptors).

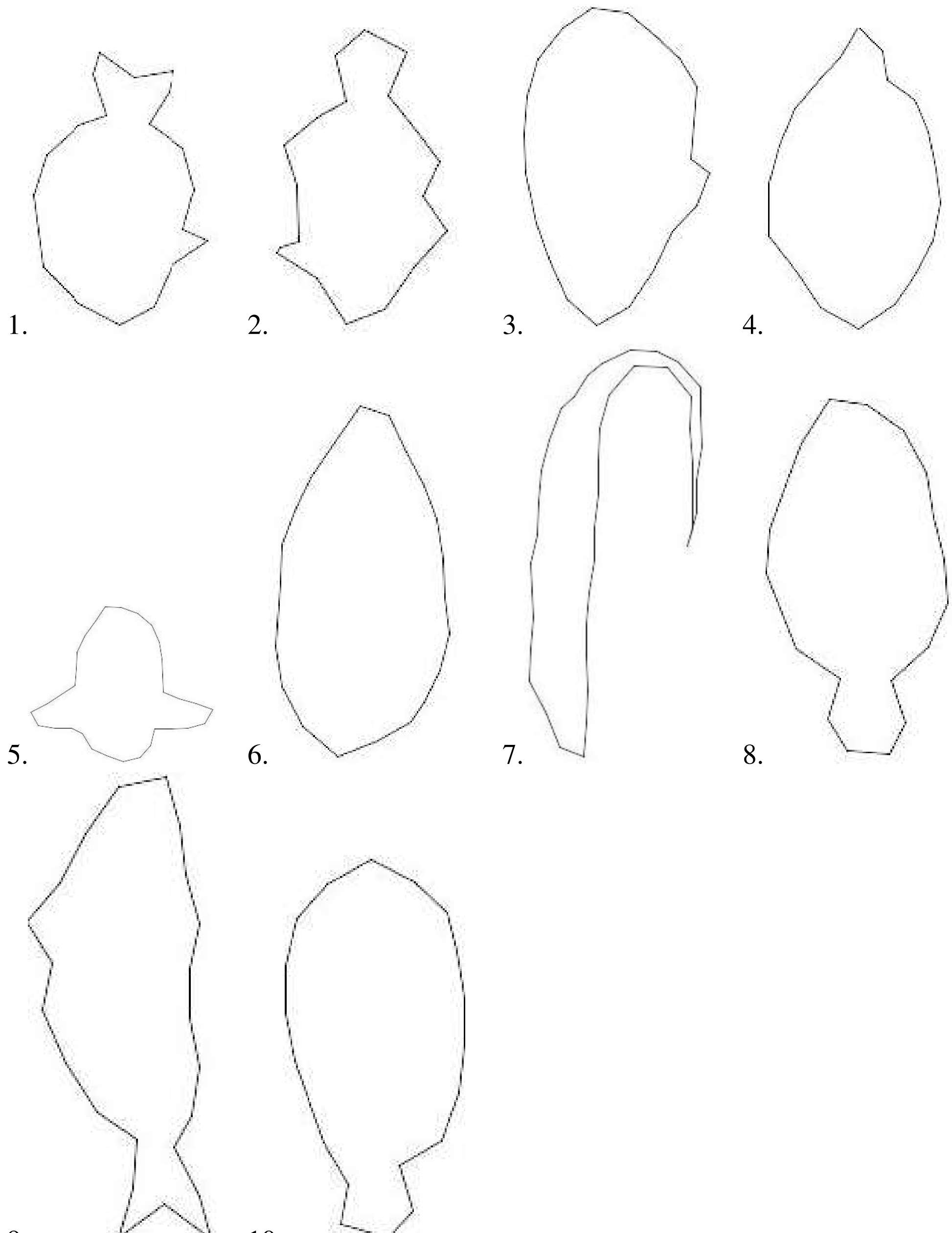


Table 12: Similar images to Figure 4.22 retrieved by the method based on Distance Histogram

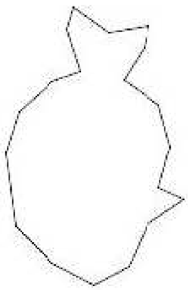
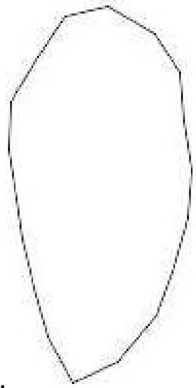
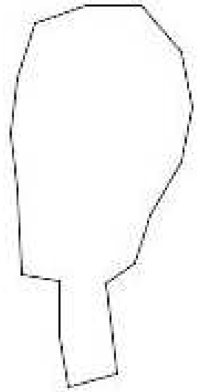
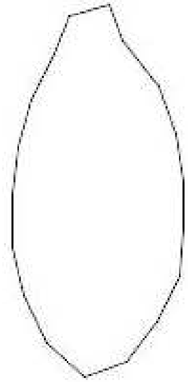
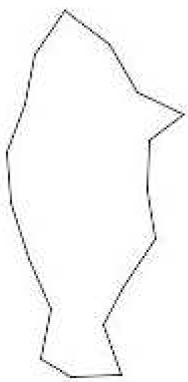
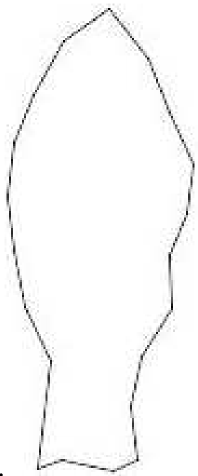
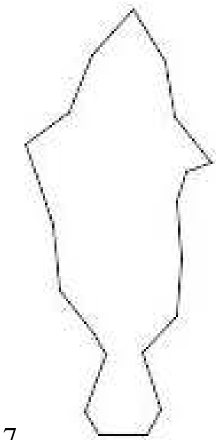
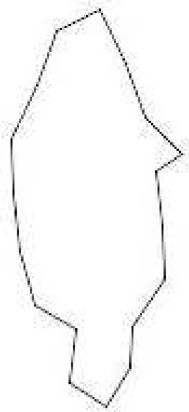
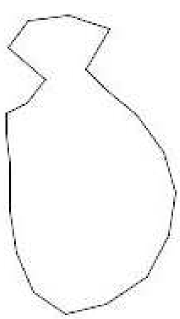
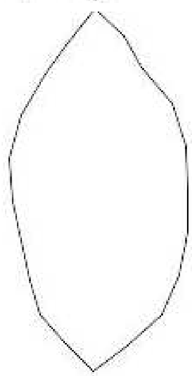
1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 

Table 13: Similar images to Figure 4.22 retrieved by the method based on Fourier descriptors

5.2 Future work

In this thesis, we proposed a novel approach to shape representation, and carried out a number of experiments which yielded positive results from the use of our method. However, we think there is still some space for us to improve our method. We would suggest the following enhancements for future work:

- In our method, the distance calculated, based on centroid and radius of shape, is always a positive value. However, we can apply negative value in distance calculation. The definition can be: if the centroid is outside the shape's boundary, the distances will be negative; if the centroid is inside the shape's boundary (including the centroid in the boundary), the distances will be positive. In this way, we may be able to distinguish convex and concave polygons better.

- Currently our approach can only compare and retrieve the shape of a single object. In the next step, we wish to extend our approach to handle shapes of multiple objects. If these objects are connected together, we can treat them as a single object and calculate the boundary for representing their shapes. If they are isolated from each other, we can add several lines to connect the centroids of each object, and then treat them as a single object.

- In terms of processing speedup, our technique can readily make use of high-dimensional indexing structures, e.g., the A-tree [Sa00] and SR-tree [KS97], since n-bin histograms can be naturally mapped into the n-dimensional space.

- Finally, we consider that another venue for future work would be to further reduce the space overhead of Distance Histograms by using the approach presented in [Ch01]. In that research, a color histogram is encoded as a much smaller bit-string. Furthermore, an indexing technique, namely the S-tree [TNM00], could also be applied to index such a bit-string, speeding up query processing as well.

References

[ACH+91] E. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, and J. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209-215, 1991.

[AFS93] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *Proceedings of the 4th International Conference on Foundations of Data Organizations and Algorithms (FODO '93)*, pages 69-84, Chicago, IL, USA, October 1993.

[AN00] A. Aboulnaga, and J.F. Naughton. Accurate estimation of the cost of spatial selections. *16th IEEE International Conference on Data Engineering*, pages 123-134, 2001.

[BBP00] S. Berretti, A.D. Bimbo, and P. Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on Multimedia*, 2(4):225-240, December 2000.

[BG80] E. Bribiesca and A. Guzman. How to describe pure form and how to measure differences in shape using shape numbers. *Pattern Recognition*, 12(2):101-112, 1980.

[BKK96] S. Berchtold, D. Keim, and H.P. Kriegel. The X-tree: an index structure for high dimensional data. In *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB '96)*, pages 28-39, Mumbai, India, August 1996.

[BKK97] S. Berchtold, D.A. Keim, and H.P. Kriegel. Using extended feature object for partial similarity retrieval. *VLDB Journal*, 6(4):333-348, December 1997.

[BKSS90] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger. The R* tree: an efficient and robust index method for points and rectangles. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '90)*, pages 322-331, Atlantic City, NJ, USA, May 1990.

[BP97] A.D. Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Transaction on Pattern Analysis Machine Intelligence*, Vol. 19, pages 121-132, Febuary 1997.

[Bri81] E. Bribiesca. Arithmetic operations among shapes using shape numbers. *Pattern Recognition*, 13(2):123-138, 1981.

[BSA91] S.O. Belkasim, M. Shridhar, and M. Ahmadi. Pattern recognition with invariants: a comprehensive study and new results. *Pattern Recognition*, 24:1117-1138, 1991.

[Ch99] I.L. Cheng. Image databases: A content-based type system and query by similarity match. Masters thesis, Department of Computing Science, University of Alberta. Available as Technical Report TR-99-03, 1999.

[Ch01] V. Chitkara. Color-based image retrieval using compact binary signatures. Masters thesis, Department of Computing Science, University of Alberta, Summer 2001.

[CIT94] W. Chu, I. Ieong, and R. Taira. A semantic modeling approach for image retrieval by content. *VLDB Journal*, 3(4):445-477, 1994.

[CJ90] S.K. Chang and E. Jungert. Pictorial data management based upon the theory of symbolic projection. *Journal of Visual Languages and Computing*, 2(3):195-215, 1990.

[CPZ97] P. Ciaccia, M. Patella, and P. Zezula. M-tree: an efficient access method for similarity search in metric spaces. In *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB '97)*, pages 426-435, Athens, Greece, 1997.

[CSW95] S.F. Chang, J.R. Smith, and H. Wang. Automatic feature extraction and indexing for content-based visual query. *Technical Report CU/CTR*, 414-95-20, Columbia University, 1995.

[Efg01] *http://www.efg2.com/Lab/Graphics/PolygonArea.htm*, June 2001.

[EG99] J.P. Eakins and M.E. Graham. Content-based image retrieval A report to the JISC technology applications programme. *http://www.unn.ac.uk/iidr/research/cbir/report.html#Heading2*, January 1999.

[Exp97] Tenth Planet Explorations, Inc. Spatial relationships. Half Moon Bay, Calif.: Tenth Planet Explorations, Inc., c1997.

[FBF+94] C. Faloutsos, R. Barber, M. Flickner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information System*, 3(3/4):231-262, July 1994.

[FFN+93] C. Faloutsos, M. Flickner, W. Niblack, D. Petkovic, W. Equitz, and R. Barber. The Qbic Project: Efficient and effective querying by image content. IBM Res. Div. Almaden Res. Center, Res. Rep. 9453, August 1993.

[GCG96] Y. Gong, C.H. Chuan, and X. Guo. Image indexing and retrieval based on color histograms. *Multimedia Tools and Applications*, 2(2):133-156, 1996.

[GG98] V. Gaede and O. Gunther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170-231, June 1998.

[GM90] W.I. Grosky and R. Mehrotra, Index-based object recognition in pictorial data management. *Comput. Vis. Graph. Image Process.*, Vol.52, pages 416-436, 1990.

[GNM92] W.I. Grosdy, P. Neo, and R. Mehrotra. A pictorial index mechanism for model-based matching. *Data Knowledge Engineering*, vol. 8, pages 309-327, 1992.

[GR96] V.N. Gudivada and V.V. Raghavan. Content-based image retrieval systems. *Computer*, September:18-22, 1996.

[GSF77] T. Gonzalez, S. Sahni, and W.A. Franta. An efficient algorithm for the Kolmogorov-Smirnov and Lilliefors test. *ACM Trans. Math. Software*, Vol.3, pages 60-64, 1977.

[GSP95] R.C. Gonzalez, T. Seppanen, and M. Pietikainen. An experimental comparison of aoutoregressive fourier-based descriptors in 2D shape classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 201-207, 1995.

[Gut84] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '84)*, page 47-57, Boston, June 1984.

[GW92] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison Wesley, 1992.

[HCP95] W. Hsu, T.S. Chua, and H.K. Pung. An integrated color-spatial approach to content-based image retrieval. In *Proceedings of ACM Multimedia*, pages 305-313, 1995.

[HMK00] S. Hatipoglu, S.K. Mitra, and N. Kingsbury, Image texture description using complex wavelet transform. in *Proceedings of IEEE International Conference on Image Processing*, vol.2 pp.530-533, Vancouver, B.C., Canada, September 2000.

[Hu62] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179-187, 1962.

[Hul94] J.J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550-554, 1994.

[Iss96] *http://www.issco.unige.ch/ewg95/node216.html*, 1996.

[IV96] G. Iannizzotto and L. Vita. A new shape distance for content based image retrieval. In *Proceedings of the 22nd VLDB Conference*, pages 371-386, 1996.

[Jag91] H.V. Jagadish. A retrieval technique for similar shapes. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '91)*, pages 208-217, Denver, May 1991.

[JMM95] H.V. Jagadish, A.O. Mendelzon, and T. Milo. Similarity-based queries. In *Proceedings of the 14th ACM SIGACT-SIGART Symposium on Principles of Database Systems (PODS '95)*, pages 36-45, San Jose, May 1995.

[KCH95] P.M. Kelly, T.M. Cannon, and D.R. Hush. Query by image example: the CANDID approach. In *SPIE Storage and Retrieval for Image and Video Databases III*, pages 238-248, 1995.

[KS97] N. Katayama and S. Satoh. The SR-tree: an index structure for high-dimensional nearest neighbor queries. *Proceedings of ACM SIGMOD' 97 International Conference*, pages 369-380, 1997.

[KSF+96] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas. Fast nearest neighbor search in medical image databses. In *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB '96)*, pages 215-226, Mumbai, India, 1996.

[KSP95] H. Kauppinen, T. Seppanen, and M. Pietikainen. An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):201-207, 1995.

[LCH+97] W.S. Li, K.S. Candan, K. Hirata, and Y. Hara. SEMCOG: an object-based image retrieval system and its visual query language. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 521-524, Tucson, Arizona, May 1997.

[LJF94] K.I. Lin, H.V. Jagadish, and C. Faloutsos. The TV-tree-an index structure for high=dimensional data. *VLDB Journal*, 3(4):517-542, 1994.

[LOSO97] J.Z. Li, M.T. Ozsu, D. Szafron, and V. Oria. MOQL: a multimendia object query language. In *Proceedings of the 3rd International Workshop on Multimedia Informations Systems*, pages 19-28, September 1997.

[LS90] D.B. Lomet, and B. Salzberg. The hb-tree: a multiattribute indexing method with good guaranteed performance. *ACM Transactions on Database Systems*, 15(4):625-658, December 1990.

[Lu96] G. Lu. Image retrieval based on shape. In *Proceedings of the International Conference on Visual Information Systems*, Melbourne, Australia, 1996.

[MG93] R. Mehrotra, and J.E. Gary. Feature-based retrieval of similar shapes. In *Proceedings of 9th International Conference on Data Engineering (ICDE '93)*, pages 108-115, Vienna, Austria, 1993.

[MG95] R. Mehrotra, and J.E. Gary. Similar-shape retrieval in shape data management. *IEEE Computer*, 28(9):57-62, 1995.

[MM86] F. Mokhtarian, and A. Mackworth. A scale-based description and recognition of planar curves and two dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):34-43, 1986.

[MM96] B. Manjunath, and W. Ma. Texture features for browsing and retrieval of image data. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):37-42, August 1996.

[MP96] T. Minka, and R. Picard. Interactive learning using a 'society of models'. In *Proceedings of the computer vision and pattern recognition (CVPR) 96*, pages 447-452, 1996.

[NBE93] W. Niblack, R. Barber, and W. Equitz. The QBIC project: Querying images by content using color, texture, and shape. In *SPIE Proceedings of Storage and Retrieval for Image and Video Databases*, pages 173-187, 1993.

[NHS84] J. Nievergelt, H. Hinterberger, and K.C. Sevcik. The grid file: an adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems*, 9(1):38-71, March 1984.

[PF77] E. Persoon, and K.S. Fu. Shape discimination using Fourier Descriptors. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-7(2):170-179, March 1977.

[Pho01] *http://trevor.www.media.mit.edu/ tpminka/photobook/*, 2001.

[PPS94] A.P. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *Storage and Retrieval for Image and Video Databases*, pages 34-47, 1994.

[PTVF92] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

[Qbi01] *http://wwwqbic.almaden.ibm.com/stage/detailed.html*, 2001.

[Raf98] D. Rafiei. *Fourier-transform based techniques in efficient retrieval of similar time sequences*. PhD thesis, University of Toronto, 1998.

[Raf99] D. Rafiei. On similarity-based queries for time series data. In *Proceedings of the 15th International Conference on Data Engineering (ICDE '99)*, Sydney, March 1999.

[RH74] C.W. Richard, and H. Hemami. Identification of three-dimensional objects using Fourier descriptors of the boundary curve. *IEEE Transactions on Systems, Man, Cybern.*, SMC-4:371-378, July 1974.

[RM97] D. Rafiei, and A. Mendelzon. Similarity-based queries for time series data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '97)*, pages 13-24, Tucson, Arizona, May 1997.

[RM98] D. Rafiei, and A. Mendelzon. Efficient retrieval of similar time sequences using DFT. In *Proceedings of the 5th International Conference on Foundations of Data Organizations and Algorithms (FODO '98)*, pages 249-257, Kobe, Japan, November 1998.

[RM99] D. Rafiei, and A. Mendelzon. Efficient Retrieval of Similar Shapes. The VLDB Journal. To appear.

[RSH96] Y. Rui, A. She, and T. Huang. Modified fourier descriptors for shape representation - a practical approach. In *Proceedings of the First International Workshop on Image Databases and Multi-Media Search*, 1996.

[SAF94] B. Scassellati, S. Alexopoulos, and M.D. Flickner. Retrieving images by 2D shape: a comparison of computation methods with human perceptual judgements. In *Proceeding of SPIE Storage and Retrieval for Image and Video Databases*, pages 2-9, 1994.

[Sa00] Y. Sakurai. The A-tree: an index structure for high-dimensional spaces using relative approximation. *Proceeding of 26th International Conference on Very Large Data Bases*, pages 516-526, 2000.

[SC95] J. Smith, and S. Chang. Automated image retrieval using color and texture. Technical report CU/CTR 408-95-14, Columbia University, July 1995.

[SC96] J.R. Smith, and S.F. Chang. VisualSEEK: a fully automated content-based image query system. *ACM Multimedia*, pages 87-98, November, 1996.

[SH91] M.J. Swain, and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11-32, 1991.

[SJ82] S. Smith, and A. Jain. Chord distribution for shape matching. *Computer Graphics and Image Processing*, Vol.20, pages 259-271, 1982.

[SL97] A. Sajjanhar, and G. Lu. A grid based shape indexing and retrieval methoed. *Special Issue of Australian Computer Journal on Multimedia Storage and Archiving Systems*, 29(4):131-140, November 1997.

[SQU01] F. Mokhtarian. *http://www.ee.surrey.ac.uk/Research/VSSP/ imagedb/demo.html*, March 2001.

[TNM00] E. Tousidou, A. Nanopoulos, and Y. Manolopoulos. Improved methods for signature-tree construction. *The Computer Journal*, 43(4):301-314, 2000.

[TT01] K-L. Tan, and L.F. Thiang. Retrieving similar shapes effectively and efficiently. *Multimedia Tools and Applications*, Kluwer Academic Publishers, the Netherlands. (Accepted for publication in 2001).

[VSE01] *http://www.ctr.columbia.edu/VisualSEEK*, 2001.

[Wa01] S. Wang. A robust CBIR approach using local color histograms. Masters thesis. Department of Computing Science, University of Alberta, 2001.

[Xu00] B. Xu. A visual query facility for DISIMA image database management system. Masters thesis. Department of Computing Science, University of Alberta, 2000.

[YA94] L. Yang and F. Albregtsen. Fast computation of invariant geometric moments: A new method giving correct results. In *Proceeding of the International Conference on Pattern Recognition (ICPR) 94*, pages A:201-204, 1994.

[You99] P. Young. Physics 114A Parseval's Theorem. *http://bartok.ucsc.edu/peter/114A/parseval/parseval.html*, May 12, 1999.

[ZR72] C.T. Zahn, and R.Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C-21(3):269-281, 1972.