# SHAPE ANALYSIS USING CONTOUR-BASED AND REGION-BASED APPROACHES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜNCE ÇİFTÇİ

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Mübeccel Demirekler
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen
Supervisor

Examining Committee Members

Assoc. Prof. Dr. A. Aydın Alatan (Chairman)

Prof. Dr. İsmet Erkmen

Assoc. Prof. Dr. Gözde Bozdağı Akar

Assoc. Prof. Dr. Aydan Erkmen

Assoc. Prof. Dr. Volkan Atalay

# ABSTRACT

SHAPE ANALYSIS USING CONTOUR-BASED AND REGION-BASED APPROACHES

ÇİFTÇİ, GÜNCE

MSc, Department of Electrical And Electronics Engineering

Supervisor: Prof. Dr. İsmet Erkmen

December 2003, 71 pages

The user of an image database often wishes to retrieve all images similar to the one (s)he already has. In this thesis, shape analysis methods for retrieving shape are investigated. Shape analysis methods can be classified in two groups as contour-based and region-based according to the shape information used. In such a classification, curvature scale space (CSS) representation and angular radial transform (ART) are promising methods for shape similarity retrieval respectively. The CSS representation operates by decomposing the shape contour into convex and concave sections. CSS descriptor is extracted by using the curvature zero-crossings behaviour of the shape boundary while smoothing the boundary with Gaussian filter. The ART descriptor decomposes the shape region into a number of orthogonal 2-D basis functions defined on a unit disk. ART descriptor is extracted using the magnitudes of ART coefficients. These methods are implemented for similarity comparison of binary images and the retrieval performances of descriptors for changing number of sampling points of boundary and order of ART coefficients are investigated. The experiments are done using 1000 images from MPEG7 Core Experiments Shape-1. Results show that for different classes of shape, different descriptors are more successful. When the choice of approach depends on the properties of the query shape, similarity retrieval performance increases.

Keywords: Scene Analysis, Shape Representation, Curvature Scale Space, Angular Radial Transform

# ÖZ

ÇEVRİT VE BÖLGE TEMELLİ YAKLAŞIMLARLA ŞEKİL ANALİZİ

ÇİFTÇİ, GÜNCE

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. İsmet Erkmen

Aralık 2003, 71 sayfa

Bir görüntü veritabanı kullanıcısı sıklıkla elindeki görüntüye benzeyen görüntülere erişmek ister. Bu tezde, şekil erişiminde kullanmak için şekil analiz yöntemleri araştırılmıştır. Şekil analizi yöntemleri, kullanılan şekil bilgisine göre, çevrit ve bölge temelli olarak iki grupta sınıflandırılabilir. Şekil erişimi için, kıvrım ölçek uzayı (KÖU) gösterimi ve açısal radyal dönüşüm (ARD) bu sınıflardaki başarılı yöntemlerdir. KÖU gösterimi şekli dışbükey ve içbükey bölümlere ayırır. KÖU betimleyicisi, şeklin sınırlarını Gauss filtresiyle düzleştirirken sınırın kıvrım davranışını kullanarak çıkartılır. ARD, şekli birim dairede tanımlı iki boyutlu dikgen taban fonksiyonlarına ayırır. ARD betimleyicisi, ARD katsayılarının büyüklüklerinden çıkartılır. Bu yöntemler ikili imgelerin benzerlik karşılaştırma problemine uygulanmış ve betimleyicilerin, değişen sınır noktaları örnekleme sayısı ve ARD katsayı derecelerine göre, performansları araştırılmıştır. MPEG7 veri setiyle yapılan bu deneylerde farklı şekil grupları için farklı betimleyicilerin daha başarılı olduğu görülmüştür. Çevrit ya da bölge temelli yaklaşım seçimi, benzeri aranan şeklin özelliklerine dayandırıldığında, benzerlik bulma performansı artmaktadır.

Anahtar Kelimeler: Sahne analizi, Şekil Gösterimi, Kıvrım Ölçek Uzayı, Açısal Radyal Dönüşüm

# ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor Prof. Dr. İsmet Erkmen for his encouragement and motivation during my studies. I also thank to Assoc. Prof. A. Aydın Alatan for introducing me the thesis topic and his useful suggestions. I am grateful to Yiğit Akkök for his interest and support in several stages of master thesis. Special thanks to Evren İmre for the useful discussions. Finally, I would like to thank to Ahmet Öztürk who has been great help during thesis writing.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| ART | : | Angular Radial Transform |
| ARTD | : | Angular Radial Transform Descriptor |
| CBIR | : | Content Based Image Retrieval |
| CSS | : | Curvature Scale Space |
| CSSD | : | Curvature Scale Space Descriptor |
| LUT | : | Look Up Table |
| MAT | : | Medial Axis Transform |
| CE-1 | : | Moving Pictures Experts Group 7 Core Experiments data set |

# CHAPTER 1

# INTRODUCTION

There has been a rapid increase on the amount of digital images around the world, due to the decreasing storage and processing costs and the Internet. Terabytes of data are being generated in the form of aerial imagery, survelliance images, fingerprints, engineering drawings, medical images, trademarks and logos, images from sports events, graphic illustrations, entertainment industry photos and videos. The increasing number of images in many applications gives rise to the difficult problem of organizing them for the best and rapid access to their information content.

Traditional text based indexing mechanisms has limitations. One is that, when the database is large, it is almost impossible to manually annotate all the images. The other is, some visual properties of images such as texture and shape are difficult to describe using words. Therefore, instead of words , researchers in computer vision have tried to use some convenient feature vectors to describe these properties. The objective is to allow users to specify an image or a part of it and the system should then find all images in the database similar to that image. Among the visual properties of an image such as shape, color and texture, shape features are very powerful when used in similarity search and retrieval. This is because the shape of objects are strongly linked to their identity and functionality: humans can recognize objects only from their shapes. This property distinguishes shape from the other elementary visual features. Therefore, shape similarity search has the potential of being the most effective search technique in many application fields.

This study addresses the problem of similar shape retrieval. The similar shape retrieval problem can be stated as retrieve or select all shapes or images that are visually similar to the query shape. The first step of designing a shape retrieval technique is shape representation. In shape representation, the objective is to find a representation that captures essential shape properties. After that, shape feature vectors are extracted from the chosen shape representation and these descriptors are used for image similarity measurement and indexing. Shape representation methods are generally classified as *contour-based* and *region-based* methods. In this study, most promising shape descriptors for each class, curvature scale space representation and angular radial transform are implemented and compared in order to observe their retrieval performances.

The following pages of this thesis are organized in six chapters. In Chapter 2, state of art shape representation and description techniques will be reviewed. In Chapters 3 and 4, the reader will find detailed presentations of curvature scale space and angular radial transform shape descriptors. In each chapter, it will be shown that both descriptors have desired properties which make them promising for shape retrieval. In Chapter 5, retrieval performances of the descriptors will be evaluated and compared by use of a developed software for shape retrieval of binary shapes. Chapter 6 summarizes the conclusions of this thesis and gives some future directions.

# CHAPTER 2

# REVIEW OF SHAPE REPRESENTATION AND DESCRIPTION

## 2.1 Introduction

Shape is an important visual feature and it is one of the primary features for image content description. However, shape description is a difficult task. Because it is difficult to define significant shape features and measure the similarity between shapes. Moreover, shape is often corrupted with noise, defection and occlusion.

Shape research is an active area for over 30 years. In the past, shape research was driven by applications of object recognition. Therefore, shape representation and description methods used targeted particular applications. Effectiveness or accuracy of methods was very important in these techniques. *Content Based Image Retrieval* (CBIR) is newly coming up multimedia application. CBIR systems are system which automatically retrieve images from an image database by color, texture, shape and text features. With this application, for the representation and description of shape , efficiency becomes equally important as effectiveness for online retrieval.

In this chapter, shape representation and description techniques in literature are reviewed. Several criteria for the evaluation of shape description techniques are listed. Promising shape descriptors for use in image retrieval are identified.

## 2.2 Classifications of Shape Representation Techniques

Shape analysis methods can be classified according to different criteria. Pavlidis [1] has proposed the following classifications. The first classification is based on the use of shape boundary points or interior of the shape. This commonly used method classifies shape representations as *contour-based and region-based* representations. Fourier transform of the boundary is an example of contour-based method. Some examples of region-based methods are moment based approaches and the medial axis transform (MAT).

Another classification of shape analysis algorithms can be made according to the result of the analysis, numeric or non-numeric. For example, MAT produces another image (a symmetric axis) for representation of a shape which is a non-numeric descriptor . Fourier and moment based methods produce numbers (scalar or vectors).

A third classification of shape analysis can be made according to information preservation property of the method. Methods which allow the accurate reconstruction of a shape from its descriptor are called *information preserving*, while methods only capable of partial reconstruction or ambiguous description are called *non-information preserving.* For example, area to perimeter square ratio is a non-information preserving description because many different shapes may have the same ratio therefore it is not possible to reconstruct the shape from its area to perimeter square ratio.

Using a hierarchical classification, first it is possible to classify the variety of shape representations contour-based method and region-based methods. Under each class, the different methods are distinguished as global and structural based on whether the shape is represented as a whole or represented by sub-parts (primitives).

The whole hierarchy of the classification is shown in Figure 2.1

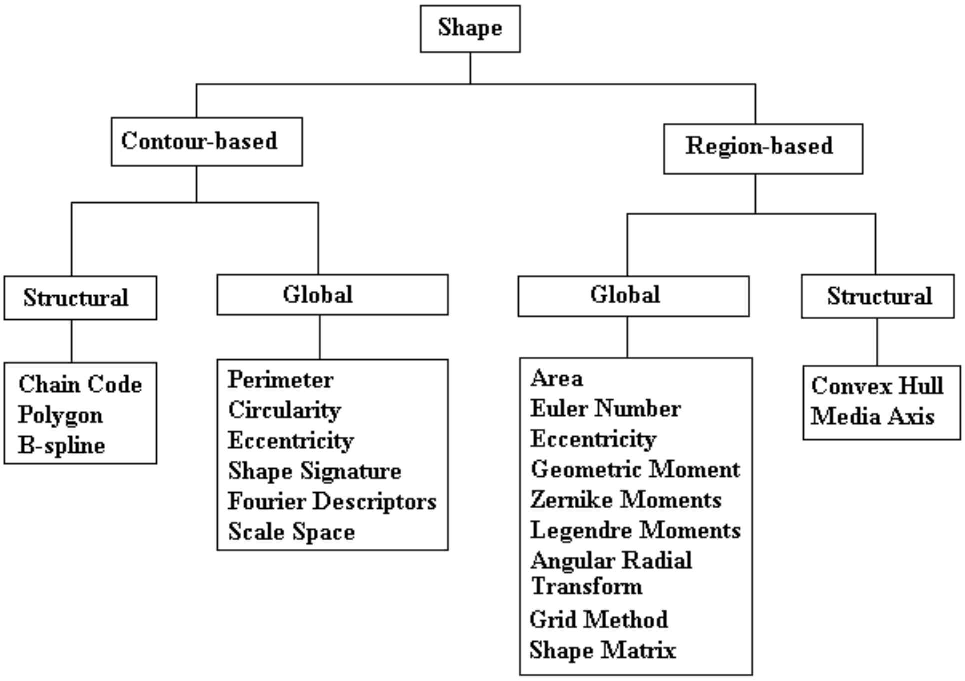


Figure 2.1: Classification of shape representation and description methods

## 2.3 Shape Description Methods Evaluation Criteria

A common discussion in shape description research is how to judge the quality of shape descripton method. All methods are not suitable for every kind of shape and application. In fact, the method chosen among shape representation methods should depend on the properties of the shape to be described and the particular application. For example, the presence of noise which is a common problem in segmented images can affect the choice of the shape representation method, such a representation should not change much with noisy images.

The essential property of a shape representation is the invariance of the representation with the transformations of the object like scaling, rotation and translation. Because such transformations do not change the shape of an object.

Several authors proposed criteria for the evaluation of the quality of a shape descriptor. A number of criteria have been proposed in [2, 3, 4] to define a representation as reliable, they are:

- Uniqueness: The representation should uniquely specify a single shape.If there are two different shapes, their representations also should be different. It will therefore be possible to say that two objects have the same shape by observing that they have the same representation.

- Invariance: If two objects have the same shape, they should also have the same representation. Since transformations like uniform scaling, rotation and translation do not change the shape of an object, a good shape descriptor should be invariant under uniform scaling, rotation and translation.

- Stability: Stability describes how sensitive a shape descriptor is to small changes in shape. If two objects have a small shape difference, their representations should also have a small difference. A small change in some part of a shape should create a small local change in the representation.

- Efficiency: For a representation to be suitable for practical recognition tasks, the representation should be computable efficient and storable. It is also called as accessibility which describes how easy or difficult it is to compute a shape descriptor in terms of memory requirements and computational time.

- It may be useful to determine some properties of the shape using its representation. For example, if a curve has a symmetric shape, it may be desirable to be able to determine that fact from its representation (the symmetry criteration). Moreover, if the shape of a part of an object is the same as the shape of part of another object, it may be useful to be able to determine this relationship using their representations (the part/whole criteration).

The methods mentioned above define desired properties of a good shape descriptor conceptually but the quality of a shape descriptor can not be numerically expressed. So it is not possible to make an exact numerical comparison of shape description methods. Moreover, shape representation methods proposed in computational vision may also fail to satisfy one or more of the criteration above. At the same time, these methods can be suitable for special-purpose shape representation tasks. Therefore, the specific application puts the criteria for the evaluation of a shape descriptor. For example, in the development of MPEG-7 which aims to standardize CBIR, six requirements has been set to measure a shape descriptor, they are: good retrieval accuracy, compact features, general application, low computation complexity, robust retrieval performance and hierarchical coarse to fine representation.

## 2.4 Contour-Based Shape Representation Techniques

Contour-based shape representation techniques extract shape features from shape boundary. There are generally two approaches for contour shape representation: Global and structural.

With global approaches a feature vector derived from the whole boundary is used to describe the shape. The shape similarity is done by point-based matching or feature-based matching. Structural approaches break the shape boundary into segments, called *primitives* using a particular criterion. The final representation is usually a string or a graph (or tree), the similarity measure is done by string matching or graph matching. These two types of approaches are discussed in the following.

### 2.4.1 Global Methods

#### 2.4.1.1 Simple Shape Descriptors

Common simple shape descriptors are perimeter, area, circularity ($perimeter^2/area$), eccentricity ($length\ of\ major\ axis/length\ of\ minor\ axis$), major axis orientation and bending energy [5]. These simple descriptors are different for shapes with large dissimilarities. So they are usually used to filter shapes with large dissimilarities. They are not used as standalone descriptors. Because same representations with these descriptors do not guarantee shape similarity. For example, the eccentricity of the shape in Figure 2.2 (a) is close to 1. A circle has also the eccentricity value 1, therefore eccentiricity is not enough to correctly describe the shape in the figure. The circularity of two shapes in figure 2.2 (b) and (c) are nearly the same, however they are two different shapes.

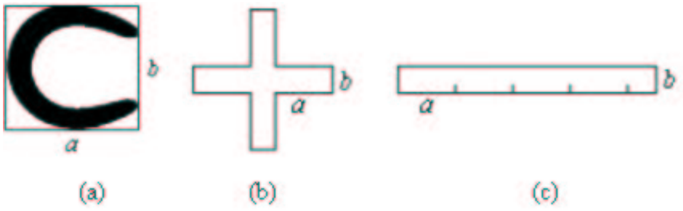


Figure 2.2: Shape eccentiricity and circularity

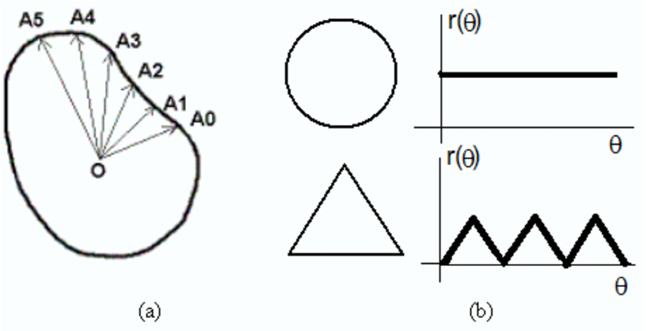


Figure 2.3: (a)Centroid to boundary distance approach (b) Shape signatures for a circle and triangle

#### 2.4.1.2 Shape Signature

Shape signatures represent shape by a one dimensional function derived from shape boundary points. Many shape signatures exist, including complex coordinates, polar coordinates, central distance, tangent angle, curvature and chord-length. Shape signatures have been extensively studied in [6, 7]. Using shape centroid for the representation of the boundary is an example of shape signature. One variation of idea is to use the distances between shape centroid and boundary points as values of 1-D function. Boundary points are selected so that the central angles are equal. Another idea is to use the distances between these subsequent boundary points for the 1-D function values. Shape signatures are usually normalized into being translation and scale invariant. Shape signatures are sensitive to noise, slight changes in the boundary can cause large error in the matching. Therefore, it is undesirable to directly use shape signature for shape retrieval.

#### 2.4.1.3 Fourier Transform of Boundary

Fourier transform has played a key role in image processing for many years and also applied to many shape representation applications. This method uses the Fourier transform of the 1-D boundary representation of the boundary of a shape. The fundamental principle behind Fourier transform is that a shape boundary can be treated as a signal and it can be represented by basic components of the signal. If we can define basic components to represent shape under analysis, the significant basic components found can be used for the description of the whole shape. Zahn and Roskies method [8] represents a closed boundary as a function of tangent angles versus the distance between the boundary points from which the angles were determined. The Fourier transform is applied to the boundary function and the resulting coefficients are used for shape description. If we do arc length normalization, the descriptor becomes invariant to scale change. The shape descriptor is invariant to translation because tangent angle versus arc length function is invariant to shape position. Rotation of the object (variation of the starting point) causes a phase change in the Fourier transform. Looking at the magnitudes of the coefficients will ensure rotation invariance. The nice characteristics of Fourier descriptor, such as simple derivation, simple normalization, simple to do matching, robustness to noise, make it a popular shape descriptor. The disadvantage is Fourier descriptor does not give local shape information. After the transform, local shape information is distributed to all coefficients.

#### 2.4.1.4 Scale Space Techniques

Sensitivity of shape descriptors to scale (image resolution) is an undesirable feature of shape desciptors. Different results can be found at different resolutions. Witkin [9] proposed a scale-space filtering approach which provides a useful representation of the significant shape features. The representation was created by tracking the position of inflection points in signals while they are filtered by low-pass Gaussian filters of variable widths. The inflection points in the representation were expected to be significant object characteristics. Babaud et al. [10] proved that the Gaussian kernel is the most useful filter for scale-space filtering. The Gaussian filter has the desirable property of saving inflection points when the width of the filter is increased.

Asada and Brady [11] introduced a representation called Curvature Primal Sketch. This is a scale-space approach for the representation of curvature. The shape boundary is filtered with Gaussian functions of increasing width to get a multi-scale representation for shape boundary. The curvature is then computed at different scales to obtain curvature primal sketch.

Mokhtarian and Mackworth [3] applied the scale space approach to the description of planar curves using the shape boundary. The curvature along the contour was computed and smoothed with variable width Gaussian filters. The scale space image of the curvature function was used as a descriptor which is invariant to scale, translation and rotation and robust to noise. The method is later extended for shape retrieval [12, 13].

### 2.4.2 Structural Methods

With structural approach, shapes are broken down into boundary segments called primitives. Structural methods differ in the selection and organization of the *primitives.* The boundary can be decomposed using polygonal approximation, curvature decomposition and curve fitting [14]. The result is encoded and represented as a string of general form

$$S = s_1, s_2, ..., s_n \tag{2.1}$$

where $s_i$ may be an element of a chain code, a side of a polygon, a quadratic arc, a spline, etc. $s_i$ may contain a number of attributes like length, average curvature, maximal curvature, bending energy, orientation etc. The resulting string can be directly used for description or can be used as input to a higher level syntactic analyzer. Such structural methods of shape representation and description are described in the following.

#### 2.4.2.1 Chain Encoding

Chain codes describe a shape by a sequence of unit-size line segments with a given orientation. The basis were introduced in 1961 by Freeman [15] who described a method for encoding arbitrary geometric shapes. In this approach, an arbitrary curve is represented by a sequence of small vectors of unit length and a limited set of possible directions. A chain code is an ordered sequence of vectors written in the form of $A = A_1 A_2 ... A_n$ .

If the chain code is used for matching, it must be independent of the choice of the starting pixel in the sequence. One method for normalizing the chain code is to find the sequence of the numbers so that when the chain is interpreted as a base four number the resulting chain will give the minimum integer number. An 8-directional chain code can also be defined which represents the relative directions of boundary elements of a shape measuring $90^o$ and $45^o$ direction changes. N-directional ($N > 8$ and $N = 2k$) chain code is also possible, which is called as general chain code [16].

A chain code is very sensitive to noise and scaling. It is not desirable to use chain code directly for shape description because of its sensitivity to boundary noise and resolution. It is often used as an input to higher level analysis for example it can be used for polygonal approximation and for finding boundary curvature.

#### 2.4.2.2 Syntactic Techniques

Syntactic techniques view the composition of a natural scene as a composition of a language that is sentences are built up from phrases, phrases are built up from words and words are built up from alphabets, etc [17]. Syntactic methods represent shapes with a set of predefined primitives. The set of predefined primitives is called codebook and the primitives are called codewords. For example, given the codewords in the right of Figure 2.4, the chromosome shape in the left of Figure 2.4 can be represented as a grammatical string of S:

$$S = d\ b\ a\ b\ c\ b\ a\ b\ d\ b\ a\ b\ c\ b\ a\ b \qquad (2.2)$$

The matching between shapes can use string matching by finding the minimal number of operations to convert one string into another.

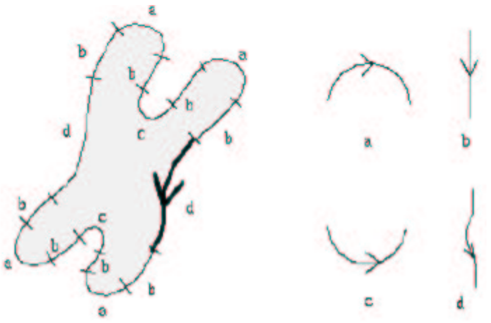


Figure 2.4: Structural description of chromosome shape [5]

The representation can be formulated as a string grammar. Each primitive is interpreted as an alphabet of some grammar, where a grammar is a set of rules of syntax of how sentences are generated from the symbols of the alphabet. After grammars have been established, matching is straightforward. For a sentence which represents an un-

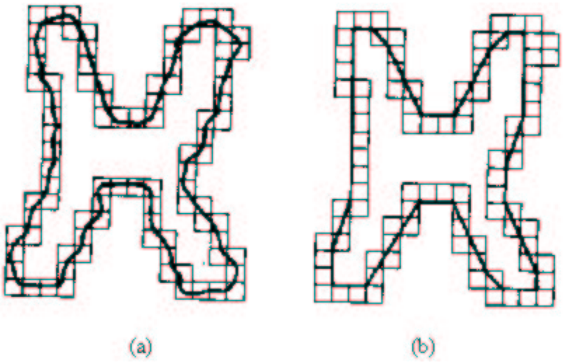


Figure 2.5: (a) Object boundary enclosed by cells (b) Minimum perimeter polygon [19]

known shape, the matching task is to decide in which language the shape represents a valid sentence.

Syntactical methods attempts to use the structural and hierarchical nature of human vision system. However, it is not practical in general application because building up a pattern grammar which generates valid patterns (shapes) is not possible. In addition, this method needs a priori knowledge for the database in order to define codewords or alphabets which is usually unavailable.

#### 2.4.2.3 Boundary Approximations

Two most popular methods for boundary approximation are polygonal and spline approximations.

Polygonal approximatons are used to approximate the shape boundary using polygonal lines. The objective is to capture the essence of the boundary shape with some approximation criteria. Such criteria can be the use of the minimal error, the minimal polygon perimeter, the maximal internal polygon area or the minimal external polygon area. One of the most popular methods in this group is the split-and-merge algorithm [14]. In this approach, a curve is split into segments until some acceptable error is obtained. At the same time, split polygonal segments are merged if the resulting segment approximates the curve within some maximum error. For the description, the polygon vertices are used as primitives. The feature for each primitive is expressed with a four element string which consists of internal angle, distance from the next vertex, and its x and y coordinates. After the shape is represented as string of line segments, line segments are organized into tree data structure and similarity between shapes can be investigated.

Splines are also popular for curve approximations. B-splines are piecewise polynomial curves whose shape is closely related to their control polygon a chain of vertices giving a polygonal representation of a curve. They have good representation properties: They change their shape less than their control polygon. If a control polygon vertex changes its position, the resulting change of the spline curve occurs in a small neighbourhood of that vertex. The drawback of a spline representation is its high sensitivity to change in scale [5].

## 2.5 Region-Based Shape Representation Techniques

In region based methods, shape features are extracted from the whole region of a shape. Common region based methods use moment descriptors to describe a shape. Other region based methods include grid method, shape matrix, convex hull and media axis.

### 2.5.1 Global Methods

#### 2.5.1.1 Geometric Moments

The use of moments for shape description was initiated by Hu [18]. His approach was based on the two-dimensional Cartesian moment $m_{pq}$ of order p+q of a function f(x,y) :

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q \rho(x, y) dx dy \qquad p, q = 0, 1, 2, ... \tag{2.3}$$

Using nonlinear combinations of the lower order moments, a set of moment invariants (usually called geometric moment), which has the desirable properties of being invariant under translation, scaling and rotation, are derived. The use of higher order moments for pattern analysis has not been addressed. Geometric moment invariants has attracted wide attention [5, 19, 20, 21, 22, 23] and has been used in many applications.

The main problem with geometric moments is that only a few invariants derived from lower order of moments is not sufficient to accurately describe shape. Higher order invariants are difficult to derive. It has been shown that similar moments do not guarantee similar shapes. Uniqueness criteria may not be satisfied for each moment invariant.

#### 2.5.1.2 Orthogonal Moments and Other Moments

The algebraic moment transform of equation (2.3) can be extended to generalized form by replacing the conventional transform kernel $x_p y_q$ with a more general kernel of $P_p(x) P_q(y)$.

Teague [21] uses this idea to introduce orthogonal moments -Legendre moments and Zernike moments- by replacing $x_p y_q$ in equation (2.3) with Legendre polynomial and Zernike polynomial respectively. Since both Legendre and Zernike polynomials are complete sets of orthogonal basis, Legendre and Zernike moments are called orthogonal moments. Other orthogonal moments are pseudo-Zernike moments. Orthogonal moments allow for accurate reconstruction of the described shape.

Teh and Chin [22] have made a detailed study of orthogonal moments -Legendre moments, Zernike moments, psudo-Zernike moments, and non-orthogonal moments-geometric moments, complex moments, rotation moments. Their results show that geometric moments, complex moments and pseudo-Zernike moments are less affected by noise and Legendre moments are more severely affected by noise. Zernike moments and pseudo-Zernike moments have more reconstruction power than Legendre moments for both noisy and normal images. They conclude that Zernike moments and pseudo-Zernike moments are the preferred shape descriptors to others. Liao and Pawlak [23] extend Teh and Chin's work by introducing techniques to increase the accuracy and efficiency of moments.

Moment shape descriptors are usually concise, robust, easy to compute and match. The disadvantage with many moment methods is that it is difficult to correlate high order moments with shape physical features. Shape description using Zernike moments proves to be very promising [24, 25].

#### 2.5.1.3 Grid Based Method

Grid shape descriptor is proposed by Lu and Sajjanhar [26] and has been used in [27] and [28]. Basically, a grid of certain number of cells is overlaid on a shape, the grid is then scanned from left to right and top to bottom. The result is a bitmap. The cells covered by shape are assigned 1 and those not covered by shape are assigned 0. The shape can then be represented as a binary feature vector. In order to be invariant to translation, rotation and scaling of the shape, the shape is first normalized before the scanning. The shape is scaled into a fix sized rectangle, shifted to the upper left of the rectangle and rotated so that the major axis of the shape are horizontal. Mirrored shape and flipped shape should be considered separately. The advantages of grid descriptor are its simplicity in representation, conforming to intuition. However, the computation of the descriptor is expensive.

#### 2.5.1.4 Shape Matrices and Vectors

Normal shape methods use grid sampling to acquire shape information. Goshtasby proposes the use of shape matrix which is derived from a circular raster sampling technique [29]. The idea is similar to normal raster sampling. However, rather than overlaying the normal square grid on a shape image, a polar raster of concentric circles and radial lines is overlaid in the center of the mass (Figure 2.6(a)). The binary value of the shape is sampled at the intersections of the circles and radial lines. The shape matrix is formed so that the circles correspond to the matrix columns and the radial lines correspond to the matrix rows. The result matrix representation is invariant to translation, rotation, and scaling.

Since shape matrix is a sparse sampling of shape, it is easily affected by noise. Besides, shape matching using shape matrix is expensive.

Parui et al. propose shape description based on the relative areas of the shape contained in concentric rings located in the shape center of the mass [30, 31]. Let L be the maximum radius of the shape S to be described, $C_k$, be the $k$th ring of n concentric rings obtained by sectioning the maximum radius L into n equal segments.

An area-ratio invariant is defined as

$$x = \frac{A(S \cap C_i)}{A(C_i)} \tag{2.4}$$

where $A(\cdot)$ is the area function. The shape descriptor is the feature vector of $x = [x_1, x_2 \dots x_n]^T$. Although the area ratio descriptor is more compact and robust than the shape matrix, its drawback is to ignore the shape distribution within a ring. Consequently, the two shapes in Figure 2.6 (b) and (c) will be the same under this descriptor.

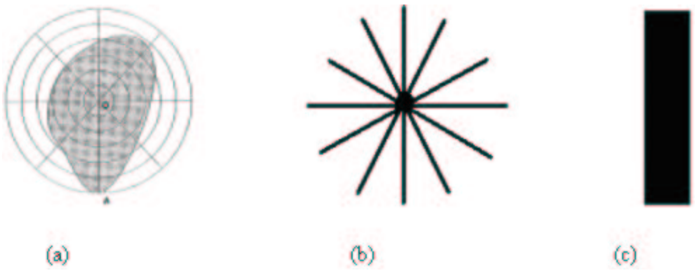


Figure 2.6: (a) shape matrix concept [33] (b) a star shape formed by line strips (c) a rectangle shape.

### 2.5.2 Structural Methods

#### 2.5.2.1 Convex Hull

A region R is convex if and only if any two points $x_1, x_2 \in R$, the whole line segment $x_1 x_2$ s inside the region. The convex hull of a region is the smallest convex region H which satisfies the condition $R \subset H$. The difference $H - R$ is called the convex deficiency D of the region R.

Since shape boundaries tend to be irregular because of digitization, noise and variation in segmentation, this results convex deficiency that has small components

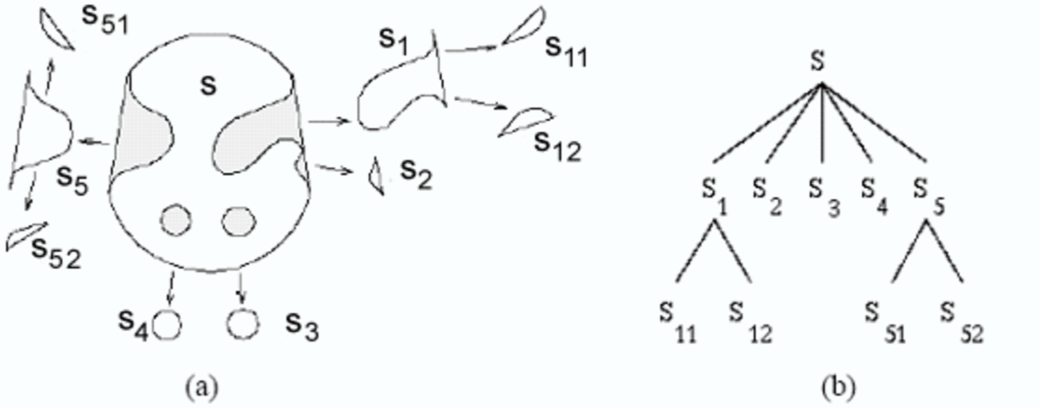


Figure 2.7: (a) Convex hull and it concavities (b) Concavity tree representation of convex hull [5]

throughout the boundary. Common practice is to first smooth a boundary prior to partitioning (i.e.polygon approximation). The extracting of convex hull is the process of finding significant convex deficiencies along the boundary. The shape can then be represented by a string of concavities. A better representation of the shape is obtained by a recursive process which results in a concavity tree. A convex hull of the whole region is constructed first, and then the convex hulls of remaining concave parts are found. It goes on until all remaining convex deficiencies are convex. Figure 2.7(a) illustrates this process. The shape is then represented as a concavity tree (Figure 2.7(b)). Each concavity can be described by its area, bridge length (the line which connects the cut of the concavity), maximum curvature, distance from maximum curvature point to the bridge. The matching between shapes becomes a string matching or a graph matching.

#### 2.5.2.2 Medial Axis Transform

Like the convex hull, region skeleton can also be employed for shape representation and description. The basic idea of the skeleton is that eliminating redundant information while retaining only the topological information concerning the structure of the object that can help with recognition. The skeleton methods are represented by Blum's medial axis transform (MAT) [23]. The MAT of a region R with border B is as follows: For each point $p$ in $R$, we find its closest neighbour in $B$. If $p$ has more than one such a neighbour, it is said to belong to the medial axis (skeleton) of R.

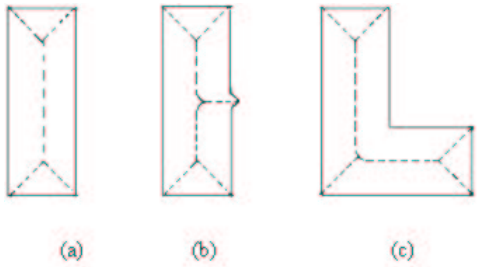


Figure 2.8: Medial axes of three simple regions

The skeleton can then be decomposed into segments and represented as a graph according to certain criteria. The matching between shapes becomes a graph matching problem.

Although the MAT of a region gives an intuitively pleasing skeleton, implementation of the definition is a computational burden because it involves calculating the distance from every interior point to every point on the boundary of a region. In addition, medial axis tends to be very sensitive to boundary noise and variations.

## 2.6 Evaluation of Shape Representation Methods

In this chapter, a selection of most characteristic shape representations are discussed. There are mainly two approaches: contour-based and region-based. Contour-based methods are generally more popular than region-based methods because it is believed that human beings discriminate shapes by contour features. Contour-based methods generally use less computation than region-based methods. However, contour shape descriptors are more easily affected by noise and variations than region-based shape descriptors. Because they use less shape information than region-based methods. Region-based methods are usually more robust and application independent. But they involve more computation and need more storage area than contour-based methods.

The choice of shape representation directly depends on the type of shapes and application it will be used. For image description and retrieval, MPEG-7 proposed some criteria for the suitability of a shape description technique which are compactness of the descriptor, good retrieval accuracy, general application, low computation complexity, robust retrieval performance and hierarchical coarse to fine representation.

According to these criteria, structural approaches are weak because of their computational complexity and strong application dependence. Global methods which result in numeric descriptors are more suitable for image retrieval. In contour-based methods Fourier descriptors and curvature scale space descriptor are more promising than other methods for image retrieval. The perceptual meaning and compact features of curvature scale space makes it a good candidate for shape description and online retrieval. In region-based methods moment-based techniques are the most promising. In the next chapters, these most promising methods, curvature scale space descriptor and moment-based techniques will be studied in detail.

# CHAPTER 3

# CONTOUR-BASED ANALYSIS: CURVATURE SCALE SPACE REPRESENTATION

In this chapter, the contour shape descriptor is presented: underlying theory of curvature scale space representation, the algorithm to extract CSS descriptor is explained and its properties are discussed.

## 3.1 Scale Space

### 3.1.1 Background

The philosophy behind scale space modeling is that we observe objects in different scales and objects are meaningful only under certain scales. A simple example is the concept of a branch of a tree, which makes sense only at a scale from, say, a few centimeters to a few meters. It is meaningless to discuss a tree concept at nanometer and kilometer level [32]. For example, in physics the world is described at several levels of scales; at fine scale, particle physics and quantum mechanics, at coarse scale, astronomy. The physical description depends on the scale the world is modelled. An instrument (e.g.eye, camera) is needed to do an observation. The range an instrument can see is bounded on two sides: the outer scale is the coarsest detail that an object is discriminated, like a minumum size of a window that completely contains the object or the structure; the inner scale is the finest detail seen which depends on the instrument, e.g. one CCD element of digital camera, a cone or rod on the retina. The type of information that can be obtained from an image is determined by the relationship between the size of the structures in the image and the size of resolution.

While these qualitative aspects have been well known, the concept of scale has been hard to be formalized into a mathematical theory. A method that has been proposed for handling the notion of scale is to represent measured signals at multiple scales. In certain situations, the suitable scales for analysis may be known a priori, however,in the most of vision problems, the scale of the scene under analysis is not known. The main idea for scale space is that if no prior knowledge is available about the appropriate scale for a given data set, then the only reasonable approach is to represent the input data at multiple scales [32]. A major reason for representing a signal at multiple scales is to represent the multi-scale aspect of real-world data. Another aim is to remove unnecessary and disturbing details so that further processing tasks can be simplified.

Witkin [9] first related image structures at different scales and introduced the term scale-space. He embedded a signal $f(x) : R_N \to R$ into a continuous family $L(x,t) \mid t > 0$ of gradually smoother versions of it. It means a signal $f(x)$ is convolved with a family of continuous functions $g(x;\ t)$:

$$L(x,t) = g(x,t) * f(x) \tag{3.1}$$

where $g(x,t)$ is the smoothing function or kernel function, $L(x,t)$ is the smoothed signal, and * means convolution. For a continuous function $f(x)$, (3.1) is expressed as:

$$L(x;\ t) = \int_{\xi \in R^N} g(\xi;\ t) f(x - \xi) d\xi \tag{3.2}$$

### 3.1.2 Gaussian Scale Space

It has been shown that Gaussian is the most useful smoothing function for generating scale space [10].

Given a signal $f(x) : R^N \to R$, the scale-space representation $L : R^N X R_+ \to R$ is defined such that the representation at zero scale is the original signal:

$$L(x;\ t) = f(x) \tag{3.3}$$

and the representations at coarser scales are given by convolution with Gaussian kernels of increasing width:

$$L(x;\ t) = \int_{\xi \in R^N} \frac{1}{(2\pi t)^{N/2}} e^{-x^T x/2t} f(x - \xi) d\xi \tag{3.4}$$

where $x = (x_1, ..., x_N)^T \in R^N$. For one-dimensional signal, the scale-space formulation and its discrete form are given as

$$L(x, t) = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi t}} e^{-\xi^2/2t} f(x - \xi) d\xi \tag{3.5}$$

$$L(x, t) = \sum_{n=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi t}} e^{-n^2/2t} f(x - n) \tag{3.6}$$

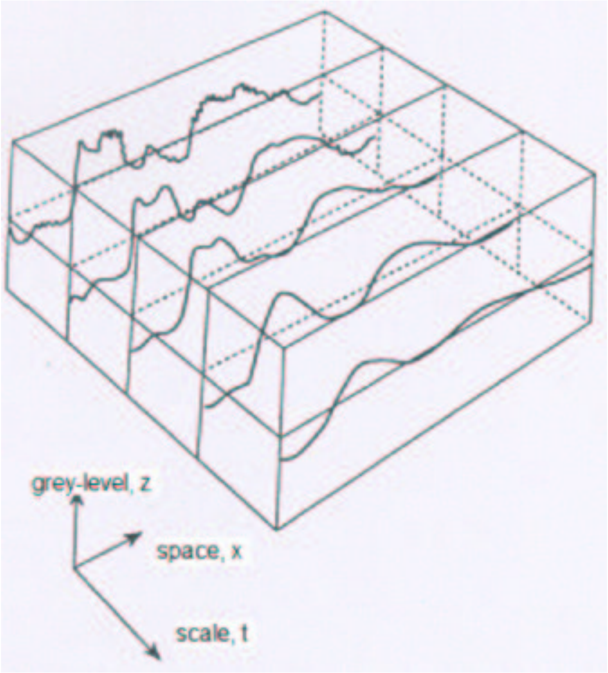


Figure 3.1: The scale-space behaviour of a one dimensional signal [35].

### 3.1.3 Property of non-creation of new features

The most important property of scale space is non-creation of new features. It means that the transformation from a fine scale to a coarse scale can be considered as a simplification of the signal, so that fine-scale features disappear monotonically with increasing scale.

Particularly useful features are zero-crossing points of the $n$th-order spatial derivatives. In practice, second order derivatives of the signal are used widely in pattern analysis, because the second derivatives represent the curvatures of the signal. When Witkin [9] introduced scale-space, he observed that the number of zero-crossings in the second derivative decreased monotonically with scale and took that as a basic characteristic of the representation. The zero-crossing points of the second order derivatives are called as inflection points which are significant features of the pattern. For one

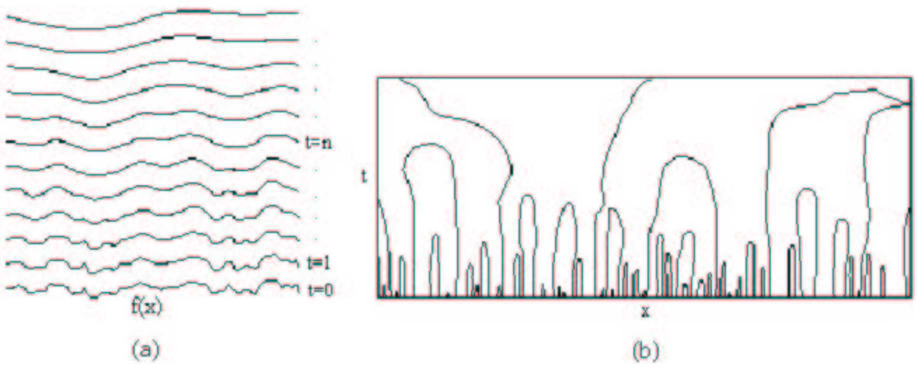


Figure 3.2: (a) an original signal f(x) at the bottom and its successively smoothed versions on the top of it. t is the scale of the smoothing function; (b) interval tree derived from the zero crossing of the second derivatives of the smoothed signals in the left, each (x,t) in the interval tree corresponds to a zero-crossing point at position x and scale t of the signal[10].

dimensional signal, with the application of Gaussian scale space, the zero-crossing points of signal at all scales form the so called fingerprints or interval tree. (Figure 3.2(b)).

With Gaussian smoothing, moving from coarse to fine scale, new zero-crossings appear but existing one never disappear.

Asada and Brady [11] first extracted peaks from the interval tree, introduced a representation called Curvature Primal Sketch. From the detected peaks they interpreted high level primitive events as physical features such as corner, join, end, crank etc.

Mokhtarian and Mackworth [3] adopt Asada and Bradys interpretation method and extend it to shoreline registration. They called the resulting scale space signature as Curvature Scale Space (CSS) contour image. The peaks of individual branches in the CSS are detected, and they are used for matching two curves under analysis. The method is later used for shape retrieval [12, 13]. Curvature Scale Space Descriptor has been chosen as MPEG-7 contour based shape descriptor.

## 3.2 Curvature Scale Space Descriptor

Basically CSS method treats shape boundary as a 1-D signal and analyze this 1-D signal in scale space. The concavities/convexities of shape contour are found by examining zero-crossings of curvature at different scales. These concavities/convexities are useful for shape description because they represent perceptual features of shape contour. The block diagram of computing CSSD is shown in Figure 3.3.

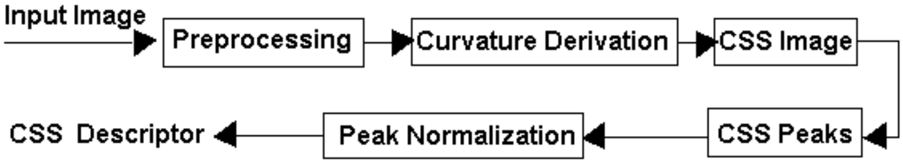


Figure 3.3: Block diagram of computing Curvature Scale Space Descriptor

### 3.2.1 Contour Tracing

After segmentation of a given image, first step of the process is finding the boundary points of the input shape.

Among most common contour tracing algoritms, Square Tracing Algorithm has the advantage of simple implementation procedure and therefore used frequently to trace the contour of a given pattern. The idea behind the algorithm is: Given a digital pattern i.e. a group of black pixels, on a background of white pixels i.e. a grid; locate a black pixel and declare it as the " start" pixel. Locating a "start" pixel can be done in a number of ways; for example starting from the top left corner of the grid, scanning from left to right starting from the top row and proceeding with bottom rows, until the starting pixel is encountered. In order to extract the contour of the pattern: when

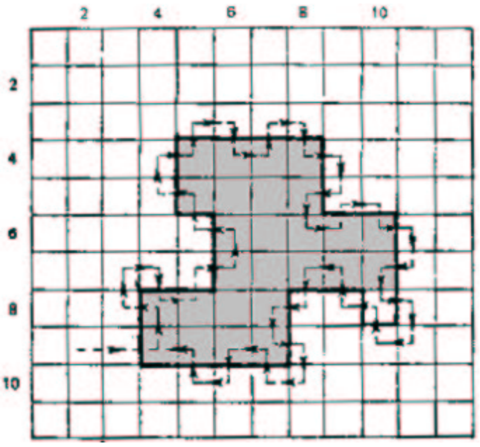


Figure 3.4: Contour tracing algorithm.When a black pixel is encountered, turn left. When a white pixel is encountered, turn right.

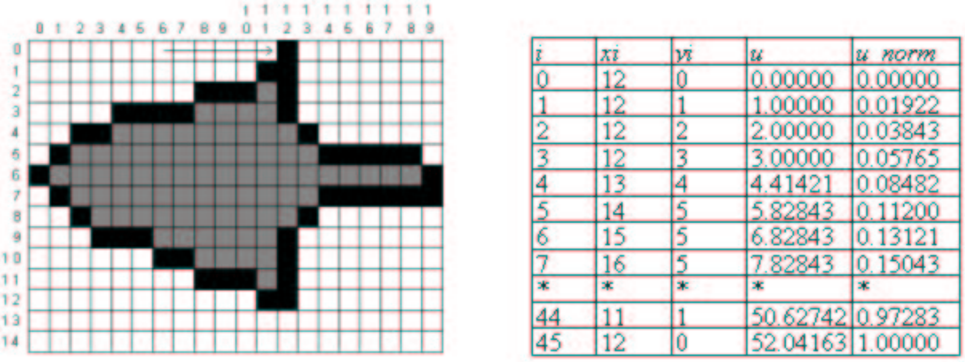
black pixel is found, turn left and go on with that pixel, otherwise white pixel is found, turn right and go on with that pixel. This goes on until the start pixel is encountered again. Whenever black pixels are found, the coordinates are stored as the contour of the pattern. The pseudo code of this algorithm is provided in Appendix 1.

### 3.2.2 Arc-length parametrization and scale normalization

After contour tracing, let's say the list of contour points of the shape form a set of n elements:

$$X = \{(x_1, y_1), (x_2, y_2), ....(x_n, y_n)\}$$

In the following steps, it is desired to find zero-crossings in curvature of the curve at varying levels of detail, that is, for varying degrees of the smoothing of the curve. A planar curve does not behave like a single valued function so a parametrization of the curve is needed in order to compute the curvature of the curve. Such a parametrization can be done by considering arc length variable along the curve and expressing the curve in terms of $x(u)$ and $y(u)$:



| i | xi | yi | u | u norm |
|---|---|---|---|---|
| 0 | 12 | 0 | 0.00000 | 0.00000 |
| 1 | 12 | 1 | 1.00000 | 0.01922 |
| 2 | 12 | 2 | 2.00000 | 0.03843 |
| 3 | 12 | 3 | 3.00000 | 0.05765 |
| 4 | 13 | 4 | 4.41421 | 0.08482 |
| 5 | 14 | 5 | 5.82843 | 0.11200 |
| 6 | 15 | 5 | 6.82843 | 0.13121 |
| 7 | 16 | 5 | 7.82843 | 0.15043 |
| * | * | * | * | * |
| 44 | 11 | 1 | 50.62742 | 0.97283 |
| 45 | 12 | 0 | 52.04163 | 1.00000 |

Figure 3.5: An example for arc-length parametrization of boundary coordinates

$$C = \{x(u), y(u)\}$$

where u is the arc length changing over the closed interval [0,1].

The x and y coordinates of the boundary pixels are parametrized by the curve arc-length parameter u, and u is normalized to interval [0-1]. Figure 3.5 is an example of a contour and shows the set of contour points, and the corresponding values of the parameter u and u_norm (normalized to interval [0,1] ).

The functions $x(u)$ and $y(u)$ are then resampled to $N$ equidistant points. This makes a normalization effect so that shapes with different number of boundary points can be compared.

### 3.2.3 Calculating Curvature

The curvature $K$ of a planar curve at a point $P$ on the curve is defined as the instantaneous rate of change of the slope angle $Q$ of the tangent at point $P$ with respect to arc-length $u$.

$$K(s) = \lim_{h \to 0} \frac{\theta}{h}$$

where $\theta$ is the angle between tangent $t(s)$ and $t(s+h)$. It is possible to compute the curvature of a planar curve from its parametric representation.

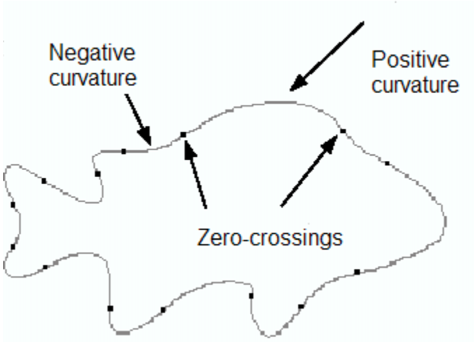


Figure 3.6: Inflection points of a fish shape

It is shown in Appendix 2 that the curvature $K$ on curve $C$ is given by:

$$K(j) = (x'(j) * y''(j) - y'(j) * x''(j))/(x'^2 + y'^2)^{3/2}$$

where $j$ is index of $N$ equidistant points, $x'$, $y'$ and $x''$, $y''$ are the first and second derivatives at location $j$.

### 3.2.4 Gaussian Filtering

The idea is to compute the convolution of the parametric coordinate functions of the curve $C$ with a 1-D Gaussian kernel with a progressively larger width $\sigma$, which is equivalent to low-pass filtering the original contour with a filter with a progressively lower bandwidth. This can be implemented by repetitive application of the lowpass filter with kernel (0.25, 0.5, 0.25). After each pass of the low-pass filter, the curvature zero-crossings are found and located on the CSS image. Zero crossings can be found by finding k for which $K(k) * K(k+1) < 0$ (Figure 3.6). CSS image shows how the inflection points change when filtering is applied. CSS image x-axis, x_css corresponds to the position on the contour (clockwise, starting from any arbitrary point).The number of filtering applied makes up the y_css coordinates at the CSS graph. Any black point on the CSS image means that at that position on the contour, at that scale, there is an inflection point. Smoothing of the curve goes on until the curve becomes convex that is there are no zero-crossings of curvature. CSS image formation can be observed from Figure 3.7.

### 3.2.5 Properties of CSS Image

The significant peaks in the CSS graph give valuable information about a shape. The concept of CSS representation is based on the observation that when comparing shapes, humans tend to decompose shape contours into concave and convex sections. The overall similarity depends on the similarity of the corresponding sections: for example, how prominent they are, their length relative to the contour length and their position and order on the contour. The CSS representation decomposes the contour into convex and concave sections by determining the inflection points where curvature goes from negative to positive or vice versa. Figure 3.8 shows a shape of a fish and its CSS image. There are three peaks in the CSS image (marked as A,B,C), this means that the fish shape has three concave parts. The CSS image shows that the concave parts of the shape A and C are approximately has similar perceptual importance because their peak heights are similar. Concavity B is less prominent. From the CSS image, we can say that the concavity B is between concavities A and C and find relative distances between inflection points.

In figure 3.9, the left column shows the contour at different stages of curve evolution (after 20 and 80 passes of filter). Next to the contour is the CSS image obtained from the evolution of the curve, corresponding curvature zero-crossings are shown.

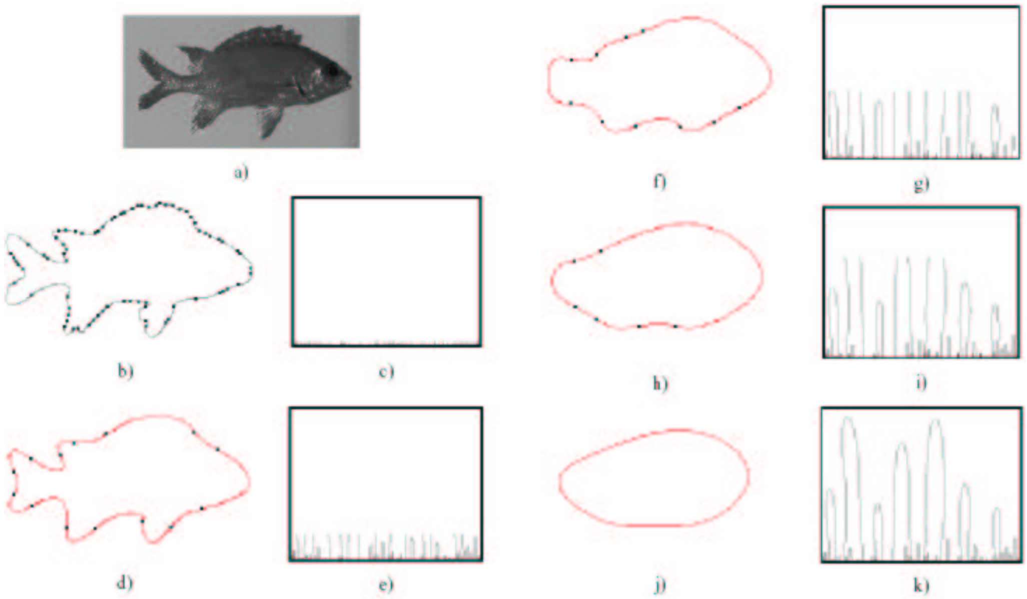


Figure 3.7: Fish object: a) original image; b), d), f), h) and j) contours with progressive amounts of low-pass filtering (after 3, 13, 30, 45 and 60 passes of the low-pass filter, respectively); c), e), g), i) and k) corresponding progressive formation of the CSS representation[36]

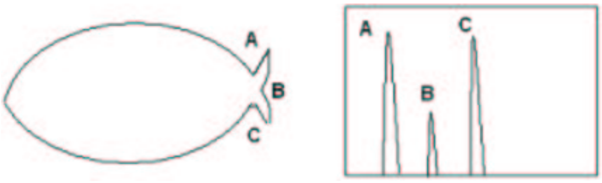


Figure 3.8: Concavities of a fish shape and corresponding peaks at its CSS image

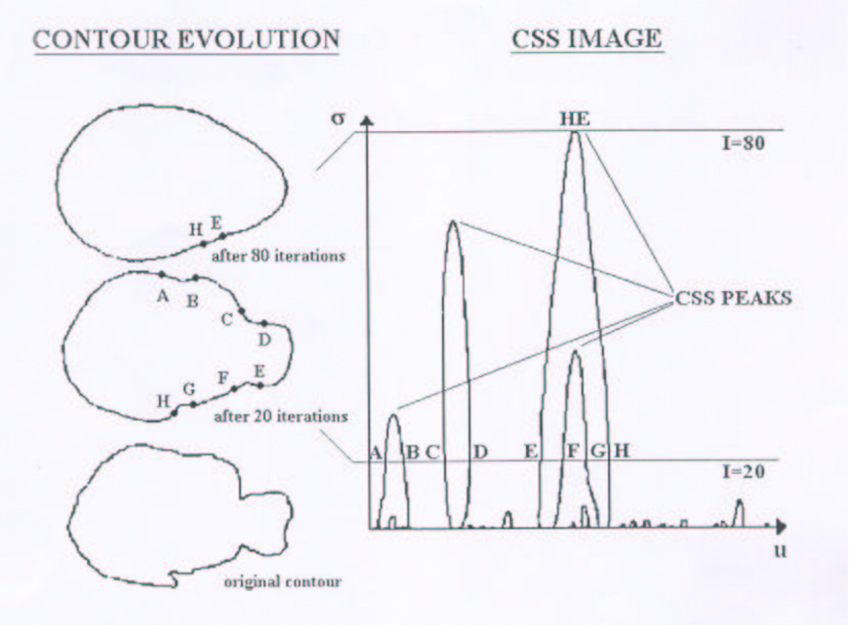


Figure 3.9: CSS image formation and zero-crossing of curvature

### 3.2.6 CSS Matching and Similarity Measurement

The characteristic property of the CSS image is the maxima points called as peaks. The CSS coordinates of a maximum has information on the location and the scale of the corrresponding contour. The matching algorithm should find the correct correspondence between two sets of maxima from each CSS image.

#### Peak Extraction

CSS image is usually connected everywhere except possibly in a neighbourhood of its maximum. In fact, the exact maxima can be seen in the CSS image with a very fine sampling of the curve and the $\sigma$ parameter, but fine sampling will result in a large CSS image and greater computation cost. So the actual maximum of a CSS contour usually falls in the gap at the top of the contour. In order to find the gaps, the zero-crossing points must be very close to each other and none of them must have a zero-crossing neighbour at the next higher scale. When such a pair is found, the maximum is assumed to be the midpoint of the line segment joining the pair.

### Peak Normalization and Quantization

The location points are normalized so that they are invariant to the number of the sampling points and the size of the contours. The peaks are ordered on the basis of decreasing values y_css, transformed using a nonlinear transformation and quantized. The quantization details are explained in Appendix 3.

### Similarity Measurement

The matching is done by evaluating the similarity measure between the descriptor of the query shape and the descriptors of the model shapes in the database of shapes.

Firstly, it is a good idea to compare global parameters of the query and model shapes. If there is a significant difference, comparison is not performed for the dissimilar shapes. This step reduces the amount of computation that would be needed to find a similarity measure between dissimilar shapes. Simple shape descriptors eccentricity and circularity are used for this purpose.

Circularity is a dimensionless quantitity and thus is insensitive to scale changes and orientation and is minimal for a disk-shaped region.

$$circularity = \frac{perimeter^2}{area} \tag{3.7}$$

Eccentiricity measure can be defined as the ratio of main axes of inertia [33]. Eccentricity is also scale and orientation invariant.

$$eccentricity = \frac{(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}}{area} \tag{3.8}$$

where $\mu_{p,q}$ is the central moment of order $p, q$.

If the calculated circularity and eccentricity are in thresholds, further similarity is to be measured for the model shapes.

$$\frac{\mid e_q - e_m \mid}{MAX(e_q, e_m)} \leq Th\_e \tag{3.9}$$

$$\frac{\mid c_q - c_m \mid}{MAX(c_q, c_m)} \leq Th\_c \tag{3.10}$$

where $e_q$ and $e_m$ are the eccentricity values for the query shape and model respectively, and $c_q$, $c_m$ are the circularity parameters of the query shape and model $Th\_e$ and $Th\_c$ are the thresholds used, $Th\_e = 0.6$ and $Th\_c = 1.0$ were used.

The similarity measure $M$ is computed as a weighted sum of similarity measure between global shape parameters and the similarity measure $Mcss$ between the peaks of CSS representation.

$$M = 0.4 \times \frac{\mid e_q - e_m \mid}{MAX(e_q, e_m)} + 0.3 \times \frac{\mid c_q - c_m \mid}{MAX(c_q, c_m)} + Mcss \tag{3.11}$$

The similarity measure between two sets of CSS peaks is an L2 measure between the matching peaks. For two peaks to be matched, the L2 measure between their x-coordinates should be below a threshold taken to be 0.1. For unmatched peaks, the similarity measure increases with the height of the unmatched peaks.

$$Mcss = \sum_1 ((xpeak[i] - xpeak[j])^2 + (ypeak[i] - ypeak[j])^2) + \sum_2 (ypeak[i])^2 \tag{3.12}$$

where $\sum_1$ is summation over all matched peaks (i and j are indices of query and model peaks that match), and $\sum_2$ is summation over all unmatched query and model peaks.

# CHAPTER 4

# REGION-BASED ANALYSIS: ANGULAR RADIAL TRANSFORM

In this chapter, the region shape descriptor is presented: the ART algorithm used to extract ART descriptors of a shape is explained and its properties are discussed.

## 4.1 Background

Among region-based methods used for description of shape, various types of moments have been used for shape description in many applications. Orthogonal moments are more suitable than geometric moments for shape description because uniqueness criteria defined for a reliable description holds for orthogonal moments.

Teh and Chin [22] have made a detailed study of orthogonal moments and non-orthogonal moments. Sensivity of these moments to image noise and image representation abilities were compared. They concluded that Zernike moments and pseudo-Zernike moments are the preferred shape descriptors to others.

Kim [34] presented the Zernike moment descriptor for retrieval of images from a large image database. It is shown that the Zernike moment descriptor has many desirable properties such as rotation invariance, robustness to noise, expression efficiency and fast computation. The real parts of Zernike basis functions up to order 8 are shown in Fig.4.1. The radial and angular directional complexities of Zernike basis functions vary depending on the values of n and m. As shown in Fig.4.1, Zernike basis functions do not describe equally the radial and angular directional complexities, but they rather tend to emphasize the radial directional complexities because the Zernike basis functions are defined only when $n \geq m$.

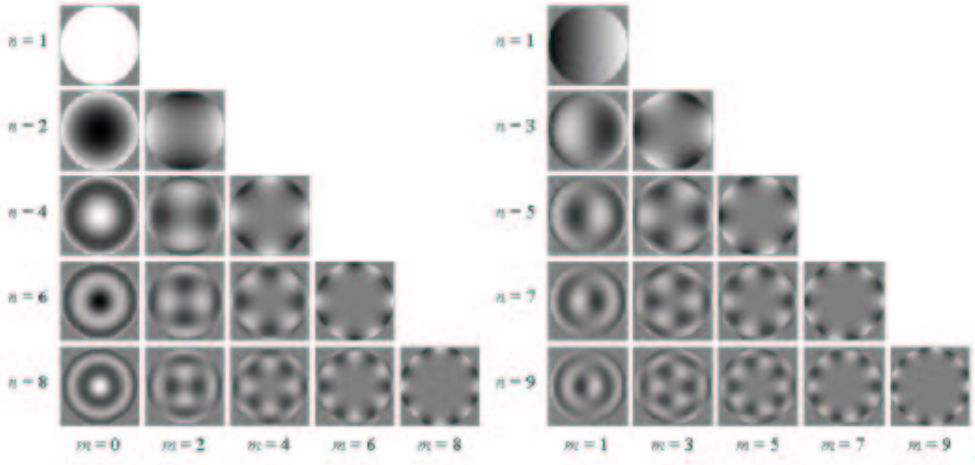


Figure 4.1: Real parts of Zernike basis functions up to order 8

With the motivation keeping the two desirable properties of Zernike basis function, the orthogonality and their rotation invariance of magnitude, with additionally taking into account both complexities in radial and angular direction, ART shape descriptor has been proposed [35]. Experimental results showed that the proposed ART descriptor totally improved retrieval accuracy when compared with Zernike moment descriptor. ART descriptor has been chosen as MPEG-7 region based shape descriptor.

## 4.2 Angular Radial Transform

### 4.2.1 The definition of ART

ART is an orthogonal 2-D complex transform defined on a unit disk in polar coordinates:

$$\begin{aligned} F_{nm} &= \langle V_{nm}(\rho,\theta), f(\rho,\theta)\rangle & (4.1) \\ &= \int_0^{2x}\int_0^1 V^*_{nm}(\rho,\theta), f(\rho,\theta)\rho d\rho d\theta & (4.2) \end{aligned}$$

where $F_{nm}$ is an ART coefficient of order $n$ and $m$, $f(\rho,\theta)$ is an image function in polar coordinates, and $V_{nm}(\rho,\theta)$ is the ART basis function that are separable along angular and radial directions, i.e.,

$$V_{nm}(\rho,\theta) = A_m(\theta)R_n(\rho) \tag{4.3}$$

inorder to achieve the rotation invariance, an exponential function is used for the angular basis function,

$$A_m(\theta) = \frac{1}{2\pi}exp(jm\theta) \tag{4.4}$$

The radial basis function defined by a cosine function,

$$R_n(\rho) = \begin{cases} 1 & \text{if } n = 0 \\ 2cos(\pi n\rho) & \text{if } n \neq 0 \end{cases} \tag{4.5}$$

where $n$ and $m$ are respectively radial and angular indices which define the order of coefficient $F_{nm}$.

### 4.2.2 Region Description with ART

ART basis functions $V_{nm}(\rho, \theta)$ are complex functions. In Fig.4.2, the real parts of the first 36 basis functions are shown. Their imaginary parts are similar except for quadrature phase difference. While for exact reconstruction of an image an infinite number of ART coefficients are needed, finite, usually small number of coefficients are enough for image description.

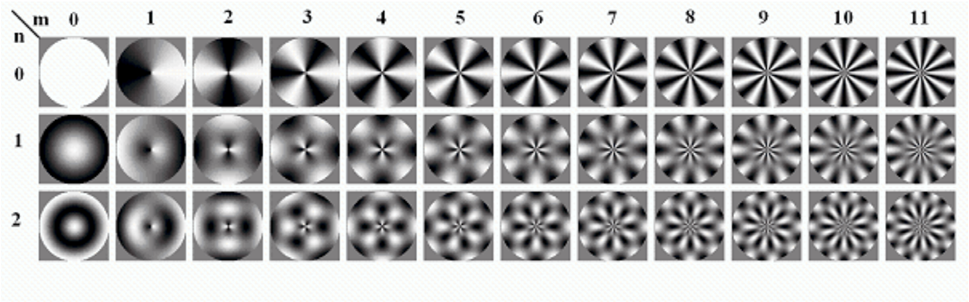


Figure 4.2: Real parts of ART basis functions (N=3, M=12)

Figure 4.3 summarizes how a region descriptor can be formed with ART transform. The steps for finding the ART descriptor will be explained in detail. The pseudo-code for computing the ART descriptor is provided in Appendix 4.

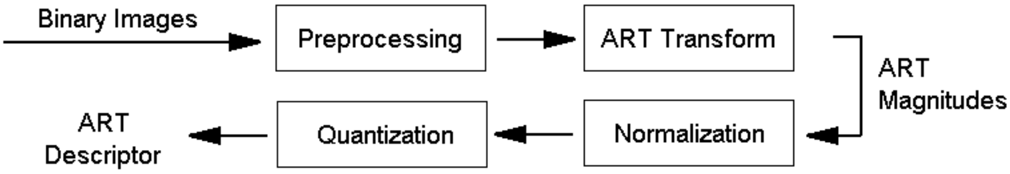


Figure 4.3: Block diagram of computing ART descriptor

#### Preprocessing

The normalized region is projected onto the basis functions to compute the corresponding ART coefficients. Due to reasons of simplicity and efficiency, instead of converting every image to polar coordinates, the basis functions are computed directly in Cartesian coordinates.

$$F_{nm} = \int_y \int_x V^*_{nm}(x, y) f(x, y) dx dy \tag{4.6}$$

First, the set of ART basis functions $V_{nm}(x, y)$ is stored in a lookup table (LUT). In the next step, the binary image is normalized. ART transform is defined on a unit disk so the shape should be enclosed in a circle. A radius $R$ of a circle is determined to enclose the shape completely from the center of mass (centroid) of shape to the outermost pixel of the shape. If the diameter of the shape is different from the LUT, the centroid of the shape is aligned to coincide with that of the LUT and linear interpolation is applied to map the image onto the LUT size.

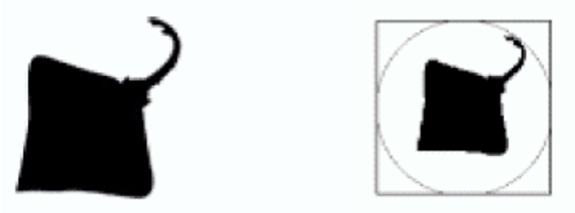


Figure 4.4: An original shape (left) and its normalized shape (right).

### ART transformation

The real and imaginary parts of the ART coefficients are computed by summing up the multiplication of a pixel in an image to each corresponding pixel in the LUT, on raster scan order.

Table 4.1: Quantization table of ART Magnitudes.

| | | | |
|---|---|---|---|
| $0.00000000 \leq ArtM < 0.00358547$ | 0000 | $0.03850817 \leq ArtM < 0.04592658$ | 1000 |
| $0.00358547 \leq ArtM < 0.00741841$ | 0001 | $0.04592658 \leq ArtM < 0.05449051$ | 1001 |
| $0.00741841 \leq ArtM < 0.01153552$ | 0010 | $0.05449051 \leq ArtM < 0.06461948$ | 1010 |
| $0.01153552 \leq ArtM < 0.01598233$ | 0011 | $0.06461948 \leq ArtM < 0.07701635$ | 1011 |
| $0.01598233 \leq ArtM < 0.02081630$ | 0100 | $0.07701635 \leq ArtM < 0.09299868$ | 1100 |
| $0.02081630 \leq ArtM < 0.02611131$ | 0101 | $0.09299868 \leq ArtM < 0.11552452$ | 1101 |
| $0.02611131 \leq ArtM < 0.03196467$ | 0110 | $0.11552452 \leq ArtM < 0.15403269$ | 1110 |
| $0.03196467 \leq ArtM < 0.03850817$ | 0111 | $0.15403269 \leq ArtM$ | 1111 |

**Normalization of magnitude of ART coefficients and quantization**

After finding the ART coefficients, the magnitudes of each ART coefficient is calculated. The magnitudes of ART coefficient are used as descriptor because of their rotation invariance property. The magnitudes of coefficents are normalized with $F_{00}$ which is the highest valued element.

$$ArtM[n,m] = \mid F_{nm} \mid / \mid F_{00} \mid \tag{4.7}$$

For keeping the descriptor size minimum, quantization is applied to each coefficient using four bits per coefficient. Quantization table is constructed based on an exponential distribution model [36]. The resulting quantized magnitude of ART coefficients is called ART descriptors. Since $ArtM[0,0]$ is 1 after normalization, it is not used. Therefore, the descriptor is made up of an array of (n*m)-1 normalized and quantized magnitudes of ART coefficients of a shape.

### 4.2.3 Similarity matching with ART

The similarity distance of two shapes is calculated by summing up the absolute differences of two sets of reconstructed values of ART descriptors in Table 4.2. Thus, the similarity distance between two shapes A and B is determined by:

$$D(A,B) = \sum_{i=0}^{n*m-1} \mid InverseQuantize(ArtDE_A[i]) - InverseQuantize(ArtDE_B[i]) \mid \tag{4.8}$$

Table 4.2: Reconstruction values for ART descriptors.

| ArtDE | Quantisation center |
|---|---|
| 0000 | 0.001763817 |
| 0001 | 0.005468893 |
| 0010 | 0.009438835 |
| 0011 | 0.013714449 |
| 0100 | 0.018346760 |
| 0101 | 0.023400748 |
| 0110 | 0.028960940 |
| 0111 | 0.035140141 |
| 1000 | 0.042093649 |
| 1001 | 0.050043696 |
| 1010 | 0.059324478 |
| 1011 | 0.070472849 |
| 1100 | 0.084434761 |
| 1101 | 0.103127662 |
| 1110 | 0.131506859 |
| 1111 | 0.192540857 |

### 4.2.4 Properties of ART Transform

With the algorithm defined for ART transform, ART has the desirable shape description properties of translation, scaling and rotation invariance. Translation and scaling invariance property comes with the preprocessing step where the centroid of the shape is aligned to coincide with that of the LUT table and the size of the shape is scaled to be enclosed in the circle with the diameter of the LUT size. The rotation invariance is achieved by taking the magnitudes of ART coefficients as descriptor. The magnitudes of any ART coefficients of a rotated shape and a reference shape are equal as shown in Appendix 5.

# CHAPTER 5

# SIMULATIONS AND RESULTS

## 5.1 Simulations

In the previous two chapters, the contour shape descriptor CSSD and region shape descriptor ARTD are explained in detail. The descriptor extraction methods described in these chapters are realized by software developed using C++ language with Borland C++ Builder. In order to give a good understanding of the descriptor extraction methods for CSSD and ARTD, they are implemented separately and then combined in a third program which analyzes a group of image similarity: the user can choose a binary image and ask the system to find all images similar to it using these contour and region based approaches.

The performance of both descriptors are examined through a number of experiments. Performance evaluation of contour-based descriptor CSSD and region-based descriptor ARTD was in mainly three tests: Similarity based retrieval, robustness to scaling, robustness to rotation.

For the similarity based retrieval tests, 1000 images from MPEG-7 Core Experiments-

Shape-1-B dataset, one of the test-sets that were defined during the standardization process of MPEG-7 [35] are used.The dataset consists of binary masks of natural objects with significant shape variability and arbitrary shape distortions including rotation, scaling, arbitrary skew and streching, defection.

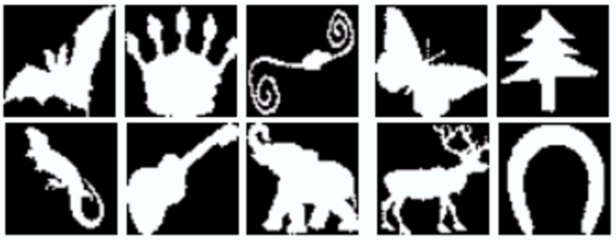


Figure 5.1: Some of the images in similarity based retrieval dataset.

The data set used consists of 50 classes of similar shapes with 20 images in each class. Some classes are illustrated in Figure 5.1. While there is a variability of shapes in each class, the degree of similarity between each class is greater than the similarity between two shapes from different classes. Using this property, a similarity retrieval performance can be evaluated: a query with a shape from, say class x , should ideally return all the remaining shapes in that class at top ranks, before shapes from other classes.

Considering each shape as an input query, we observed n outputs of the system and found the number of correctly retrieved shapes, m. A shape is considered to be correctly retrieved if it is in the same class with the query shape. The success rate of a query is calculated :

$$Success\ rate\ for\ a\ query = \frac{m}{m_{max}} \times 100$$

where $m_{max}$ is the maximum possible value of m in n outputs of the system. The

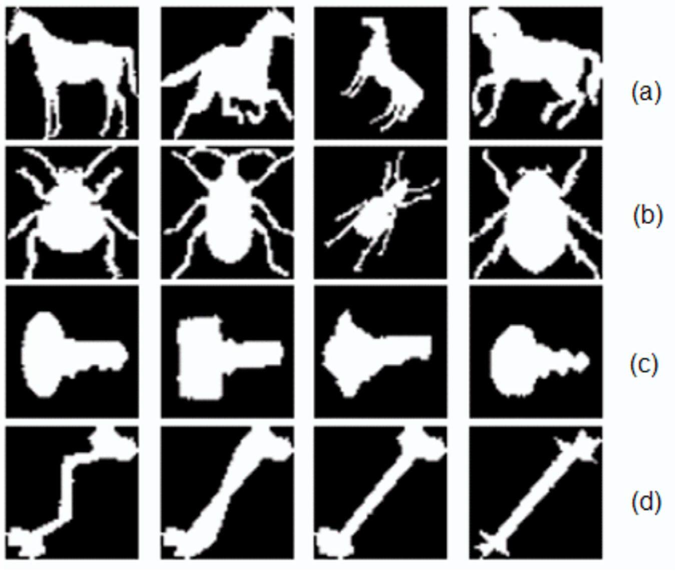


Figure 5.2: Example shape classes (a)horses, (b)beetles, (c)keys and (d)bones.

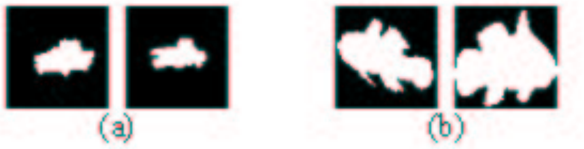


Figure 5.3: Shapes from differenet classes (a)car and truck, (b)imfish and fish.

success rate of the descriptor for the whole dataset is the average of the success rates for each query. Note that in the dataset, there are different classes with similar shapes. Therefore, one does not expect to reach a retrieval rate of 100 % (examine Figure 5.3).

### 5.1.1 The effect of the order of ART coefficients to similarity based retrieval performance

The number of angular and radial functions of the ART descriptor determines how accurately the shape is represented. If the selected number of ART descriptors is not large enough, it will affect retrieval performance. However, if the number of ART coefficients is larger than necessary, it will cost more storage area and feature extraction time.Therefore, retrieval using different number of ART coefficients is studied. The retrieval performance achieved by using different numbers of angular and radial functions is presented in Table 5.1.

Table 5.1: Retrieval results of the ART descriptor obtained by employing different number of angular and radial basis functions.

| n | m | processing time (ms) | Top 20(%) | Top 30(%) | Top 40(%) |
|---|---|---|---|---|---|
| 2 | 3 | 79,86 | 21.58 | 27.34 | 32,07 |
| 2 | 6 | 99.45 | 40.21 | 46.61 | 51.04 |
| 2 | 12 | 157.71 | 44.00 | 50.13 | 54.47 |
| 2 | 24 | 361.35 | 43.73 | 49.48 | 53.65 |
| 3 | 3 | 86.53 | 30.71 | 37.06 | 41.54 |
| 3 | 6 | 119.95 | 45.66 | 51.63 | 55.82 |
| 3 | 12 | 228.17 | 47.82 | 53.37 | 57.45 |
| 3 | 24 | 581.27 | 47.00 | 52.58 | 56.55 |
| 5 | 3 | 112.44 | 38.92 | 44.95 | 48.75 |
| 5 | 6 | 193.86 | 48.38 | 54.33 | 57.98 |
| 5 | 12 | 502.32 | 49.34 | 54.66 | 58.45 |
| 5 | 24 | 930.00 | 48.59 | 53.83 | 57.70 |
| 10 | 3 | 192.29 | 41.16 | 46.37 | 49.81 |
| 10 | 6 | 501.56 | 48.68 | 53.86 | 57.46 |
| 10 | 12 | 960.84 | 49.32 | 54.66 | 58.24 |
| 10 | 24 | 1870.00 | 48.93 | 54.15 | 57.76 |

From Figure 5.4 it is observed that the retrieval performance does not improve significantly above the angular basis m=12 for all the selected values of n. The radial complexity of the basis functions above n=5 does not improve retrieval performance for the dataset, either. As far as the retrieval performance is concerned m=12, n=5 gives the highest performance with 58.45 % retrieval success.

From the Figure 5.5, for the (m,n) values of ARTD performing best for the dataset, the ones with retrieval performance over 55 % are good candidates. From the point of

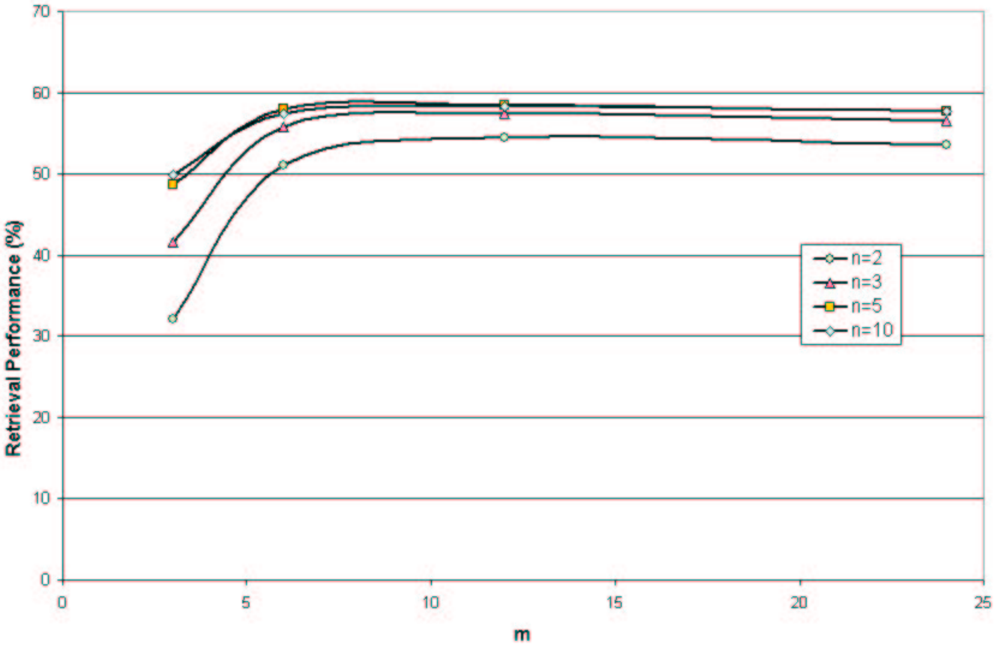


Figure 5.4: Effect of (m,n) on retrieval performance

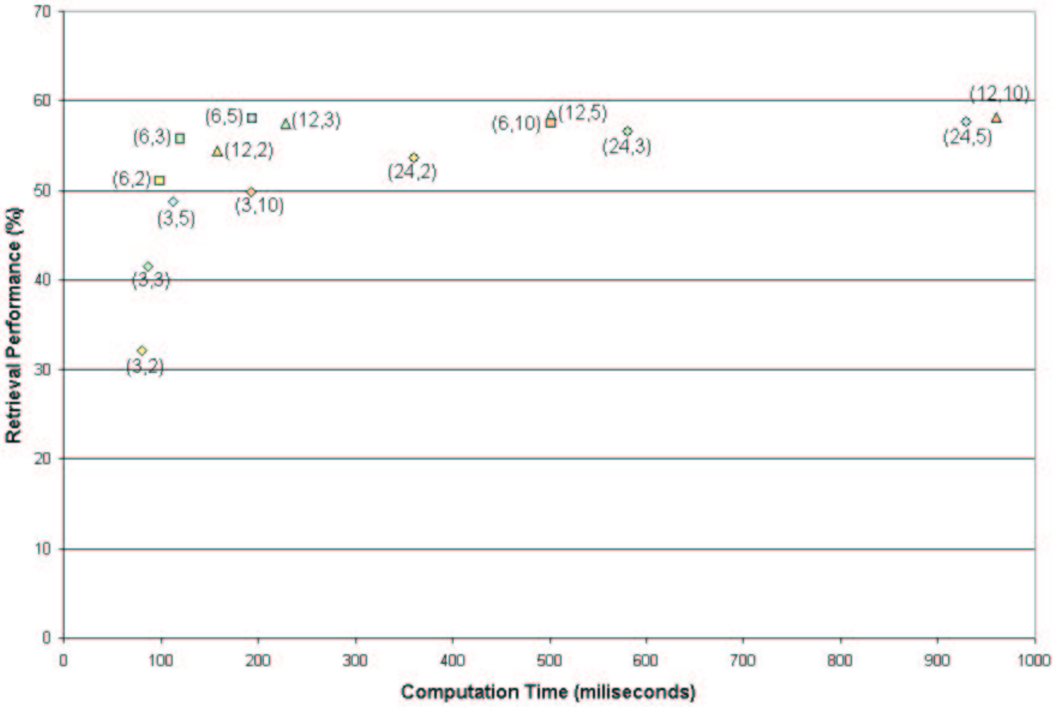


Figure 5.5: Effect of (m,n) on retrieval performance and computation time

view of feature extraction time and storage area, the ones with the smallest number of coefficients are advantageous. Between (6,5) and (12,3) which have a very small performance difference, it can be said that employing 6 angular and 5 radial basis functions offers a good tradeoff between retrieval performance (57.98 %) and compactness of the descriptor (number of basis functions = 30).

### 5.1.2 The effect of number of sampling points in CSS to similarity based retrieval performance

In the algorithm to find the CSS representation of a binary image, after finding boundary points, the next step is to sample the boundary points into a fixed number of points, N. Then these points are used for low-pass filtering of the boundary. Often in literature, the number of sampled points is the power of two and is usually small, for example, 32 points or 64 points. However there is a concern that the sampling may result in loss of boundary features. To see its effect, similarity retrieval experiments using CSSD are conducted for varying number of sampling points N = 32, 64, 128, 256 ,512.

Table 5.2: Retrieval performance with changing number of sampling points

| N | time(ms) | Top 20(%) | Top 30(%) | Top 40(%) |
|---|---|---|---|---|
| 32 | 109.42 | 41.76 | 50.21 | 56.24 |
| 64 | 116.88 | 37.32 | 44.79 | 50.30 |
| 128 | 155.29 | 34.25 | 41.76 | 47.67 |
| 256 | 517.22 | 31.95 | 38.37 | 42.75 |
| 384 | 1263.81 | 32.32 | 38.32 | 43.32 |
| 512 | 2316.82 | 32.28 | 38.86 | 43.59 |

From the results of the simulation, it is seen that the coarser boundary feature extraction results with a higher classification performance. With a fine sampling of the boundary, the small differences of shapes in the same class also begin to be represented in the descriptor. For the calculation of similarity distance between peaks of CSS images, these insignificant peak characteristics are added to unmatched peaks and increase the similarity distance.

### 5.1.3 Similarity based retrieval performances of CSSD and ARTD

The similarity based retrieval performances of CSSD and ARTD are shown in Table 5.3. The orders of n=5, m=6 for ART basis functions and for the boundary N=32 resampled points are used respectively for ART and CSS representation methods. In order to compare the computation times of the descriptors, the feature extraction and retrieval times are tested on the Windows platform of 1.8 GHz AMD PC with 512 MB memory. Note that further speed optimization of the code can reduce proccessing times. The time taken for the feature extraction and the retrieval of the shape database using developed software is given in Table 5.4

Table 5.3: Similarity based retrieval performance of CSSD and ARTD using N=32, (n,m)=(5,6) in top 20, 30, 40 retrievals.

| | Top 20(%) | Top 30(%) | Top 40(%) |
|---|---|---|---|
| CSSD | 41.755 | 50.21 | 56.24 |
| ARTD | 48.375 | 54.33 | 57.98 |

Table 5.4: Computation times for CSSD and ARTD using N=32, (n,m)=(5,6)

| | Average feature extraction time of each shape (ms) | Average similarity measurement time (ms) |
|---|---|---|
| CSSD | 109.415 | 4.1 |
| ARTD | 193.86 | 1.76 |

### 5.1.4 Rotation Effect

For testing the descriptors against rotation variance, a database of 250 shapes is used. The robustness of each approach to changes of the shape due to digital rotation and to the quantization of the representations is investigated. In the dataset, there are 50 basic shapes from MPEG-7 CE1 set and four derived shapes from each basic shape.These shapes are digitally rotated versions of each shape by angles 9, 45 ,80, 150 degrees. Each of 250 shapes are used for query. The number of correct matches was computed in the top 5,7,10 retrievals.

Table 5.5: Retrieval performance for rotation set in top 5, 7, 10 retrievals.

| | Top 5(%) | Top 7(%) | Top 10(%) |
|---|---|---|---|
| CSSD | 80.48 | 86.56 | 89.6 |
| ARTD | 99.92 | 100 | 100 |

### 5.1.5 Scaling Effect

For testing the descriptors to changes of the shape due to digital scaling and quantization of the representations, a database of 250 shapes is used. From 50 basic shapes, four more shapes are derived by scaling each basic shape by factor 0.2, 0.4, 0.6 and 2. Each of 250 images are used for query. The number of correct matches was computed in the top 5 ,7 ,10 retrievals.

Table 5.6: Retrieval performance for scaling set in top 5, 7, 10 retrievals.

| | Top 5(%) | Top 7(%) | Top 10(%) |
|---|---|---|---|
| CSSD | 77.6 | 83.44 | 85.52 |
| ARTD | 90.16 | 93.12 | 94.24 |

## 5.2 Results

Results of the simulations reveal some important properties of descriptors CSSD and ARTD. In this section, the properties of the descriptors will be discussed.

It was mentioned that both representations are invariant to translation, rotation and scale. As we deal with digital images, for example digital scaling of images to a small size as compared with the sampled grid may result significant distortions in their shape. In order to test descriptor behaviour to such effects, rotation and scaling experiments are done. It can be seen from rotation and scaling tests (Table 5.5 and Table 5.6) that ARTD offers very good rotational and scaling invariance. CSSD scaling behaviour is also acceptable (approximately 4 successful retrievals in the first 5) as some images are severely distorted when scaled by factor of 0.2.

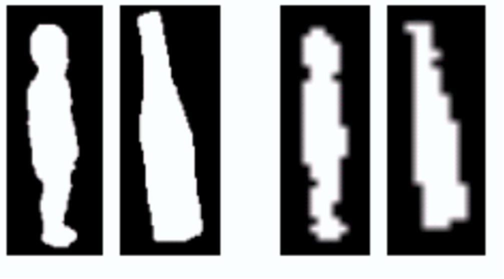


Figure 5.6: Shapes of child and bottle and their scaled-down versions by 0.2.

Other properties of CSSD are as follows:

- It is robust to mirroring: invariance to mirroring of the shape contour is considered by mirroring the significant peaks in similarity measurement algorithm.
- It is robust to noise: noise influence is eliminated by thresholding short peaks in CSS representation.
- It is robust to significant non-rigid deformations. (Figure 5.7-a)

- It can find the strong similarity between shapes distorted due to perspective transformations.(Figure 5.7-b)

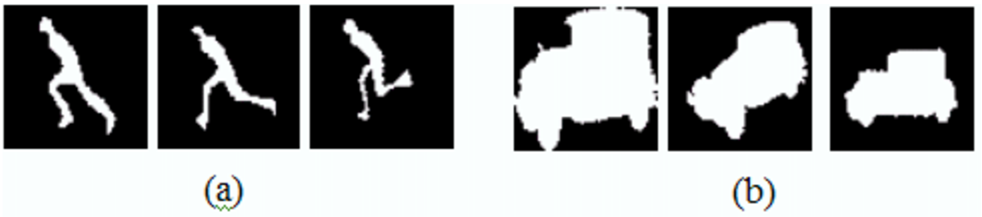


Figure 5.7: Examples of shapes for CSSD properties.

The properties of ARTD are as follows:

- It can describe objects with or without holes, as well as shapes consisting of disconnected regions (Figure 5.8-a).
- It can cope with errors in segmentation in which an object is split into disconnected subregions (Figure 5.8-b).
- It is robust to minor non-rigid deformations shown as in Figure 5.8-c .

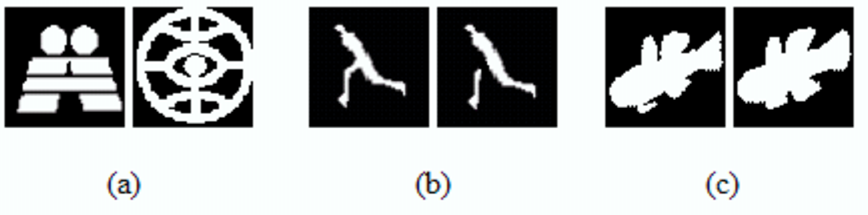


Figure 5.8: Examples of shapes for ARTD properties.

For similarity based retrieval, it is observed from the simulations that the region based ARTD performed better on average about 4 % compared with CSSD using the best performing parameter for N and n,m. While for the top 20 retrievals , ARTD gives better performance, for top 40 retrievals, the performances of the descriptors are very close. It is observed from the test results that for some classes of shapes CSS performed better. A class-by-class analysis illustrates the differences between contour and region based similarity notions. Considering the shape classes in which CSSD performance is better than ARTD by 10 % or more, some example classes are shown in Figure 5.9 .

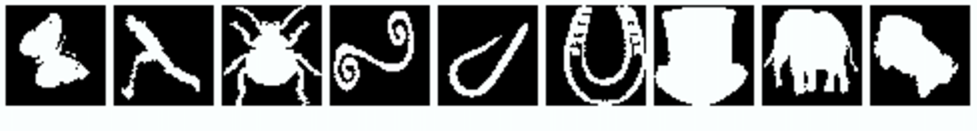


Figure 5.9: Example of shapes where contour descriptor success rate is better than region descriptor.

For example for *Butterfly*, some characteristic features are in its antenna, which have a relatively small area and not sufficiently emphasized. The second class includes images of a running person where significant non-rigid deformations occur. CSSD is more robust to such deformations. *Beetles* and *Elephants* are examples in which the variability of the shapes is better captured by the contour rather than the region.

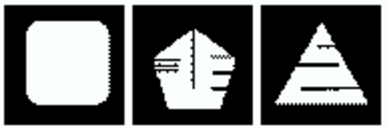


Figure 5.10: Example of shapes where region descriptor success rate is better than contour descriptor.

Three example classes in which ARTD has significantly better performance than CSSD are shown in Figure 5.10. In this classes basic shapes like square (Device-3), triangle (Device-4) and pentagon (Device-6) is cut out variable number of openings. Such shapes have a similar distribution of pixels, but different contours, because the number and shape of the openings are different in class members.

A disadvantage of the CSSD is for convex shapes. A convex shapes contour carries a very little characteristic information that can be represented by CSS. For example, CSSD algorithm can not represent a perfect circle because there are no curvature zero-crossings. For the rounded square in Figure 5.10, CSSD can be calculated however CSS may not discriminate it from other convex objects.

As for dimensions of descriptors, we can say that both descriptors are compact. Once the order of ART basis function is chosen, the number of ART features is constant $(n*m-1)$ The dimension of CSS descriptor varies with each shape according to contour complexity (number of significant peaks in CSS representation). For the similarity retrieval dataset, the average number of CSSD features needed to describe shape is seven with eccentiricity and circularity values.

The computation process of CSSD is more complex than that of ARTD. The extraction of the CSSD feature takes three processes i.e equal arc-length parametrization, CSS image computation , CSS peak extraction. For ARTD, once the basis functions are calculated , size normalization is done and ART coefficients are extracted.

In the algorithms used for extracting descriptors from shapes, among the parameters and thresholds used ,the order of the ART basis functions and the number of equidistant points used for sampling of a contour are investigated. For ARTD, the only parameter used is the number of the basis functions. On the other hand, CSSD extraction uses many parameters : the number of the sampling points, the thresholds to eliminate short peaks, thresholds for eccentiricity, circularity, the tolerance value for peak position matching. When more parameters are involved in the descriptor extraction algorithm, the retrieval performance of the descriptor depends more on these parameters. These parameters are generally defined emprically for a set of databases. For general purposes and wide variety of databases the defined parameter may not give the best retrieval performances. Therefore , ARTD performance can be considered to be more stable for general applications.

From the aspect of online matching computation, ART is simpler, the similarity measurement calculation of ARTD is done by calculating the L1 distance between two feature vectors. The online matching of two sets of CSSD involves finding the matching and unmatching peaks, calculating the L2 distance between matching peaks and adding additional penalty for unmatched peaks. Because of these reasons, similarity matching time for CSSD takes longer than ARTD as shown in Table 5.4.

# CHAPTER 6

# CONCLUSION

## 6.1 Conclusions

The user of an image database often wishes to retrieve all images similar to the one (s)he already has. In this thesis, techniques for similar shape retrieval are investigated. Generally, similar shape retrieval has three issues - shape representation, similarity measure, indexing structure. This thesis is mainly focused on shape representation. State of art shape representation methods have been reviewed. The purpose of the review is to understand the problems and issues involved in shape representation, to identify advantages and disadvantages of different shape descriptors.

Generally, there are two types of approaches in shape representation : contour-based and region-based. Contour-based approches make use of the boundary points of shape while region-based methods depend on the pixel distribution of the shape. Among contour-based representation methods, for image description and retrieval, using Curvature Scale Space representation is promising. The CSS representation works by decomposing the shape into convex and concave sections. CSS descriptor is extracted by using the curvature zero-crossings behaviour of the contour while smoothing the contour with low-pass filters.Among region based shape representations Angular Radial Transform which belongs to a class of moment based techniques is the most promising. The ART descriptor works by decomposing the shape into a number of orthogonal 2-D basis functions defined on a unit disk.

The CSSD and ARTD shape feature extraction methods are implemented for a similar shape retrieval system by software developed in C++. For similarity retrieval performances of descriptors, a dataset consisting of 1000 images with significant shape variability and distortions is used.

ARTD has the main strength of application independence. It can describe any object with holes, without holes , shapes consisting of disjoint regions and convex shapes. CSSD can also be used for describing holes in a shape or disjoint parts of shapes, however, similarity measurement of such shapes with CSSD will involve additional computation. Convex shapes can not be discriminated from each other using CSSD because of the insufficient characteristic information of the contour.

CSSD's main strength is to emulate well the shape similarity perception of the human visual system. Many objects have distinguishing features in their contour while having similar region properties.The contour-based descriptor CSSD performs better than ARTD for objects for which characteristic shape features are contained in their contour. Both descriptors have advantages and disadvantages. When the choice of approach depends on the properties of the query shape, similarity retrieval performance will increase.

In similar shape retrieval, one of the issues is to describe and capture essential properties of any shape. As far as the usage for general application is concerned in which query shape property is not known , using both contour-based and region-based description of shapes can be meaningful. Therefore, for general applications, if instead of choosing one approach to the other, a method for employing both descriptors to shape similarity measurement is considered, higher similarity retrieval performances can be achieved.

## 6.2 Future Work

Content-based image retrieval is a difficult task. There exists several directions in this area for future work by researchers. While employing shape description methods to images, it is assumed that the shape object has already been available and existed in binary form. In reality , segmentation is a main issue for CBIR systems. The shapes of segmented objects from general image can be more complex and inaccurate than shapes from a homogenous shape database. Therefore, a practical shape description and retrieval system should take segmented objects into consideration.

One of the important issues is to describe image using higher level features. Combining low level features shape, color and texture will make it possible to reach semantic descriptions of images leading to more effective searches in CBIR systems.

Retrieval effectiveness is the main focus in this thesis. Retrieval efficiency issue has been avoided due to the limited scope of the research. Retrieval efficiency is a crucial factor in image retrieval,considering image databases are distributed and large in characteristic. Further research is needed to organize the feature indices into efficient data structure for remote and fast retrieval.

# REFERENCES

[1] T. Pavlidis, "A review of algorithms for shape analysis," *Comput. Graphics Image Process*, vol. 7, pp. 243–258, 1978.

[2] D.Marr and H.Nishiara, "Representation and recognition of the spatial organization of three dimensional shapes," in *Proc.Roy.Soc.London, B200*, pp. 269–294, 1978.

[3] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE PAMI-8(1)*, pp. 34–43, 1986.

[4] F. Mokhtarian and A. Mackworth, "Theory of multiscale, curvature based shape representation for planar curves," *IEEE PAMI-14(8)*, pp. 789–805, 1992.

[5] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision.* Chapman and Hall Computing, 1993.

[6] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities.* Academic Press, 1997.

[7] P. J. van Otterloo, *A Contour-Oriented Approach to Shape Analysis*, pp. 90–108. Prentice Hall International (UK) Ltd, 1991.

[8] C.Zahn and R.Roskies, "Fourier desciptors for plane closed curves," *Comput. Graphics Image Process*, vol. 21, pp. 269–281, 1978.

[9] A. P. Witkin, "Scale space filtering," in *Proc. 8th. Int. Joint Conf. On Artificial Intelligence*, pp. 1019–1022, 1983.

[10] J. Babaud, A. Witkin, M. Baudin, and R. Duda, "Uniqueness of the gaussian kernel for scale space filtering," *IEEE PAMI-8(1)*, pp. 26–33, 1986.

[11] H. Asada and M. Brady, "The curvature primal sketch," *IEEE PAMI-8(1)*, pp. 2–14, 1986.

[12] F. Mokhtarian, S. Abbasi, and J. Kittler, "Robust and efficient shape indexing through curvature scale space," in *Proc. British Machine Vision Conference, Edinburgh, UK*, pp. 53–62, 1996.

[13] F. Mokhtarian, S. Abbasi, and J. Kittler, "Efficient and robust retrieval by shape content through curvature scale space," in *Int. Workshop on Image DataBases and Multimedia Search, Amsterdam, The Netherlands*, pp. 35–42, 1996.

[14] T. Pavlidis, *Algorithms for Graphics and Image Processing.* Springer Berlin, 1982.

[15] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. on Electronic Computer*, vol. EC-10, pp. 260–268, 1961.

[16] H. Freeman and A. Saghri, "Generalized chain codes for planar curves," in *Proceedings of the 4th International Joint Conference on Pattern Recognition, Kyoto, Japan, November 7-10*, pp. 701–703, 1978.

[17] K. S. Fu, *Syntactic Methods in Pattern Recognition.* Academic Press, 1974.

[18] M. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, vol. 8, pp. 179–187, 1962.

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

[20] B. Jahne, *Digital Image Processing Concepts, Algorithms and Scientific Applications.* Springer-Verlag Berlin, 1995.

[21] M. R. Teague, "Image analysis via the general theory of moments," *Journal of Optical Society of America-70(8)*, pp. 920–930, 1980.

[22] C. H. Teh and R. T. Chin, "On image analysis by the methods of moments," *IEEE Trans. PAMI-10(4)*, pp. 496–513, 1988.

[23] S. X. Liao and M. Pawlak, "On image analysis by moments," *IEEE Trans. PAMI-18(3)*, pp. 254–266, 1996.

[24] Y. S. Kim and W. Y. Kim, "Content-based trademark retrieval system using a visually salient feature," *Image and Vision Computing, 16*, pp. 931–939, 1998.

[25] Y. S. Kim and W. Y. Kim, "A region-based shape descriptor using zernike moments. signal processing," *Image Communication, 16*, pp. 95–102, 2000.

[26] G. J. Lu and A. Sajjanhar, "Region-based shape representation and similarity measure suitable for content-based image retrieval," *Multimedia Systems, 7(2)*, pp. 165–174, 1999.

[27] K. Chakrabarti, M. O. Binderberger, K. Porkaew, and S. Mehrotra, "Similar shape retrieval in mars," in *Proc. of IEEE Int. Conf. On Multimedia and Expo (CD-ROM), New York, USA*, 2000.

[28] M. Safar, C. Shahabi, and X. Sun, "Image retrieval by shape: A comparative study," in *Proc. of IEEE Int. Conf. On Multimedia and Expo (CD-ROM), New York, USA*, 2000.

[29] A. Goshtasby, "Description and discrimination of planar shapes using shape matrices," *IEEE Trans. PAMI-7*, pp. 738–743, 1985.

[30] S.Parui, E.Sarmaand, and D.Majumder, "How to discriminate shapes using the shape vector," *Pattern Recognition Lett., 4*, pp. 201–204, 1986.

[31] S. Loncaric, “A survey of shape analysis techniques,” *Pattern Recognition, 31(8)*, pp. 983–1001, 1998.

[32] T. Lindeberg, *Scale-space Theory in Computer Vision*, pp. 8–15. Kluwer Academic, Boston, 1994.

[33] D. Ballard and C. Brown, *Computer Vision.* Prentice Hall, NJ, 1982.

[34] W. Kim and Y. Kim, “A rotation invariant geometric shape descriptor using zernike moment,” *ISO/IEC JTC1/SC29/WG11 P687*, 1999.

[35] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Language.* Wiley, New York, 2001.

[36] “Extraction and use of mpeg-7 description,” *ISO/IEC JTC1/SC29/WG11 N4579*, pp. 289–296, 2002.

# APPENDIX A

## 1. Contour Tracing

The following is a formal description of the square tracing algorithm used for tracking boundary points of a shape:

**Input**: A square tesellation **T**, containing a connected component **P** of black cells.

**Output**: A sequence $B(b_1, b_2, ..., b_k)$ of boundary pixels i.e. the contour.

Begin

- Set **B** to be empty.
- From left to right and top to bottom scan the cells of **T** until a black pixel, s, of **P** is found.
- Insert **s** in **B**.
- Set the current pixel, **p**, to be the starting pixel, **s**.

```
• Turn left i.e. visit the left adjacent pixel of p.
• Update p i.e. set it to be the current pixel.
• While bf p not equal to bf s do
    – If the current pixel p is black
        ∗ Insert p in B and turn left (visit the left adjacent pixel of p).
        ∗ Update p i.e. set it to be the current pixel.
    – else
        ∗ Turn right (visit the right adjacent pixel of p).
        ∗ Update p i.e. set it to be the current pixel.
• end While
End
```

**2. Curvature Derivation**

Let **r** be a smooth parametrization of a smooth curve $C$ with tangent **T** and normal **N**. The velocity and acceleration of an object moving along $C$ with position **r** are given by

$$v = \|v\|T \quad and \quad a = a_T T + a_N N$$

Then since $T \times T = 0$ we find that

$$\begin{aligned} v \times a &= (\|v\|T) \times (a_T T + a_N N) \\ &= [\|v\|T \times (a_T T)] + [\|v\|T \times (a_N N)] \end{aligned}$$

$$= (\|v\| a_N)(T \times N)$$

Since $a_N = \|v\| \|dT/dt\|$ and since **T** and **N** are perpendicular unit vectors,

$$\|v \times a\| = \|v\| a_N = \|v\|^2 \|\frac{dT}{dt}\|$$

However,

$$\kappa = \frac{\|dT/dt\|}{\|dr/dt\|} = \frac{\|dT/dt\|}{\|v\|}$$

so that

$$\kappa = \frac{\|v \times a\|}{\|v\|^3}$$

**r** represents an object moving on a curve in the $xy$ plane, we have

$$\begin{aligned} r &= xi + yj \\ v &= \frac{dx}{dt} i + \frac{dy}{dt} j = x'(t)i + y'(t)j \\ a &= \frac{d^2x}{dt^2} i + \frac{d^2y}{dt^2} j = x''(t)i + y''(t)j \end{aligned}$$

Therefore curvature $\kappa$ becomes:

$$\kappa = \frac{\mid x'(t)y''(t) - x''(t)y'(t) \mid}{[(x'(t))^2 + (y'(t))^2]^{3/2}}$$

### 3. Peak Transformation and Quantization

After finding the peaks of the CSS image, the x_css and y_css coordinates are transformed using a nonlinear transformation and quantized.

Re-scale x_css coordinates to lie in the range [0.0, 1.0]. The re-scaled coordinates of the peaks are referred to as xpeak[i]. The transformation and quantization algorithm [36] used is as below:

- Transform all peak heights according to the equation $ypeak[i] = 3.8 \times (\frac{y\_css[i]}{Nsamples^2})^{0.6}$, where $Nsamples$ is the number of equi-distant points from the contour used for smoothing.
- Shift all peaks so that the highest peak after transformation is at the x_css coordinate 0.0.
- If the highest peak has a height of less than 0.09, remove all peaks.
- For any peaks which have a height of less than $ypeak[0] \times 0.05$, remove them.
- Linearly quantize all X-coordinates of the peaks to 6-bits.
- Linearly quantize the first (maximum) peak using a maximum value of 1.7, and 7-bits. If the height of this first peak is greater than 1.7, it will be saturated to 1.7.
- Linearly quantize the other peaks, in descending order of height, using a maximum value of the previous peak height (quantized), and 3bits.

### 4. Angular Radial Transform

The algorithm for extracting ART magnitudes of a shape are given below for m=12, n=3.

- **Step 1 - Basis Function Generation:** Since the basis functions are separable, $V_{nm}(x, y)$ is computed directly in Cartesian coordinate rather than converting it after computing $V_{nm}(\rho, \theta)$ in polar coordinate. To do so, first construct a set of complex basis functions of ART in two 4-dimensional arrays, $BasisR[12][3][LUT\_SIZE][LUT\_SIZE]$ and $BasisI[12][3][LUT\_SIZE][LUT\_SIZE]$ for real and imaginary part, respectively.

```
$cx = cy = LUT\_SIZE/2;$ // center of basis function
$for(y = 0;\ y < LUT\_SIZE;\ y++)$
$for(x = 0; x < LUT\_SIZE;\ x++)\{$
          $radius = sqrt((x - cx) * (x - cx) + (y - cy) * (y - cy));$
          $angle = atan2(y - cy, x - cx);$
          $for(m = 0;\ m < 12;\ m++)$
          $for(n = 0;\ n < 3;\ n++)\{$
                    $temp = cos(radius * \pi * n/(LUT\_SIZE/2));$
                    $BasisR[m][n][x][y] = temp * cos(angle * m);$
                    $BasisI[m][n][x][y] = temp * sin(angle * m);$
          $\}$
$\}$
```

- **Step 2 - Size Normalization**: The center of mass or the centroid of the object in the image is aligned to coincide with that of the lookup table. If the size of the image and that of lookup table are different, linear interpolation is applied to map the image onto the corresponding lookup table. Here, the size of the object is defined as twice the maximum distance from the centroid of the object.

- **Step 3 - ART Transformation**: The real and imaginary parts of ART coefficients, $ArtR[12][3]$ and $ArtI[12][3]$ are computed by summing up the multiplication of a pixel in an image to each corresponding pixel in the lookup table in raster scan order, respectively. Twelve angular and three radial basis functions are used.

```
$for(y = 0;\ y < LUT\_SIZE;\ y++)$
$for(x = 0;\ x < LUT\_SIZE;\ x++)\{$
        $for(m = 0;\ m < 12;\ m++)$
        $for(n = 0;\ n < 3;\ n++)\{$
                $ArtR[m][n]+ = Image[x][y] * BasisR[m][n][x][y];$
                $ArtI[m][n]- = Image[x][y] * BasisI[m][n][x][y];$
        $\}$
$\}$
```

- **Step 4 - Area Normalization**: For area normalization, the magnitudes of ART coefficient $ArtM[m][n]$ is divided by $ArtM[0][0]$ which corresponds to the area of an object.

```
$for(m = 0;\ m < 12;\ m++)$
$for(n = 0;\ n < 3;\ n++)$
        $ArtM[m][n] = sqrt(ArtR[m][n] * ArtR[m][n] + ArtI[m][n] * ArtI[m][n]);$
$for(m = 0;\ m < 12;\ m++)$
$for(n = 0;\ n < 3;\ n++)$
        $ArtM[m][n] = ArtM[m][n]/ArtM[0][0];$
```

- **Step 5 - Quantization**: The magnitudes of ART coefficients excluding $ArtM[0][0]$ are then quantized to 4 bits or 16 levels using the following quantization table as in Table 4.1 . The resulting quantized magnitude of ART coefficients is called MagnitudeOfART.

```
$index = 0;$
$for(m = 0;\ m < 12;\ m++)$
$for(n = 0;\ n < 3;\ n++)$
          $if(m! = 0 || n! = 0) MagnitudeOfART[index++] = Quantize(ArtM[m][n]);$
$cx = cy = LUT\_SIZE/2;$ // center of basis function
$for(y = 0;\ y < LUT\_SIZE;\ y++)$
$for(x = 0; x < LUT\_SIZE;\ x++)$
{
        $radius = sqrt((x - cx) * (x - cx) + (y - cy) * (y - cy));$
        $angle = atan2(y - cy, x - cx);$
        $for(m = 0;\ m < 12;\ m++)$
        $for(n = 0;\ n < 3;\ n++)$
        {
                $temp = cos(radius * \pi * n/(LUT\_SIZE/2));$
                $BasisR[m][n][x][y] = temp * cos(angle * m);$
                $BasisI[m][n][x][y] = temp * sin(angle * m);$
        }
}
```

## 5. Proof of Rotation Invariance of ART Magnitudes

Let the image $f^{\alpha}(\rho,\theta)$ be the rotated image of $f(\rho,\theta)$ by an angle $\alpha$ about its origin,

$$f^{\alpha}(\rho,\theta) = f(\rho,\alpha+\theta).$$

The ART of the rotated image are the given as

$$F_{nm} = \int_0^{2x}\int_0^1 V_{nm}^{*}(\rho,\theta), f(\rho,\alpha+\theta)\rho d\rho d\theta$$

or

$$F_{nm}^{\alpha} = F_{nm}exp(-jm\alpha)$$

Here, it is seen that the magnitude of the ART of rotated image and the reference image is the same.

$$\|F_{nm}^{\alpha}\| = \|F_{nm}\|$$