# SHAPE : REPRESENTATION, DESCRIPTION, SIMILARITY AND RECOGNITION

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

BY

NAFİZ ARICA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

SEPTEMBER 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 Doctor of Philosophy.

Prof. Dr. Ayşe Kiper
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 Doctor of Philosophy.

Prof. Dr. Fatoş Yarman - Vural
Supervisor

Examining Committee Members

Prof. Dr. Fatoş Yarman - Vural

Prof. Dr. Enis Çetin

Assoc. Prof. Dr. Volkan Atalay

Assoc. Prof. Dr. Gözde Bozdağı

Assoc. Prof. Dr. Sibel Tarı

# ABSTRACT

SHAPE : REPRESENTATION, DESCRIPTION, SIMILARITY AND RECOGNITION

Arıca, Nafiz
Ph.D., Department of Computer Engineering
Supervisor: Prof. Dr. Fatoş Yarman - Vural

September 2003, 157 pages

In this thesis, we study the shape analysis problem and propose new methods for shape description, similarity and recognition. Firstly, we introduce a new shape descriptor in a two-step method. In the first step, the 2-D shape information is mapped into a set of 1-D functions. The mapping is based on the beams, which are originated from a boundary point, connecting that point with the rest of the points on the boundary. At each point, the angle between a pair of beams is taken as a random variable to define the statistics of the topological structure of the boundary. The third order statistics of all the beam angles is used to construct 1-D Beam Angle Statistics (BAS) functions. In the second step, we apply a set of feature extraction methods on BAS functions in order to describe it in a more compact form. BAS functions eliminate the context-dependency of the representation to the data set. BAS function is invariant to translation, rotation and scale. It is insensitive to distortions. No predefined resolution or threshold is required to define the BAS functions.

Secondly, we adopt three different similarity distance methods defined on the BAS feature space, namely, Optimal Correspondence of String Subsequences, Dynamic warping and Cyclic Sequence Matching algorithms. Main goal in these algorithms is to minimize the distance between two BAS features by allowing deformations.

Thirdly, we propose a new Hidden Markov Model (HMM) topology for boundary based shape recognition. The proposed topology called circular HMM is both ergodic and temporal. Therefore, the states can be revisited in finite time intervals while keeping the sequential information in the string which represents the shape. It is insensitive to size changes. Since it has no starting and terminating state, it is insensitive to the starting point of the shape boundary.

Experiments are done on the dataset of MPEG 7 Core Experiments Shape-1. It is observed that BAS descriptor outperforms all the methods in the literature. The circular HMM gives higher recognition rates than the classical topologies in shape analysis applications.

Keywords: Shape Representation, Shape Description, Shape Similarity, Matching, Circular Hidden Markov Models

# ÖZ

ŞEKİL : SUNUM, BETİMLEME, BENZERLİK VE TANIMA

Arıca, Nafiz
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Fatoş Yarman - Vural

Eylül 2003, 157 sayfa

Bu tezde, şekil analizi üzerine çalışmalar yapılarak, şekil betimleme, şekil benzerliği ve tanıması için yeni yöntemler önerilmiştir. İlk olarak, iki boyutlu şekil bilgisini tek boyutlu fonksiyonlarla ifade eden yeni bir şekil betimleme yaklaşımı sunulmaktadır. Bu amaçla, iki aşamalı yöntem uygulanır. İlk aşamada, iki boyutlu şekil bilgisi çözünürlük seviyesinden bağımsız olarak tek boyutlu moment fonksiyonlarına dönüştürülür. Dönüşüm, bir sınır noktasını diğer tüm sınır noktalarına bağlayan kerterizlerin açı bilgisine dayanır. Şekil sınırı topolojik yapısının istatistiklerinin bulunmasında her sınır noktasında kerteriz çiftleri arasındaki açı, rastgele değişken olarak tanımlanır. Tüm kerteriz açılarının üçüncü derece istatistikleri ile tek boyutlu Kerteriz Açısı İstatistikleri (KAİ) fonksiyonları oluşturulur. İkinci aşamada, tek boyutlu KAİ fonksiyonları değişik öznitelik çıkarma yöntemleri kullanılarak sıkıştılır. KAİ foksiyonları, sunumun veri setine bağımlı olmasını ortadan kaldırmaktadır. Ayrıca, dönmeye, yer değişimine ve ölçeğe değişimsiz, bozunuma da hassas değildir.

İkinci olarak, KAİ öznitelik uzayında tanımlanan üç değişik benzerlik ölçümü yöntemi adapte edilmektedir. Bunlar, Dizinlerin Eniyi Eşleşmesi, Dinamik Eşleştirme ve Periyodik Seri Eşleştirme algoritmalarıdır. Bu algoritmalardaki temel amaç, bozunuma izin vererek iki KAİ özniteliği arasındaki uzaklığı anaza indirgemektir.

Üçüncü olarak, sınır tabanlı şekil tanıma için yeni bir Saklı Markov Modeli (SMM) topolojisi önerilmektedir. Dairesel SMM olarak adlandırılan topoloji, hem şekli ifade eden dizindeki sıra bilgisini saklamakta hemde model durumlarının belirli aralıklarla tekrar ziyaret edilmesine imkan sağlamaktadır. Önerilen Dairesel SMM vektör uzunluğuna hassas değildir. Ayrıca, başlangıç ve bitiş model durumu olmadığından, şekil sınırının başlangıç noktasına bağımsızdır.

MPEG-7 veri seti üzerinde yapılan deneylerde önerilen şekil betimleyicisi, raporlanan en iyi yöntemlerden daha iyi sonuçlar vermiştir. Dairesel SMM ise klasik topolojilerden daha yüksek tanıma oranına sahiptir.

Anahtar Kelimeler: Şekil Betimlemesi, Şekil Benzerliği, Şekil Tanıma, Eşleştirme, Dairesel Saklı Markov Modelleri

# ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my thesis supervisor, Prof. Dr. Fatoş Yarman Vural, for her tremendous effort in guiding me throughout my study. She did not only guide me in my academic life, but also supported me to draw the mission of my life that I always will follow.

I am grateful to Prof. Dr. Volkan Atalay, Prof. Dr. Gözde Bozdağı for all the guidance they gave me during my Ph.D. study.

This thesis is a product of Image Processing and Pattern Recognition Laboratory at METU, where the members at the time I studied, Pınar, Ebru, Neslişah, Özge, Turan, Murat, Ulaş, Aykut, Mutlu, Erkut, and Gülşah, supported me with their friendship and patience. I thank them all.

Thanks are also to Prof. Dr. Hüseyin Vural and Dr. Faruk Yarman for their kind and warm spiritual support.

Friendly athmosphere in the Department of Computer Engineering flourished me with their faculty members and staff. To be with them was one of the most important motivations to start such a hard journey.

I would like to give my sincere thankfullness to the Turkish Navy, my branch chiefs, Levent, Semih, Cengiz and my friends, Altay, Saadettin, Gürkan, Selcuk.

This thesis was only possible with the support of my father Mustafa and my mother Göksel. Lastly, thanks to my sister Ifakat, my brother Serdar and my dear nephew for their support and love.

To my family and to the memory of İfakat Arıca and Cemile Zehra Türker-Yarman

# TABLE OF CONTENTS

# LIST OF TABLES

TABLE

# LIST OF FIGURES

FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| 1-D | One-dimensional |
| 2-D | Two-dimensional |
| 3-D | Three-dimensional |
| AR | Autoregressive |
| ART | Angular Radial Transformation |
| AWT | Approximation by Wavelet Transforms |
| BAS | Beam Angle Statistics |
| CBIR | Content Based Image retrieval |
| CHMM | Circular Hidden Markov Models |
| CR | Character Recognition |
| CSC | Cyclic Sequence Comparison |
| CSS | Curvature Scale Space |
| CWT | Continuous Wavelet Transform |
| DAG | Directed Acyclic Graph |
| DC | Direct Coefficient |
| DW | Dynamic Warping |
| DWP | Dynamic Warping with Penalty |
| DWT | Dyadic Wavelet Transform |
| FD | Fourier Descriptor |
| FIR | Finite Impulse Response |
| HMM | Hidden Markov Models |
| HVS | Human Visual System |
| MAT | Medial Axis Transform |
| MPEG | Moving Picture Expert Group |
| MST | Morphological Signature Transform |
| OCS | Optimal Correspondence of String Subsequences |
| PA | Polyline Approximation |
| PCA | Piecewise Constant Approximation |
| PDL | Picture Description Language |
| PLAWTMM | Piecewise Linear Approximation by Wavelet Transform Modulus Maxima |
| QMF | Quadrature Mirror Filter |
| SFT | Sampling by Fourier Transformation |
| SOM | Self Organizing Map |
| WTMM | Wavelet Transform Modulus Maxima |

# CHAPTER 1

# INTRODUCTION

Machine simulation of human functions has been a very challenging research field since the advent of digital computers. One of these functions is visual information analysis or shortly vision, which is not only an attribute of humans, but also a major characteristic in wide range of animals. Vision, more than any other sense, provides us with information about the surrounding environment and guides most of our activities and interactions. One of the major goal, in computer vision, is to take an image (or sequence of images) from a scene and process it in such a way that relevant information (depending on the task of application) about the scene is inferred. Vision has been motivated by a wide range of important applications such as robotics, biology, Internet, document analysis, visual arts, industrial inspection, security etc.

Visual information is produced by the interaction between light and the objects and attained by the ability of sight. Therefore, anything that appeals to the sight can be related to the visual information. Among different aspects underlying visual information, shape of an object plays a crucial role. Shape represents the extends of object and can be conveyed by color or texture information. Since many properties of objects in our world are strongly determined by geometric properties, shape is one of the most important visual attributes of an object. Shape features provide a powerful clue to object identity and functionality, opening an important door for object recognition. Humans can recognize objects solely from their shapes, because; shape often carries semantic information [91], [138]. This distinguishes shape from other low level visual features such as color and texture, which in most cases is not

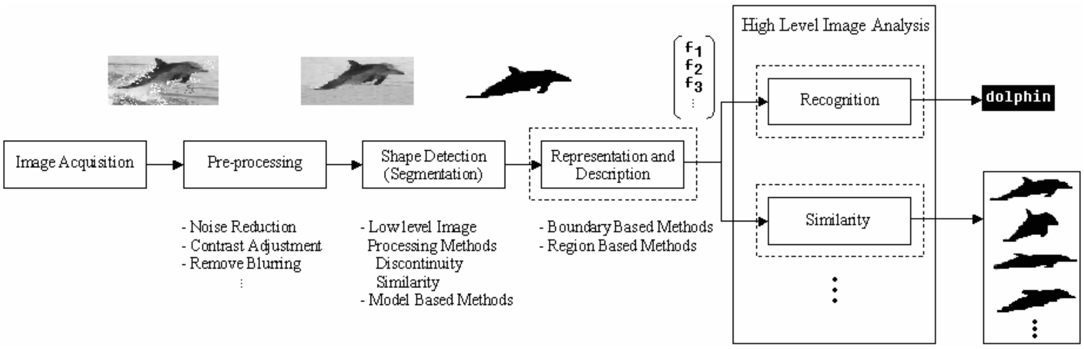


Figure 1.1: Basic stages in shape analysis.

sufficient to reveal object identity.

In a typical application, image processing starts with the image acquisition, yielding a digital image that is to be pre-processed to accentuate some visual information related to the problem domain. The objects of interest are then detected and represented for high level image analysis. The basic stages in shape analysis application is demonstrated in figure 1.1.

After the image acquisition stage, various methods of pre-processing can be applied to the image to produce an output which in some way represents an "improvement" to the original image. This stage may remove noise, increase the contrast of the image, remove blurring caused by movement of the camera during image acquisition, etc.

Usually the next stage is shape detection, which involves locating the object of interest in the image. Shape detection is also considered as an outproduct of the segmentation process, which is one of the most difficult tasks in computer vision. Segmentation is an important stage, because; the extent one can reach in separation of objects affects the result of shape analysis. Segmentation using traditional low-level image processing techniques is generally based on one of the two main properties of image: discontinuity in between-class-pixels and similarity among within-class-pixels. The techniques using discontinuity, partition an image, based on the abrupt changes in the pixel values. In this category of techniques, edges provide strong visual clues that can help the segmentation. Therefore, the segmentation is generally performed by first detecting the edges and, then, extracting the boundary through linking the edges. Another category of techniques group pixels together into regions of similarity.

The principal approaches in this category are thresholding, region growing, region splitting and merging.

However, the result of segmentation techniques using low level image processing, does not always give satisfactory results for various reasons, including the existence of noise, shape complexity, variations within and across objects in the image, etc. In order to improve the quality of segmentation, one should constrain the problem by exploiting the prior knowledge, which partly eliminates the ambiguities in detecting shapes. This task is generally performed by model based approaches, which use deformable shape templates. These techniques combine the segmentation and recognition processes in order to use the information about the objects that are to be located in the image.

Once the shape of interest has been acquired in the image, a set of techniques can be applied in order to extract information from the shape, so that it can be analyzed further. This process is called shape description and generally results in a feature vector (descriptor vector). Feature extraction is performed in two steps: In the first step, shape is represented in such a way that the "important" properties of the shape are preserved for a specific application. The shape representation approaches are classified in two groups:

1. Boundary based techniques,

2. Region based techniques.

Boundary based techniques represent the shape by its outline (external characteristics), while region based techniques treat the shape in terms of its 2-D region (internal characteristics). In the second step, shape description methods are employed to retain maximum amount of information in a more compact form. The goal of description is to uniquely characterize the shape using its shape descriptor.

High level image analysis algorithms are very much dependent on the application domain. For example; in Content Based Image Retrieval (CBIR) systems, given a query shape, the similarity measurement is performed in order to retrieve the similar shapes in the database. In object recognition applications, such as character recognition, target identification and medical diagnosis, the shape classes are predefined and the objective is to assign a given shape to one of these classes.

Shape recognition and retrieval applications generally use different techniques and give different emphasis on subjects of image analysis. In a typical recognition problem, we expect a comparison to be successful for shapes very close to the model and unsuccessful for shapes different from the query. The degree of similarity of shapes different from the model is of no interest as long as it remains below a suitable acceptance threshold. On the other hand, shape retrieval applications require a similarity distance that accurately measures perceptual similarity for all shapes in the database, ”reasonably similar” to the query.

In this study, we focus on three aspects of shape analysis problem (dotted boxes in figure 1.1): representation (and also description), similarity measurement and recognition. First, we introduce a boundary based shape representation method, which transforms 2-D shape information into a set of 1-D functions. Then, various feature extraction methods are employed for compact description of shapes. The resulting shape descriptors are used for two different purposes: First, our aim is to develope a shape similarity scheme for CBIR applications. For this purpose, some improvements are proposed on the available similarity measurement techniques in order to calculate the distances between shapes. Second, we use the proposed description method in object recognition problem. As a recognition scheme, we propose a new topology of Hidden Markov Models, which is designed for boundary based object recognition.

## 1.1 Motivation

Shape representation and analysis are two of the oldest and most common areas in computer vision literature, which have been studied extensively over the past fifty years. However, despite all the intensive research, the available techniques are very poor compared to human visual abilities. The techniques developed for specific applications have met limited success on the restricted problem domains.

One of the major difficulties in shape analysis is the representation of 3-D objects in 2-D images. Since the 2-D object in an image is a projection of 3-D object, in real world, the shape is disturbed due to change of view point with respect to the camera. Non-rigid object motion is another source of problem. The noise added in the digitization and segmentation processes further complicates the representation and analysis. In order to suppress the undesired changes in object shapes, current representation techniques simplify the shapes by a smoothing process, which results in decreased resolution, depending on the predefined threshold values. Therefore, these methods require procedures, which determine the necessary level of details or use heuristics to select the parameters for sufficient resolution of shape. However, the determination of the resolution is highly dependent on the context of the image. The level of detail, selected for removing distortions in a shape may not be suitable for the others in a different context and may cause to the loss of important characteristics of shapes. There is a trade off between the resolution and the ability to capture different aspects of shape.

The major motivation of this study, is to find a representation of the 2-D shape information without a predefined resolution or context-dependent scale. The representation should, also, be invariant to affine transformations and robust to small non-rigid deformations. For this purpose, a mapping, which transforms the 2-D shape boundary information into a set of 1-D functions called Beam Angle Statistics (BAS) is developed. The transformation utilizes the moment theorem for preserving the information at all scales. The use of statistical information makes the resolution of representation independent from a predefined threshold. It, also, provides information preservation and compaction. The BAS functions are consistent with the human visual system. In other words, the functions represent all the convexities and concavities of the shapes.

Choosing a representation is only a part of the solution for transforming data into a form, suitable for shape analysis. A method must, also, be specified for describing the shape data so that features of interest are as compact as possible. This process is required for efficient time and space complexity for the implementation of algorithms. In CBIR systems, the increase in vector size results in the huge amount of data to be stored, yielding inefficiency of image retrieval, in large database. A variety of descriptors have been proposed in the literature, including Fourier and Wavelet based methods, corner detection and curve evolution methods. In this study, we propose the utilize Shannon's Sampling theorem to find the minimum size of the proposed shape descriptor. We also detect the "dominant" points in BAS functions to reduce the size. These approaches describe the proposed 1-D functions compactly, preserving the characteristics of shape. Shortly, this study transforms 2-D shape information into a unique vector whose size is as small as possible.

Although a sound mathematical basis exists, similarity measurement between the shape features is a serious problem in the shape retrieval and indexing applications. Most of the mathematical similarity metrics are not convenient for shape similarity, even if they minimize the distance between feature vectors by allowing deformations. Another problem in distance measurement is the complexity of the algorithms. Similarity distance is calculated between the query and each database shapes, thus the time complexity of the process is crucial for large databases. Current studies on shape similarity use exhaustive search in matching the starting point of the shape descriptors so that the minimum distance can be found. The motivation in similarity measurement is to develope an efficient distance calculation method, which is consistent with the visual system in perceiving similarity and possesses a remarkable robustness to distortions. In this study, we propose two different similarity measurement algorithms which are adopted for BAS features. Since the starting point of the representations may differ in shapes, the proposed algorithms find the best correspondence of items by shifting one of the feature vectors. We, also, propose an efficient method which utilizes the cyclic nature of the shape boundary and eliminates the shifting operation.

Shape recognition is one of the open research area in shape analysis. Hidden Markov Models are widely used mathematical tools in shape recognition problems. All the approaches in the literature presume a model size and topology according to the application domain. Unfortunately, there is no simple, theoretically correct way of making the choice of model size and topology, These choices must be made based on trial and error depending on the process being modeled. The vast amount of HMM topologies, used in various applications, can be divided in two categories: ergodic and temporal topologies. Ergodic topologies enable the revisits of each state and do not impose a sequential or temporal order. The temporal topologies, on the other hand, constrain the state transitions and do not allow the revisits to the previous states. Our motivation is to develope an HMM topology which is computationally efficient and suitable for boundary based features. We propose a new HMM topology, called circular HMM which is both temporal and ergodic. Therefore, the states can be revisited in finite time intervals, while keeping the sequential information in the string which represents the shape.

## 1.2 Contribution

The contributions of this thesis are grouped in three categories. Accordingly, we focus on three major tasks of shape analysis problem and propose new methods for each them:

- **Shape Representation and Description based on Beam Angle Statistics (BAS) :** We develope a new shape description scheme using the boundary information. The proposed descriptor is based on the beams, which are the lines connecting the reference point with the rest of the points on the boundary. The characteristics of each boundary point is extracted by using the beam angles in a set of neighborhood systems. The angle between each pair of beams is taken as the random variable at each point on the boundary. Then, the moment theorem provides all the statistical information. In the first moment, each valley and hill corresponds to a concave and convex visual part of the object shape.

  The main contribution of this representation is to eliminate the use of any heuristic rule or empirical threshold value of shape boundaries in a predefined scale. BAS, also, gives globally discriminative features to each boundary point by using all other boundary points. Another advantage of BAS representation is its simplicity, yet consistency with human perception through preserving visual parts of the shapes. The representation is scale, rotation and translation invariant. It is also insensitive to noise, occlusion.

  In order to describe the 1-D BAS functions in a compact way, we adopt various feature extraction methods. These methods compress the data by highlighting important characteristics of the representation. The methods are developed both in space and transform domains. In space domain, piecewise constant approximation and polyline approximation are employed. In transform domain, Fourier and Wavelet Transformations are utilized.

- **Shape Similarity Measurements :** Two different sequence comparison methods are revised and adopted for the proposed feature vectors, extracted from the BAS functions:

  - *Optimal Correspondence of String Subsequences (OCS) Algorithm:* developed for the infinite alphabet string matching,

  - *Dynamic Warping (DW) :* used for the sequences obtained by time sampling of continuous functions.

The above methods are widely used in many areas including the shape similarity problems. However, none of them are consistent with the Human Visual System. Since the first method satisfies the metric property, it is implemented for comparison purposes to the non-metric DW algorithm, which results in better performance. In order to improve the performance of DW as a similarity matching tool in shape analysis, some simple modifications are proposed:

Firstly, it is noted that the DW algorithm stretches and squishes the shapes arbitrarily to minimize the distances between two objects. This may result in meaningless similarities between distinct objects. In this study, we propose to revise the cost function of DW with a penalty to control the deformations on the shape boundary.

Secondly, it is observed that the nature of the closed shape boundary represents a cyclic pattern. To ensure a consistent description of shapes, a unique starting point must, therefore, be defined for each shape. Since this is impractical to achieve, the alignment computation must determine the amount of cyclic shift that has taken place in order to find the optimal solution. In the above mentioned methods, this is performed by shifting any sequence, one item at a time and recomputing the sequence comparison. However, this computation increases the complexity of similarity measurement and may be cumbersome especially for large databases. For this reason, we propose an efficient cyclic sequence comparison algorithm which approximates the optimal solution in minimization of the total distance between the patterns.

- **Shape Recognition by Circular Hidden Markov Model :** A new topology, called circular HMM, is proposed for shape recognition. This topology is a modification of left-to-right HMM model, where the initial and terminal states are connected through the state transition probabilities. This simple connection eliminates the need to define a starting point of a closed boundary, in the description problem. Since the circular HMM is both ergodic and temporal, its structure enable one to decide on the optimal state order by simple experiments on the training data and requires no size normalization.

The circular HMM has many superiorities compared to the classical topologies in the literature. First of all, the circular HMM does not require to increase the number of states as the size of the boundary increases. Therefore, it is size invariant. Secondly, circular HMM does not require as many non-zero state transition probabilities as the classical topologies. Therefore, the computational complexity of the circular HMM is relatively less then the available topologies for the same recognition rates.

## 1.3 Organization of the Thesis

The thesis is organized as follows:

- *Chapter 2* provides an overview of shape description, similarity measures and recognition methods. The available techniques with their superiorities and weaknesses are reviewed.

- *Chapter 3* presents the proposed shape representation based on Beam Angle Statistics (BAS). Mathematical foundations together with the properties of BAS are given.

- *Chapter 4* explains two classes of feature extraction methods for BAS representation, namely, space domain and transform domain methods.

- *Chapter 5* describes the similarity measurement algorithms for BAS descriptors.

- *Chapter 6* introduces the Circular Hidden Markov Model together with parameter estimation and recognition algorithms.

- *Chapter 7* evaluates the performance of the proposed BAS descriptor, the revised similarity measures and Circular HMM. Experiments and test data are described.

- *Chapter 8* concludes the thesis and gives the future research direction.

# CHAPTER 2

# OVERVIEW OF SHAPE DESCRIPTION, SIMILARITY AND RECOGNITION

Shape analysis involves several important tasks, starting from image acquisition, reaching to shape classification. In this chapter, we give an overview of three major tasks of shape analysis problem:

- *Shape Description*, which characterizes the shape and generates a shape descriptor vector (also called feature vector) from a given shape.
- *Shape Similarity*, which establishes criteria to allow objective measures of how much two shapes are similar to each other.
- *Shape Recognition*, which labels the class of an input shape.

After giving an overview of shape description techniques in section 2.1, shape similarity measurement is discussed in section 2.2. Finally, the available shape recognition algorithms are presented in section 2.3.

## 2.1 Shape Description

The problem of shape analysis has been pursued by many authors, thus, resulting in a great amount of research. Recent review papers [85], [99] as well as books [37], [41] provide a good resource of references.

In most of the studies, the terms shape representation and description are used interchangeably. Since some of the representation methods are inherently used as shape descriptors, there is no well-defined separation between the shape representation and description. However, shape representation and description methods are defined in [85] as follows. Shape representation result in a non-numeric values of the original shape. Shape description refers to the methods that result in a numeric values and is a step subsequent to shape representation. For the sake of simplicity, we consider the representation and description together throughout the section and refer them as shape description methods.

Shape description methods can be classified according to the use of shape boundary points or the interior of the shape. The two major classes of methods are known as boundary based methods and region based methods. Since, the proposed BAS description is based on shape boundary, we rather concentrate on the boundary based methods. In addition, our focus is on the current studies, which are evaluated in MPEG-7 experiments and shown to outperform the other studies in the literature.

### 2.1.1 Region Based Methods

Region based shape descriptors express pixel distribution within a 2-D object region. It describes a complex object consisting of multiple disconnected regions as well as a simple object with or without holes. Since it is based on the regional property of an object, the descriptor is insensitive to noise that may be introduced inevitably in the process of segmentation.

#### 2.1.1.1 Moments

Two dimensional Cartesian moment $m_{pq}$ of order $p+q$ of the function $f(x, y)$ is defined as:

$$m_{pq} \triangleq \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} x^p y^q f(x, y). \tag{2.1}$$

The use of moments for shape description was initiated in [59], where it is proved that moment-based shape description is information preserving. The zeroth order moment $m_{00}$ is equal to the shape area assuming that $f(x, y)$ is a silhouette function with value one within the shape and zero outside the shape. First order moments can be used to compute the coordinates of the center of mass as $x_c = m_{10}/m_{00}$ and $y_c = m_{01}/m_{00}$. Second order moments are called moments of inertia and can be used to determine the principal axes of the shape.

Based on the moments formulated in equation 2.1, a number of functions can be defined that are invariant under certain transformations such as translation, scaling and rotation. The moment invariants are then put into a feature vector. Global object features such as area, circularity, eccentricity, compactness, major axis orientation, Euler number, concativity tree and algebraic moments can all be used for shape description [105].

A generalization of moment transform to other transform kernels is also possible by replacing a conventional transform kernel $x^p y^q$ by a form $P_p(x)P_q(y)$ such as orthogonal polynomials. In this case, the moments produce minimal information redundancy, which is important for optimal utilization of the information available in a given number of moments [85]. Some of the orthogonal polynomial systems include Legendra and Zernika polynomials [125]. The advantage of moment methods is that they are mathematically concise. The disadvantage is that it is difficult to correlate high order moments with shape features.

#### 2.1.1.2 Angular Radial Transformation

Although this approach belongs to the moment based methods, we prefer to explain it under a different section; because it is an MPEG-7 region based shape descriptor [26]. Angular Radial Transformation (ART) is a 2-D complex transform defined on a unit disk in a polar coordinates [72]. The method extracts a set of ART coefficients, $F_{nm}$ of order $n$ and $m$, from each shape using the following formula:

$$\begin{aligned} F_{nm} &= \langle V_{nm}(p,\theta), f(p,\theta) \rangle \\ &= \sum_{0}^{2\pi} \sum_{0}^{1} V^{*}_{nm}(p,\theta), f(p,\theta)p, \end{aligned} \tag{2.2}$$

where $f(p,\theta)$ is an image intensity function in polar coordinates and $V_{nm}(p,\theta)$ is the ART basis function that are separable along the angular and radial directions, i.e.,

$$\begin{aligned} V_{nm}(p,\theta) &= \frac{1}{2\pi} exp(jm\theta) R_n(p), \\ R_n(p) &= \begin{cases} 1 & n = 0 \\ 2cos(\pi np), & n \neq 0. \end{cases} \end{aligned} \tag{2.3}$$

One of the advantages of ART method is the compactness and efficiency in describing the properties of multiple disjoint regions, simultaneously. It describes the objects that may be split into disconnected sub-regions in the segmentation process.

#### 2.1.1.3 Shape Decomposition

This approach assumes that shape description is a hierarchical process and decomposes an object shape into a union of primitives being the simplest elements, which form the region. Given the decomposition, a graph object representation is constructed. Syntactic pattern recognition techniques are, then, applied.

Shape decomposition methods generally use mathematical morphology. A morphological shape decomposition technique is proposed in [102] where the binary objects are decomposed into a union of simple objects. A structuring element (disk) is defined as the simplest object component and the image is analyzed as a union of the disks. The representation is shown to be unique and invariant under rotation, translation and scaling. Another morphological decomposition method, which is called Morphological Signature Transform (MST) is developed in [86]. The MST method utilizes multi-resolution morphological image processing by multiple structuring elements. The idea of this approach is to process decomposed, multiple shapes instead of the original ones. The decomposed shapes, called signature shapes, contain substantial information about the whole shape. The method calculates the area of the shape signatures obtained from multiple structuring elements and multiple object scales to generate shape descriptor. Multiple structuring elements are obtained by rotating single or multiple structuring elements.

One of the problems in shape decomposition is the definition of the components. The result of the decomposition does not always correspond to the human intuitive shape representation. Furthermore, the results are not always unique. The decomposition of similar shapes may result in different elements. In many cases, the derivation of the decomposition is rather tedious and the computational complexity is relatively high compared to the other region based shape description methods such as ART and moments.

#### 2.1.1.4 Shape Matrices and Vectors

Shape matrix and vector approaches locate shape in a grid and extract global shape information to create a numerical description of shape. In [50] a matrix is used to represent the pixel values corresponding to a polar raster of coordinates centered in the shape center of mass. A polar raster of concentric circles and radial lines is positioned with respect to the center of mass. 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. This method is invariant to translation, rotation and scaling.

In [124], shapes are described by matrices and the comparison of matrices is performed to classify unknown shapes into one of the known classes. For this purpose, a scheme for weighting matrix entries is developed. The reason for weighting is based on the fact that sampling density is not constant with the polar sampling raster. Since the sampling density is much higher in the center of the shape, the importance of innermost and peripheral shape samples are equalized by weighting process. Another method proposed in [98] uses shape vectors. This method describes the shape, based on the relative areas of the shape contained in concentric rings located in the shape center of mass. The area of each shape part contained in a ring, relative to the area of the ring itself, is used as an entry of the shape vector.

#### 2.1.1.5 Medial Axis Transform

The idea of Medial Axis Transform (MAT) is to represent the shape by a graph in which the important shape features are preserved [24]. The shape graph is based on the region skeleton and the first step is the skeleton construction. The skeleton is the set of all region points, which have the same minimum distance from the region boundary for at least two separate boundary points.

One of the weaknesses in medial axis transform is its sensitivity to noise in the shape boundary. In order to overcome this problem, a generalized medial axis transform is proposed in [25]. This approach defines the touching circle, which is a circle, tangent to the shape boundary without intersecting it. The *r-symmetric* axis of a shape is, then, defined as the union of all points that have a touching circle of radius greater than $r$ and at least two points that touch the boundary.

The approach starting in late 80's is the introduction of regularization to the symmetry extraction process itself [27]. Alternative regularizations are introduces in 90's [103], [73], [122]. In [103], hierarchy by scale is used to construct a series of skeletons. Hierarchical representation using axial shape description is proposed in [112]. First, the original shape is broken into parts at negative curvature minima of shape boundary. Then, the parts are represented by using smooth local symmetry ribbons, which are objects from their symmetry axis.

Another problem of medial axis transform is the difficulty in the realization of discrete space. In [80], a dynamic multi-scale skeleton representation is developed in order to resolve the discretization problem. The method combines the use of snakes for active contour representation, high curvature points on the boundary and symmetric axis transform.

There are also alternative symmetry representations in the literature. In these representations, symmetry is, generally, defined in terms of a condition that has to be satisfied for two points to form a symmetry. The line connecting two points is called the line of symmetry. The axis of symmetry is formed by the union of middle points in the line of symmetry. In *skew symmetry*, the lines connecting the boundary points are perpendicular to the axis of symmetry which is a mirror-type symmetry [67]. Two curves are said to be *parallel symmetric*, if there exists a connected, point wise correspondence function such that corresponding points have parallel tangent vectors [131]. The axis of parallel symmetry is the locus of mid points of segments connecting parallel symmetric points (lines of parallel symmetry). Another type of symmetry is *smooth local symmetry* [27], where two points form local symmetry if the angles between the curve normals at two given points and the line connecting the points are the same. In [122], the axes of local symmetries are determined by using the Ambrosio Tortorelli (AT) edge strength function [8], where its level curves correspond to the smoothed propagating glassware front. In [121], it has been shown that for the AT edge strength function, the gradient vector coincides with the main principal axis, providing a numerically robust curvature detection.

#### 2.1.1.6 Bounding Regions

Another class of region based shape description methods use a region that encloses the shape of interest [37]. The Feret Box is defined as the smallest rectangle that encloses the shape [80]. The rectangle is usually oriented with respect to the coordinate axis. The Minimum Enclosing Rectangle, on the other hand, is the smallest rectangle oriented along any direction [29].

The convex hull are used to describe the region shape properties to build a tree structure of region concavity. A region is called convex if and only if for any two points in that region, the whole line segment defined by these two points, lies inside the region. The Convex Hull is the smallest convex set that contains the shape [48].

#### 2.1.1.7 Scalar Shape Descriptors

A large number of shape description techniques is presented by heuristic approaches, which yield acceptable results for description of simple shapes in some restricted applications. Some examples are given below, for the sake of completeness:

- *Area :* The number of pixels of which the region consists.
- *Rectangularity :* The maximum of ratio between the region area and the bounding rectangle area in any direction.
- *Elongatedness :* The ratio between the length and with of the region bounding rectangle.
- *Eccentricity :* The ratio of major and minor axes of an object.
- *Euler's Number :* The difference between the number of contiguous parts and the number of holes in the object.
- *Projections :* The projection of number of pixels in vertical or horizontal direction.
- *Compactness :* The ratio between the square of region border length and the area.

Note that, the above descriptors can not be used for region reconstruction and are far from being sufficient to represent complex shapes.

### 2.1.2 Boundary Based Methods

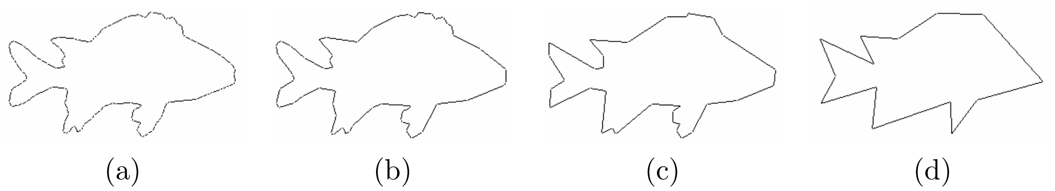
Boundary based shape description methods exploit only shape boundary information. The shape properties of object boundary are crucial to human perception in judging shape similarity and recognition. Many authors, who study on the human visual perception system, agree on the significance of high curvature points of the shape boundary in visual perception [91], [12]. In the psychological experiments, it is suggested that corners have high information content and, for the purpose of shape description, corners are used as points of high curvature. Therefore, the shape boundary contains more information than the shape interior, in terms of perception.

#### 2.1.2.1 Polygon Approximation

Polygon approximation is one of the most popular shape representation methods, where the continous shape is represented by a set of vertices of a polygon [85]. The main idea is to eliminate the redundant and insignificant shape features by reducing the number of boundary points. This is performed by searching the break points on the boundary, based on an approximation criteria, such as minimal error, minimal polygon perimeter and maximal internal polygon area.

In the approaches, which use split and merge algorithm [100], [22] while the curve is split into segments until some acceptable error, the split segments are merged if the resulting segment approximates the curve within some maximum error.

Currently, there are two widely used polygon approximation methods which use curve evolution [10], [77] and wavelet based corner detection methods [106], [79] respectively. Curve evolution reduces the influence of noise and simplifies the shapes by removing the irrelevant features, keeping relevant ones. In this method, a digital curve is regarded as a polygon, where each boundary point is assumed to be a vertex, at the beginning of evolution. In every evolution step, a pair of consecutive line segments is substituted with a line segment, joining the end points (see figure 2.1).



(a) (b) (c) (d)

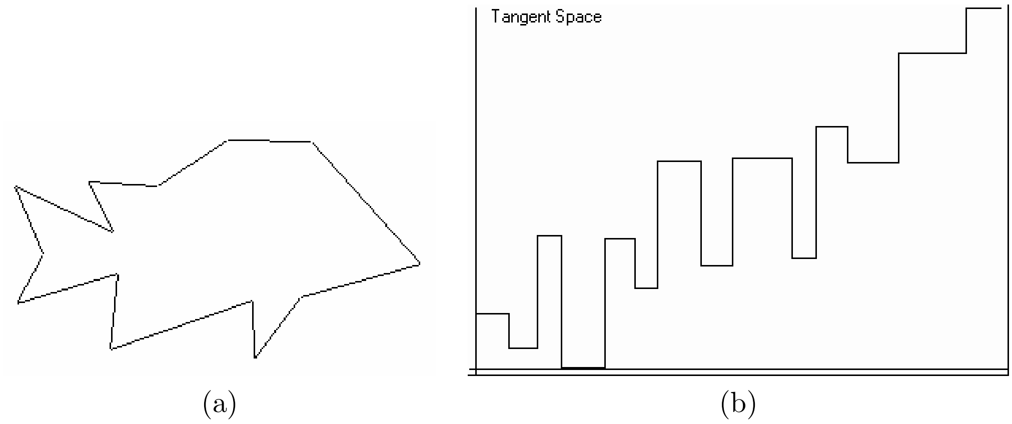
Figure 2.1: A few stages of discrete curve evolution.

The substitution is performed according to a relevance measure given by:

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)} \tag{2.4}$$

where $\beta(s_1, s_2)$ is the turn angle at the common vertex of segments $s_1$, $s_2$ and $l$ is the length function normalized with respect to the total length of the boundary. Since the relevance measure is in direct proportion to the turn angle and the length of the curve segment, the higher relevance measure means that the vertex has larger contribution to the shape of the curve. The evolution process halts when the difference between the evolved polygon and the original curve is higher than a given threshold. Curve evolution method assumes that the shape boundary consists of various sources of distortions, which should be removed by smoothing. The process of smoothing depends on the predefined threshold value, which halts the evolution.

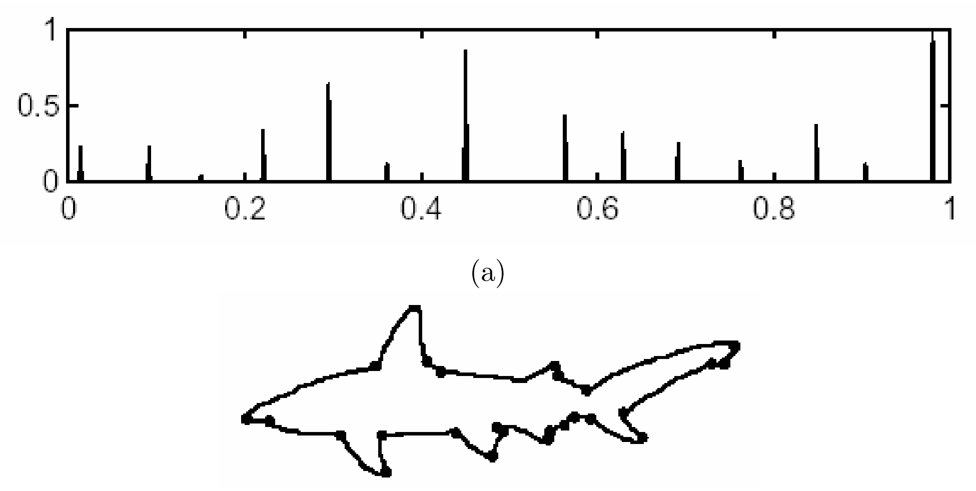
Finally, each polygon is represented by a tangent function (see figure 2.2). Tangent function gives the angle between the counterclockwise tangent and the x-axis as a function of the arc length. Tangent function is a piecewise constant function, increasing or decreasing at the vertices and constant between two consecutive vertices. Tangent space representation is invariant to translation and scale. Shifting vertically and horizontally provides rotation and starting point invariance.



(a) (b)

Figure 2.2: An example of tangent function (a) A polygonal curve, (b) its tangent function.

Corner detection is another approach for polygonal approximation. Popular techniques for corner detection use Wavelet Transform Modulus Maxima (WTMM) [107]. First, the shape boundary is represented by a 1-D signal such as contour's orientation profile. The singularities on that signal are, then, detected by determining the local maxima of the wavelet transform of the signal. These techniques extract high curvature points and can be broadly classified into two categories. In the first category, the corner detection is performed at one scale [107]. Those techniques might suffer from finding a lot of insignificant details, while at the same time missing large rounded corners. The second category of techniques make use of different scales [128], avoiding the problems in one scale techniques and provide additional information about the structural importance of the high curvature points. The corners are, then, described by using various features, such as the angle at the vertex and lengths of the consecutive line segments.



(a)

(b)

Figure 2.3: Corner detection by Wavelet Transform Modulus Maxima (WTMM), (a) WTMM at dyadic scale $s = 2^j, j = 2$,(b) High curvature points detected at level $j = 2$.

#### 2.1.2.2 Scale Space Filtering

Scale space representations are based on tracking the position of inflection points in the boundary, filtered by low-pass Gaussian function of variable widths [139]. The inflection points, remained in the representation are expected to be significant object characteristics [85].

Among many other studies such as [14],[11], Curvature Scale Space (CSS) approach proposed in [96] is the most popular one in this class of methods. It is, also, selected as MPEG-7 contour based shape descriptor.

Mathematically speaking, let $\Gamma$ be a closed planar curve and $u$ be the normalized arc length parameter on $\Gamma$:

$$\Gamma = \{(x(u), y(u))|u \in [0,1]\}. \tag{2.5}$$

In order to smooth the curve, functions $x(u)$ and $y(u)$ is convolved with a one-dimensional Gaussian kernel at different levels,

$$\begin{aligned} X(u,\sigma) &= x(u) * g(u,\sigma), \\ Y(u,\sigma) &= y(u) * g(u,\sigma), \end{aligned} \tag{2.6}$$

where $g(u,\sigma)$ is a Gaussian kernel of width $\sigma$. The curvature of the smooted curve is then computed as,

$$\kappa(u,\sigma) = \frac{\dot{X}(u,\sigma)\ddot{Y}(u,\sigma) - \ddot{X}(u,\sigma)\dot{Y}(u,\sigma)}{(\dot{X}^2(u,\sigma) + \dot{Y}^2(u,\sigma))^{\frac{3}{2}}}, \tag{2.7}$$

where $\dot{X}$, $\dot{Y}$ and $\ddot{X}$, $\ddot{Y}$ are the first and second derivatives of $x(u,\sigma)$ and $y(u,\sigma)$. As $\sigma$ increases, the shape of $\Gamma_\sigma$ changes. This process of generating ordered sequences of curves is referred to as the evolution of $\Gamma$. During evolution, the locations of curvature zero crossings of every $\Gamma_\sigma$ is determined. The points are, then, plotted in $(u,\sigma)$ plane. The result of this process is represented as a binary image, called CSS image of the curve (see figure 2.4).

Finally, shapes are described by the positions of their CSS contour maxima. These positions projected on the simplified object contours give the positions of the mid points of the maximal convex/concave arcs, obtained during the curve evolution. Since the small contours on the CSS image represent some information about the existing noise on the actual object, those maxima which are lower than a threshold are discarded. The shape similarity measure between two shapes is computed by relating the positions of the maxima of the corresponding CSSs.

CSS image representation is invariant to scale, rotation and translation. It is also robust to significant non-rigid deformations and perspective deformations. The basic drawback of this representation is the difficulty in determination of a threshold value in order to remove small contours in CSS image. This process results in decreased resolution, depending on the predefined threshold value. Therefore, it requires procedures, which find the necessary level of details or use empirical parameters for sufficient resolution of the shape contour.

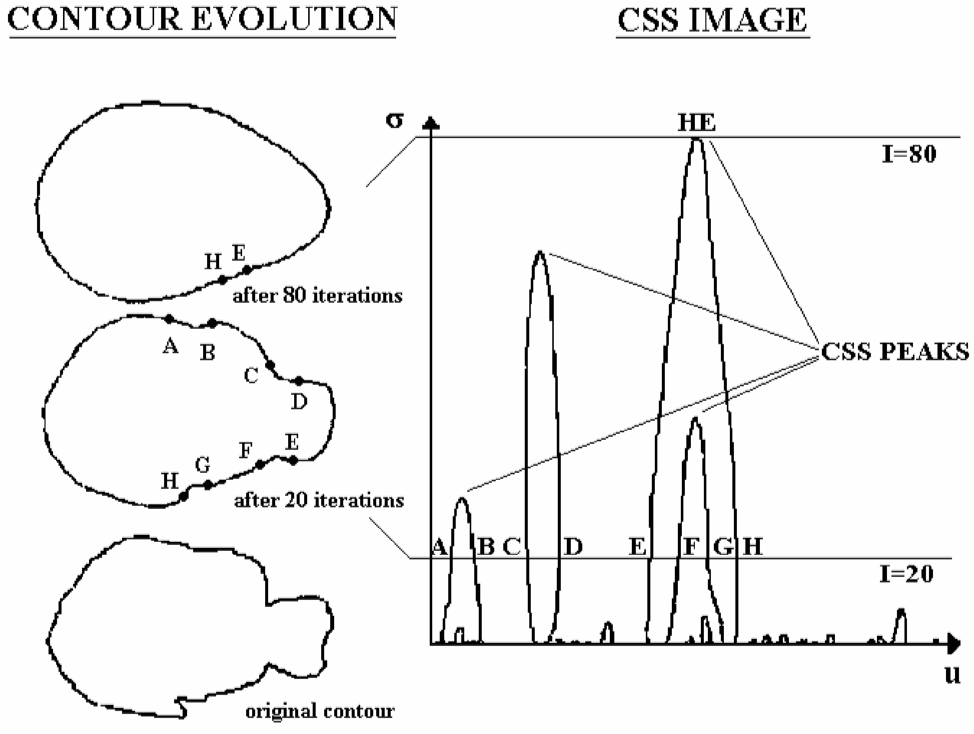


Figure 2.4: An example of Curvature Scale Space image, a sample shape and its CSS image (reproduced from [26]).

#### 2.1.2.3 Stochastic Representation

These methods are based on the stochastic modeling of one dimensional function obtained from the shape boundary [85]. The idea is to interpret the one dimensional function as a a stochastic process realization and use the model parameters obtained by estimation, as shape descriptors.

Autoregressive (AR) model is proposed in [68] to represent the closed shape boundary. One dimensional representation is obtained using centroid distance function. The AR model is, then, characterized by a set of unknown parameters and an independent noise sequence. Given the function $u(s)$, the stochactic process is defined by,

$$u(s) = \alpha + \sum_{j=1}^{m} \theta_j u(s-j) + \sqrt{\beta}\omega_s \qquad , \tag{2.8}$$

where $\theta_j$ is the AR coefficients to be estimated, $\alpha$ is a constant to be estimated, $\sqrt{\beta}$ is the variance of prediction noise and $\omega_s$ is the independent random noise sources. $\theta_1, .., \theta_m, \alpha, \beta$ are called the model parameters. The shape descriptor vector is constructed using these parameters estimated by Maximum Likelihood Method.

The same AR model is used in [40] with additional methods for improving classification performance. In [38], the AR model is extended to the bivariate case and the classification results are further improved.

The disadvantage of AR model is the sensitivity to shape occlusion [55]. The reason for this drawback is that it models the whole shape with only one set of predictive parameters. If the shape contains a large number of sample points and the contour varies radically, the shape may seem unpredictable. Therefore, an AR model with finite number of parameters is not adequate for the whole shape. In order to overcome this drawback, the AR model is combined with the Hidden Markov Model in [55]. This approach segments closed shape into segments, describes each segment by AR modeling and finally analyzes the resulting vectors by Hidden Markov Model.

#### 2.1.2.4 Boundary Approximation

These methods represent the curves by piecewise polynomial interpolation, which results in a set of smooth curves inflected at control points. Splines are used for the interpolation of functions to approximate shape segments. The power of splines comes from the approximation of a given function with a curve having the minimum average curvature. An overview of spline applications for shape representation can be found in [61].

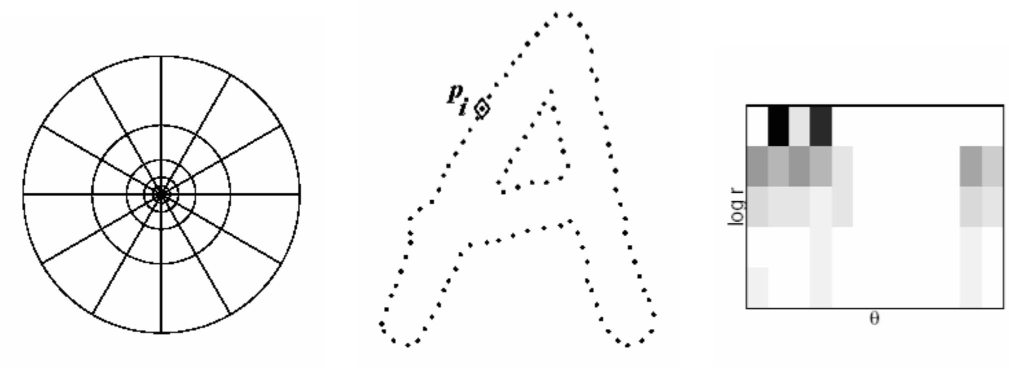
Main disadvantage of splines is that the local function value modification changes the complete spline representation [85]. For this reason, B-splines are constructed so that the local function value change does not spread to the rest of the intervals. B-splines are piecewise polynomial curves, whose shape is closely related to their control polygon which is a chain of vertices giving a polynomial representation of a curve. If a control polygon vertex changes its position, a resulting change of the spline curve occur only in a small neighborhood of that vertex.

In [33], curve representation and matching is performed by B-splines. Another method [35] use splines for object modeling and shape estimation. Shape preserving approximation based on splines is proposed in [57].

#### 2.1.2.5 Set of Boundary Points

In this category of methods, the shape boundary is represented as a set of points without any special order among them. These methods assign a descriptor to each boundary point, which discriminates it from the other points. The correspondence problem between two shapes is, then, solved based on the descriptors of points. Since there is no order among the points, the description scheme should incorporate the whole shape information into local descriptor. The methods in this group can also be used for object localization by determining correspondence between the points in shape model and image.

In [60], the relative positions of points in a shape model and an image are used for calculating the Hausdorff distance between them. The work on range images [66] introduces a representation for matching dense clouds of oriented 3-D points, called the spin image. In order to obtain the spin image, a plane around a normal vector on the object surface is spinned and the points that fall inside bins in the plane are counted. The resulting 2-D histogram is then used as spin image.



(a) (b) (c)

Figure 2.5: Shape Context computation, (a) Diagram of log-polar histogram bins, (b) Sampled edge points of a shape, (c) Shape Context for the reference boundary point (reproduced from [21]).

A recent study [21] proposes shape contexts descriptor for measuring similarity between shapes and exploits it for object recognition. The shape context at a reference point captures the distribution of the remaining points relative to it. Given a shape with $N$ points, for a point $p_i$, a coarse histogram $h_i$ of the relative coordinates of the remaining $N-1$ points is calculated as follows,

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\} \qquad , \tag{2.9}$$

where $bin(k)$s are the bins that uniform in log-polar space (see figure 2.5). This histogram is defined to be the shape context of $p_i$. The correspondence problem between the points of two shapes is, then, solved using Bipartite graph matching algorithm.

#### 2.1.2.6 Fourier Descriptors

The Fourier representation decomposes a shape contour into its frequency components (Fourier descriptors) obtained via its Fourier transform. Fourier transform is applied to the boundary function and the resulting coefficients are used for shape description.

For a given closed curve, which in turn is represented by a one dimensional function $u(s)$, the discrete Fourier transform is defined by;

$$a_n = \frac{1}{N} \sum_{s=0}^{N-1} u(s) \quad exp(-j2\pi is/N). \tag{2.10}$$

The coefficients $a_n$, $n = 0, 1, ..., N-1$, are used to derive Fourier descriptors. In order to achieve, translation and rotation invariance, phase information of $a_n$ is ignored and only the magnitude $\mid a_n \mid$ is used. The magnitudes are divided by the DC component $\mid a_0 \mid$, to achieve the scale invariance.

In order to obtain one dimensional function from the shape boundary, various methods have been suggested such as tangent angle [143] and centroid distance [145]. Classical method is to express the shape as a sequence of coordinates in the complex plane, namely $u(s) = x(s) + jy(s)$. Affine invariant Fourier descriptors are proposed in [9]. A modified Fourier descriptors are used in [113] for efficiently compute the feature matching.

The Fourier descriptors represent a global information about the boundary. Therefore, local spatial information about the shape is not readily available and the level of shape detail can only be controlled on a global basis.

#### 2.1.2.7 Coding

Coding methods describe an object by a sequence of unit size line segments with a given orientation. One of the most popular coding schema is Freeman's chain code

[44], which is essentially obtained by mapping the shape boundary into a 2-dimensional parameter space, which is made up of codes. There are many variations of Freeman chain codes.

A generalized chain code is proposed in [45], where the nodes surrounding a center node are enumerated counterclockwise in ascending order from inside out. Another study in [98] modifies the chain code in order to search for the symmetric axis position. The shape boundary is represented in a hierarchical way so that at the highest level, a lower number of polygon vertices is used, while at the lowest level the finest polygonal approximation represents the shape.

#### 2.1.2.8 Simple Boundary Functions

The following descriptors are mostly based on geometric properties of the boundary. Because of the discrete character of digital images, all of them are sensitive to image resolution. In the following, popular geometric descriptors are provided for the sake of completeness:

- *Bending Energy :* Given a curvature function $C(s)$ for a shape boundary, the bending energy is defined as,

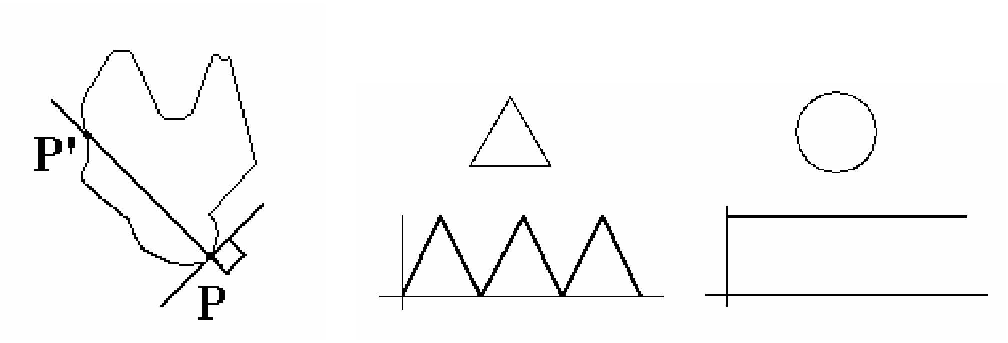
$$E = \frac{1}{N}\sum_{s=1}^{N} C(s)^2. \tag{2.11}$$

- *Signature :* Signature of a shape boundary is computed for each boundary point as a function of the boundary path length. For each boundary point $p$, the signature is the distance between $p$ and another boundary point $p'$, such that $pp'$ is perpendicular to the tangent vector at $p$ (see figure 2.6).

- *Chords :* Chord is a line joining any two points of the shape boundary. The distribution of lengths and angles of all chords on a shape boundary is used for shape representation. The chord distribution $D_{chord}$ is computed as:

$$D_{chord}(\triangle x, \triangle y) = \sum_x \sum_y B(x,y)B(x+\triangle x, y+\triangle y), \tag{2.12}$$

  where $B(x,y) = 1$, if $(x,y)$ is a boundary point and $B(x,y) = 0$ for all other points.



(a) (b)

Figure 2.6: Calculation of signature function, (a) Signature at point $P$, (b) Sample shapes and their signature functions.

- *Rotation-independent Radial Distribution :* It is computed by taking the sum of chord distribution over all angles,

$$D_{radial}(r) = \sum_{\theta=-\pi/2}^{\pi/2} D_{chord}(\triangle x, \triangle y)r, \tag{2.13}$$

  where $r = \sqrt{\triangle x^2 + \triangle y^2}$ and $\theta = \tan^{-1}(\frac{\triangle y}{\triangle x})$.

- *Angular Distribution :* In order to make the radial distribution scale-independent, the angular distribution $D_{angular}$ is calculated as follows:

$$D_{angular}(\theta) = \sum_{r=0}^{\max(r)} D_{chord}(\triangle x, \triangle y). \tag{2.14}$$

- *Centroid Distance :* The centroid distance function is expressed by the distance of the boundary points from the centroid $(x_c, y_c)$ of the shape:

$$D_{centroid}(s) = \sqrt{(x(s) - x_c)^2 + (y(s) - y_c)^2}. \tag{2.15}$$

## 2.2 Shape Similarity Measurements

Many pattern matching and recognition techniques are based on a similarity measures between patterns. A similarity measure is a function defined on pairs of patterns indicating the degree of resemblance between the patterns [134]. In this section, first, some properties of shape similarity measures are discussed and, then, the basic similarity measures are described. Further information on this subject can be found in [133].

### 2.2.1 Desired Properties of Shape Similarity Measures

Unfortunately, it is not clear what properties perceptual similarity has [17]. It is possible that Human Visual System operates different similarity methods between classes than the ones within classes. It is also possible that our prior knowledge of objects plays a significant role in our similarity judgments, a role which may vary considerably depending on the shapes we view. Since perceptual similarity is not a well-known phenomenon, none of the available similarity measures are fully consistent with the Human Visual System.

In this section, we list some desirable properties of similarity measures. Depending on the application, a property, which is useful in some cases, may be undesirable in some other cases. Combinations of properties may be contradictory [133]. While some of the properties are satisfied by the distance function and the algorithm used in similarity calculation, the others are inherently satisfied by the shape representation.

Let us define the distance function $d$ on a feature space of shapes $S$ as $d : S \times S \rightarrow \Re$ and $A$, $B$, $C$ as the shape descriptors on $S$. In the following subsections, we describe the properties of shape similarity measures.

#### 2.2.1.1 Continuity Properties

It is debatable whether human comparison between shapes are smooth [17]. It is, also, not clear how one should formally define continuity to be relevant to human perceptions. However, the continuity of similarity measurement can be investigated in the property of robustness, a form of continuity. Such properties are useful to be robust against the effects of discretization [134]. Four types of robustness properties are defined in [53] as follows,

- *Noise Robustness :* For each point on the shape, applying changes in a "sufficiently small" open neighborhood of that point results in shape that has an "arbitrary small" distance to the original one.

- *Deformation Robustness :* If a shape is transformed by a deformation "sufficiently close" to the identity transformation, then the deformed shape have an "arbitrary small" distance to the original one.

- *Blur Robustness :* Adding new pieces of boundary of a shape "sufficiently close" to the original boundary results in a shape that has an "arbitrary small" distance to the original one.

- *Crack Robustness :* Changes in a shape that are "sufficiently small" to a crack result in a shape with an "arbitrary small" distance to the original one.

#### 2.2.1.2 Invariance

A distance function $d$ is invariant under a chosen group of transformations $G$ if for all $g \in G$, $d(g(A), g(B)) = d(A, B)$ [133]. Traditional approaches in shape similarity aim to build invariance under the following transforms:

- *Euclidean Transformation,* which preserves invariant size of the shape and change only shape position in the@article Ambrosio, author = "L. Ambrosio and V. M. Tortorelli", title = "On the Approximation of Free Discontiuity Problems", journal = "Bolletino Della Unione Matematica Italiana", volume = "6", pages = "105-123", year = "1990" space.

- *Affine Transformation,* which changes the positions of the shape points by applying a linear combination of translation, rotation, scaling and/or shearing. It preserves lines and parallelism (maps parallel lines to parallel lines).

- *Projective Transformation,* which maps shape lines to lines but do not necessarily preserve parallelism.

Invariance to Euclidean and affine transformation is desirable especially, for shape recognition. However, it depends on the application whether or not a large group of transformation is required. In CBIR systems, the scale, translation and rotation invariances are important properties. On the other hand, the invariance under shear transformation may not be desirable, because; the different objects can be transformed into each other by using shear transformation. Projective invariance, on the other hand, is the most difficult property, that can be satisfied by the features extracted from the 2-D shape information.

### 2.2.2 Metric Axioms

In mathematical literature, *distance* is ordinarily used to indicate a function $d$, which satisfies the metric axioms:

1. Nonnegativity, $d(A, B) \geq 0$,

2. Zero Property, $d(A, B) = 0$, if and only if $A = B$

3. Symmetry, $d(A, B) = d(B, A)$, for all $A$ and $B$,

4. Triangle Inequality, $d(A, B) + d(B, C) \geq d(A, C)$, for all $A, B$ and $C$.

Although it is generally desirable to use a function satisfying the above axioms, the metric distance function defined on feature space of shapes may not be suitable with Human Visual System. For instance, symmetry is not always a desired property. Human perception does not always find that shape $A$ is equally similar to $B$, as $B$ is to $A$. In particular, a variant $A$ of prototype $B$ is often find more similar to $B$ than vice versa [130]. For example, an ellipse may be percieved as more similar to a circle (it may be the image of a slanted circle) than a circle is to an ellipse (it is very uncommon for a circle to be the image of an ellipse) [17]. In addition, the triangle inequality may not hold for human comparison of shape. The similarity measure for partial matching, giving a small distance $d(A, B)$ if a part of $A$ matches a part of $B$, do not obey the triangle inequality. For example, people perceive a horse and a man to have very different appearance, while a centaur may be quite similar to both.

### 2.2.3 Similarity Measures

In this section, the distance measures defined on the point sets are described. In what follows, we denote $A$ and $B$ as two point sets in $\Re^k$ , $\mid A \mid= N$ and $\mid B \mid= M$ as the number of points in sets.

#### 2.2.3.1 Minkowsky Distance

Given two points $a$ and $b$ in $R^k$, the Minkowsky ($L_P$) distance is defined as;

$$d(a, b) = (\sum_{i=0}^{k} \mid a_i - b_i \mid^P)^{1/P}. \tag{2.16}$$

For $P = 1$, the distance is called the Manhattan or city block distance $L_1$. For $P = 2$, $L_2$ is called Euclidean distance.

If $N = M$ and the correspondence between the points in two sets is known, Minkowsky distance is calculated between the points and the total distance between two patterns $A$ and $B$ is simply found by summing all the distances. However, in many applications the number of points in the patterns are not the same and the correspondence is not known in advance. Thus, the similarity measure should first find the correspondence between the points of two patterns and then calculate the distance. The remaining part of this section gives commonly used distance measures in shape similarity. Further information about distance measures can be found in [134] and [133].

#### 2.2.3.2 Hausdorff Distance

Hausdorff distance is the maximum distance of a set to the nearest point in the other set. It is often used when the two point sets are of different size, so that no one-to-one correspondence exists between all points. The asymmetric Hausdorff distance from $A$ to $B$ is

$$\overrightarrow{d}_H(A,B) = \max_{a \in A} \min_{b \in B} d(a,b). \tag{2.17}$$

It should be noted that $\overrightarrow{d}_H(A,B)$ is not equal to $\overrightarrow{d}_H(B,A)$. A more general Hausdorff distance is defined as,

$$d_H(A,B) = \max(\overrightarrow{d}_H(A,B), \overrightarrow{d}_H(B,A)). \tag{2.18}$$

For finite point sets, it can be computed using Voronoi diagrams with a time complexity of $O((M+N)log(M+N))$ [4]. Because of high complexity, there is also some approximate solutions defined for shape matching under translation and rigid motion [3].

Since a single point can determine the distance value, the Hausdorff distance is very sensitive to noise. In order to reduce the noise sensitivity, the Partial Hausdorff distance and $P^{th}$ Order Hausdorff distance are proposed in the literature [133].

#### 2.2.3.3 Bottleneck Distance

The bottleneck distance is the minimum over all one-to-one correspondences between $A$ and $B$ of the maximum distance. Let, $F$ be the set of all one-to-one correspondences $f$ from $A$ to $B$. The asymmetric bottleneck distance from $A$ to $B$ is defined as [135],

$$\overrightarrow{d}_B(A,B) = \min_{f \in F} \max_{(a,b) \in F} d(a,b). \tag{2.19}$$

The symmetric bottleneck distance $d_B$ can, then, be defined by taking the maximum of $\overrightarrow{d}_B(A,B)$ and $\overrightarrow{d}_B(B,A)$. The bottleneck distance can be calculated by a time complexity of $O(N^{1.5}logN)$ using parametric search or k-nearest neighbors. Variations on the bottleneck distance are the minimum weight distance, the most uniform distance and the minimum deviation distance [133].

#### 2.2.3.4 Turning Function Distance

Turning function distance is defined for the dissimilarity of curves. As we mention in section 2.1.2, any polygonal curve can be represented by a turning function which is a piecewise constant function, increasing or decreasing at the vertices and constant between the consecutive vertices. The distance between two turning functions $T_A$ and $T_B$, is defined in [10] as,

$$d_T(A,B) = (\sum_s \mid T_A(s) - T_B(s) \mid^P)^{1/P}. \tag{2.20}$$

In the above definition the size of the polygons are scaled so that they have the equal length (see figure 2.7).

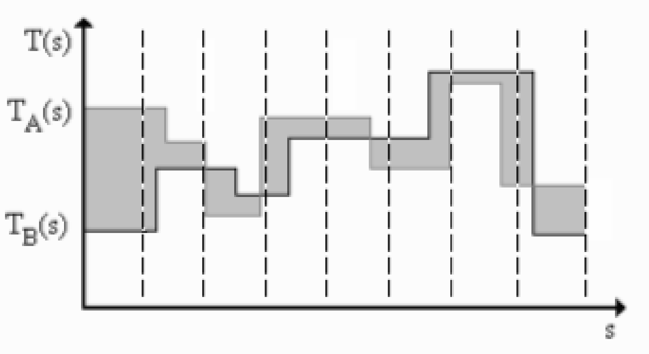


Figure 2.7: Tangent function distance.

In [78], in order to make the turning function distance more robust to local distortions, the matching parts in the tangent functions is performed by allowing non-uniform stretching of the parts. This result the best possible correspondence of visual parts in shape boundary.

Formally speaking, let $a$ and $b$ be simple polygonal arcs that are parts of curves $A$ and $B$. Let also $T_a$ and $T_b$ denote their tangents functions uniformly scaled so that their projections have length one. The arc similarity measure is given by

$$d_T(a,b) = (\sum_0^1 (T_a(s) - T_b(s) + \theta_0)^2) \times \max(l(a), l(b)) \times \max(\frac{l(a)}{l(b)}, \frac{l(b)}{l(a)}), \tag{2.21}$$

where $l$ is the relative arclength of an arc with respect to the boundary length of the curve and $\theta_0$ is a translation of $T_b$ that minimize the summation.

#### 2.2.3.5 Frechet Distance

The Frechet distance is a distance measure between curves. In contrast the other distance measures, it does not treat the curves as a set of points, but respects the one dimensional structure of the curves. As a popular illustration of the Frechet distance, suppose a man is walking with his dog, on one curve and the dog on the other. Both are allowed to control their speed, but, are not allow to go backwards. The Frechet distance of the curves is the minimal length of a leash that is necessary.

Formally speaking, let $A(\alpha(s))$ and $B(\beta(s))$ two parameterized curves and let their parameterization $\alpha$ and $\beta$ be continuous functions of the same parameter $s \in [0,1]$, such that $\alpha(0) = \beta(0) = 0$ and $\alpha(1) = \beta(1) = 1$. The Frechet distance is the minimum over all monotone increasing parameterization $\alpha(s)$ and $\beta(s)$ of the maximal distance $d(A(\alpha(s), B(\beta(s)))$ [134],

$$d_F(A,B) = \inf_{\alpha,\beta} \max_{s\in[0,1]} d(A(\alpha(s), B(\beta(s))). \tag{2.22}$$

In [6], the Frechet distance between two shapes represented by polygonal curves, with the number of points $M$ and $N$, is calculated in time $O(MN\log(MN))$. Other examples of Frechect distance like minimizing the Frechet distance under translation [7], and a generalized Frechet distance between a curve and a graph [5] can be given. Some variations of Frechet distance can be obtained by dropping the monotonic condition of the parameterization or considering partial matching [133].

#### 2.2.3.6 Nonlinear Elastic Matching Distance

Nonlinear Elastic Matching Distance is a string matching technique and is developed for the finite point sets of ordered contour points. Note that more detailed information about the string matching techniques is given in chapter 5.

Given two ordered contour point sets $A = \{a_1, a_2, ..., a_N\}$ and $B = \{b_1, b_2, ..., b_M\}$, let $f$ be a correspondence between all points in $A$ and all points in $B$ such that there are no $f(a_i) > f(a_{i+1})$, the Nonlinear Elastic Matching Distance is defined as;

$$d_N(A, B) = \min_f \sum_i s(a_i, f(a_i)) + d(a_i, f(a_i)), \tag{2.23}$$

where $s(a, f(a_i))$, the stretch of $(a_i, f(a_i) = b_j)$ is 1 if either $f(a_{i-1}) = b_j$ or $f(a_i) = b_{j-1}$, or 0 otherwise. It can be efficiently computed by using dynamic programming. It is not a metric distance, since it does not obey the triangle inequality.

#### 2.2.3.7 Reflection Distance

The reflection metric, introduced in [54], is defined on finite unions of curves in the plane. If $A$ and $B$ are such unions, the reflection distance first converts the patterns $A$ and $B$ into real valued functions $\rho_A$, $\rho_B$: $\mathbb{R}^2 \to \mathbb{R}$. The functions are defined using a strong form of visibility, called reflection-visibility. These functions are then compared using normalized difference;

$$d_R(A, B) = \frac{\sum_{R^2} \mid \rho_A(x) - \rho_B(x) \mid}{\sum_{R^2} \max(\rho_A(x), \rho_B(x))}. \tag{2.24}$$

The reflection metric can be computed in $O(r(N+M))$, where $r$ is the the complexity of the overlay of the reflection-visibility partitions of curve unions $A$ and $B$ which contain $N$ and $M$ segments, respectively [54].

The reflection metric is invariant under all affine transformations. It is also robust to deformation, blur, crack and noise.

## 2.3 Shape Recognition

Shape analysis systems extensively use the methodologies of pattern recognition, which assigns an unknown sample into a pre-defined class. Numerous techniques for pattern recognition can be investigated in four general approaches [62]:

1. Template Matching,

2. Statistical Techniques,

3. Structural Techniques,

4. Neural Networks.

The above approaches are neither necessarily independent nor disjoint from each other. Occasionally, a recognition technique in one approach can also be considered to be a member of other approaches. This section gives an overview of above approaches by focusing on the shape recognition problem.

### 2.3.1 Template Matching

The simplest way of shape recognition is based on matching the stored prototypes against the unknown shape to be recognized. Generally speaking, matching operation determines the degree of similarity between two vectors in the feature space. Based on the features, recognition is done by matching the feature vectors in order to decide to which pattern class they belong. Matching techniques can be studied in three classes:

- **Direct Matching and Correlation Techniques:** A gray-level or binary input shape is directly compared to a standard set of stored prototypes. According to a similarity measure, a prototype matching is done for recognition. The comparison methods can be as simple as one-to-one comparison, or as complex as decision tree analysis in which only selected pixels are tested. A template matcher can combine multiple information sources, including match strength and k-nearest neighbor measurements from different metrics [129], [47]. Although direct matching method is intuitive and has a solid mathematical background, the recognition rate of this method is very sensitive to noise. An alternative method is the use of deformable templates, where shape deformation is used to match an unknown shape against a database of known shapes. In [64], two shapes are matched by deforming the contour of one, to fit the edge strengths of the other. A dissimilarity measure is derived from the amount of deformation needed, the goodness of fit of the edges and the interior overlap between the deformed shapes (see figure 2.8).

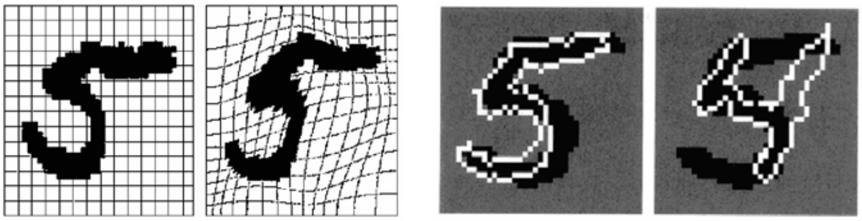


Figure 2.8: Deformable templates, (a) Deformations of a sample digit, (b) Deformed template superimposed on target image, with dissimilarity measures in [64]

- **Feature Analysis and Matching Techniques:** These techniques are based on matching on feature spaces, which are distributed on an n-dimensional hyperplane. A set of features those represents a characteristic portion of a shape, or a group of shapes is compared to the feature vector of the ideal shape class. The description that matches most "closely" according to a distance measure provides recognition [127].

- **Flexible Matching Techniques:** Some flexible matching techniques are proposed as a better alternative to direct or feature analysis matching techniques.

  - *Curve and Elastic Matching:* A shape is recognized by matching a representative curve against those of prototype shapes. It is a linear alignment of the points of the curve. However, due to nonlinearities, the best fit is usually an elastic matching. The basic idea of elastic matching is to optimally match the unknown symbol against all possible elastic stretching and compression of each prototype. Once the feature space is formed, the unknown vector is matched using dynamic programming and a warping function [58], [120].

  - *Relaxation Matching:* It is a symbolic level image matching technique that uses feature-based description for the object image. First, the matching regions are identified. Then, based on some well-defined ratings of the assignments, the image elements are compared to the model. This procedure requires a search technique in a multi-dimensional space, for finding the global maximum of some functions [104], [141].

The matching methods mentioned above are sometimes used individually or combined in many ways as part of the recognition schemes.

### 2.3.2 Statistical Techniques

Statistical decision theory is concerned with statistical decision functions and a set of optimality criteria which maximizes the probability of the observed pattern given the model of a certain class [39]. Statistical techniques are, mostly, based on three major assumptions:

- Distribution of the feature set is Gaussian or, in the worst case, uniform,
- There is sufficient statistics available for each shape class,
- Given ensemble of shapes, one is able to extract a set of features $\{fi\} \in F$, $i \in \{1, ..., n\}$, which represents each distinct class of shapes.

The measurements taken from n-features of each shape can be thought to represent an n-dimensional vector space and the vector, whose coordinates correspond to the measurements taken, represents the original shape. The major statistical approaches, applied in the shape recognition are the followings:

- **Non-parametric Recognition:** This method is used to separate different pattern classes along hyper planes defined in a given hyperspace. The best known method of non-parametric classification is the Nearest Neighbor (NN) and is extensively used in shape recognition [118]. It does not require a priori information about the data. An incoming pattern is classified using the cluster, whose center is the minimum distance from the pattern over all the clusters.
- **Parametric Recognition:** When a priori information is available about the object shapes in the training data, it is possible to obtain a parametric model for each shape class [20]. Once the parameters of the model, which is based on some probabilities, are obtained, the shapes are classified according to some decision rules such as Maximum Likelihood or Bayes method.
- **Clustering Analysis:** The clusters of shape features, which represent distinct classes, are analyzed by way of clustering methods. Clustering can be performed

either by agglomerative or divisive algorithms. The agglomerative algorithms operate step-by-step merging of small clusters into larger ones by a distance criterion. On the other hand, the divisive methods split the shape clusters under certain rules for identifying the underlying shape [142].

- **Hidden Markov Modeling (HMM):** Hidden Markov Models are the most widely and successfully used technique for shape recognition applications, such as handwritten character recognition [30], [75], [94]. It is defined as a stochastic process generated by two interrelated mechanisms; a Markov Chain having a finite number of states and a set of random functions, each of which is associated with a state [108]. At discrete instants of time, the process is assumed to be in some state and an observation is generated by the random function corresponding to the current state. The underlying Markov chain then changes states according to its transitional probabilities. Here, the job is to build a model that explains and characterizes the occurrence of the observed symbols. The output corresponding to a single symbol can be characterized as discrete or continuous. Discrete outputs may be characters from a finite alphabet or quantized vectors from a code-book, while continuous outputs are represented by samples from a continuous waveform. There are two basic approaches to shape recognition systems using HMM:

  *Model Discriminant HMM:* A model is constructed for each shape class, in the training phase. States represent cluster centers for the feature space. The goal of classification is then to decide on the model, which produces the unknown shape observation.

  *Path Discriminant HMM:* In this approach, a single HMM is constructed for the whole shape classes. Modeling is supported by the initial and transitional probabilities on the basis of observations from a random experiment. Recognition consists of estimation of the optimal path for each class using Viterbi algorithm, based on dynamic programming.

- **Fuzzy Set Reasoning:** Instead of using a probabilistic approach, this technique employs fuzzy set elements in describing the similarities between the features of the shapes. Fuzzy set elements give more realistic results, when there is no a priori knowledge about the data and therefore the probabilities cannot be calculated. The shapes can be viewed as a collection of strokes, which is compared to reference patterns by fuzzy similarity measures. Since, the strokes under consideration are fuzzy in nature; the concept of fuzziness is utilized in the similarity measure. In order to recognize a shape, an unknown input shape is matched with all the reference shapes and is assigned to the class of the reference shape with the highest score of similarity among all the reference shapes [31], [1].

### 2.3.3 Structural Techniques

The recursive description of a complex pattern in terms of simpler patterns based on the shape of the object was the initial idea behind the creation of structural pattern recognition. These patterns are used to describe and classify the shapes. The shapes are represented as the union of the structural primitives. It is assumed that the shape primitives are quantifiable and one can find the relations among them. The following structural methods are applied to the shape recognition problems:

- **Grammatical Methods:** In mid 1960's, researchers started to consider the rules of linguistics for analyzing the speech and writing. Later, various orthographic, lexicographic and linguistic rules were applied to the recognition schemes. The grammatical methods create some production rules in order to form the shapes from a set of primitives through formal grammars. These methods may combine any type of topological and statistical features under some syntactic and/or semantic rules, [123], [115].

  In grammatical methods, training is done by describing each shape by a grammar $G_i$. In the recognition phase, the string, tree or graph of shape is analyzed in order to decide to which pattern grammar it belongs [19]. Top-down or bottom-up parsing does syntax analysis. In shape recognition, Picture Description Language (PDL) is used to model each shape in terms of a set of strokes and their relationship.

- **Graphical Methods:** Shapes are represented by trees, graphs, di-graphs or attributed graphs. The shape primitives (e.g. strokes) are selected by a structural approach, irrespective of how the final decision making is made in the recognition [71], [119]. For each class, a graph or tree is formed in the training stage to represent shape primitives or shapes. Recognition stage assigns the unknown graph to one of the classes by using a graph similarity measure.

### 2.3.4 Neural Networks

A Neural Network is defined as a computing architecture that consists of massively parallel interconnection of adaptive 'neural' processors. Because of its parallel nature, it can perform computations at a higher rate compared to the classical techniques. Because of its adaptive nature, it can adapt to changes in the data and learn the characteristics of input shape. A neural network contains many nodes. The output from one node is fed to another one in the network and the final decision depends on the complex interaction of all nodes. In spite of the different underlying principles, it can be shown that most of the neural network architectures are equivalent to statistical pattern recognition methods [111].

Several approaches exist for training of neural networks [63]. These include the error correction, Boltzman, Hebbian and competitive learning. They cover binary and continuous valued input, as well as supervised and unsupervised learning. On the other hand, neural network architectures can be classified into two major groups, namely, feed-forward and feedback (recurrent) networks. The most common neural networks used in the CR systems are the multilayer perceptron of the feedforward networks and the Kohonen's Self Organizing Map (SOM) of the feedback networks.

Multilayer perceptron, proposed by R. Rosenblatt [23] and elaborated by Minsky and Papert [93], is applied in shape recognition by many authors. Mohiuddin et. al. use multi-network system in shape recognition by combining contour direction and bending point features [90], which is fed to various neural networks. Neocognitron of Fukushima et. al. [46] is a hierarchical network consisting of several layers of alternating neuron-like cells. S-Cells are for feature extracting and C-Cells allow for positional errors in the features. Last layer is the recognition layer. Some of the connections are variable and can be modified by learning. Each layer of S and C cells are called cell planes. This study proposes some techniques for selecting training patterns useful for deformation-invariant recognition of a large number of shape classes. The feedforward neural network approaches to shape recognition problem is proved to be successful, in [13], where the neural network is trained with a large number of shape classes and tested in large database of shapes. A recent study, proposed by Maragos and Pessoa, incorporates the properties of multilayer perceptron and morphological rank neural networks for shape recognition. They claim that this unified approach gives higher recognition rates than multi-layer perceptron with smaller processing time [101].

Recent developments on shape recognition research are concentrated on Kohonen's Self Orginizing Map [74]. SOM integrates the feature extraction and recognition steps in large training set of shapes. It can be shown that it is analogous to k-means clustering algorithm. An example of SOM on CR systems is the study of Liou and Yang [83], which presents a self organization matching approach. In [110], Reddy and Nagabhushan propose a combination of modified SOM and Learning Vector quantization to define three dimensional neural network model for shape recognition. They report higher recognition rates with shorter training time then other SOMs reported in the literature. Jabri et. al. realized the adaptive-subspace self organizing map to built a modular classification system [144].

# CHAPTER 3

# SHAPE REPRESENTATION BASED ON BEAM ANGLE STATISTICS : BAS

Can we define a shape descriptor, which represents **all** the visual information of a shape boundary in a set of 1-D functions ? If the answer is yes, then it is possible to quantize these functions to define a feature vector, which bears **almost all** the visual information. This representation should carry the concavities and convexities of the shape boundary.

Practically speaking, this task may be achieved by assuming a wire shape, cutting the wire at some point and slightly opening both ends to lie the wire on a flat surface, without disturbing the concavities and convexities. Then, the next question to ask is how to define a mathematical formalism in order to obtain the wire shapes in a 1-D representation.

In this chapter, we introduce a new shape representation technique, Beam Angle Statistics (BAS), based on the above motivation. BAS is derived by navigating on the boundary, taking the point of bearings, and calculating the statistics of the angles between a pair of beams, joined at the point of bearing, at various neighborhood systems.

After giving the relevant background in Section 3.1, Section 3.2 introduces the BAS shape representation. The consistency of the proposed representation with Human Visual System, invariance and robustness of the representation under various transformations as well as noise and occlusion, are discussed in sections 3.3 and 3.4, respectively. Computational complexity is analysed in section 3.5. Finally, a discussion about the proposed representation is given in section 3.6.

## 3.1 K-Curvature

As mentioned in chapter 2, the significance of curvature information at object boundaries has been an inspiration of many studies related to shape analysis [85]. A variety of algorithms use high curvature points as corners and approximate the shape by a polygon [77]. Another curve approximation scheme is based on splines, which posses the beneficial property of minimizing curvature [34]. In some studies, the shape is decomposed into parts at points of high negative curvature [116]. Scale space techniques are applied to shape boundary by filtering with Gaussian kernels of increasing width to obtain a multi-scale representation of curvature [96]. The curvature function is directly used to represent the shape boundary in [132].

The curvature function basically describes how much a curve bends at each point of shape boundary and is defined as the rate of change of the curve slope $\theta(\mathbf{s})$ with respect to its length $\mathbf{s}$. The slope angle of a curve, $\Gamma(\mathbf{s}) = (x(\mathbf{s}), y(\mathbf{s}))$, is given by

$$\theta(\mathbf{s}) = \tan^{-1} \frac{dy}{dx}. \tag{3.1}$$

The curvature function, $\kappa(\mathbf{s})$, is then calculated by the rate of change of $\theta(\mathbf{s})$:

$$\kappa(\mathbf{s}) = \frac{d\theta(\mathbf{s})}{d\mathbf{s}} = \frac{\dot{x}(\mathbf{s})\ddot{y}(\mathbf{s}) - \dot{y}(\mathbf{s})\ddot{x}(\mathbf{s})}{(\dot{x}(\mathbf{s})^2 + \dot{y}(\mathbf{s}))^{\frac{3}{2}}}. \tag{3.2}$$

The above calculation of curvature requires second derivatives, which is not practical for discrete curves. For this purpose, curvature based shape description methods use some heuristics. In [96], the shape boundary is filtered by Gaussian function and the curvature is calculated directly along the smoothed curve definition. Another study uses the edge gradient at each point by the arctangent of its Sobel difference, in a $3x3$ neighborhood [84].

The curvature function can be computed as the derivative of the contour's slope function. In order to prevent noise in the slope function caused by fluctuations on the boundary, smoothing is applied into the slope evaluation. This can be performed by $K$-slope method. $K$-slope at a boundary point is defined as the slope of a line connecting that point with its Kth right neighbor. Then, the $K$-curvature at a boundary point is defined as the difference between the $K$-slope at that pixel and the $K$-slope of its $K^{th}$ left neighbor.

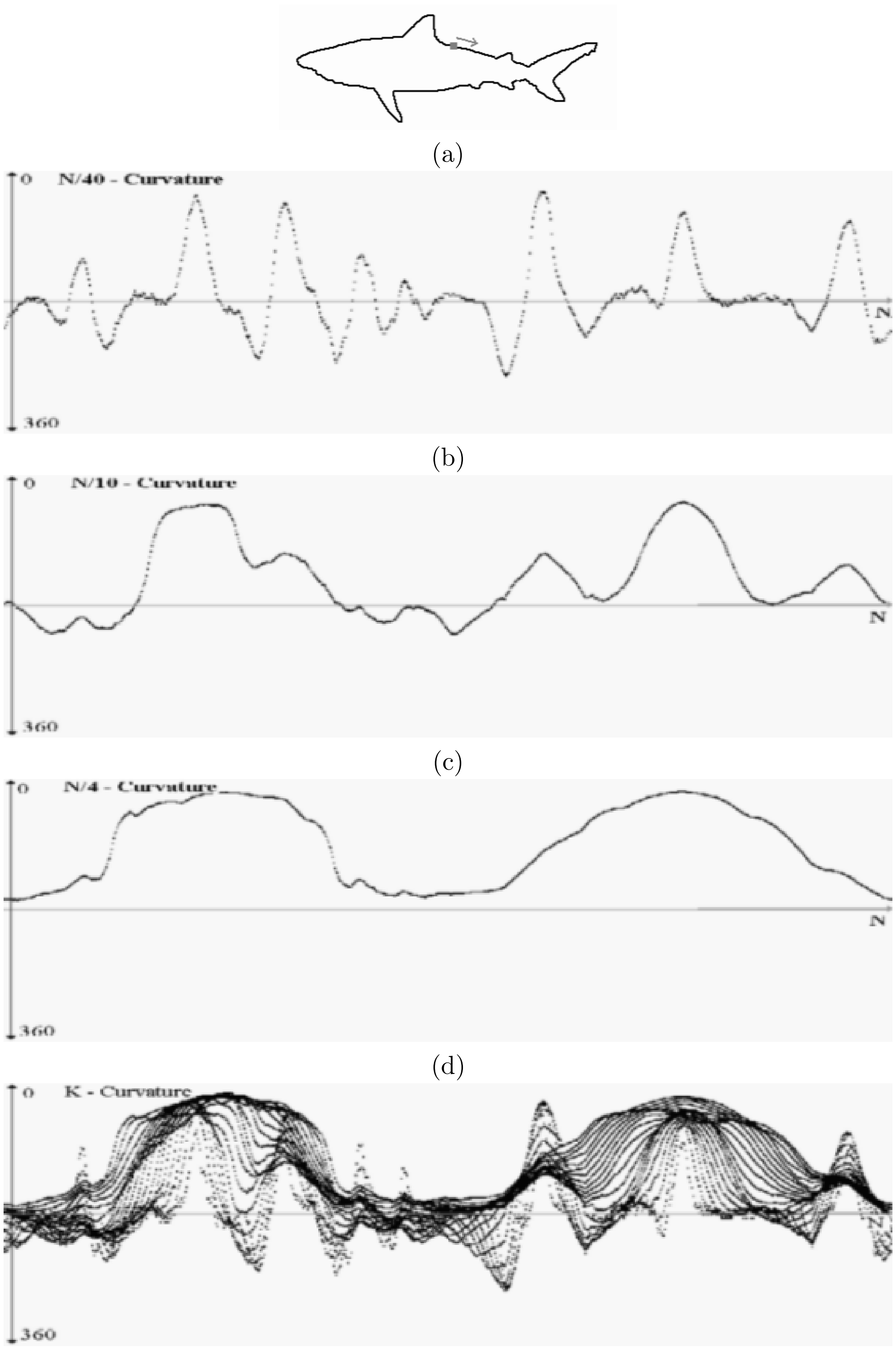
The $K$-curvature function may be exploited to form an appropriate shape descriptor, if one could identify an optimal value for the parameter $K$. Because; for a fixed $K$, the plot of $K$-curvature function represents a one-dimensional function, which extracts the concavities and convexities of the shape at a predefined scale settled by $K$.

At this point, we need a rigorous technique to identify the scale parameter $K$, which discriminates a wide range of shapes in large image databases. In [2], the curvature function is obtained by using a fixed $K$ value and then it is filtered in order to stress the main features. Another study [132] uses an adaptive $K$ value by changing it according to the distance between relevant points. It is also possible to estimate an "optimum" value for $K$ using the popular training methods. However, choosing a single $K$ value cannot capture the exact curvature information for all varieties of shapes in the database. In addition, the heuristics used in estimation of $K$ may not be valid for some shapes.

## 3.2 BAS: A Shape Descriptor with Beam Angles Statistics

Let us start by investigating the effect of the parameter $K$ on the resolution of the $K$-curvature representation. Figure 3.1 b, c and d indicate the plot of $K$-curvature function for a sample shape given in figure 3.1 a, with various $K$ values. Note that the plots for small $K$ capture fine details. By increasing $K$, it is possible to plot fine to curse representation of a closed boundary as a one-dimensional function. Examining these figures shows that each peak and valley of the curvature function plot corresponds to a convexity, and a concavity of the shape. For this particular example, the shape information is preserved in the peaks corresponding to the head, tail and fins. The remaining peaks and valleys of the curvature plot are rather redundant details, that result from the noise in the shape, leading mismatches in the image database. Therefore, one needs to select an "appropriate" $K$ to avoid this redundant information.

Visual inspection of Figure 3.1 indicates that in order to capture the perceptu-



(a)

(b)

(c)

(d)

(e)

Figure 3.1: A sample shape boundary of a fish taken from MPEG-7 database and its various K-Curvatures, (a) A sample shape, its curvature function for (b) K=N/40, (c) K=N/10, (d) K=N/4 and (e) the superposition of K = N/40...N/4 from top to bottom respectively.

ally meaningful concavities and convexities, $K = N/10$ may be an appropriate choice to represent the silhouette of the fish. As we decrease the value of $K$ to $N/40$, we obtain too many spurious peaks, whereas increasing the $K$ value to $N/4$ yields a representation, which lacks the peaks and valleys corresponding to important perceptual information. Therefore, selection of $K$ defines the amount of smoothing of the shape and highly depends on the context of the image. If $K$ smoothes the ripples corresponding to some context information, this will result in important information loss. On the other hand, if the ripples correspond to noise, keeping them will increase the number of convexities and concavities, which may carry superfluous shape information. As a result, the problem of selecting a generic $K$, which resolves the necessary and sufficient information for all the images, has practically no solution, due to the diversity of the shape context in large databases.

The above discussion leads us to somehow find a representation, which employs the information in $K$-curvature function for **all** values of $K$. This representation should be compact enough to yield a shape descriptor. Superposition of the plots of $K$-curvature function for all $K$, yields impractically large data with no formal way of representation, as indicated in figure 3.1.e.

In this study, we attack this problem by modeling the shape as the outcomes of a stochastic process, which is generated by the same source at different scales. In this model, Figure 3.1.e shows the possible outcomes of the shape curvature plot, which generates the fish shape. At a given boundary point $p(i)$, the value of $K$-curvature function is assumed to be a function of a random variable $K$ and may take one of the values indicated in the figure, depending on the values of $K$. Therefore, $K$-curvature function can be considered as the output of a stochastic process. In the following formalism, we introduce the Beam Angle which is closely related to the curvature slope.

Mathematically speaking, let the shape boundary $B = p(1), ..., p(N)$ is represented by a connected sequence of points,

$$p(i) = \{x(i), y(i)\}, \qquad i = 1...N, \tag{3.3}$$

where $N$ is the number of boundary points and $p(i) = p(i + N)$. For each point of bearings, $p(i)$, the beams of $p(i)$ is defined as the set of vectors;

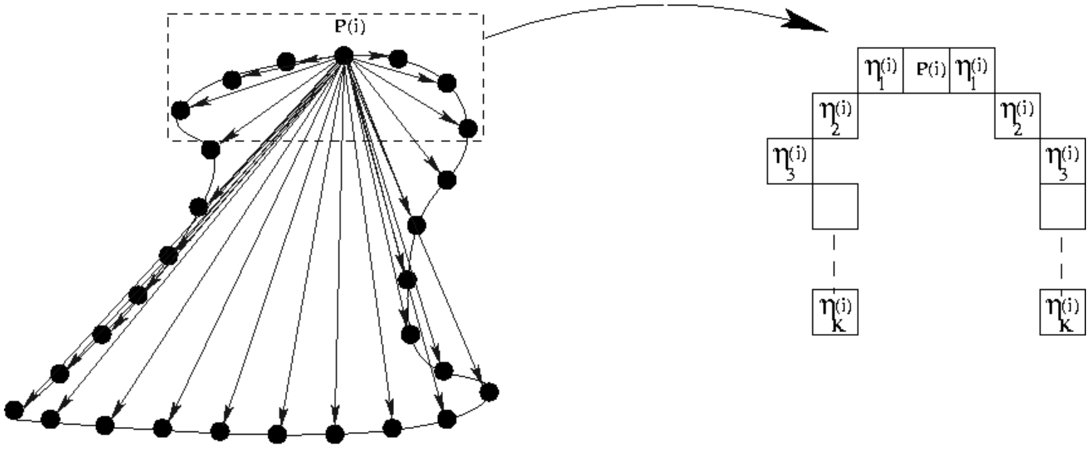


Figure 3.2: The beams of point $p(i)$, and its neighborhood systems.

$$L(p(i)) = \{V_{i+j}, V_{i-j}\}, \tag{3.4}$$

where $V_{i+j}$ and $V_{i+j}$ are the forward and backward vectors connecting $p(i)$ with the points, $p(i+j)$ and $p(i-j)$ in the boundary, for $j = 1...N/2$ . Figure 3.2.a indicates the beams of a point $p(i)$.

Also, the $K^{th}$ order neighborhood system is defined as

$$p(i \mp K) \in \eta_K(p(i)) \qquad \forall p(i), \qquad i = 1...N, \quad K = 1...N/2. \tag{3.5}$$

Note that, for each neighborhood system $K$, there is only one pair of beams, connecting $p(i)$ to $p(i+K)$ and $p(i-K)$. Figure 3.2.b indicates the pixels in different neighborhood systems.

The slope of each beam , $V_{i+k}$ is then calculated as,

$$\theta_{V_{i+l}} = \tan^{-1}\frac{\Delta y_{i+l}}{\Delta x_{i+l}} \qquad , \qquad l = \pm k. \tag{3.6}$$

For the point of bearings, $p(i)$, the beam angle between the forward and backward beam vectors in the $K^{th}$ order neighborhood system, is then computed as (see Figure 3.3)

$$C_K(i) = (\theta_{V_{i-l}} - \theta_{V_{i+l}}). \tag{3.7}$$

Note that, beam angle at a neighborhood system $K$ is nothing, but the $K$-curvature function, which takes values between 0 and $2\pi$.

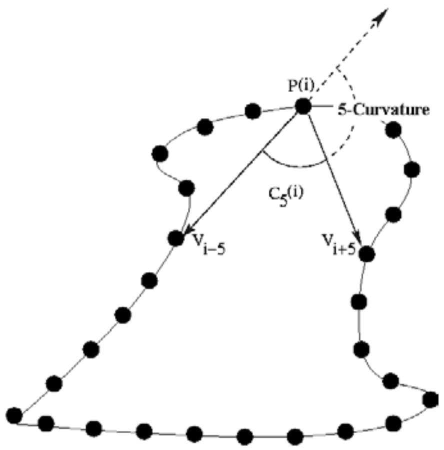


Figure 3.3: The 5-curvature and the beam angle at the neighborhood system $\eta_5$ for the boundary point p(i).

Now, for each point of bearings, $p(i)$, on the curve $\Gamma$, the beam angle $C_K(i)$ in the neighborhood system $\eta_K$ can be taken as a **random variable** with the probability density function $P_K(C_K(i))$ and $C_K(i)$ vs. $i$ plot for each $K$ becomes an outcome of the stochastic process, which generates the shape at different scales. Therefore, **Beam Angle Statistics** (BAS), may provide a compact representation for a shape descriptor. For this purpose, the mean of the random variable $C_K(i)$ is defined as:

$$\mu = E(C^1(i)) = \sum_{K=1}^{N/2} C_K P_K(C_K(i).) \tag{3.8}$$

In the above formula $E$ indicates the expected value operator and $P_K(C_K(i))$ is the probability density function of $C_K(i)$. Note that, the maximum value of $K$ is $N/2$, where N represents the total number of boundary points. Note also that, the value of $C_K(i)$ approaches to 0 as $K$ approaches to $N/2$. During the implementations, $P_K(C_K(i))$ is approximated by the histogram of $C_K(i)$ at each boundary point $p(i)$.

The absolute central moments are, then, defined by the following equation.

$$E(C^m(i)) = \sum_{K=1}^{N/2} \mid C_K^m - \mu \mid^m P_K(C_K(i)), \qquad m = 2, 3, ... \quad . \tag{3.9}$$

The moments describe the statistical behavior of the beam angle at the boundary point $p(i)$. Each boundary point is, then, represented by a vector, whose components are the moments of the beam angles:

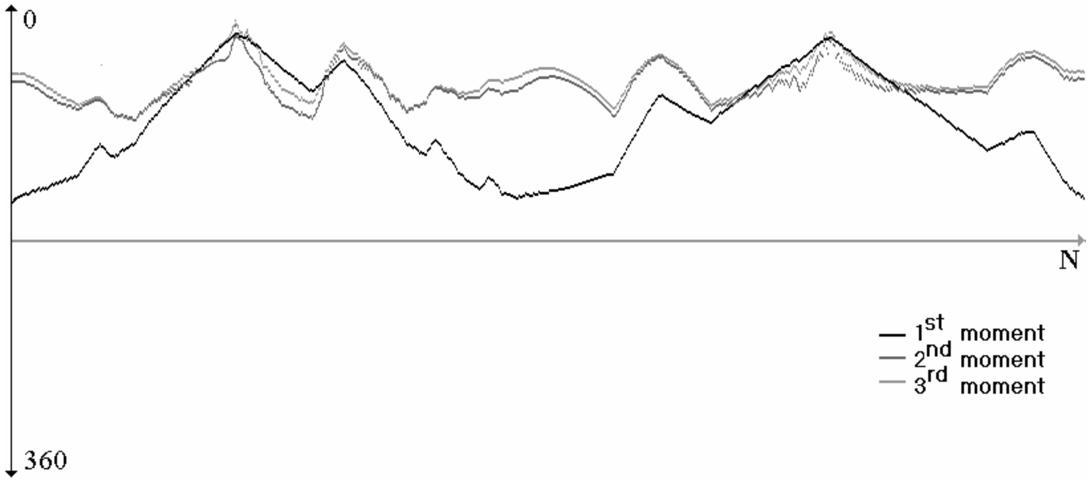


Figure 3.4: Third order statistics of beam angle for the shape in figure 3.1.a.

$$\Gamma(i) = (E(C^1(i)), E(C^2(i)), ...). \tag{3.10}$$

The shape boundary is, finally, represented by the plot of the moments of $C_K(i)$'s vs $i$. for all the boundary points. In the proposed representation, the first moment preserves the most significant information of the object. Figure 3.4, indicates the first three moments of boundary points for the sample shape of figure 1.a. For this particular example, it should be noted that the first moment suffice to represent the convexities and concavities of the shape. The second moment increases the discriminative power of the representation. The third moment, on the other hand, does not bring any additional significant information to the representation for this particular shape. The order of the statistics, naturally, depends on the characteristics of the probability density function, $P_K(C_K(i))$. Central limit theorem provides us a strong theoretical basis to assume Gaussian distribution for each point $p(i)$ for large N. This implies that second order statistics is sufficient for representing most shapes, provided that we have "enough" samples on shape boundary.

The BAS representation can be summarized by the following algorithm, which calculates the first and second moments of the Beam Angles. Recall that, $P_K(C_K(i))$ is approximated by the histogram of $C_k(i)$ at each point.

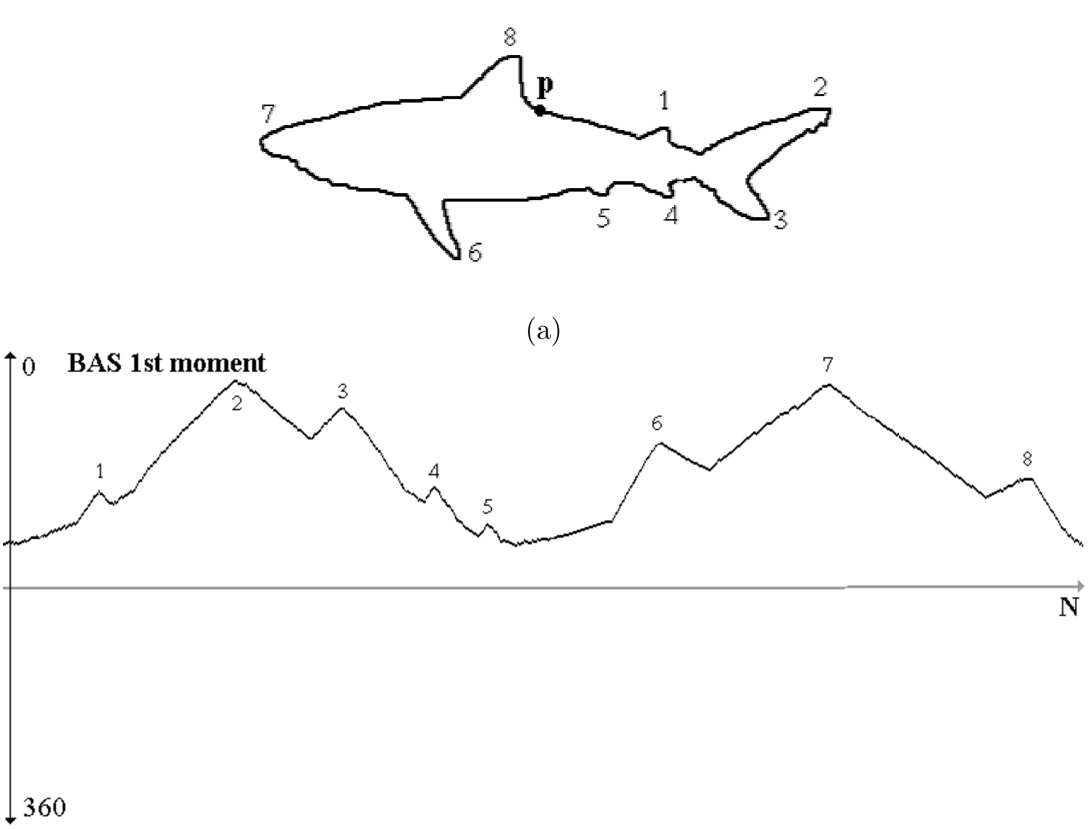
```
                BAS REPRESENTATION ALGORITHM

1. Boundary Extraction:
        $for1 \leq i \leq N$
            $p_i = (x_i, y_i)$
2. Iteration:
        $for1 \leq i \leq N$
            $for1 \leq k \leq N/2$
                $\theta_{V_{i-k}} = arctan \ \ (y_{i-k} - y_i) \ / \ (x_{i-k} - x_i)$
                $\theta_{V_{i+k}} = arctan \ \ (y_{i+k} - y_i) \ / \ (x_{i+k} - x_i)$
                $C_k(i) = (\theta_{V_{i-k}} - \theta_{V_{i+k}})$
            $\Gamma^1(i) = \frac{1}{\frac{N}{2}-1} \sum_{k=1}^{\frac{N}{2}} C_k(i)$
            $\Gamma^2(i) = \sqrt{\frac{1}{\frac{N}{2}-1} \sum_{k=1}^{\frac{N}{2}} (C_k(i) - \Gamma^1(i))^2}$
```

## 3.3 Consistency with Visual Information

There is a strong evidence that Human Visual System perceives shapes based on the ”parts” of the overall object [116]. In other words, shape perception is accomplished through, first, breaking the shape into parts, then evaluating the individual parts and finally, measuring information by the individual parts. Part based recognition allows robustness under distortion and occlusion. In cognitive experiments, it is shown that contours are segmented into visual parts at negative curvature minima [56].

In BAS representation, the transformation of 2-D shape information into a set of 1-D functions is expected to preserve the convex and concave parts of the shape. Therefore, BAS representation inherently carries parts information. This is shown in figure 3.5 by the correspondence of convex parts between the shape boundary and the BAS function.

Notice that, the first moment BAS function resembles a 2-D wire shape, which is cut at point **p** and is slightly opened at both ends for lying on a flat surface, yielding



(a)

(b)

Figure 3.5: Correspondence of visual parts, (a) A sample shape from MPEG-7 database, (b) Correspondence of visual parts in first moment BAS function.

a 1-D representation.

## 3.4 Invariance and Robustness

As we discussed in chapter 2, it is not well-understood what properties are desirable for human shape similarity judgment. It depends on the application whether the invariance under a large group of transformation is desired or not and until which degree the robustness under distortions and occlusions is desirable. Scale, translation and rotation invariance are usually required for shape analysis problems. In recognition problems, rotation invariance impedes the recognition performance, e.g., when distinguishing 6 from 9 rotation invariance would be completely inappropriate. On the other hand, by using shear transformation, two different object shape can be transformed into each other [126]. In this section, we evaluate the BAS representation under affine transformation, noise and occlusion.

### 3.4.1 Affine Invariance

Affine transformations change the positions of shape points by applying a linear combination of translation, rotation, scaling and shearing. The general affine transformation of a vector $V_1 = (x_1, y_1)$ to $V_2 = (x_2, y_2)$ is defined as,

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} x \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} \alpha \\ \beta \end{bmatrix}. \tag{3.11}$$

where $a$, $b$, $c$ and $d$ are the affine parameters.

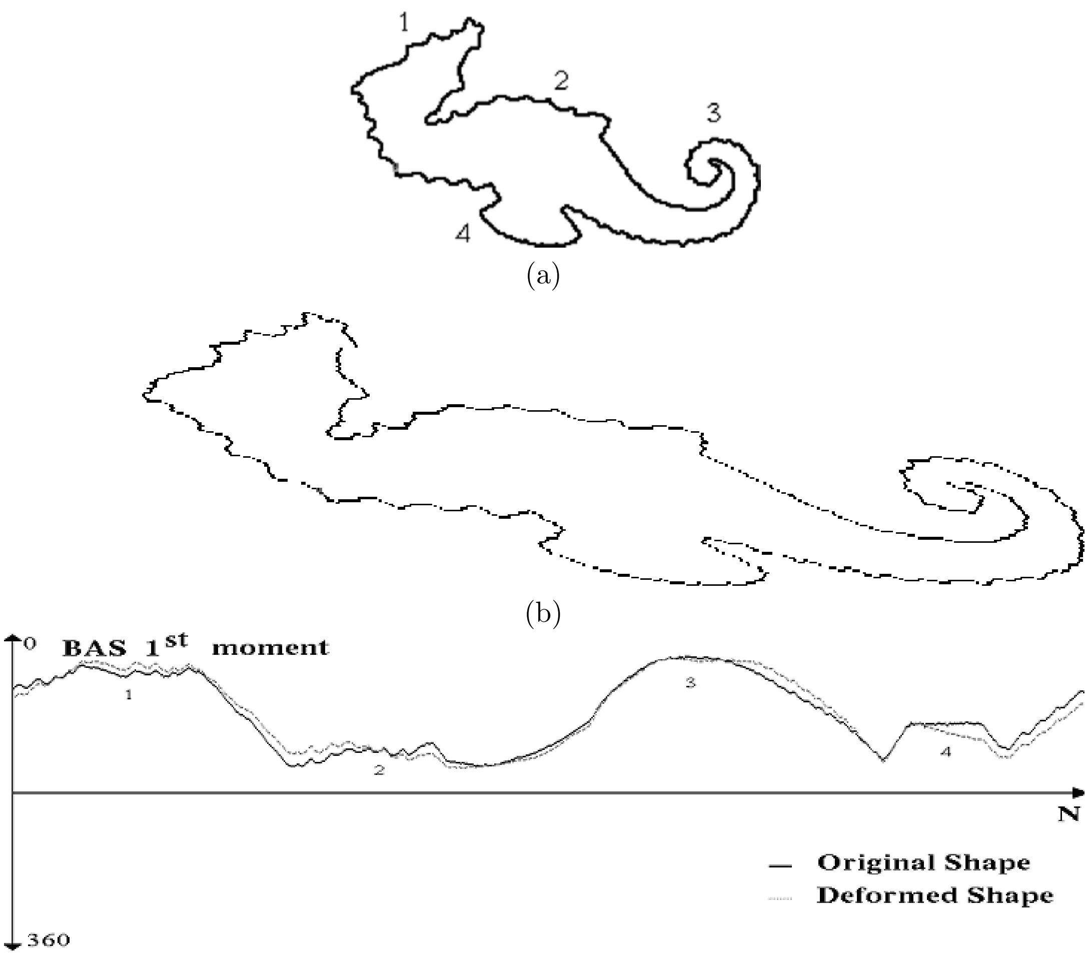
- **Scale :** The scale invariance of the BAS function can be provided by normalizing the boundary of the shapes into equal lengths. Since the number of points in shape boundaries may depend on the shape size, the same number of points can be sampled from the boundary. By this way, scale difference between the shapes is eliminated.

- **Translation :** Invariance to translation is intrinsic to the definition of BAS function. The beams originating from a boundary point are defined as lines connecting that point with the others. The orientation of lines are independent of the point coordinates. Instead, the calculations are performed with respect to the relative positions of points. Thus, the translation of boundary points does not affect the representation.

- **Rotation :** Rotation invariance is, also, comes from the definition of BAS function. The angles between beam pairs do not change if the shape is rotated, yielding the same Beam Angle Statistics for rotated shapes. Starting from different points result in simple circular shift on the BAS function. This shift is handled in the similarity measurement algorithm discussed in chapter 5.

- **Shear :** Although the BAS function is not theoretically shear invariant, it is quite insensitive to shear transformation. This is basically, because of taking the moments over all the neighborhood systems.

  The experiments are performed in order to test the insensitivity of the BAS descriptor under shear and scale. Figure 3.6 shows a sample shape distorted by shear and scale transformation. Note that, the BAS descriptor slightly changes under shear transform, yielding very similar first moments.



(a)

(b)

(c)

Figure 3.6: Insensitivity to shear transformation, (a) A sample shape from MPEG-7 database with major perceptual parts indicated by numbers 1 through 4, (b) The sheared and rescaled version, (c) The comparison of two representations.

### 3.4.2 Robustness to Noise

We, also, claim that the proposed representation is insensitive to small deformations. During the navigation on the shape boundary, at each point of bearings, the rest of the other boundary points are taken into account. Therefore, the noise introduced in some part of the shape boundary is smoothed through the calculation of Beam Angle Statistics for various degree of neighborhood systems. The robustness of BAS representation is tested during the experiments.

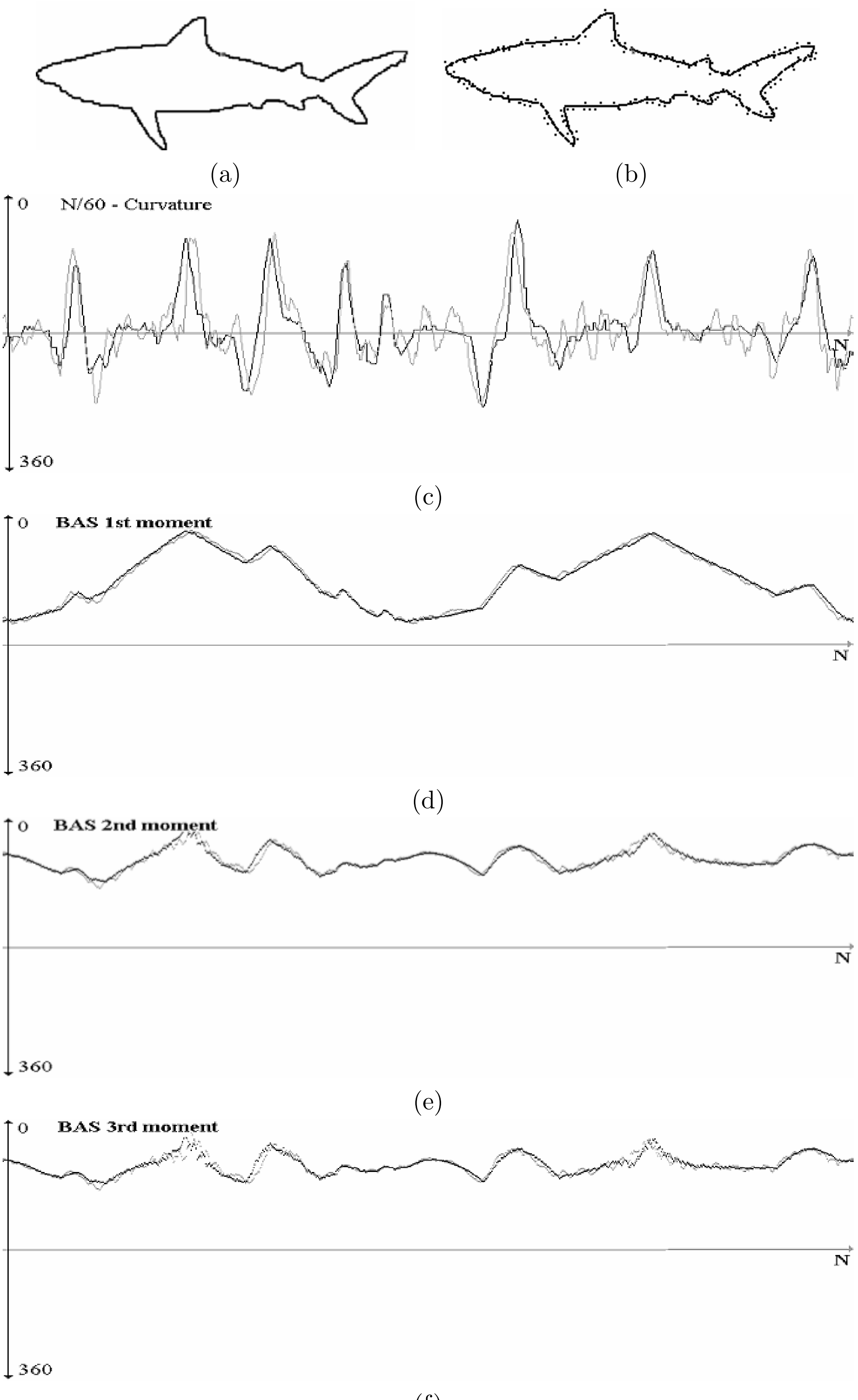
The behavior of the BAS descriptor under uniform noise is exemplified and compared to the K- curvature function plot in figure 3.7, where the fish example distorted by uniform noise. Note that the ripples appeared in the shape due to the noise, add additional peaks and valleys in the K-curvature function, if K is taken small enough to capture the ripples of the noise. The value of K should be large enough to smooth the noise and small enough to preserve the convexities and concavities of the context. On the other hand, the BAS representation smoothes the noise in the first order statistics, yielding almost the same plots for both the original and noisy images. The changes in the second and third moments are rather insignificant.

Conversely, the BAS representation is tested under smoothing effects. Curve evolution method is used for smoothing and the BAS representation of the original shape and evolved shape is compared in figure 3.8. The degree of smoothing is decided by visual inspection, which preserves the maximal amount of context information. Note that, the BAS representation yields very similar first, second and third moments, indicating the robustness of the proposed representation under smoothing.

### 3.4.3 Robustness to Occlusion

One of the crucial issues in shape analysis problem is matching under occlusion. The proposed shape representation assumes that occluded shapes are obtained by, first, deleting a portion of the boundary, then connecting two end points by a straight line. An example of occlusion is given in figure 3.9.a, where a portion of the shape boundary is deleted. The corresponding first moment BAS function differs from the original shape representation mainly on the occluded part. The rest of the BAS functions remains almost unchanged. As long as the occluded portion of the shape is visually insignificant, i.e. does not contain important concavities and convexities, the BAS moment function remains almost the same in the non-occluded portions due to the inherent definition of moments.

Note that the proposed representation cannot handle the shapes with multiple objects, occluding each other or where parts of one shape may match parts of another shape. This is a difficult problem and requires further effort, which is beyond the scope of this thesis.



(a) (b)

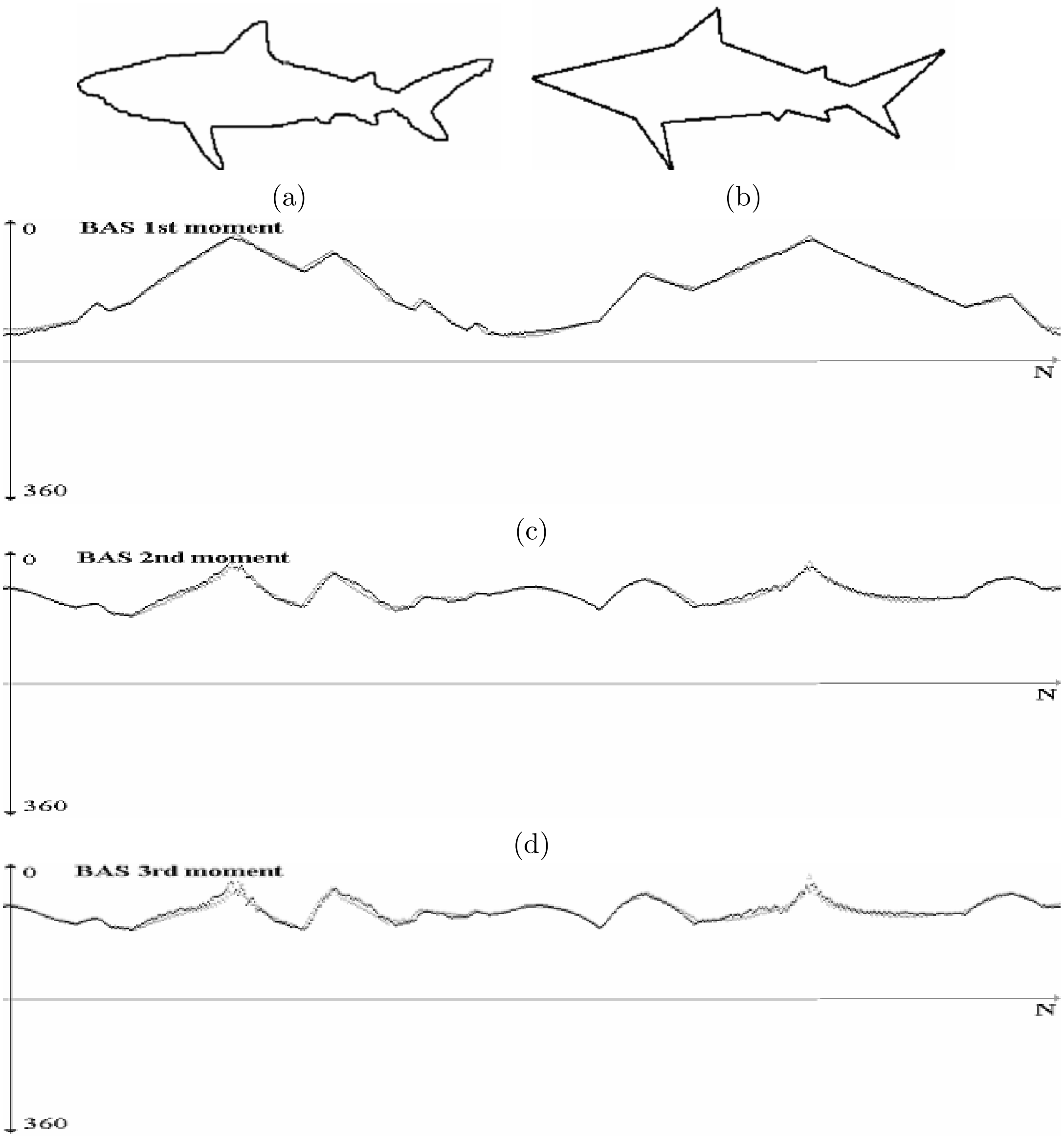
(c)

(d)

(e)

(f)

Figure 3.7: Robustness to Noise, (a) Original shape, (b) Noise added shape, (c) N/60 curvature functions, (d) BAS $1^{st}$ moment function, (e) BAS $2^{nd}$ moment function (f) BAS $3^{rd}$ moment function.



(a)

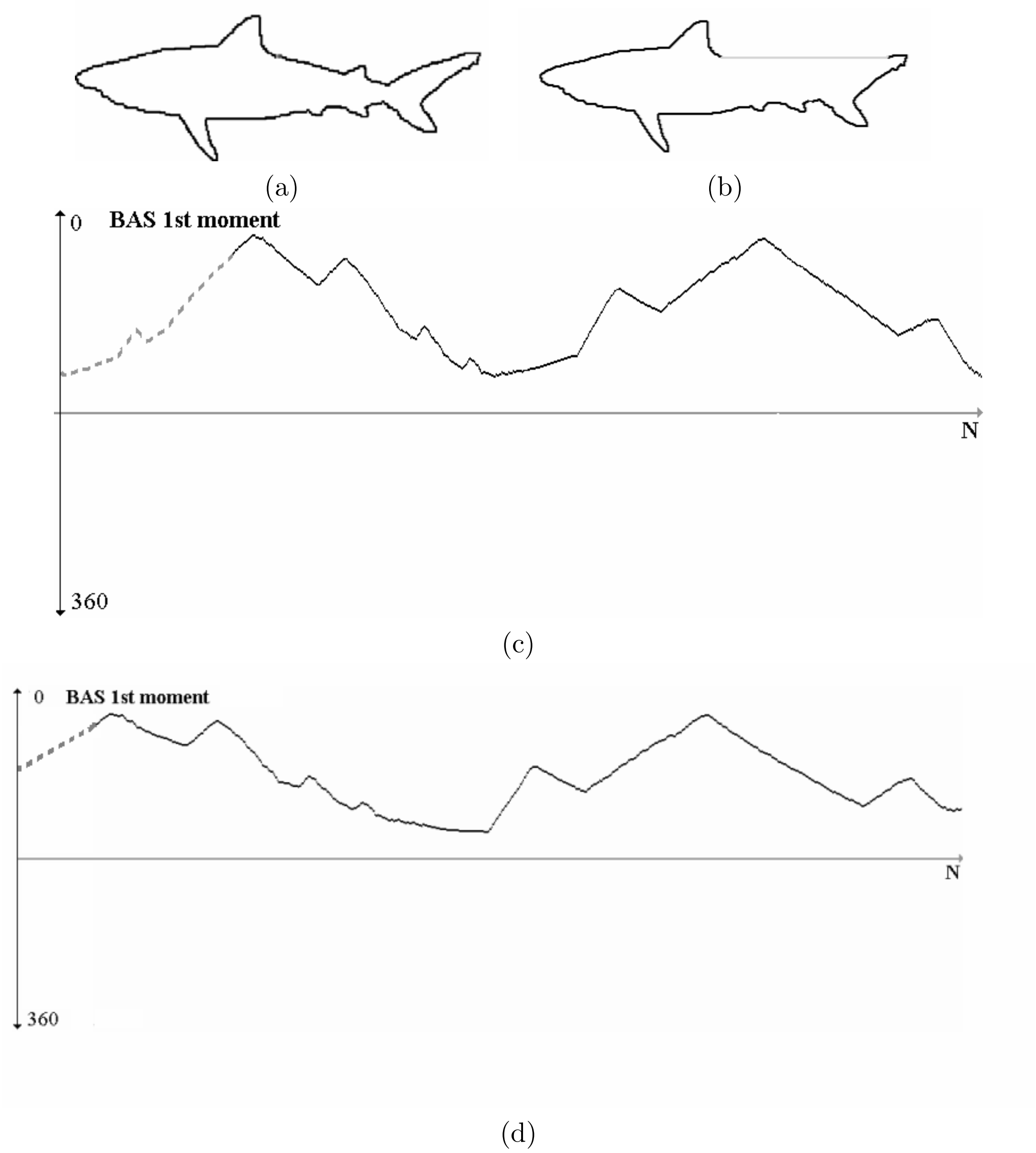
(b)

(c)

(d)

(e)

Figure 3.8: Robustness to smoothing, (a) Original shape (b) Smoothed shape (c) BAS $1^{st}$ moment function (d) BAS $2^{nd}$ moment function (e) BAS $3^{rd}$ moment function.



(a)

(b)

(c)

(d)

Figure 3.9: Robustness to occlusion, (a) Original shape (b) Occluded shape (c) BAS $1^{st}$ moment function of original shape (d) BAS $1^{st}$ moment function of occluded shape.

## 3.5 Computational Complexity of BAS Representation

The time complexity of the BAS representation is dominated by the number of boundary points for each shape. BAS function requires the calculation of beam angle statistics for each boundary point. Since each angle is calculated by a pair of beams, the BAS of a point is found in $N/2$ iterations, where $N$ is the number of boundary points. In each iteration, the direction of two beams and the angle between them are calculated. The same process is performed for each boundary point in representing the whole shape. This results in a time complexity of $O(N^2)$.

## 3.6 Discussion about BAS Representation

In this chapter, we introduce a robust shape descriptor, which transforms two-dimensional shape information into a set of one-dimensional functions, called Beam Angle Statistics. In BAS representation, the transformation of 2-D shape information into 1-D function preserves the convex and concave parts of the shape. It is invariant to scale, translation and rotation transformation. It is also insensitive to small deformations, noise and occlusion.

The motivation of this study is to represent the silhouette of an object by a descriptor, which captures the perceptual information of its shape, in a set of 1-D functions. This representation should be independent of the resolution of the shape boundary. The two-dimensional object silhouettes are mapped into a set of 1-D BAS moment functions, which capture the perceptual information using the statistical information based on the beams of individual points. At each point, the angle between a pair of beams is calculated to extract the topological structure of the boundary. If we assume that the shape boundary is made out of wire, breaking the wire at a point and kindly opening each end without disturbing its topological properties, would yield the desired representation.

The representation avoids smoothing, in order to eliminate the spurious peaks and valleys on the boundary. It also avoids the selection of a threshold value to represent the "sufficient" resolution of the boundary. Since the selection of the threshold, thus the level of details depends on the shape, it somehow eliminates the context-dependency of the representation to the data set. Therefore, rather than using a single representation of the boundary, at a predefined scale, the proposed representation uses the statistics of the representations at **all** scales. It gives globally discriminative features to each boundary point by using all other boundary points. Since it preserves the concave and convex parts of the shapes, it is consistent with the Human Visual System.

# CHAPTER 4

# FEATURE EXTRACTION

Shape description, subsequent to shape representation, is a process performed for uniquely characterizing the shape. The method generates a compact feature vector (also called shape descriptor vector) from a given shape representation. In its original form, the BAS functions can be used as a shape descriptor. However, the use of the row BAS functions may be expensive, when shapes carry a large number of boundary points. In order to reduce the computational complexityof the algorithms, a more compact description is required. For this purposes, a set of features is extracted for each shape, which carries almost all the information of the BAS representation.

This chapter introduces the feature extraction methods used for compactly describing the BAS moment functions. The main goal is to reduce the size of the shape representation without loosing ”important” information. The methods used in this thesis can be divided in two categories: Space Domain Techniques and Transform Domain Techniques.

Space Domain Techniques attempt to extract feature vector directly utilizing the BAS function. Two different methods is developed in this category.

- **Piecewise Constant Approximation :** The BAS function is segmented into equal-size frames and each frame is represented by the segment mean.

- **Polyline Approximation :** This method simplifies the BAS function by an evolution process. In each step, a pair of consecutive line segments on the BAS function is replaced by a single line. At the end of this process the ”maximal” convex parts of the function are preserved.

Transform Domain Techniques, on the other hand, use Fourier and Wavelet Transformations in feature extraction process.

- **Fourier Based Techniques :** Fourier Transformations are utilized in two different ways:
  - *Fourier Descriptors :* Spectral coefficients are used as the entries of the feature vector.
  - *Sampling by Fourier Transformations :* Inverse Fourier Transformation of the low pass filtered signal in spectral domain is calculated and used as the samples of the BAS functions.
- **Wavelet Based Techniques :** Wavelet Transformations are used in two different ways:
  - *Approximation by Wavelet Transforms :* The approximation coefficients of the wavelet transformation at a given scale smoothes the BAS function.
  - *Piecewise Linear Approximation by Wavelet Modulus Maxima :* The maxima and minima on the BAS function is detected by using Wavelet Transform Modulus Maxima (WTMM) approach.

Note that, the above methods perform the dimension reduction on the shape representation either by identifying Nyquist rate or detecting dominant points of the BAS functions. In each cases, the number of boundary points $N$, is reduced to a value $T$, where $T \ll N$. $T$ is taken as a database dependent variable during the experiments. An effort is spent to find the optimal rate for the shapes in the database.

In the following sections, the above methods are described in detail. After explaining the space domain techniques in section 4.1. Fourier and Wavelet Transformation methods are presented in section 4.2 and section 4.3, respectively. Finally, the discussion about the methods and the value of vector size $T$ is given in section 4.4.

## 4.1 Space Domain Techniques

This category of methods are based on the curve approximation approaches by interpolating the BAS moment functions. Rather than representing the whole boundary by a single function or string, it is generally more interesting to derive piecewise representations which approximate each portion of the representation by a geometric primitive such as a straight line segment. Basically, this approach can be divided into two steps: segmentation and approximation. The points which define the segmentation boundary are called dominant points and can be found in number of ways.

In this study, two space domain methods for BAS function approximation are used. First, the segmentation is performed by simply taking the equal length frames on the BAS function. Second, the dominant points on the BAS function is determined by a recursive evolution process as explained in the following subsections.

### 4.1.1 Piecewise Constant Approximation

Recall that, a BAS function is denoted as $\Gamma^m(i)$, $i = 1...N$, where $N$ is the number of points on the shape boundary. Let $T$ be the dimensionality of the feature space we wish to extract $(1 \leq T \leq N)$. For convenience, $T$ is assumed to be a factor of $N$. Although, this is not a requirement for feature extraction, it simplifies the notation.

The discrete BAS function $\Gamma^m$ of length $N$ can be represented in $T$ dimensional space by a vector $\overline{\Gamma}^m = \overline{\Gamma}^m(1), ..., \overline{\Gamma}^m(T)$.

The $j^{th}$ element of $\overline{\Gamma}^m$ is calculated as follows:

$$\overline{\Gamma}^m(j) = \frac{T}{N} \sum_{i=\frac{N}{T}(j-1)+1}^{\frac{N}{T}j} \Gamma^m(i). \tag{4.1}$$

Intuitively, in order to reduce the dimension of BAS function from $N$ to $T$, the data is divided into $T$ equal-size frames. The mean value of the data falling within a frame is calculated and the value obtained at each segment becomes the entries of the feature vector with dimension $T$. The representation can best be visualized as an attempt to model the original BAS function with linear combination of box basis functions, as shown in figure4.1.

### 4.1.2 Polyline Approximation

This method performs the approximation of BAS function by a segmentation process which detects the dominant points, representing the significant convex/concave parts of the BAS function, obtained by an evolution process. In each iteration of the

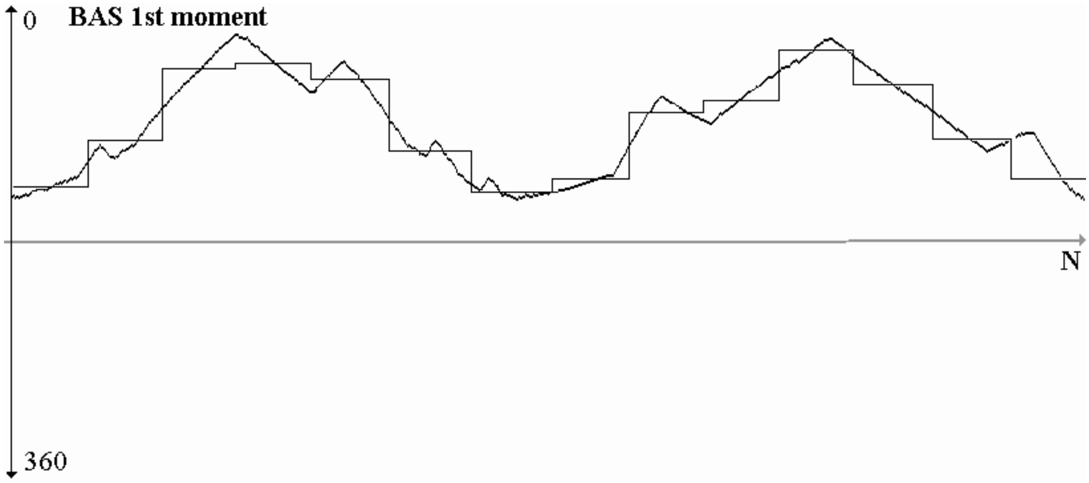


Figure 4.1: Piecewise Constant Approximation of BAS function.

evolution, the BAS function is simplified by removing the small convex parts on the function.

This approach is first introduced in [77] for polygon approximation of 2-D shape boundary. In [77], the shapes are simplified by a digital curve evolution process in order to reduce the influence of digitization noise as well as segmentation errors. Their idea is based on the fact that visual parts are somehow related to convexities of the shape and the significant convex parts can be determined by neglecting the small concavities. This is performed by a discrete curve evolution method in which a significant visual part becomes a convex part at some level of the evolution.

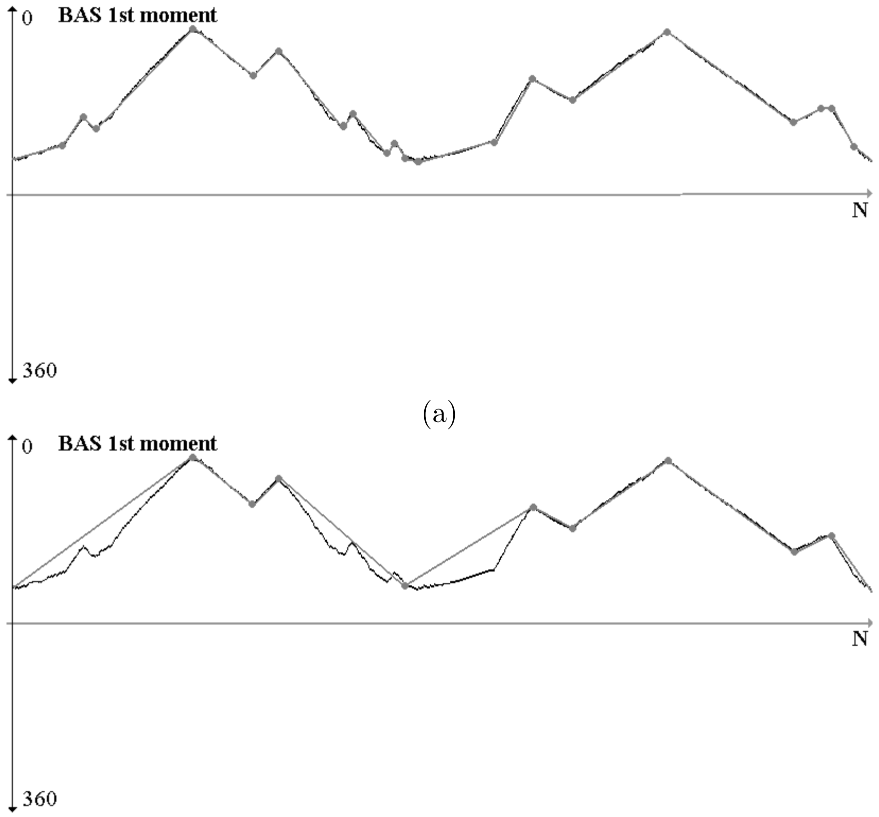
We adopt this idea for the approximation of 1-D BAS functions. Since the discrete BAS function can be considered as a polyline with connecting all points on the BAS function, the evolution of polylines is performed as follows:

- In each evolution step, a pair of consecutive line segments $s_1$, $s_2$ is replaced by a single line segment joining the end points of $s_1 \cup s_2$.

In this evolution, the crucial point is the order of substitution, which is done according to a relevance measure $K$ given by ;

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)}, \tag{4.2}$$

where $\beta(s_1, s_2)$ is the turn angle at the common vertex of segments $s_1$, $s_2$ and $l$ is the length function normalized with respect to the total length of the boundary.



(a)

(b)

Figure 4.2: Polyline Approximation : Two different stages of BAS function evolution, number of iterations (a) 235 (b) 245

In the above relevance measure, the higher the value of $K(s_1, s_2)$, the larger is the contribution of the shape of the curve of arc $s_1 \cup s_2$.

A few stages of BAS function evolution is illustrated in figure 4.2. It is shown that as the iteration of evolution increases, the small convex parts of the BAS function is eliminated. Note that the position of the end points of a convex parts, obtained in the course of the evolution, is exactly the same as the position of the end points on the original BAS function.

The evolution process is performed only on the first moment BAS function. The dominant points extracted on the first moment are, then, used for the rest of the moment functions. The algorithm of polyline evolution is given below.

```
POLYLINE EVOLUTION ALGORITHM

1. Initialization:
      $for 1 \leq i \leq N$
          $K(i) = Calculate\_Relevence\_Measure(i)$
          $Point\_list(i) = 1$
2. Induction:
      $for 1 \leq t \leq N \ - \ T$
          $m = arg\_min_i(K(i))$
          $(j, k) = Find\_prev\_Next\_Points(m)$
          $Point\_list(m) = 0$
          $K(j) = Calculate\_Relevence\_Measure(j)$
          $K(k) = Calculate\_relevence\_measure(k)$
3. Termination:
      $for 1 \leq t \leq T$
          $\overline{\Gamma}^m(t) = \Gamma^m(i) \quad if(Point\_list(i) == 1) \quad 1 \leq i \leq N$

function $Calculate\_Relevance\_Measure(i)$
    $(j, k) = Find\_Prev\_Next\_Points(i)$
    $Q_{backward} = arctan(\frac{\Gamma^1(i) - \Gamma^1(j)}{(i-j)})$
    $Q_{forward} = arctan(\frac{\Gamma^1(k) - \Gamma^1(i)}{(k-i)})$
    $\beta(i) = abs(Q_{forward} - Q_{backward})$
    $measure = (\frac{\beta(i) \ (k-i) \ (i-j)}{(k-j)})$
$return(measure)$
```

In the above algorithm, the lists *PointList* and $K$ keep the remaining points in the evolution and their relevance measures, respectively. The evolution starts with all the points of BAS function by initializing the lists *PointList* and $K$. In each iteration of evolution, first the point with minimum relevance measure is found and removed from the *PointList*. Then, the relevance measures of its neighbor points $j$ and $k$ are updated. The evolution procedure iterates $N-T$ times, where $T$ is the dimension of the feature space we wish to extract. Algorithm terminates by loading the values of all the moment functions in the corresponding points.

## 4.2 Fourier Based Features

Fourier Transformation, with its well-establish theory, provides a convenient tool for characterizing the signals. It has been widely used for shape description and multiscale representation.

In this thesis, the Fourier Transform is used for feature extraction of BAS functions. For a given $m^th$ moment BAS function $\Gamma^m(i)$, the Fourier transformation is calculated by

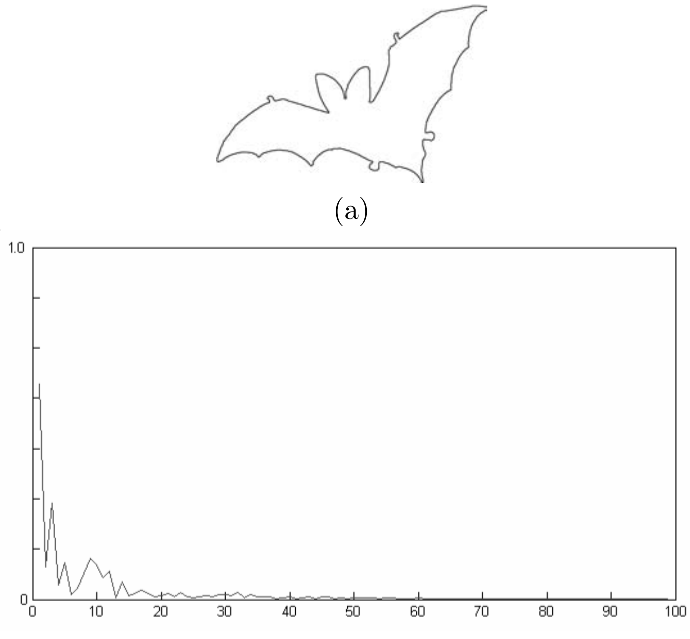
$$a_t^{(m)} = \frac{1}{N}\sum_{i=0}^{N-1} \Gamma^m(i) \quad exp(-j2\pi it/N). \tag{4.3}$$

The spectral coefficients $a_t^{(m)}$ is, then, used either directly as a feature vector or in sampling of BAS function by taking its inverse transform using the low frequency components. The main idea in this approach is the use of energy compaction along the low-frequency coefficients. It is well-known that most of the energy of signal is concentrated in the low-frequency coefficients. This fact is depicted in figure 4.3, indicating the power spectrum of BAS mean function, extracted from a sample image. The plot shows the rapid decay of the spectral coefficients which becomes zero after n=60.

Power spectrum of the BAS moment functions can serve for finding the optimal value for $T$. A preliminary analysis in the power spectrum of BAS moments functions for the shapes in the image database indicates that a certain value of $t = T$, the spectral coefficients become zero for all the BAS moment functions. This value, which is the Nyquist rate, is taken as the optimal dimension for the feature space.

### 4.2.1 Fourier Descriptors

Fourier Descriptors (FD) are commonly used techniques for characterizing the shape boundary. The basic idea consists of representing the shape in terms of a 1D or a 2D signal, followed by taking the Fourier transform of this signal and calculating the



(a)

(b)

Figure 4.3: Energy compaction in low-frequency coefficients, (a) A sample shape from MPEG-7 database, (b) the power spectrum of its BAS mean function.

FD's from the Fourier representation.

The main advantage of FD method is the ability to characterize a shape boundary with a small size descriptors. Besides, FDs are easy to implement and computationally efficient.

After calculating the spectral coefficients, $a_t^{(m)}$, of the $m^{th}$ moment BAS function $\Gamma^m(i)$, phase information is ignored and only the magnitudes $\mid a_t^{(m)} \mid$ are used in order to achieve invariance in starting point of boundary extraction, The coefficients are, also, normalized by dividing the magnitudes with the DC component,$\mid a_0^{(m)} \mid$. Then, the $T$ lowest frequency Fourier coefficients are used to construct the feature vector. $T$ is taken as a variable in the experiments and used to identify the Nyquist rate. Finally, the feature vector for a given BAS function is formed as follows:

$$F^{(m)} = \left[ \frac{\mid a_1^{(m)} \mid}{\mid a_0^{(m)} \mid}, \frac{\mid a_2^{(m)} \mid}{\mid a_0^{(m)} \mid}, \ldots \frac{\mid a_T^{(m)} \mid}{\mid a_0^{(m)} \mid} \right], \qquad m = 1, ..., M. \tag{4.4}$$

The shape boundary is, then, represented by concatenating $F^{(m)}$'s for each BAS functions, $\Gamma^1(i)$,..., $\Gamma^M(i)$. The similarity between two shapes is measured by Euclidean distance between features $[F^{(1)}, ..., F^{(M)}]$ extracted from the BAS moment functions.

### 4.2.2 Sampling by Fourier Transforms

As mentioned in section4.1.1, the easiest way for feature extraction from BAS functions is to perform sampling with equal distance. However, there is a trade off between the sampling rate and the accuracy of the representation. As the sampling rate decreases, the method loses information about the visual parts (convexities and concavities). In order to find an optimal sampling rate, Fourier analysis is utilized in this thesis.

The feature extraction is performed in two steps. In the first step, the Fourier Transform of the BAS functions $\Gamma^m(i)$, are calculated as shown in Equation 4.3. In the second step, Inverse Fourier transformation of the first $T$ coefficients is calculated.

$$\overline{\Gamma}^m(i) = \sum_{t=0}^{T-1} a_t^{(m)} \quad exp(j2\pi it/T). \tag{4.5}$$

Note that, the higher frequency coefficients are removed in this representation.

The major difference between the Fourier descriptors and sampling by inverse Fourier Transformation lies in the phase information. As we described in the previous section, Fourier descriptors ignores the phase information and uses only the magnitudes in the transform domain. This provides the independence of starting point. However, visual consistency is somehow lost. Sampling by inverse transformation, on the other hand, preserves almost all the visual information about the shape. This results in a consistent description of the BAS function.

## 4.3 Wavelet Based Features

The wavelet transform is widely used in many interesting image analysis applications. The discrete wavelet transform, in particular, is very popular due to its compact representation and efficient implementation [69].

Wavelet transform decomposes a function $f(x)$ into a family of functions, which are the translation and dilation of a unique function $\psi(x)$, called the mother wavelet, satisfying certain requirements [88]. The continuous wavelet transform of $f(x)$ is defined as;

$$\mathbf{CWT}f(s,u) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{s}}\psi(\frac{x-u}{s})f(x)dx, \tag{4.6}$$

where $s \in R^+$ and $u \in R$ are called scale and shift variables respectively. Essentially, the wavelet transform measures the similarity between $f(x)$ and scales and shifts of the mother wavelet function.

Dyadic Wavelet Transform is a special case of Continuous Wavelet Transform, where the scale parameter is discretized along the dyadic grid $2^j$, $j \in Z$, i.e.,

$$\mathbf{DWT}f(2^j, 2^j n) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2^j}} \psi(\frac{x - 2^j n}{2^j}) f(x) dx = \langle \psi_{j,n}(x), f(x) \rangle \tag{4.7}$$

where $\psi_{j,n}(x) = 2^{j/2}\psi(t/2^j - n)$. The function $\psi(x)$ is called wavelet basis function and has a companion, the scaling function $\phi(x)$. These functions satisfy the following relations;

$$\phi(x) = \sqrt{2} \sum_{u=0}^{L-1} h_u \phi(2x - u), \tag{4.8}$$

$$\psi(x) = \sqrt{2} \sum_{u=0}^{L-1} g_u \phi(2x - u), \tag{4.9}$$

where $h_u$ and $g_u$ are called low pass and high pass filter coefficients, respectively and

$$g_u = (-1)^u h_{L-u-1} \quad u = 0...L-1, \tag{4.10}$$

$$\int_{-\infty}^{\infty} \phi(x) dx = 1. \tag{4.11}$$

Wavelet transform can be implemented with two Finite Impulse Response (FIR) filters, called "Quadrature Mirror Filter" (QMF) pair. In other words, QMF pair, namely, a low pass filter $h$ and a high pass filter $g$, characterize the discrete dyadic wavelet transform.

Starting with $S_1 f = f$ $h_1 = h$ and $g_1 = g$, the recursive algorithm is defined as;

$$W_{2^{j+1}} f = g_{2^j} * S_{2^j} f, \tag{4.12}$$

$$S_{2^{j+1}} f = h_{2^j} * S_{2^j} f. \tag{4.13}$$

For any coarse scale $2^J$, the sequence of discrete functions;

$$\{S_{2^J} f, (W_{2^j} f)_{1 \le j \le J}\} \tag{4.14}$$

is called the discrete wavelet transform of $S_1 f = f$. Here, $S_{2^J} f$ is the last approximation coefficients and the set of sequence $(W_{2^j} f)_{1 \le j \le J}$ is the wavelet (detail) coefficients at levels $2^1 \le 2^j \le 2^J$.

### 4.3.1 Approximation by Wavelet Transforms

In many practical applications, wavelet decomposition provides satisfactory approximations of the functions, allowing multi-resolution analysis. Therefore, wavelet transforms can be applied on the BAS functions for feature extraction and dimensionality reduction.

After the BAS functions $\Gamma^m(i)$ are upsampled and interpolated in a way to have the same number of points for each shape, the dyadic discrete wavelet transformation is applied. The approximation coefficients $S_{2^J}\Gamma^m$ at a level $J$ is, then, used as the shape descriptor. The feature vector is constructed as;

$$\overline{\Gamma}^m(i) = S_{2^J}\Gamma^m. \tag{4.15}$$

Various family of wavelets are used in the experiments, including Haar, Daubechies, Symlets, Coiflets, B-splines biorthogonal and quadratic spline wavelets. The performance tests of the proposed wavelet families are given in Chapter 7.

### 4.3.2 Piecewise Linear Approximation by Wavelet Modulus Maxima

Another feature extraction method is the approximation of BAS function by the minima anf maxima points. The detection of maxima/minima points is performed by the Wavelet Transform Modulus Maxima (WTMM). The WTMM algorithm gives the local maxima in the wavelet transform of the corresponding signal. This approach is first introduced in [89] and applied for multiscale edge detection in images. Later in [88], it is proved that all singular points of a signal correspond necessarily to local maxima in the wavelet transform of that signal.

Mathematically speaking, WTMM can be defined as follows:

**Wavelet Transform Modulus Maxima :** Let, $Wf(s,u)$ be the wavelet transform of a function $f(x)$. We call modulus maxima, any point $(s_0, u_0)$ such that

$$\mid Wf(s_0,u) \mid \quad < \quad \mid Wf(s_0,u_0) \mid, \tag{4.16}$$

when $u$ belongs to either a right or the left neighborhood of $u_0$, and

$$\mid Wf(s_0,u) \mid \quad \leq \quad \mid Wf(s_0,u_0) \mid,$$

(4.17)

when $u$ belongs to the other side of the neighborhood of $u_0$.

In other words, a modulus maxima $(s_0, u_0)$ of the wavelet transform is a strict local maxima of the modulus either on the right or the left side of the $u_0$.

As an example, figure 4.4 (a) and (b) shows the plot of a discrete function of 256 samples and its discrete dyadic wavelet transform computed on nine scales. Figure 4.4 (c) gives the location and values of the local maxima of the wavelet transform modulus. At each scale $2^j$, each modulus maximum is represented by a Dirac that has the same location and whose amplitude is the value of $W_{2^j}f(x)$. The modulus maxima detection is an adaptive sampling that finds the sharp variation points of a function.

The WTMM is used for corner detection of a shape boundary by indexing the high curvature points. This task is achieved in [107] by determining the local maxima of the wavelet transform of the contour's orientation profile, where the high curvature points correspond to the positive or negative transitions of the function. This property of the orientation profile allows to use WTMM directly, in detecting the corners.

We adopt this idea for piecewise linear approximation of BAS function by detecting the maxima and minima points. For this purpose, a two step algorithm is proposed. In the first step, WTMM algorithm is applied to the BAS function. The result of this step is the detection of positive and negative transitions on the function. The second step searches for the maxima and minima points on the BAS function based on the result of WTMM.

The quadratic spline mother wavelet $\psi(x)$ is used in WTMM algorithm. A fast dyadic wavelet transform is applied by a filter bank algorithm proposed in [88]. The coefficients of the low-pass, $h[n]$ and high-pass, $g[n]$ filters, which generate the quadratic spline scaling function and wavelet, are given in table 4.1.

The WTMM algorithm is applied on the first moment BAS function. The result of this algorithm is a sequence of points, which take positive for positive and negative values, depending on the variations of the function and take zero values for the rest of the points. The consecutive positive and negative value in the sequence gives a clue for the existence of a maximum in the corresponding interval of BAS function. The maximum point is, then, easily found by taking the maximum value in this interval.

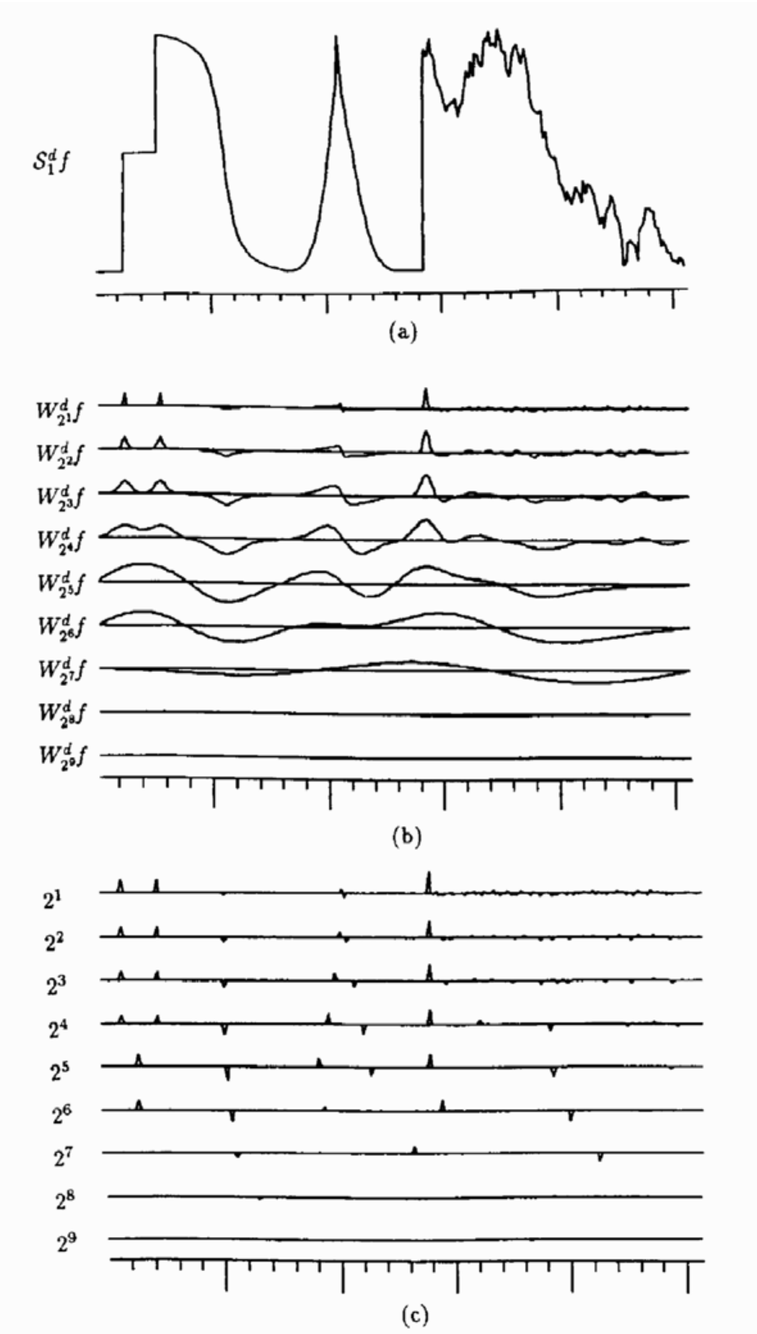


Figure 4.4: Wavelet Transform Modulus Maxima, (a) A function of 256 samples, (b) Discrete Wavelet Transform of signal computed on nine scales, (c) Modulus Maxima of the dyadic wavelet transform shown in (b), (reproduced from [88]).

Table 4.1: Coefficients of the filters for the quadratic spline wavelets.

| n | $h[n]/\sqrt{2}$ | $g[n]/\sqrt{2}$ |
|---|---|---|
| -1 | 0.125 | |
| 0 | 0.375 | -0.5 |
| 1 | 0.375 | 0.5 |
| 2 | 0.125 | |

Similar approach is applied for finding the minimum point by taking the minimum value in the interval of the points with negative and positive values. The procedure for detecting maxima and minima points on the BAS function can be summarized by the following algorithm.

```
ALGORITHM FOR DETECTION OF MAXIMUM/MIMIMUM POINTS
BY WTMM

$WTMM\_List = WTMM\_Algorithm(\Gamma^1);$    $i = 0;\ j = 0;$
$while(i < Length)$
    $if(WTMM[i].y > 0)$
        $start = WTMM[i].x;$
        $while(WTMM[i].y \leq 0)\ \ i++;$
        $end = WTMM[i].x;$
        $Max\_point = \arg\max_{start \leq k \leq end}(\Gamma^1(k))$
        $\overline{\Gamma}^m(j) = \Gamma^m(Max\_point);$    $j++;$
    $else\ \ if(WTMM[i].y < 0)$
        $start = WTMM[i].x;$
        $while(WTMM[i].y \geq 0)\ \ i++;$
        $end = WTMM[i].x;$
        $Min\_point = \arg\min_{start \leq k \leq end}(\Gamma^1(k))$
        $\overline{\Gamma}^m(j) = \Gamma^m(Min\_point);$    $j++;$
```

The above algorithm initially applies WTMM to the first moment BAS function and loads $WTMM\_List$. The iteration starts from the first point of the $WTMM\_List$ goes until a positive or negative value. If the positive value is detected, the beginning of the maximum value interval, *start* is signed. Iteration continuous until a negative

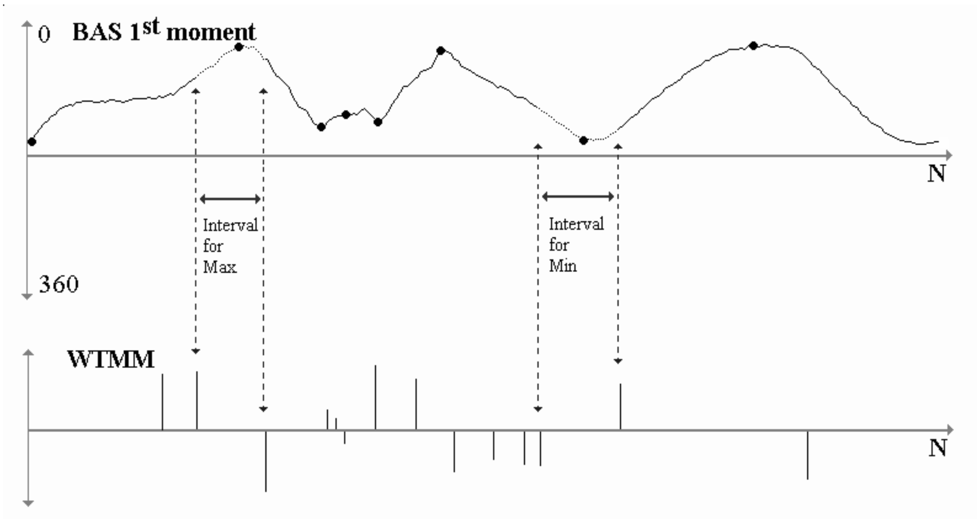


Figure 4.5: Maxima and Minima Detection.

value is found and this point is signed as the end of the interval, *end*. The maximum point in this interval is then found by the *max* operation. If the negative value is detected, the beginning and end point of the minimum value interval is found and *min* operation is performed. The illustration of the algorithm is given in figure 4.5.

## 4.4 Discussion about the Feature Extraction Methods

The goal of the feature extraction is to reduce the size of the BAS shape representation. This is mainly performed in two different approaches. The first approach samples the BAS function in the Nyquist rate. Piecewise constant approximation, Fourier based methods and wavelet approximation method can be considered in this category. The second approach detects the maxima and minima on the BAS function. Polyline approximation and piecewise linear approximation by WTMM use this approach.

Note that, the vector size $T$ is a critical parameter in all the proposed methods, which is determined by compromising the accuracy and complexity of the representation. Unlike the available methods in the literature, rather then empirically identifying the vector size of the feature space, we effectively identified the smallest possible $T$ for the highest possible performance using the Nyquist rate. The selection of $T$ is further discussed in chapter 7.

Note also that the proposed feature extraction methods can be applied on directly

2-D boundary information. However, utilization of the BAS function significantly improves the performance of the shape descriptor. Therefore, BAS function can be considered as a preprocessing step prior to feature extraction for preserving the information as much as possible in a compact 1-D representation.

# CHAPTER 5

# SIMILARITY MEASUREMENT

In Content Based Image Retrieval Systems (CBIR), given a query image, which is just a partial model of the user's inquiry, the system searches for all images in the database that are "similar" to it according to some predefined criterion. Usually, all images within a given distance from the query or the first few images that have the smallest distance are retrieved as a result of the query operation. The similarity measurement is performed by a distance function defined on the feature space of the images. The algorithms used in calculation of similarity, often depends on the desired properties of distance function and the nature of feature vector.

The objective in most of the similarity measurement methods is to minimize the distance between two vectors by allowing deformations. This is achieved by, first, solving the correspondence problem between the elements in feature vectors and, then, computing the distance as a sum of matching errors between corresponding items.

In chapter 2, we present the common distance measures for similarity of two vectors. Unfortunately, none of the mathematical distances are fully consistent with the Human Visual System. In particular, when the shape similarity is under consideration, measuring the similarity between two shapes with the available mathematical distances does not always give satisfactory results. The shapes, which are visually similar may not be close to each other in the vector space, defined over a particular distance. Recently, this problem brought the cognitive and computer scientists together and currently it is a hot research topic [116], [17], [134]. Additionally, the time and space complexity of the similarity measurement algorithms may create problems in the large size image databases, specially, when the dimension of the vector space gets large.

In this chapter, we describe the algorithms used in similarity measurement of BAS feature vectors. Since the feature extraction is performed by traversing the shape boundary sequentially, the resulting feature vector is considered as a sequence. Thus, sequence comparison methods can be applied as a distance function of BAS features. Two different sequence comparison methods are adopted for the proposed descriptors, extracted from the BAS function.

- *Optimal Correspondence of String Subsequences (OCS) Algorithm:* It is developed for the infinite alphabet string matching in [137]. This algorithm is used in similarity measurement of BAS features because of its metric property.

- *Dynamic Warping With Penalty (DWP):* DW is used for the sequences obtained by sampling of continuous functions. The algorithm is modified by adding penalty to the expansion and compression of the BAS elements in the matching process. This modification results in a substantial improvement in the similarity rates.

Since the shape boundary is a cyclic sequence, detection of a unique starting point is crucial in the matching process. Unfortunately, this is not an easy task. Therefore, instead of determining the starting point on the boundary, the sequences are shifted one item at a time and the matching values are recomputed over and over again. The recomputation increases the complexity of similarity measurement. In order to avoid this complexity, we propose an efficient Cyclic Sequence Comparison (CSC) algorithm, which approximates the optimal solution. The original algorithm proposed in [49], is developed for the partial matching of cyclic sequences. In this thesis, we improve the performance of the algorithm in calculating BAS similarity.

In this chapter, first a brief introduction to the sequence comparison methods is provided, in section 5.1. The point distance function is described in section 5.2. The OCS and DWP algorithms are explained in section 5.3 and 5.4, respectively. Section 5.5 discusses the methods for reducing the complexity of the OCS and DWP algorithms. After CSC algorithm is presented in section 5.6, a discussion about the similarity measurement algorithms is given in 5.7.

## 5.1 Sequence Comparison

Given two or more sequences, comparison of them is a process by which one attempts to measure the extent to which they differ. The problem is also called string matching (approximate or exact), when it is related to text processing. Sometimes there is a natural correspondence between items in one sequence and those in the other and the comparison is only based on the corresponding items. In such situations, some well known methods such as Minkowsky distances can be easily applied. However, in some situations the correspondence is not known in advance, because one or both of the sequences have suffered some kind of undesirable corruptions or simply because the sequences have different lengths. Then, the appropriate correspondence must be found by optimizing over all possible correspondences [114]. Efficient methods for performing the optimization to find the distance and the corresponding optimum analysis are the central topic of sequence comparison.

Sequence comparison methods have wide range of application areas, such as molecular biology, speech recognition, error and code control etc. Depending on the application, various algorithms have been proposed in the literature. The theory and practice of the sequence comparison algorithms for different applications can be found in [114].

In this section, we briefly introduce the main concepts and give the basic algorithms.

### 5.1.1 Sequence Matching

The distance between two sequences is calculated by some set of elementary operations. In most of the applications, the set of operations is restricted to :

- substitution (replacements) of one element for another at the same position,
- deletion of elements,
- insertion of elements.

The above operations are allowed in transforming one sequence into another and each operation has an associated cost. The distance between the sequences is defined as the minimum cost of total operations that is qualified by associating a cost function.

Note that there are, also, other operations used in various applications, such as compressions, expansions and transposition. Compression and expansion arises primarily for connecting the sequences obtained by time sampling from continuous functions of time as in the speech processing. Transposition, which means the interchange of two adjacent elements of a sequence is particularly relevant to typing errors.

Formally speaking, let us define $A$ and $B$ as two different sequences, $A_i$, $i = 1, .., N$ and $B_i$, $j = 1, .., M$ as the elements of $A$ and $B$, where $N$ and $M$ are the lengths of the sequences, respectively. The operations with their associated cost function is defined as,

- Substitution $A_i \rightarrow B_j$, with the cost of $w(A_i, B_j)$,
- Deletion of $A_i$, with the cost of $w(A_i, \phi)$,
- Insertion of $B_j$, with the cost of $w(\phi, B_j)$.

Note also that, substitution is expressed in terms of insertion and deletion or conversely insertion and deletions can be expressed in terms of substitution, where the substitution occurs with a NULL element. However, the distinction between the operations is important in sequence matching; because there may be a different cost associated with each of the operations.

Many distance functions have been proposed in the literature by giving restriction on the operations and the associated costs. **Levenshtein** (Edit) distance allows substitution, insertion and deletion. All the operations costs are assumed to be 1. In other words, the distance is defined as the minimal number of insertions, deletions and substitutions to make two sequences equal. **Hamming** distance allows only replacements with cost 1. **Episode** distance allows only insertions with cost 1. **Longest Common Subsequence** distance allows only insertion and deletion, all costing 1.

### 5.1.2 Basic Algorithm

The calculation of distance between two sequences attempts to minimize the total cost of operations that transform one sequence into another. The minimization problem can be solved by dynamic programming approach. For this purpose, a minimum distance table is constructed, which accumulates the partial distances between the sequences.

Mathematically speaking, let us define $D(i,j)$ as the minimum distance of the matching of $A_1...A_i$ against $B_1...B_j$. The initial value for the empty sequences $A_0$ and $B_0$ is $D(0,0)$ and the $D(N,M)$ is the total distance between the sequences. The recurrence starts from the obvious value $D(0,0) = 0$, which provides the entry for $(0,0)$ cell. The forward recursion finds the distances $D(i,j)$ for successively larger $i$ and $j$, finally reaching $D(N,M)$, which is the desired distance. The recurrence relation for cell $(i,j)$ is based on the values in three predecessor cells, $(i-1,j)$, $(i-1,j-1)$ and $(i,j-1)$. The calculation is performed by

$$D(i,j) = min \begin{cases} D(i-1,j) + w(A_i,\phi), \\ D(i-1,j-1) + w(A_i,B_j), \\ D(i,j-1) + w(\phi,B_j). \end{cases} \tag{5.1}$$

The intuitive meaning of the terms in the above recurrence can be understood as follows: The first term leaves $A_i$ unmatched and delete it, the second term substitutes $A_i$ with $B_j$ and the last term allows to skip $B_j$ without matching any character from $A$.

Computation of $D(N,M)$ necessitates iterating over $i$ and $j$. Therefore, it is easy to see that the complexity of the algorithm is $O(NM)$ in time and space.

## 5.2 Point Distance Function

Comparison of two BAS feature vectors is achieved via a pair wise comparison of the points in each vector. The absolute value distance (also called city block distance) is used for the point distance function. This distance is simply the average difference across dimensions.

Given two BAS feature vectors $\Gamma_q$ and $\Gamma_t$, namely query and test features, the absolute value distance between the points $i$ and $j$ from each vectors respectively, is defined as,

$$d(\Gamma_q(i),\Gamma_t(j)) = \sum_{k=1}^{M} \mid \Gamma_q^k(i) - \Gamma_t^k(j) \mid \qquad , \tag{5.2}$$

where $M$ is the number of BAS moments used in the representation of shape boundary.

Note that, it is possible to use other distance functions, such as Euclidean distance. However, evaluating the square root operation increases the computational cost, with no or marginal improvement on the similarity measurement process.

## 5.3 Optimal Correspondence of String Subsequences

One of the similarity measurement algorithm used for BAS feature vector is Optimal Correspondence of String Subsequences (OCS) proposed in [137]. The OCS problem is derived from the classical Longest Common Subsequences and approximate string matching problem, where the matching is defined as a notion of exact match or of compatible relation on a finite alphabet. In [137], the OCS extends the finite alphabet editing error minimization matching to the infinite alphabet penalty minimization matching. It is, also, proved that the string matching derived from OCS is a metric.

The OCS problem is defined by the following entities.

- *Alphabet :* The alphabet is an infinite set $\mathbb{N}$, which contains all characters, and lifted $\mathbb{N}$, denoted as $\mathbb{N}_\perp$, is $\mathbb{N} \cup \{\epsilon\}$, where $\epsilon$ is a null character and $\perp$ is a number or null character. In many applications, the reasonable choice of $\mathbb{N}$ is an interval of real numbers in $\mathbb{R}$.

- *Distance :* $d(x, y)$ is a distance function, if $d : \mathbb{N}_\perp \times \mathbb{N}_\perp \to \mathbb{R}$, $\forall x, y, z \in \mathbb{N}$ such that

  $d(x, y) \geq 0,$
  $d(x, x) = 0,$
  $d(x, y) = d(y, x),$
  $d(x, y) + d(y, z) \geq d(x, z).$
  In addition,
  $d(x, \epsilon) = d(\epsilon, x) = \tau$, where $\tau$ is a constant in $\mathbb{R}$

  Note that if $\tau$ is maximum (or upper bound) of $\mathbb{R}$, then $d(x, y)$ is a metric over $\mathbb{N}_\perp$. This can be proved by only verifying that

  $$d(x, y) + d(y, z) \geq d(x, z) \qquad for \qquad y = \epsilon. \tag{5.3}$$

  Since $\tau$ is the maximum, $d(x, \epsilon + d(\epsilon, z) = 2\tau \geq d(x, z)$.

- *Association Function :* $F$ is an association function of string $A = A_1, A_2 ... A_M$ and $B = B_1, B_2 ... B_N$, if and only if there is a mapping,

  $$F : \{1, 2, ..., M\} \to \{1, 2, ..., N, \perp\}, \tag{5.4}$$

such that

$$F(i) = \perp \tag{5.5}$$

or

$$(F(i) \neq \perp \wedge F(j) \neq \perp \wedge i < j) \Rightarrow (F(i) < F(j)). \tag{5.6}$$

The number of nontrivial correspondence of $A$ and $B$, denoted as $\# \mid F \mid$, is the number of $F(i)$, which

$$F(i) \neq \perp \qquad and \qquad 1 \leq i \leq M. \tag{5.7}$$

The OCS problem is, then, defined as follows: given strings $A$ and $B$ over alphabet $N$, find an association function $F$ of $A$ and $B$ such that the total distance

$$S_F(A, B) = (M + N - 2q)\tau + \sum_{\forall i F(i) \neq \perp} d(A_i, B_{F(i)}) \tag{5.8}$$

for $q = \# \mid F \mid$, is minimized.

Given two strings $A$ and $B$, a dynamic programming technique is used to build a minimum distance table, which accumulates the information of correspondence. The recurrence relation is defined as

$$D(i,j) = min \begin{cases} D(i-1, j-1) + d(A_i, B_j), \\ D(i-1, j) + \tau, \\ D(i, j-1) + \tau, \end{cases} \tag{5.9}$$

where $D(i, j)$ is the minimum distance of match of $A_1, ....A_i$ against $B_1, ...B_j$ and $d(A_i, B_j)$ is the distance of the difference between characters $A_i$ and $B_j$.

The distance between two BAS feature vectors $\Gamma_q$ and $\Gamma_t$, with lengths $N$ and $M$ respectively, is calculated by the following OCS algorithm.

```
                    OCS ALGORITHM

1. Initialization:    $D(0,0) = 0;$
     $for 1 \leq i \leq N \quad D(i,0) = i * \tau;$
     $for 1 \leq j \leq M \quad D(0,j) = j * \tau;$
2. Iteration :
     $for 1 \leq i \leq N$
         $for \leq j \leq M$
             $D(i,j) = \min\{D(i-1,j-1) + d(\Gamma_q(i), \Gamma_t(j));$
                             $D(i-1,j) + \tau, \quad D(i,j-1) + \tau\}$
3. Termination:
     $return(D(N,M))$
```

Note that, in the above algorithm, if $\tau$ increases, then the number of correspondences of OCS would increase. For $\tau \geq \mathbb{R}$, the number of correspondences is equal to $\min(N, M)$. Therefore, the value of $\tau$ effects the similarity results. In this study, the value for $\tau$ is taken as the upper bound in $\mathbb{R}$ in order to make the distance metric.

## 5.4 Dynamic Warping With Penalty

Another sequence comparison method used in this study is Dynamic Warping, which is applied to the sequences obtained by samples of BAS functions. The matching process needs to compensate the slight variations on the BAS functions appropriately by compressing the function at some places and expanding it at others. The classical DW algorithm achieves this goal by finding an optimal match between two sequences, which allows stretching and compression of the sequences.

In order to align two sequences $A$ and $B$ with lengths $N$ and $M$ using DW, we construct an $N$-by-$M$ matrix, where each element $(i, j)$ of the matrix contains the distance between the points $A_i$ and $B_j$. The goal is to find a path through the matrix, which minimizes the sum of the local distances of the points. Illustration of alignment between two sequences is shown in figure 5.1. The path from $(1, 1)$ to $(N, M)$ in the DW matrix is called warping path

$$W = w_1, w_2, ..., w_K \tag{5.10}$$

and it is subject to several constraints.

- *Boundary Conditions :* This requires the warping path to start at $w_1 = (1, 1)$ and finish at $w_K = (N, M)$.
- *Continuity :* Given $w_k = (a, b)$, this constraint requires

  $$w_{k-1} = (c, d), \tag{5.11}$$

  where

  $$\begin{aligned} a - c &\leq 1, \\ b - d &\leq 1, \end{aligned} \tag{5.12}$$

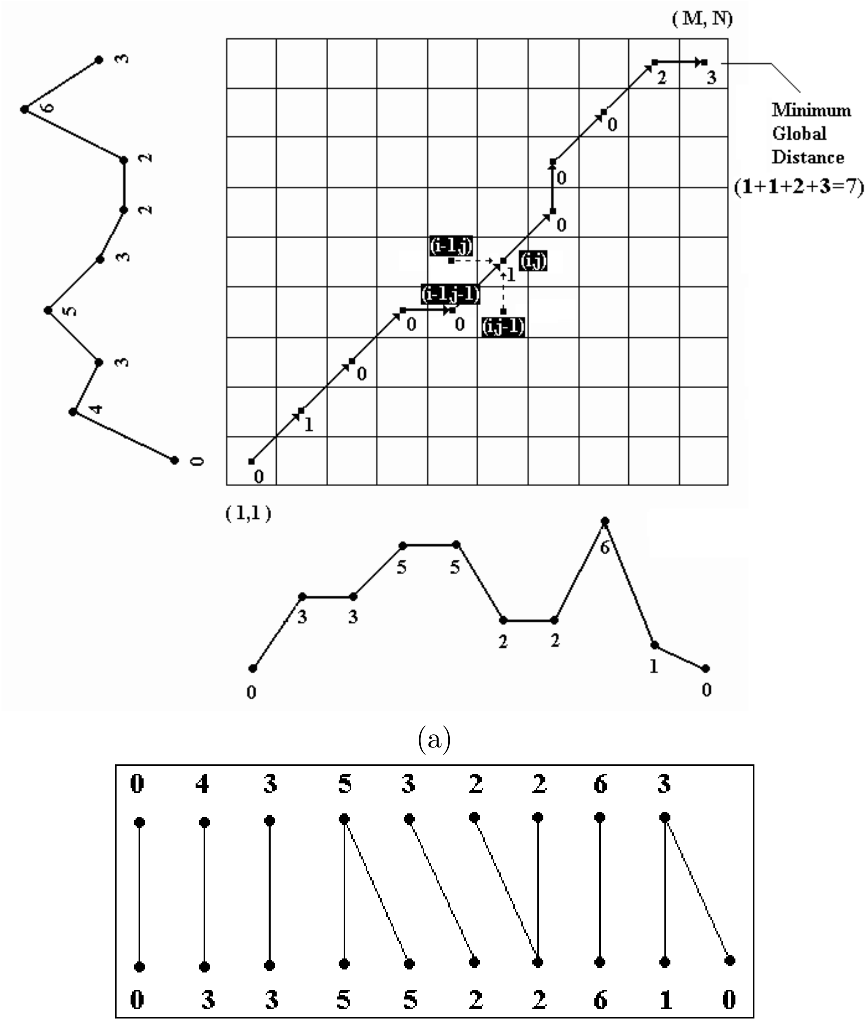
  restricting the allowable steps in the warping path.
- *Monotonicity :* Given $w_k = (a, b)$ and $w_{k-1} = (c, d)$, this constraint insures that

  $$\begin{aligned} a - c &\geq 0, \\ b - d &\geq 0. \end{aligned} \tag{5.13}$$

  The above inequalities forces the points in $W$ to be monotonically spaced in time.

The warping path on the DW matrix is found by dynamic programming algorithm, which accumulates the partial distances between the sequences. If $D(i, j)$ is the global distance up to $(i, j)$ and the local distance at $(i, j)$ is given by $d(i, j)$, then DW algorithm uses the following recurrence relation:

$$D(i, j) = d(A_i, B_j) + min \begin{cases} D(i-1, j-1), \\ D(i-1, j), \\ D(i, j-1). \end{cases} \tag{5.14}$$



(a)

(b)

Figure 5.1: Illustration of Dynamic Warping, (a) DW Matrix and the warping path (b) Correspondence of items.

Given $D(1, 1) = d(A_1, B_1)$ as the initial condition, we have the basis for an efficient recursive algorithm for computing $D(i, j)$. The algorithm starts from $D(1, 1)$ and iterates through the matrix by summing the partial distances until $D(N, M)$, which is the overall matching score of the sequences $A$ and $B$.

In figure 5.1.a, given two sequences

$$A = [0, 4, 3, 5, 3, 2, 2, 6, 3],$$

$$B = [0, 3, 3, 5, 5, 2, 2, 6, 1, 0],$$

the DW algorithm finds an optimal match between the entries of the sequences with an overall matching score. The optimal matching result is shown figure 5.1.b.

DW algorithm as described so far, while not permitting changes in the ordering of the sequence items, allows unconstrained compression and expansion of the items of the two sequences. This may be suitable for some applications, such as speech recognition, where the sequences of the same class may be traced out more slowly during one portion of the speech and more quickly during another portion. In order to allow such variations, it is necessary to distort the time axis appropriately. For this reason, the DW algorithms used for speech recognition does not need to penalize the expansion and compression of the sequences in the matching process.

However, in shape boundary, the expansion and compression of some part of the sequence change the visual appearance of the shape. These variations carry meaningful information used to distinguish visually different parts of the shape boundary. This leads to the consideration of constraining the warping path in such a way to limit the amount of compression and expansion to a certain extent in the matching process. In this study, we propose to assign penalties for expansion and compression of the BAS function. For this purpose, the horizontal and vertical moves in the DW matrix are penalized by a constant. This minor modification improves the performance of the similarity measurements in BAS descriptors.

The DWP algorithm for calculation of distance between two BAS vectors $\Gamma_q$ and $\Gamma_t$, is given below.

```
ALGORITHM FOR DWP

1. Initialization:   $D(1,1) = d(\Gamma_q(1), \Gamma_t(1));$
      $for 2 \leq i \leq N$   $D(i,1) = D(i-1,1) + d(\Gamma_q(i), \Gamma_t(1));$
      $for 2 \leq j \leq M$   $D(1,j) = D(1,j-1) + d(\Gamma_q(1), \Gamma_t(j));$

2. Iteration :
      $for 2 \leq i \leq N$
         $for 2 \leq j \leq M$
            $D(i,j) = d(\Gamma_q(i), \Gamma_t(j)) + \min\{D(i-1,j-1),$
                                                   $D(i-1,j) + penalty,$
                                                   $D(i,j-1) + penalty\};$

3. Termination:
      $return(D(N,M))$
```

Initially, the total distance of the cell $(1,1)$ is just its local distance $d(\Gamma_q(1), \Gamma_t(1))$. Then, the total distance for each successive cell in column 1 and raw 1 is the local distance for the cell, plus the global distance to the cell previous to it. The algorithm iterates by calculating the global distance of the rest of the cells in the matrix using the recurrence relation. At the cell $(i,j)$, the total distance is the local distance at $(i,j)$ plus the minimum total distance at either $(i-1,j)$, $(i-1,j-1)$ or $(i,j-1)$. The overall distance between the sequences is the value stored in the cell $(N,M)$.

Note that, the DWP algorithm differs from OCS algorithm in terms of recurrence relation. In OCS, the recurrence relation uses deletion and insertion operations in string matching and gives a constant cost $\tau$ without adding the local distances to the total distance. On the other hand, DWP uses expansion and compression operations and the cost of each operation is local distance. In addition, it is proved that the OCS algorithm is metric. However, the triangle inequality of metric axioms fails in DWP algorithm.

## 5.5 Reducing the Computational Time

The OCS and DWP algorithms find the simple alignment distance between the sequences $A$ and $B$ with lengths $N$ and $M$, in a computing time, whose dominant term is $3MNt$, where $t$ is the time required to carry out a simple basic task (point distance function, three addition and minimization).

Because the calculation time reduces to $3N^2t$ if $M = N$, which is quadratic in $N$, the algorithms are referred to as $time - quadratic$. In large image databases, the possibility of finding the minimum in calculation time that is merely quadratic in sequence length is considered as a major advance.

One way of decreasing the time complexity is to reduce the computation of the simple task, $t$. We perform this operation by using absolute value distance instead of Euclidean distance as we discussed in section 5.2.

Another way of decreasing the time complexity of the sequence comparison algorithms is to reduce the number of cells in the distance table that must be processed during the calculation. This is performed by eliminating the cells in the remote corners of the table. In other words, the corners that are remote from the beginning and the ending of the path are removed (see figure 5.2). In many cases, however, the excluded alignments are considered relatively implausible, so excluding them is not a great loss. In fact, the excluded alignments may be unrealistic that their exclusion is actively desired; because it makes more sense to define the distance as the minimum amount of realistically achievable change.

## 5.6 Cyclic Sequence Comparison

The BAS function, which is based on the shape boundary is considered as a cyclic sequence. In order to align two BAS feature vectors, the starting boundary point is to be matched. This requires to define a unique starting point for each shape, which is not practically possible. For this reason, the alignment computation must determine the amount of cyclic shift that has taken place in order to find the optimal match.

Mathematically speaking, let us denote a cyclic sequence $A$ and its shifted version by $A'$. The length of the sequence is $\mid A \mid$. Then

$$A' \equiv A \Leftrightarrow A' = \sigma^k(A), \qquad for 1 \leq k \leq N, \tag{5.15}$$

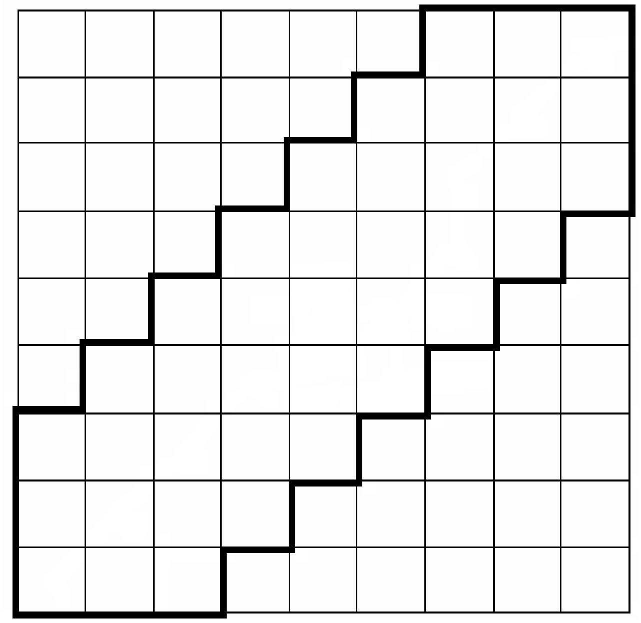


Figure 5.2: Cutting corners for reducing the computational complexity.

where $\sigma^k(A) = A_{k+1}A_{k+2}...A_{|A|}A_1...A_k$. Given two cyclic sequences $[A]$ and $[B]$, the cyclic distance $D_C$ between them is defined as

$$D_C([A],[B]) = \min\{D(\sigma^k(A), \sigma^l(B)) \ \mid 1 \leq k \leq\mid A \mid, \ 1 \leq l \leq\mid B \mid\}. \tag{5.16}$$

The easiest method of solving cyclic sequence comparison problem optimally is to shift any of the sequences one item at a time and recompute the alignment. This can be formulated as follows:

$$D_C([A],[B]) = \min_{1 \leq l \leq |B|}\{D(A, \sigma^l(B))\}. \tag{5.17}$$

The time complexity of the above computation is $O(\mid A \mid\mid B \mid^2)$. In this study, the OCS and DW algorithms are implemented by equation 5.17, in order to find an optimal solution to BAS feature similarity measurement.

There are also other studies in the literature for optimal solution to cyclic sequence comparison. A Divide and Conquer Method is presented in [87] to efficiently compute the optimal cyclic alignment with a computational complexity $O(\mid A \mid\mid B \mid log \mid B \mid)$ in the worst case. Another algorithm is introduced in [52], which uses a channeling technique to reduce the complexity of each alignment and a shift elimination technique to reduce the number of alignments carried out. The computation complexity of this algorithm is also $O(\mid A \mid\mid B \mid log \mid B \mid)$ in the worst case. A Branch and Bound strategy based on the approach used in [87] is proposed in [92], to explore only those promising branches that may lead to the optimal solution. The optimal solution is found by a guided search that discards candidate cyclic shifts as suboptimal on the basis of bounds on the corresponding alignment costs, which results in a data dependent computation complexity that varies between $O(\mid A \mid\mid B \mid)$ and $O(\mid A \mid\mid B \mid^2)$ [51].

Searching for strict optimality is not practical and efficient in image databases, which contain large number of shapes. Therefore, in practical problems, it is worth to find a suboptimal solution by approximate distance measures, rather than exact solution. The approximate techniques may serve as realistic alternatives to the optimal matching.

The approximate aproaches [49], [28], [97] double one of the sequence and then find the subsequence therein that best resembles the other sequence which computes in time $O(\mid A \mid\mid B \mid)$. A lower bound estimation of cyclic distance is computed in [28] by working on an edit graph. It is defined by a quadratic set of nodes of $(\mid A \mid +1)$ rows and $(2 \mid B \mid +1)$ columns and a set of arcs, where horizontal arcs correspond to insertions, diagonal arcs to substitution and vertical ones to deletions. This approach builds partial edit sequences between $A$ and $B^2$ (the concatenation of $B$ with itself) of uncontrolled lengths and takes the minimum weighted sequence as its approximate value. The extensions to this approach are reported in [97]. In the first extension, an upper bound estimation is computed by building complete edit sequences. The second extension combines the lower and upper estimations in the weighted solution.

In this study, we improve the classical cyclic sequence algorithm proposed in [49] to the cyclic BAS comparison. Our approach assigns some penalties in the recurrence relation. Note that, the classical cyclic sequence algorithm is developed for partial matching of sequences. However, our aim in BAS comparison is to find a matching between all the items of feature vectors. Assigning penalties achieves this task by controlling the length of the warping path in the algorithm.

To begin the formulation of cyclic sequence comparison, let us recall the notations. Given two sequences $A = A_1...A_N$ and $B = B_1...B_M$, a minimum distance table with $M$ columns and $2N$ rows is constructed by concatenating the sequence $A$. Paths start from the first $N$ entries of the first column and end at various points in the last column. Also, define $D(i,j)$ to be total distance on the minimum distance path from the $(i,j)$ entry of the distance table to the end of the path at some point in the last column. The value of $D(i,j)$ is evaluated as

$$D(i,j) = min \begin{cases} d(A_{i+1}, B_{j+1}) + D(i+1, j+1), \\ d(A_{i+1}, B_j) + D(i+1, j) + penalty \\ d(A_i, Bj+1) + D(i, j+1) + penalty, \end{cases} \tag{5.18}$$

for $i = 1, ..., 2N-1$ and $j = 1, ..., M-1$. The boundary conditions are

$$D(i,M) = 0, \quad i = 1, ..., 2N \text{ and}$$

$$D(2N,j) = d(2N, j+1) + D(2N, j+1), \quad j = 1, ..., M-1$$

Finally let

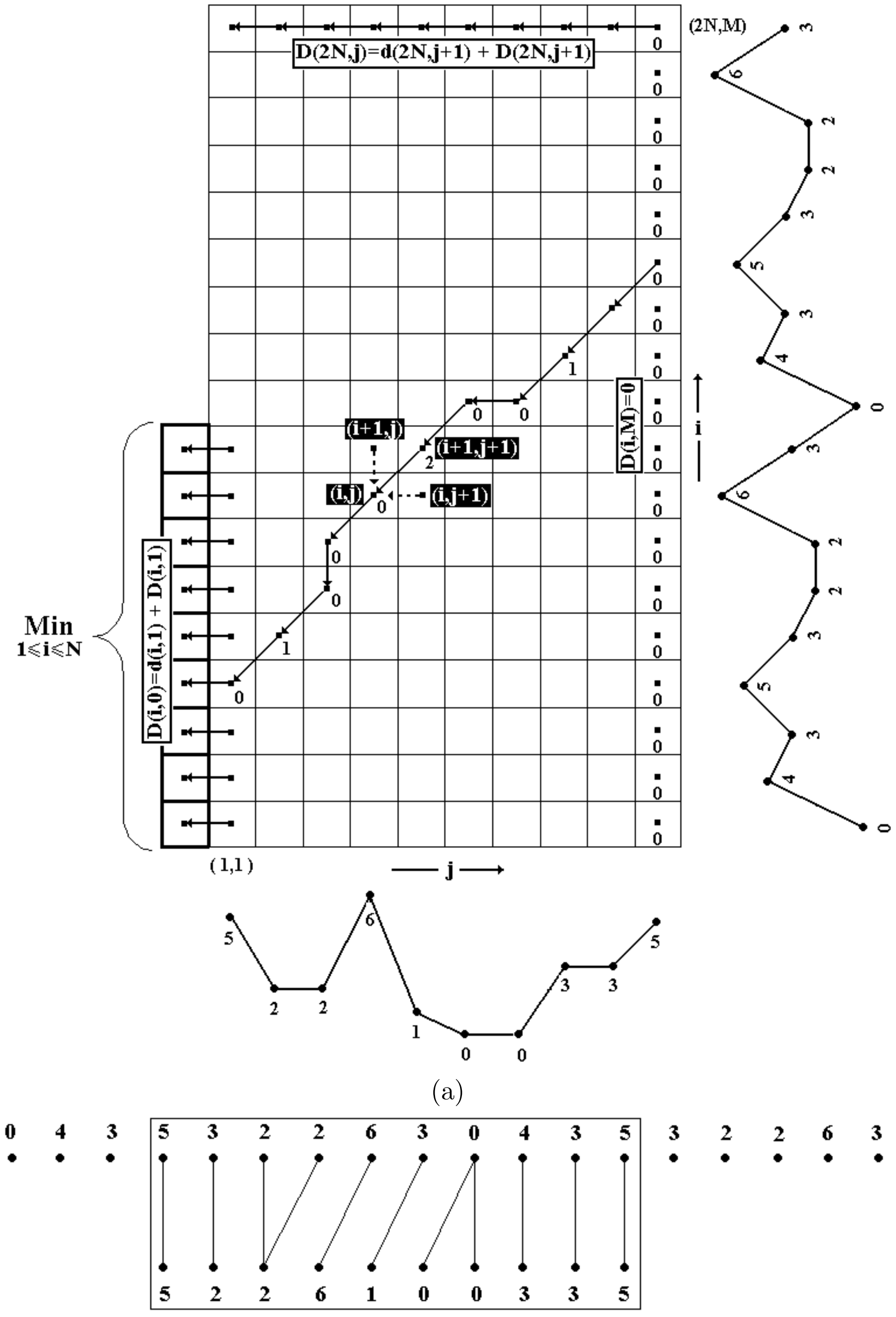
$$D(i,0) = d(i,1) + D(i,1).$$

The values of $D(i,0), \quad i = 1, ...N$ are the total distances of the paths through the minimum distance table from each starting point $i$ in $A$ running from the first point to the last point of $B$. The path with the lowest $D(i,0)$ is the minimum distance path in the table.

Illustration of the cyclic sequence comparison is shown in figure 5.3. The example uses the same sequences as in the figure 5.1. The sequence

A=[0,4,3,5,3,2,2,6,3] is concatenated with itself, resulting
$A^2 = [0,4,3,5,3,2,2,6,3,0,4,3,5,3,2,2,6,3]$ and
$B = [0,3,3,5,5,2,2,6,1,0]$ is shifted 4 items, resulting
$B = [5,2,2,6,1,0,0,3,3,5]$.

Now, we are ready to give CSC algorithm for the BAS features. Given two BAS feature vectors $\Gamma_q$ and $\Gamma_t$, the cyclic sequence comparison method can be summarized by the following algorithm:



(a)

(b)

Figure 5.3: Illustration of Cyclic Sequence Comparison, (a) The warping path (b) Correspondence of items.

```
CSC ALGORITHM FOR BAS FEATURES

1. Initialization:
    $for 1 \leq i \leq 2N-1 \quad D(i,M)=0;$
    $for 1 \leq j \leq M-1 \quad D(2N,j)=D(2N,j+1)+d(\Gamma_q(2N),\Gamma_t(j+1));$
2. Iteration :
    $for 2N-1 \geq i \geq 1$
        $for M \geq j \geq 1$
            $D(i,j)=\min\{D(i+1,j+1)+d(\Gamma_q(i+1),\Gamma_t(j+1)),$
                      $D(i+1,j)+d(\Gamma_q(i+1),\Gamma_t(j))+penalty,$
                      $D(i,j+1)+d(\Gamma_q(i),\Gamma_t(j+1))+penalty\};$
    $for 1 \leq i \leq N$
        $D(i,0)=d(\Gamma_q(i),\Gamma_t(1))+D(i,1);$
3. Termination:
    $return(\min_{1\leq i\leq N}\{D(i,0)\})$
```

Initially, the boundary conditions are calculated for $i = 1, ..., 2N-1$ and $j = 1, ..., M-1$. Then, the algorithm iterates by evaluating the recurrence relation. Finally, the minimum of $D(i,0)$, $i$ from 1 to $N$ is taken as the overall matching score. Note that, we change the recurrence relation by giving a constant penalty to the horizontal and vertical moves through the matrix. Although there is no theoretical reason for giving penalties, the discussion about this revision on the algorithm can be given by the following arguments:

Given two sequences $A$ and $B$, the algorithm given in [49], computes

$$D_C(A,B) = D(Z,B), \tag{5.19}$$

where $Z$ is a subsequence of $A^2$, which is most similar to $B$. In the computation, there is no control over the length of minimum distance path. Therefore, the length of the subsequence $Z$, $\mid Z \mid$ may tend to go far from $\mid A \mid$. This leads to the following consequences in the matching process.

- CSC algorithm calculates a partial matching of $A$ against $B$. As a matter of fact, it is developed for partial shape recognition. However, our aim is to measure the overall distance between the sequences. This requires to find a complete correspondence between items of the sequences.
- CSC algorithm approximates the optimal solution and estimates a lower bound of the exact cyclic distance [97]. This is because of taking the subsequence $Z$ of $A^2$, which is most similar to $B$, as an alignment result. This is depicted in our examples as well. In DW algorithm, the overall matching score is calculated as 7 for the particular sequences. On the other hand, cyclic sequence comparison algorithm calculates the distance as 4.

The proposed algorithm gives penalties for horizontal and vertical moves and controls the length of the path, $\mid Z \mid$. This heuristic enforces the path to go through the diagonal and approximate $\mid Z \mid$ to $\mid A \mid$. By this way, The complete correspondence between the items of two sequences is computed and the optimal solution is approximated by stimulating the matching score.

Another reason for using penalties lies within the same proposition as in the DW case. Since the expansion and compression of some part of the sequence change the visual appearance of the shape boundary, the warping path is to be constrained in such a way that the amount of compression and expansion is limited in the matching process. This is performed by giving penalties for expansion and compression of the sequences.

## 5.7 Discussion about the Similarity Measurement Algorithms

Although there is a concise mathematical base in measuring the similarity between vectors, the existing distances do not coincides with the Human Visual System. In this chapter, we propose some modifications in the available algorithms used for similarity measurement of BAS feature vectors, to partially compansate for the inconsistencies. Two different sequence comparison methods are adopted for the proposed BAS descriptors:

- Optimal Correspondence of String Subsequences Algorithm,

- Dynamic Time Warping with Penalty.

We also develope an efficient cyclic sequence comparison algorithm which approximates the optimal solution in a faster computational time

- Cyclic Sequence Comparison Algorithm

The objective in all the above mentioned methods is to minimize the distance between two vectors by allowing deformations. This is achieved by first solving the correspondence problem between items in feature vectors and then computing the distance as a sum of matching errors between corresponding items.

# CHAPTER 6

# CIRCULAR HIDDEN MARKOV MODEL (CHMM)

HMM is a powerful tool which has a wide range of applications in many fields of pattern recognition such as speech and character recognition. It is a stochastic finite state automaton for data that are observed in a sequential fashion (e.g., over time). There is a tremendous amount of variations of HMM applications, which input various feature sets into various HMM topologies. Available HMM's are either ergodic, which allows to return to previous states in finite amount of iterations, or temporal, which does not allow the revisits to the previous states, preserving the order of the sequential information.

This chapter introduces a new HMM topology, developed for boundary based shape recognition. The proposed topology, called circular HMM, is both temporal and ergodic. Therefore, the states can be revisited in finite time intervals, while keeping the sequential information in the string, which represents the shape. The proposed topology eliminates the need to define a starting point of a boundary in the recognition problem. The computational complexity of the circular HMM is relatively less then the classical topologies for the same recognition accuracy. Its structure, also, enables one to decide on the optimal state order by simple experiments on the training data and requires no size normalization.

In section 6.1, we present the necessary background for HMM's. The treatment here is greatly indebted to [108], [109] and [36]. In Section 6.2, first the circular HMM is defined, then, the algorithms and the implementation issues are explained. Finally, the discussion about the proposed topology is given in section 6.4.

## 6.1 Hidden Markov Models and Shape Recognition

Markov Model is a powerfull mathematical tool for modeling a stochastic process as a sequence of observable outputs produced by an unknown source. Modeling a process this way allows one to potentially learn about the source that creates the process without having to directly examine the source itself. Such a model enables one to recognize the patterns produced by the source or to predict the most probable future observations of the process given a partial sequence of observations.

### 6.1.1 Markovian Property

Markov Model makes the assumption that the process was produced by a Markov source, a type of source in which symbols, $O_n$, currently produced are dependent only on a fixed number of symbols that have been produced, preceeding the current output.

The order of the model specifies the number of preceeding outputs, which are taken into account for the next symbol to be produced. Since the complexity of the model grows exponentially with the order and the added benefit of increasing the order of the model decreases as the order grows higher, first or second-order Markov models are assumed to be sufficient for most of the applications.

When the shape modeling is under consideration, Markov Model assumes that each shape class is generated by the same stochastic source. Therefore, each shape in a database corresponds to an instance generated by a stochastic source. Markovian property implies that a point in the shape is dependent on its neighboring pixels, where the degree of Markov Model indicates the degree of the neighborhood systems.

### 6.1.2 Types of HMM

The observation $O_n$, corresponding to a single symbol representing an entry in shape vector, can be characterized as discrete or continuous. Discrete observations may be characters from a finite alphabet or quantized vectors from a codebook, while continuous observations are represented as samples from a continuous waveform.

Although the shapes are continuous in nature, usually, they are represented by a sequence of 2-D points in digital images. Therefore, it is practical to design discrete HMM for modeling the shape. For this reason, we only use discrete model in this thesis. However, it is worth to try continuous HMM and compare the performance of the systems for both discrete and continuous cases, in a future study.

A model may exploit some known properties of the system, or it may only rely on the statistical properties of it. If the known properties of the system are used to benefit the model, then it is deterministic, and all that need be done is to estimate the parameters of the model (such as if we know the system produce sine wave). A model that relies only on the system's statistical properties is known as a statistical or stochastic model. The Markov Model is a type of stochastic model.

### 6.1.3 Characteristics of HMM in Shape Analysis

HMM is a doubly stochastic variant of the Markov Model, with an underlying stochastic process that is not observable (hidden), but can only be observed through the sequence of outputs [109]. It contains states, state transitions, and transition probabilities. In shape analysis, states of HMM correspond to certain shape primitives, such as concavities and convexities. State transition probabilities of moving from state to state in a certain shape class, corresponds to the probability of a shape primitive being next to another primitive.

Initial state probabilities for each state define the chance of the model being found in that particular state at the beginning of an observation sequence. Since shape has a cyclic pattern, it is natural to assume equal initial state probabilities. Observation densities, which indicate the probability of a particular shape value being produced by the model, are defined for each state, and these make up the observable stochastic process. In a given state, the observation probability density function for that state defines the expectancy of a shape value in that state, while the sequence of states traveled define a movement from one expectancy of observations to another. It is unknown to an outside observer what state the source (or the model representing the source) is in at any given time. All that can be observed is the sequence of symbols representing the shape boundary, that make up the stochastic process. Therefore, this type of model is referred to as *hidden*.

An entry of the shape descriptor is produced by a Hidden Markov Model by first selecting a starting state. For simplification, this can be thought of as time $t = 1$, although it may be the case that previous entry produced may have helped determine what state the model is presently in. At this time, an entry is produced, which is the first observation of the output sequence. In deciding which entry will be produced, the model randomly selects an entry from a set of all possible entries that can be produced by this state, with probability determined by the observation density function for that state of the model. At the next time segment, time $t = 2$, the model moves to a next state by selecting a state transition from the set of all possible transitions for the present state with probability determined by the present state's transition probability density function. Having arrived in this new state, the new state selects an entry to be produced from its possible shape values, with probability determined by this new state's observation density function. The model continues in this fashion where each increment in time $t$ brings a new state transition followed by the emission of another symbol. The probability of the state sequence is equal to the product of the probability of the initial state and the probabilities of all transitions that were multiplied together, and the probabilities of all entries that were emitted in each of these states are known. Therefore, one can calculate the probability of the state sequence and the boundary representation being produced by multiplying these values together. This is not the case, however, in estimating the probability that a shape sequence was produced by a model for an unknown state sequence. When using HMMs for shape recognition produced by a source, where we have no information on how these shapes were produced by the source, it may be the case that many different state sequences could be a valid choice to produce this particular shape.

Therefore, shape recognition, based on HMM, deals with the notion that shape information can be directly stored in an HMM. If we consider a boundary of a shape represented by a BAS function, each peak or valley may correspond to a state in a Markov Model and by traversing the states, the state transitions can be recorded. Thus, one can obtain a state transition model from a shape boundary by using simple frequency counts on BAS function. However, the state transition matrix does not preserve the shape information. In other words, shape reconstruction is not possible from just the state transition model. All that can be done is to obtain a model from a boundary of a shape and compare it with the model of another shape.

In recognition of shapes by model discriminant HMM, a model is constructed for each shape class. The parameters of each shape model is estimated in the training phase using a set of class samples. After the training, each model corresponds to a cluster center for the feature vectors extracted from the shapes of that class. The goal in recognition is, then, decide on the model, which produces the unknown observation sequence, which is the shape feature vector. The probability of the observation sequence produced by each model is determined and the model, which gives the highest probability, is chosen as the result of recognition.

### 6.1.4 HMM Representation

As mentioned before, this thesis devoted to the discrete HMM, where the HMM $\lambda$, is described by three tuples,

$$\lambda = (\mathbf{A}, \mathbf{B}, \Pi), \tag{6.1}$$

where $\mathbf{A}$ and $\mathbf{B}$ are state transition and observation probability matrices, and $\Pi$ is the initial state vector.

Specifically, matrix $\mathbf{A}$ is defined as the set of all state transition probability distributions for the model,

$$a_{ij} = P\{q_j \text{ at } t+1 \mid q_i \text{ at } t\}, \tag{6.2}$$

where $q_i$ is the state the model is in at some time $t$, $q_j$ is a possible next state, and $a_{ij}$ is the probability that the model will make the transition to this next state at the next time increment.

Matrix $\mathbf{B}$ is defined as the set of all observation probability distribution functions for the model,

$$b_j(k) = P\{v_k \text{ at } t \mid q_j \text{ at } t\}, \tag{6.3}$$

where $q_j$ is the state the model is in at some time $t$, $v_k$ is an observation that may be seen by the model, and $b_j(k)$ is the probability of this observation occurring in the observation sequence while in state $q_j$.

Vector $\Pi$ is defined as the initial state distribution for the model,

$$\pi_i = P\{q_i \text{ at } t = 1\}, \tag{6.4}$$

where $q_i$ is a state in the model and $\pi_i$ is the probability that the model will begin in state $q_i$ at the beginning of an observation sequence.

Throughout this chapter, the following notation is used for mathematical representation of HMM:

$O = \{O_1, O_2, ..., O_T\}$, is the observation sequence, which corresponds to the quantized value of BAS feature vectors for a particular shape instance.

$T$ is the length of the shape representation.

$Q = \{q_1, q_2, ..., q_N\}$, is the set of states in the model.

$N$ is the number of states in the model.

$V = \{v_1, v_2, ..., v_M\}$, is the set of possible observations.

$M$ is the number of observation symbols.

### 6.1.5 Three Main Problems of HMM in Shape Analysis

There are three key problems that must be solved for HMM model to be used in shape analysis problems [108].

- Given the shape representation as an observation sequence $O = \{O_1, O_2, ..., O_T\}$ and the model $\lambda = (\mathbf{A}, \mathbf{B}, \Pi)$, how to compute $P(O \mid \lambda)$, the probability of the shape instance $O$, which belongs to a model $\lambda$.
- How to adjust the model parameters $\mathbf{A}, \mathbf{B}, \Pi$ to maximize $P(O \mid \lambda)$.
- Given the shape representation $O = \{O_1, O_2, ..., O_T\}$, how to choose a state sequence, $I = i_1, i_2, ...i_T$, which is optimal in some "meaningful" sense.

These problems are discussed in detail with their solutions in section 6.2 devoted to the circular HMM.

### 6.1.6 HMM Topologies

The number of states $N$ and the non-zero state transition probabilities $a_{ij} \neq 0$ defines the topology of HMM. The topology puts some restriction on the underlying stochastic process and results in distinct behaviors of the model. The vast amount of HMM topologies, used in various application domains, can be investigated in two categories:

1. Ergodic,
2. Temporal.

Ergodic topologies enable the revisits of each state with probability 1 in finite intervals, by allowing non-zero state transition paths between any two states (see figure 6.1). However, they do not impose a sequential or temporal order. Therefore, when the observation vector is temporal or an ordered sequence as in the case of shape boundary, ergodic models do not fully utilize the sequential information of the data.

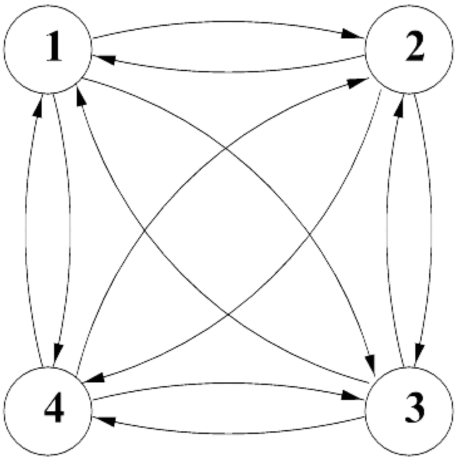


Figure 6.1: Ergodic (fully connected) HMM.

On the other hand, the temporal topologies do not allow the revisits to the previous states by constraining the state transition probabilities, $a_{ij} = 0$ for $j \geq i + k$, where $k$ is a small integer compared to the total number of the states $N$. This constraint yields a sparse state transition matrix, where the nonzero entries lie only in the few upper diagonal matrix. Therefore, when there is an ordered observation sequence such as shape boundary, it is better to use the temporal models (Bakis model) (see figure 6.2).

In temporal models, there is an initial and final states. Therefore, the underlying state sequence of this type of model is forced to start in the initial state (i.e., the leftmost state) and can only make transitions to higher states (i.e. to the right) or to the present state as time passes. In addition, the model must end in the final state (the rightmost state). This allows the sequence of states to represent the passage of time and has become the most popular form of Markov model used in various shape recognition schemes such as, handwriting recognition.

Variations of temporal topology are available in the literature. The simplest form of left-right model (also called linear model) is shown in figure 6.3. This model only allows jumps of no more than one state.

The parallel left-right models, on the other hand, allow multiple paths through the model, with each path skipping one or more model states (see figure 6.4).

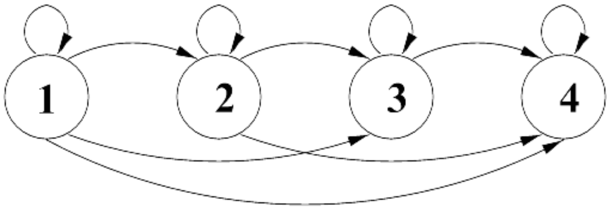


Figure 6.2: A left-right HMM.

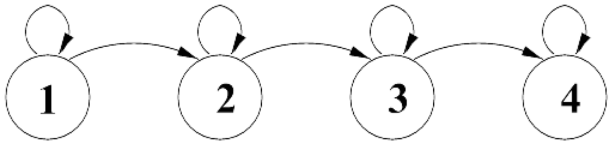


Figure 6.3: Simplest form of left-right HMM.

Another form of state transition that has been found useful is known as the Null transition. The idea of the null transition is that a state transition should be allowed to occur without requiring an observation to be absorbed. Null transitions are special kind of state transitions, with transition probabilities, that can occur without requiring the time counter, $t$, to be updated. The present observation in the sequence, $O_t$, remains the present observation in the next state.

### 6.1.7 Choice of the Topology

Unfortunately, there is no simple, theoretically consistent way of designing the topology of a model, the choice of number of states and the choice of nonzero state transition probabilities. Usually, the parameters of the topology is selected by a trial and error, throughout the training and validation stages.

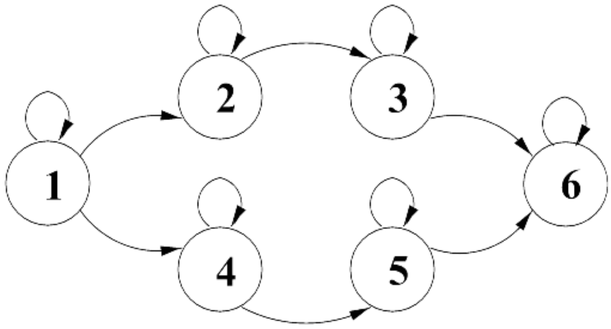


Figure 6.4: A parallel left-right HMM.

## 6.2 Circular Hidden markov Model : CHMM

As we discussed in the previous section, various HMMs with different topologies are proposed in the literature. However, the available approaches presume a topology, depending on the application domain. All of the practiced topologies have some pros and cons.

Experimental results of many studies indicate that left-right topologies are more appropriate to reach the maximum recognition rates in time series applications, such as speech and motion recognition. However, when the feature set consists of the quantized values of a closed boundary, it is impractical to identify consistent starting and ending points on a boundary of the object to represent the observation sequence. Therefore, in shape recognition problem, based on the object boundaries, the available HMM's yield relatively low recognition rates, if the feature sets do not have a geometrically meaningful starting and terminating points. However, the periodic nature of the boundary requires a topology, which is both temporal (has a sequential order) and ergodic to allow the revisits of a state, as the boundary returns to the starting point and repeats itself.

The motivation behind the circular HMM is to find a suitable topology for modeling the shape, based on the boundary. Since the shape boundary is cyclic in nature, the convex/concave parts of the shape are in a cyclic order and it can be thought as if they lie on a circle (see figure 6.5). If each portion of the shape is modeled by an HMM state, then a suitable HMM topology should incorporate the circular pattern of the shape primitives.

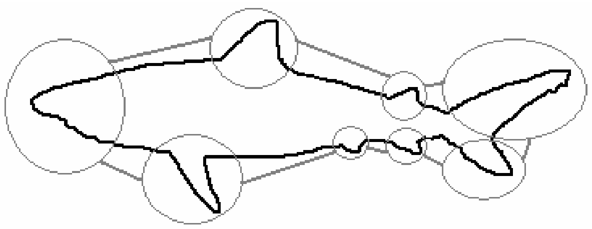


Figure 6.5: Cyclic order of convex/concave parts in shape boundary

The circular HMM topology is constructed by modifying the left-to-right HMM model, where the initial and terminal states are connected through the state transition probabilities. This connection eliminates the need to define a starting point of a closed boundary, in the description problem. The circular HMM is both temporal and ergodic. Therefore, the states can be revisited in finite time intervals while keeping the sequential information in the string, which represents the shape. This structure enables one to decide on the optimal state order by simple experiments on the training data and requires no size normalization for shape boundary.

### 6.2.1 Model Representation of CHMM

Suppose that a shape can be characterized by its discrete set of boundary points drawn from a finite alphabet or from quantized vectors of a code-book. Suppose, also, that the boundary string is the observable output of a parametric random process. For completeness, Let, $O = (O_t, O_{t+1}, O_{t+2}, ..., O_{t+T-1})$ represents the closed boundary of length $T$, over an alphabet $V = \{v_1, ...v_k, ...v_M\}$, with $\forall t, O_t = O_{t+T}$.

Our goal is to define a discrete density Hidden Markov Model, which represents the boundaries of each shape class. The circular HMM for each shape class $l = 1, .., c$, is represented by a three tupple $\lambda_l = (A_l, B_l, \Pi_l)$. The state transition probability matrix, $A_l = [a_{ij}]$, the observation probability sequence of observing the code $k$ in $i^{th}$ state for $1 \leq i, j \leq S$, $B_l = \{b_i(k)\}$ and the initial state distribution vector $\Pi_l$ satisfies the following conditions:

1. $j = j + r$, $n = 0, 1, ...R$,

2. $a_{ij} = a_{i+N,j+N}$,

3. $b_i(k) = b_{i+N}(k)$,

4. $R \ll N$ and

5. $\pi_i = 1/N$,

where $N$ represents the number of states and $R$ represents the maximum number of difference between $i$ and $j$ (see figure 6.6).

Notice that the state transition probability matrix, where each entry, $a_{ij}$, represents the probability of moving from state $i$ to $j$ is very sparse ($R \ll N$) as in the left-right HMM. For example for $R = 2$, the State transition matrix has the following form:

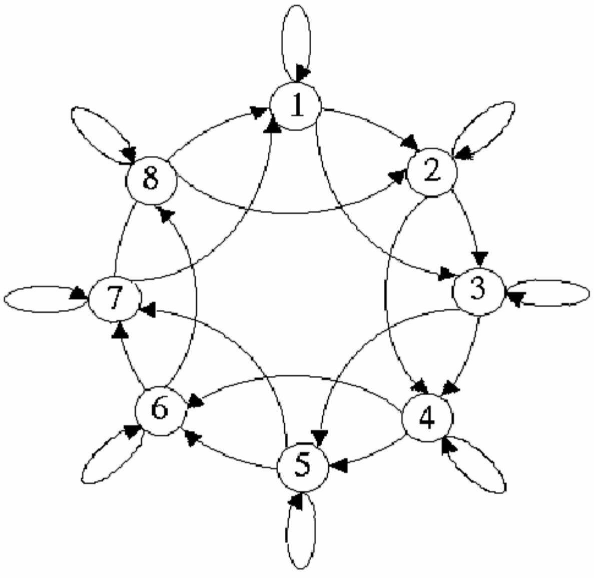


Figure 6.6: Circular HMM with 8 states, for R=2.

$$A_l = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & 0 & 0 & ... & 0 & 0 \\ 0 & a_{2,2} & a_{2,3} & a_{2,4} & 0 & ... & 0 & 0 \\ 0 & 0 & a_{3,3} & a_{3,4} & a_{3,5} & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... \\ a_{N-1,1} & 0 & 0 & 0 & 0 & 0 & a_{N-1,N-1} & a_{N-1,N} \\ a_{N,1} & a_{N,2} & 0 & 0 & 0 & 0 & ... & a_{N,N} \end{pmatrix}$$

Note that, $A_l$ is a sparse matrix with nonzero state transition probabilities lying on the main diagonal, two upper diagonals and last two lower diagonals. The only nonzero entries additional to the left-to-right model are the lower left corner of the state transition matrix. With this little extra computational cost, the temporal left-to-right model becomes an ergodic one.

Note also that, the initial state probabilities are equal for all the states. Since the starting point of boundary feature differs in observation sequences, the first symbol of the observation can be produced from any state of the model.

In the following sections, we explain the algorithms for model evaluation, parameter estimation and optimal state sequence problems of CHMM. Each section first gives the algorithm for a typical HMM and, then, discusses the modifications, needed for the CHMM.

### 6.2.2 CHMM Evaluation

The goal of classification is to decide which source this observed sequence was produced by. The BAS feature vector is presented to each model as an observation sequence and its probability of being produced by that model is evaluated. The model that has the greatest likelihood of producing a particular BAS vector then defines the chosen identity of the source which is believed to have produced it. In order to evaluate the models, we must first calculate $P(O \mid \lambda)$, the likelihood that a sequence of observations, $O$, were produced by a given model, $\lambda$. For any observation sequence produced by a model and a known sequence of states followed to produce this sequence of observations, the probability of the model producing these observations and following this state sequence can be calculated as,

$$P(O, s \mid \lambda) = \pi_{s_1} b_{s_1}(O_1) \prod_{i=2}^{T} a_{s_{i-1} s_i} b_{s_i}(O_i) \quad . \tag{6.5}$$

In most practical situations, however, the state sequence is unknown to the observer. Since there may be many state sequence paths that can produce the underlying observation sequence, the probability that $O$ was produced by model $\lambda$ is the sum of all the probabilities $P(O, s \mid \lambda)$ for all possible state sequence paths, $s$. Computing the probabilities for each of these paths would be computationally prohibitive, requiring on the order of $2TN^T$ calculations. Fortunately, an efficient algorithm using a set of variables called forward probabilities, calculates this sum of probabilities in the order of $N^2T$ calculations. This is known as the Forward-Backward algorithm [108].

Define $\alpha_t(i)$, the forward probability, as the probability of the partial observation sequence from time 1 to time $t$ occurring and the model being in state $q_i$ at time $t$. Then forward probability is defined as,

$$\alpha_t(i) = P(O_1 \; O_2 \; ... \; O_t, \; q_t = s_i \mid \lambda) \quad . \tag{6.6}$$

This can be solved inductively by the following algorithm,

*ALGORITHM FOR FORWARD PROBABILITIES*

1. *Initialization:*

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N$$

2. *Induction:*

For all $t = 1, 2, ..., T-1$, calculate

$$\alpha_{t+1}(j) = (\sum_{i=1}^{N} \alpha_t(i) a_{ij}) b_j(O_{t+1}), \quad 1 \leq j \leq N \tag{6.7}$$

3. *Termination:*

$$P(O \mid \lambda) \ = \sum_{i=1}^{N} \alpha_T(i) \tag{6.8}$$

In a similar fashion, the backward probabilities, $\beta_t(i)$, can be defined as the probability of the partial observation sequence from time $t+1$ to time $T$ and the model being in state $q_i$ at time $t$.

*ALGORITHM FOR BACKWARD PROBABILITIES*

1. *Initialization:*

$$\beta_T(i) = 1, \quad 1 \leq i \leq N$$

2. *Induction:*

For all $t = T-1, T-2, ..., 1$, calculate

$$\beta_t(i) = \sum_{i=1}^{N} a_{ij} b_j(O_{t+1}(j)), \quad 1 \leq j \leq N \tag{6.9}$$

The underlying principle of the forward- backward algorithm is the following: If the probabilities were known for each state $q_i$, of the partial sequence of observations up to time $t$, leaving the model in state $q_i$, then, the probability of the model being in some state $q_j$ at time $t+1$ could be calculated using only those probabilities and those for times before $t$ are not needed. Therefore, the forward probabilities act as accumulators for each state as $t$ is increased from 1 to $T$, and the value at time $T$ is the probability of being in that state at the end of the observation sequence.

The forward and backward probabilities have been found to be useful for other computations involving state path traversals. This will be shown in algorithms used to estimate model parameters presented in the next section.

For CHMM, no modification in forward-backward algorithm is needed. The probability that an observation sequence is produced by a given model, $P(O \mid \lambda)$ is calculated using forward-backward algorithm. The computation of forward-backward probabilities in CHMM can be performed more efficiently than the ergodic models. Since the state transitions are restricted as in the temporal models, $\alpha_{t+1}$s can be calculated using lower number of $a_{ij}$s. While the ergodic models use all the connections in the trellis structure, the CHMM use only the available state transitions in calculations.

With the help of the forward and backward probabilities, the probability that an observation sequence was produced by a given model, can be computed. In order to be able to choose a competing model over another, one must be able to compare the probabilities that the observed sequence is produced by the specified models. This can be calculated by the Bayes Rule.

### 6.2.3 Estimating CHMM Parameters

In creating a model of a source, the details of the inner structure are mostly unknown. One needs to be able to define the characteristics of the model, based on previous observation sequences or *training samples.* Therefore, a method of estimating the model parameters using these samples, is needed. The most common method of estimating the model parameters is to set them so as to maximize the probability of the training observation sequences being produced by the model [108]. This is a maximum likelihood method of training. Since an analytical solution to estimating the parameters is intractable, the parameters are computed iteratively or by method of gradient descent. A very common iterative re-estimation method is the Baum-Welch algorithm [18]:

First, let us define the probability of the model $\lambda$ being in state $q_i$ at time $t$, when presented with the sequence of observations $O$ as

$$\gamma_t(i) = P(q_i \text{at } t \mid O, \lambda)$$

and the probability of the model $\lambda$ being in state $q_i$ at time $t$ and making a transition to state $q_j$ at time $t+1$, when presented with the sequence of observations $O$ as

$$\xi_t(i,j) = P(q_i\text{at } t, q_j\text{at } t+1 \mid O, \lambda) \quad .$$

The above probabilities can be calculated using forward and backward probabilities by the Baum-Welch algorithm as follows:

*BAUM-WELCH ALGORITHM*

1. *Calculate* $\gamma_t(i)$ and $\xi_t(i,j)$;

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)} \quad 1 \leq i \leq N, \quad 1 \leq t \leq T, \tag{6.10}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(i)}{P(O \mid \lambda)} \quad 1 \leq i,j \leq N, \quad 1 \leq t \leq T. \tag{6.11}$$

2. *Update the model parameters;*

$$\bar{\pi}_i = \gamma_1(i), \tag{6.12}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{6.13}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1_{O_t=k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \tag{6.14}$$

Each iteration of the Baum-Welch algorithm has been proven to increase $P(O \mid \lambda)$ for that instance of $O$, until a local maximum has been reached [18]. Note that, the initial state probabilities, $\pi_i$ are the same for all states in CHMM. For this reason, $\pi_i$s are not updated in reestimation procedure of the Baum-Welch algorithm. In addition, the number of nonzero state transitions, $a_{ij}$s is lower than the ergodic models, resulting in a faster training stage.

### 6.2.4 Picking an Optimal State Sequence

Given a model and a sequence of observations, one problem, that needs to be addressed is the selection of the optimal sequence of states traversed to create this observation sequence. It is likely that many possible state sequences could produce a given observation sequences. One way of selecting a state sequence is to choose those states that have the highest individual probability of producing that observation. The problem with this method is that, having not taken into account the actual state connectivity, an impossible state sequence might be selected. A method of selecting the best path over state transitions is required. A popular dynamic programming algorithm known as the *Viterbi Algorithm*, is used for this purpose [136], [43]. The algorithm obtains a solution recursively as follows:

*VITERBI ALGORITHM*

1. *Initialization:*

$$\delta_1(i) = \pi_i b_i(O_1), 1 \leq i \leq N \tag{6.15}$$

$$\Psi_1(i) = 0 \tag{6.16}$$

2. *Recursion:*

$$\delta_t(j) = \max_{1 \leq i \leq N}(\delta_{t-1}(j)a_{ij})b_j(O_t),\ 2 \leq t \leq T,\ 1 \leq j \leq N \tag{6.17}$$

$$\Psi_t(j) = \arg\max_{1 \leq i \leq N}(\delta_{t-1}(j)a_{ij}),\ 2 \leq t \leq T,\ 1 \leq j \leq N \tag{6.18}$$

3. *Termination:*

$$P^* = \max_{1 \leq i \leq N}(\delta_T(i)) \tag{6.19}$$

$$i_T^* = \arg\max_{1 \leq i \leq N}(\delta_T(i)) \tag{6.20}$$

4. Path Backtracking:

$$i_t^* = \Psi_{t+1}(i_{t+1}^*),\ t = T-1, T-2, ..., 1 \tag{6.21}$$

The Viterbi algorithm is very similar to the Forward-Backward algorithm. The Viterbi Algorithm contains an additional variable to the Forward variable $\alpha_t(i)$, denoted as $\delta_t(i)$. While $\alpha_t(i)$ represents the sum of probabilities of all possible paths into a state $q_i$ at time $t$, $\delta_t(i)$ is the maximum of these different path probabilities. $\delta_T(i)$ is, then, the probability that the maximum probability path ends in state $i$, and $P^*$ is the total maximum path probability of all paths ending in any of the N states. Similarly, $i_t^*$ is the final state that the sequence will end in with probability $P^*$. Therefore, by starting at state $i_T^*$ and backtracking using $\Psi_t(i)$ to reach state $i_{t-1}^*$, from any state in the path $i_t^*$, one can completely reconstruct this path.

## 6.3 Implementation Issues for CHMM

In this section, we deal with several practical issues including scaling, multiple observation sequences, initial parameter estimatation and missing data problem.

### 6.3.1 Scaling

The goal of scaling is to keep all the parameters of the CHMM algorithms within the dynamic range of computers. In order to understand why scaling is required for implementing the re-estimation procedure of HMMs, consider the definition of $\alpha_t(i)$ in equation 6.6. It can be seen that $\alpha_t(i)$ consists of the sum of a large number of terms, each of the form

$$\left( \prod_{s=1}^{t-1} a_{q_s q_{s+1}} \prod_{s=1}^{t} b_{q_s},(O_s) \right)$$

with $q_t = S_i$. Since each $a$ and $b$ term is significantly less than 1, it can be seen that as $t$ gets big, each term of $\alpha_t(i)$ heads exponentially to zero. For sufficiently large $t$ ($t > 100$), the dynamic range of the $\alpha_t(i)$ computation will exceed the precision range of essentially any machine. Hence, the only reasonable way of performing the computation is by incorporating a scaling procedure.

The basic scaling procedure is to multiply $\alpha_t(i)$ by a scaling coefficient that is independent of $i$ (i.e., it depends only on $t$), for $1 \leq t \leq T$. A similar scaling is done to the $\beta_t(i)$ coefficients and then, at the end of the computation, the scaling coefficients are canceled out exactly.

### 6.3.2 Multiple Observation Sequences

The major problem with temporal models is that one can not use a single observation sequence to train the model. This is because the transient nature of the states within the model only allow a small number of observations for any state (until a transient is made to a successor state). The same problem arises in CHMM. Hence, in order to have sufficient data to make reliable estimates of all model parameters, one has to use multiple observation sequences.

The modification of the re-estimation procedure is straightforward and goes as follows. We denote the set of $K$ observation sequences as

$$O = (O^{(1)}, O^{(2)}, ..., O^{(k)}) \quad , \tag{6.22}$$

where $O^{(k)} = (O_1^{(k)}, O_2^{(k)}, ..., O_{T_k}^{(k)})$ is the $kth$ observation sequence. We assume that observation sequences are statistically independent. Our goal is to adjust the parameters of the model $\lambda$ to maximize

$$P(O \mid \lambda) \;=\; \prod_{k=1}^{K} P(O^{(k)} \mid \lambda) \;=\; \prod_{k=1}^{K} P_k \quad . \tag{6.23}$$

Since the re-estimation formulas are based on frequencies of occurrence of various events, the re-estimation formulas for multiple observation sequences are modified by adding together the individual frequencies of occurence for each sequence. Thus, the modified reestimation formulas for $\bar{a_{ij}}$ and $\overline{b_j(l)}$ are

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^k) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}, \tag{6.24}$$

$$\bar{b}_i(l) = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1_{O_t=v_t}}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)} \tag{6.25}$$

and $\pi_i$ is not reestimated since $\pi_i = 1/N$.

### 6.3.3 Lack of Training Data

A major problem, encountered during parameter re-estimation, due to a finite size of the training set, occurs when a model parameter is set to zero. For instance, after training the model, the probability of seeing an observation, $v_k$, in state $q_j$ is set to zero (i.e. $b_j(k) = 0$). The true distribution of the source from which the training observations were produced, may be such that it is possible for an outcome to be produced in which $v_k$ is observed in state $q_j$. However, since this case did not ever occur in the training data, the model will never recognize such an outcome, since $b_j(k) = 0$ states that this is an impossible observation. This problem can easily be overcome by restricting all parameters from falling below a minimum value. Modifications of the Baum-Welch algorithm have been formulated to take into account this constraint [81]. In our experiments a minimum value of 0 for all model parameters is enforced.

Lack of the training data often results in parameters receiving poor values and certain characteristics of the source being poorly modeled or even left out all together. This may possibly be remedied by reducing the size of the model, thus reducing the number of parameters that need to be estimated. Unfortunately, if the structure of the model was chosen because it is believed to accommodate some inherent characteristics of the source, changing this is undesirable.

### 6.3.4 Initial Parameter Values

Training algorithms are based on creating a new set of parameters using results obtained from the model with a presently defined set of parameters. Therefore, some initial set of parameters must be picked in order to start the training. Since none of the available methods of training guarantees an optimal solution due to the possibility for getting stuck at a local minima, the choice of a ”good” starting point, which assures the convergence, is crucial in the training stage.

Many of the initial parameter decisions must be made by some intuitive guesses and/or trial and error. Parameters such as the number of states and the number of mixtures used to estimate the observation distributions, have a great impact on how each training sample effects are distributed across the model. For models with a relatively small number of states, the observation density estimates for each state will be well-trained, due to the relatively large number of samples that will go into their training. However, much of the contextual information that could be trained by spreading this data over large number of states may be lost.

The topology of the model is a factor that is greatly influenced by the initial values of the state transition matrix. If an infinite number of training examples were available, we could set all state transitions more or less equal, and let the data decide which transitions are most important. Since, in the real world, lack of data is always a problem, we can greatly help the model to focus on the important parameters by choosing which of these transitions should not be allowed, and which ones will probably be chosen over others. By setting the disallowed transitions to zero and more probable transitions to higher values than the less probable transitions, the training process concentrates on the important parameters. In our experiments, it was discovered that the initial values of parameters such as those that define the model topology, the number of Gaussian mixtures, and the dimensionality of the feature vectors have an impact on the effectiveness of the model. However, the exact initial value of the parameters, such as the initial values of the Gaussian means and covariances, and small changes to the initial values of the transition matrix, tend to have little effect on the model after training.

## 6.4 Discussion about CHMM

In this chapter we propose a new HMM topology called CHMM for the recognition of cyclic patterns such as shape boundary features. CHMM is constructed by a simple modification of the left-right topology, which is widely used in many pattern recognition applications. The circular nature of the proposed topology inherits the properties of both ergodic and temporal models. Although it is simple, the CHMM has many superiorities compared to the classical topologies in the literature for shape analysis.

Firstly, CHMM eliminates the need for determining a unique starting point in recognition of cyclic patterns, e.g. the cycle of a periodic one-dimensional signal or the closed boundary of a two dimensional object. To ensure a consistent symbolic description of patterns from the same class, a unique starting point must, therefore, be defined for the recognition of patterns. This is accomplished by using some heuristic methods in feature extraction stage. The recognition by circular HMM does not require the detection of a starting point and achieves it in model evaluation process.

Secondly, CHMM is size invariant, which does not require to increase the number of states as the size of the boundary increases. The transition from the end state to start state create a model, capable of observing an arbitrary length feature vector, which represents the model. Therefore, no pre-processing is required for normalizing the features to a fixed size.

Thirdly, the CHMM does not require as many non-zero state transition probabilities as the classical topologies. Therefore, the computational complexity of the CHMM is relatively less then the other popular topologies for the higher recognition rates.

Lastly, it can also be used as a shape descriptor for identifying the similar objects in an image database. Given a query shape, the similar shapes which are modeled by a circular HMM can be catched in an image database. It pays no attention to estimate the initial points on the shape boundary. No pre-processing is required for size normalizing. The circular nature of the HMM topology makes the model topology quite stable for shape description. In other words, changing the model size and non-zero state transition in a certain range has no effect on the similarity rank. On the contrary of the classical topologies, the model topology is quite insensitive to length of the observation sequence.

# CHAPTER 7

# PERFORMANCE EVALUATION

The performance of the shape description, similarity measurement and recognition methods proposed in the previous chapters are evaluated in two steps:

- *Evaluation of BAS Descriptors :* First, the row BAS representation is tested with various sampling rates. Then, the row BAS functions are further compressed by the following feature extraction techniques:

  - Piecewise Constant Approximation (PCA),
  - Polyline Approximation (PA),
  - Fourier Descriptors (FD),
  - Sampling by Fourier Transformations (SFT),
  - Approximation by Wavelet Transforms (AWT),
  - Piecewise Linear Approximation by Wavelet Modulus Maxima (PLAWTMM)

  The resulting shape descriptors are ,then, tested for similarity measurement by the following algorithms explained in chapter 5 :

  - Optimal Correspondence of String Subsequence (OCS),
  - Dynamic Warping with Penalty (DWP),
  - Cyclic Sequence Comparison (CSC),

The performance of the BAS descriptors is compared with the recent successful algorithms in the literature. Lastly, The BAS descriptors are tested to investigate the stability under noise and occlusion.

- *Evaluation of CHMM :* The proposed Circular HMM topology developed for shape recognition is compared with the classical topologies. BAS descriptors are fed into the CHMMs as shape features. The recognition performance of each topology is, then, tested.

In the experiments, we use the data set of MPEG-7 Core Experiments (CE) Shape-1, which is developed for the evaluation of 2-D shape descriptors. Section 7.1 introduces the test data. The tests on BAS description and Circular HMM are given section 7.2 and 7.3, respectively.

## 7.1 MPEG-7 Shape Data

The goal of the MPEG 7 CE-Shape-1 is to evaluate the performance of 2D shape descriptors under the below conditions [78]:

- change of a view point with respect to the objects,
- non-rigid object motion and
- noise.

The data in MPEG 7 are restricted to the simple pre-segmented shapes defined by their outer closed contour. The main requirement is that the shape descriptors should be robust to small non-rigid deformations and scale and rotation invariant. The core experiment database is divided into three parts with the following objectives:

- **PART A : Robustness to Scaling and Rotation**
  - *A-1 Robustness to Scaling:* The database contains 420 shapes; 70 basic shapes and 5 derived shapes from each basic shape by scaling digital images with factors 2, 0.3, 0.25, 0.2 and 0.1. Each of the 420 images was used as a query image. A number of correct matches was computed in the top 6 retrieved images. Thus the best possible result is 2520 matches. Figure 7.1 indicates sample shapes from Part A1.

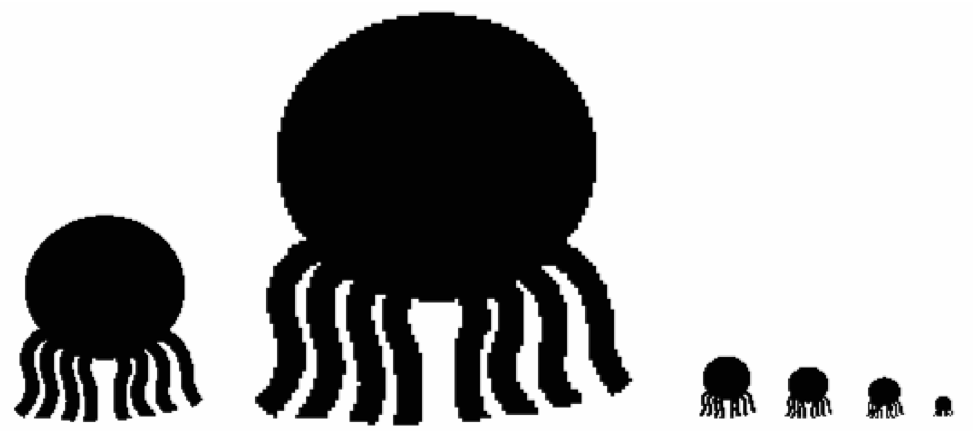


Figure 7.1: An example from PartA1: original shape, and its scaled versions with factors 2, 0.3, 0.25, 0.2 and 0.1, from left to right.

- *A-2 Robustness to Rotation:* The database contains 420 shapes; the 70 basic shapes are the same as in part A-1 and 5 derived shapes from each basic shape by rotation with angles 9, 36, 45, 90 and 150 degrees. Each of 420 images was used as a query image. A number of correct matches was computed in the top 6 retrieved images. Thus the best result is 2520 matches. Figure 7.2 indicates sample shapes from Part A2.

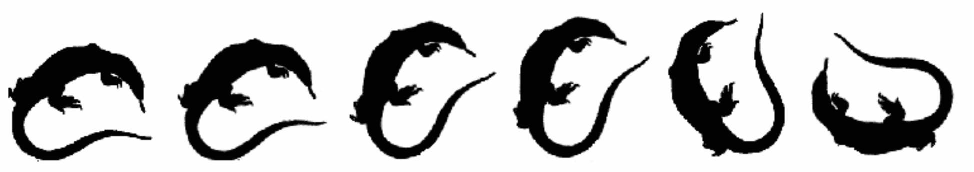


Figure 7.2: An example from PartA2: original shape, and its rotated versions with angles 9, 36, 45, 90 and 150 degrees, from left to right.

- **PART B: Similarity-based Retrieval** This is the main part of the Core Experiment CE-Shape-1. The total number of images in the database is 1400; 70 classes of various shapes, each class with 20 images. Each image was used as query and the number of similar images, which belong to the same class was counted in the top 40 matches. Since the maximum number of correct matches for a single query image is 20, the total number of correct matches is 28000.

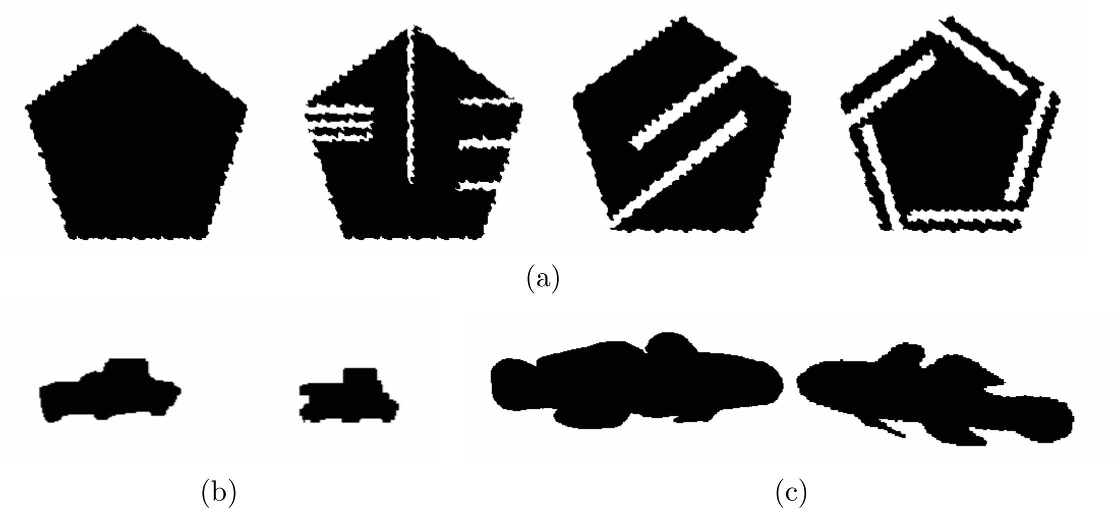


Figure 7.3: Sample shapes from MPEG CE Shape-1 Part B. (a) Shapes from the same class device6, Shapes from the different classes (b) car and truck classes, (c) fish and imfish classes.

Note that, Part B contains classes with significantly different shapes and distinct classes with similar shapes. Therefore, one does not expect to reach a retrieval rate of 100% (examine figure 7.3).

- **PART C: Motion and Non-Rigid Deformations**

  Part C adds a single retrieval experiment to part B. The database for part C is composed of 200 frames of a short video clip with a bream fish swimming, plus a database of marine animals with 1100 shapes. Fish $bream - 000$ used as query and the number of bream shapes in the top 200 shapes was counted. Thus, the maximum number of possible matches was 200. Figure 7.4 indicates sample shapes from Part C.

## 7.2 Evaluation of BAS Descriptors

In order to evaluate the power of BAS representation, first, the row BAS moment functions are used a shape descriptor. Feature vectors are formed with different sizes using various sampling rates. The performance of the row BAS representation is compared to the most successful MPEG 7 Descriptors reported in the literature. Feature extraction methods, described in chapter 4 are, then, employed in order to represent the BAS descriptors in a more compact form.

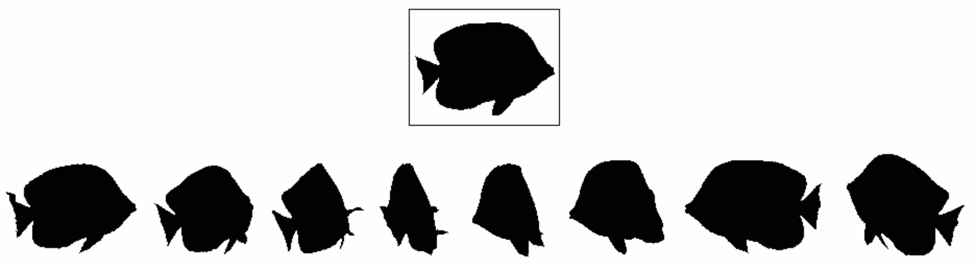


Figure 7.4: An example from PartC : Fish bream-000 and some frames of a short video clip.

### 7.2.1 Row BAS Function

BAS moment functions are discrete in nature with the length equal to the number of points in shape boundary. The length of each BAS moment functions vary between 100 and 4500, which correspond to the smallest and largest shapes in MPEG 7 database, respectively. In order to fix the size of the feature space, BAS moment functions are sampled with equal distance to extract the feature vector. Various sample sizes between 20 and 100 are tested to see the stability of the proposed descriptor to the sampling rate.

The optimal sampling rate for each database is obtained by taking the Fourier Transform of the first order BAS moment function and finding the highest frequency of the non-zero spectral coefficient, as explained in section 7.2.5.

In similarity measurement, the proposed DW with Penalty algorithm, given in chapter 5, is used for this particular experiment. Figures 7.5 and 7.6 show the similarity rate of the BAS descriptor with respect to the size of the feature vector for MPEG 7 Core Experiment Shape 1 part A1, A2, B and C. The similarity rate in each experiment is calculated by taking the ratio of correct matches in the maximum number of possible matches.

As it is seen from figure 7.5.a, the similarity rate remains almost the same after the sample size of 60, which is the Nyquist rate for MPEG 7 CE Part A1. Figure 7.5.b shows the similarity rates of MPEG 7 CE Part A2, indicating that no further improvements is achieved after the sample size of 40, which is the Nyquist rate for this particular data set. Similarly, figure 7.6.a indicates the similarity rates for MPEG 7 CE Part B, which is the most complicated dataset of MPEG 7. The improvement after the vector size 60 is, also, marginal for this datasets, indicating the Nyquist rate of 60. MPEG 7 CE Part C is the simplest data set with the same similarity rate at all vector sizes, as shown in figure 7.6.b.

Note that the sampling process is not applied on the original shape, but just on the BAS moment functions. Therefore, the sampling rate does not directly affect the scale if it is above the Nyquist rate, which is found to be 60 for this particular database.

Figures 7.5 and 7.6 indicate that using only the first moment BAS function is sufficient for Part C. Part A1 and A2 require using both the first and second moments. In Part B, which is the main part of MPEG 7 CE, the addition of third moment increases the similarity rates almost 2 percent. Addition of higher order moments does not improve the performance of the BAS descriptor, indicating that the underlying stochastic process can be modeled by a third order probability density function. Therefore, in the rest of the experiments, the first three moments are used in point wise distance function.

Row BAS features are also tested using OCS and CSC similarity measurement algorithms. The results of all the similarity algorithms with respect to the vector size are depicted in tables 7.1, 7.2, 7.3 and 7.4.

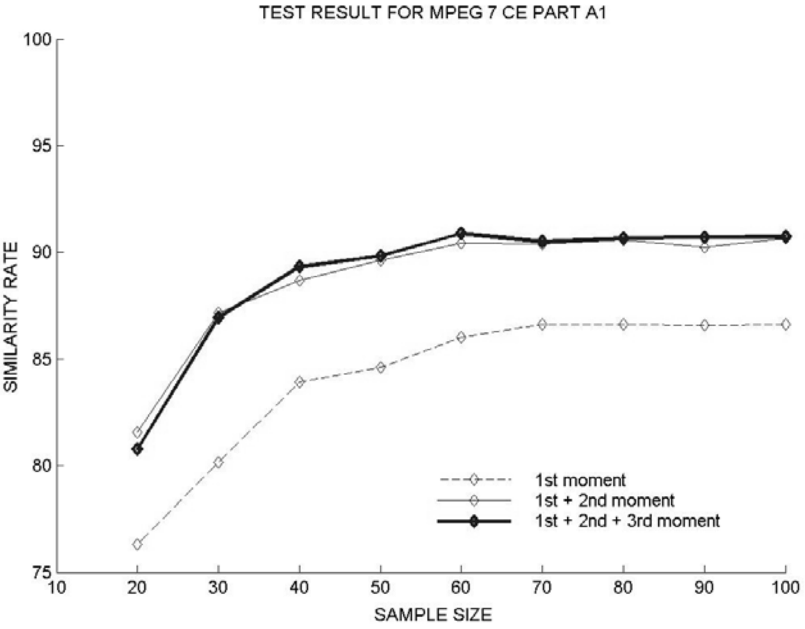
Table 7.1: Similarity rates for MPEG 7 CE Part A1 using row BAS functions with OCS, DWP and CSS Tested on various vector sizes.

| Similarity Alg. | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| OCS | 73.85 | 82.77 | 86.98 | 88.33 | 88.65 | 88.85 | 88.85 | 89.13 | 89.17 | 89.21 |
| DWP | 73.72 | 82.75 | 86.94 | 89.32 | 89.34 | 90.87 | 90.88 | 90.91 | 90.93 | **90.95** |
| CSC | 73.69 | 83.80 | 88.37 | 90.08 | 90.05 | 90.15 | 90.17 | 90.18 | 90.20 | 90.25 |

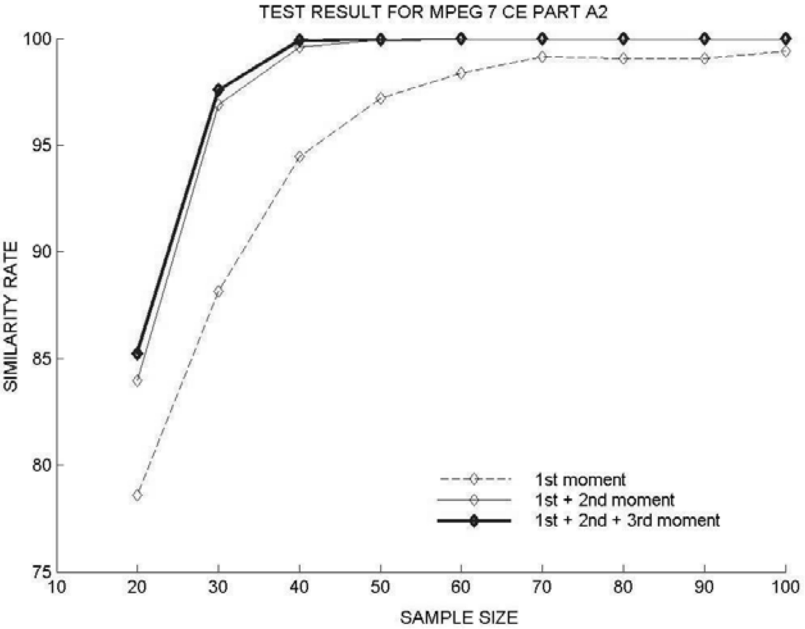
Table 7.2: Similarity rates for MPEG 7 CE Part A2 using row BAS functions with OCS, DWP and CSS tested on various vector sizes.

| Similarity Alg. | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| OCS | 76.55 | 95.75 | 99.09 | 99.44 | 99.48 | 99.50 | 99.50 | 99.52 | 99.52 | 99.52 |
| DWP | 76.56 | 95.23 | 97.57 | 99.92 | 99.96 | **100** | **100** | **100** | **100** | **100** |
| CSC | 76.55 | 95.48 | 99.29 | 99.92 | 99.92 | 99.96 | 99.96 | 99.96 | 99.96 | 99.96 |

In the tables, it is shown that the proposed DW with Penalty algorithm slightly improves the similarity rates in all the data sets. The penalty constant used in DW is taken as 10 in the experiments. The performance of OCS algorithm, on the other

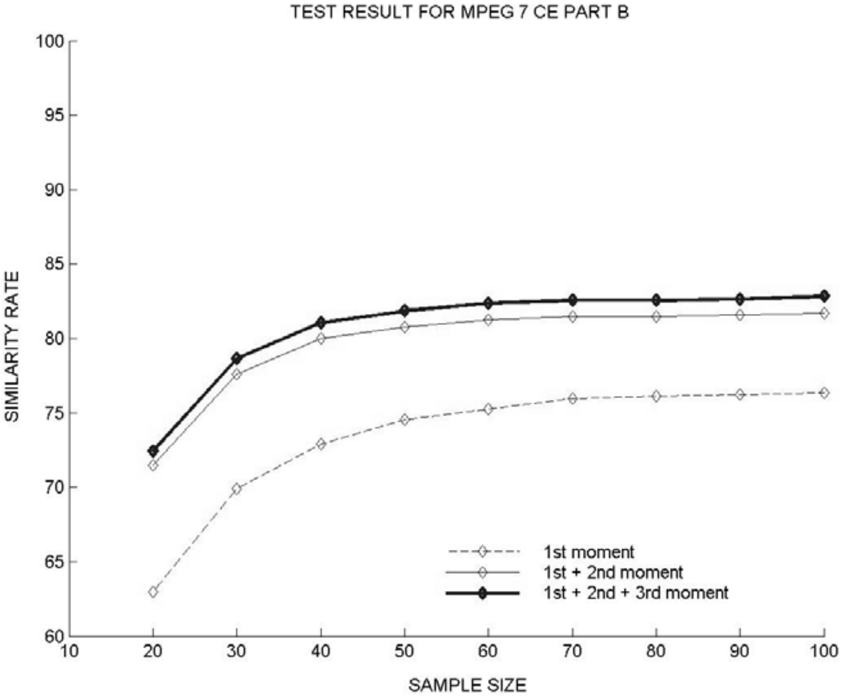


(a)

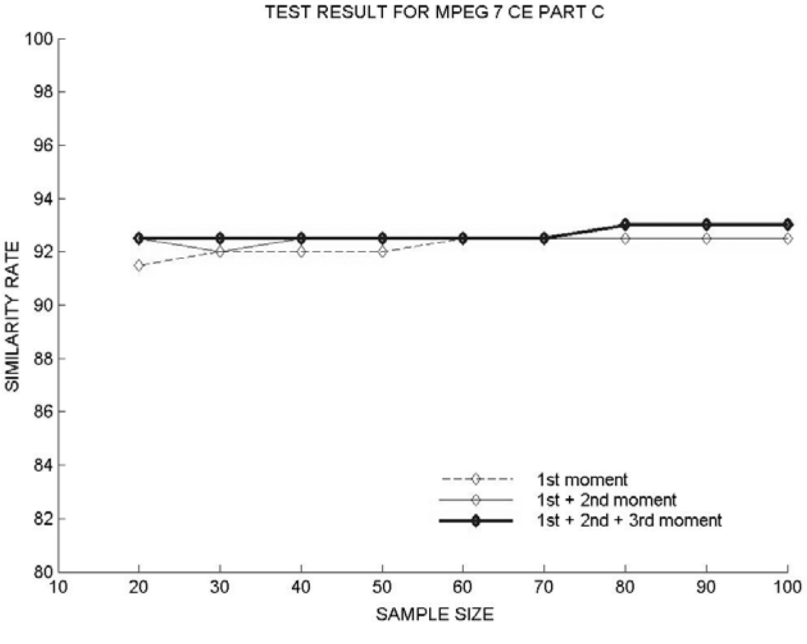


(b)

Figure 7.5: Similarity rates in MPEG CE Shape-1 data, as a function of the sample size, (a) Part A1, (b) Part A2.



(a)



(b)

Figure 7.6: Similarity rates in MPEG CE Shape-1 Part B Data, as a function of the sample size, (a) Part B, (b) Part C.

Table 7.3: Similarity rates for MPEG 7 CE Part B using row BAS functions with OCS, DWP and CSS tested on various vector sizes.

| Similarity Alg. | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| OCS | 60.40 | 71.42 | 74.48 | 75.13 | 76.01 | 76.25 | 76.54 | 76.67 | 76.80 | 76.91 |
| DWP | 62.15 | 75.94 | 79.74 | 81.19 | 81.85 | 82.37 | 82.56 | 82.56 | 82.63 | **82.81** |
| CSC | 59.15 | 74.68 | 79.50 | 80.45 | 81.06 | 81.54 | 82.04 | 82.17 | 82.29 | 82.41 |

Table 7.4: Similarity Rates for MPEG 7 CE Part C using row BAS functions with OCS, DWP and CSS tested on various vector sizes.

| Similarity Alg. | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| OCS | 63 | 89 | 90 | 90 | 91 | 90.05 | 90.05 | 91 | 91 | 91 |
| DWP | 64.5 | 92 | 91.5 | 93.5 | **93.5** | **93.5** | **93.5** | **93.5** | **93.5** | **93.5** |
| CSC | 62.5 | 89.5 | 91 | 91.5 | 92 | 92 | 92 | 92 | 92 | 92 |

hand, is not satisfactory for these particular data sets. However, note that the OCS distance is proved to be metric if the penalty is taken as an upper bound. The metric property of OCS allows the database objects to be indexed. Lastly, the CSC algorithm decreases the computational complexity with respect to the other similarity measurement algorithms, with a slight decrease in the similarity rates.

Another set of experiments is performed in order to test the effect of the modification in DW algorithm. The constant value used as a penalty in the algorithm improves the similarity rates almost 2% at all sampling rates. This is depicted in the table 7.5.

Table 7.5: Comparison of DWP and DW Algorithms using row BAS functions on various vector sizes.

| Similarity Alg. | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| DWP | 62.15 | 75.94 | 79.74 | 81.19 | 81.85 |
| DW | 59.24 | 74.8 | 78.23 | 79.98 | 80.67 |

The proposed cyclic sequence comparison algorithm is, also, compared with the original algorithm proposed in [49]. The CSC algorithm, used in this thesis, outperforms the original algorithm almost 5 % at all sample sizes.

Table 7.6: Comparison of CSC and the algorithm proposed in [49].

| Similarity Alg. | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| CSC | 59.15 | 74.68 | 79.50 | 80.45 | 81.06 |
| Gorman's Alg. | 54.90 | 69.85 | 75.38 | 76.44 | 76.98 |

### 7.2.2 Comparison of Row BAS Function with the MPEG 7 Shape Descriptors

The shape descriptors to be evaluated by the MGEG 7 CE test data, can be investigated in three main categories:

- *Contour-based descriptors:* P320 which uses the curvature scale-space [95], P567 which uses the wavelet representation [32], P298 which uses tangent space and the best possible correspondence of visual parts [77] and recent study called shape context [21].
- *Image-based descriptors :* P687, which uses the Zernike moments [70], P517 which uses the multiplayer eigenvectors.
- *Skeleton-based descriptors :* DAG, which uses Directed Acyclic Graph approach [82].

Table 7.7 indicates the comparison of the proposed descriptor with the reported results of the recent studies. Note that the BAS descriptor has a better performance in most of the experiments, especially in Part B, which is the most difficult part of the MPEG CE Shape-1.

Table 7.7: Comparison of BAS function with the recent studies.

| **Data Set** | **Shape Context** | **Tangent Space** | **CSS** | **Zernika Mom.** | **Wavelet** | **DAG** | **BAS With 40 Samples** | **BAS With 60 Samples** |
|---|---|---|---|---|---|---|---|---|
| **Part A1** | - | 88.65 | 89.76 | 92.54 | 88.04 | 85 | **89.32** | **90.87** |
| **Part A2** | - | 100 | 99.37 | 99.60 | 97.46 | 85 | **99.82** | **100** |
| **Part B** | 76.51 | 76.45 | 75.44 | 70.22 | 67.76 | 60 | **81.19** | **82.37** |
| **Part C** | - | 92 | 96 | 94.5 | 93 | 83 | 93 | 93.5 |

MPEG 7 CE Shape is evaluated with respect to two performance criteria:

- Average performance with the average over three parts (Total Score 1)

- Average over the number of queries (Total Score 2).

The performance comparison of row BAS function is given in Table 7.8. In terms of both of the performance criteria, BAS outperforms all the studies.

Table 7.8: Comparison of average performances.

| | **Tangent Space** | **CSS** | **Zernika Mom.** | **Wavelet** | **DAG** | **BAS With 40 Samples** | **BAS With 60 Samples** |
|---|---|---|---|---|---|---|---|
| **Total Score 1** | 87.59 | 88.62 | 86.93 | 84.70 | 76 | **90.80** | **91.69** |
| **Total Score 2** | 83.12 | 82.62 | 79.92 | 77.14 | 69.38 | **86.12** | **87.27** |

In the above experiments on MPEG 7 CE, it is shown that the first three moment BAS representation with only 40 samples outperform the current studies in the literature. The performances on Part A and Part C data sets are very high and almost the same in most of the methods available in the literature, including the proposed BAS representation. However, it is Part B, which makes the difference among available methods. This is because of the fact that Part B contains classes with significantly different shapes and distinct classes with similar shapes.

Therefore, in the remaining part of the experiments, we focus on the performance of the methods for Part B. Our aim is to increase the computational efficiency, decreasing the vector size. For this purpose, we perform the feature extraction methods on the BAS functions, described in Chapter 4.

### 7.2.3 Piecewise Constant Approximation

In Piecewise Constant Approximation (PCA) method, the BAS function is segmented into equal sized frames and each frame is described by the mean value of the segment. The only difference from the row BAS function is taking the mean instead of the median value in each segment.

The experiments performed on MPEG-7 Part B data set is depicted in tables 7.9. Note that the results approaches to the ones in row BAS function as the vector size increases. The main advantage of using PCA is to perform higher similarity rates in smaller vector sizes.

Table 7.9: Similarity rates of Piecewise Constant Approximation for MPEG 7 CE Part B.

| Similarity Alg. | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| DWP | 65.85 | 77.57 | 81.26 | 82.23 | 82.24 | 82.38 | 82.57 | 82.58 | 82.63 | 82.83 |

### 7.2.4 Polyline Approximation

The goal in polyline approximation is to simplify the BAS function by an evolution process. In each step, a pair of consecutive line segments on the first moment BAS function is replaced by a single line. At the end, the vertices which have the highest relevance measure value in the first moment function are preserved. This process is not applied on the higher moment functions. The second and third moment functions are sampled at the points which are extracted by the evolution of first moment function.

Note that, the resulting vertices, corresponding to minima and maxima points, do not lie in equal intervals on the BAS function. For this reason, it is necessary to consider their positions on the $x$ axis of BAS function in the similarity measurement. Therefore, the relative positions of vertices are inserted into the feature vector together with the moment values. Each vector item is, then, constructed as follows,

$$\Gamma_(i) = (\Gamma^1(i), \Gamma^2(i), \Gamma^3(i), X(i)) \tag{7.1}$$

where $X(i)$ is the $x$ coordinate of the $i^{th}$ vertex, normalized between 0 and 360 degrees.

The stop criteria of the evolution process can be determined in two ways. In the first one, a threshold value is defined on the relevance measure. The substitution of consecutive line segments is stopped if there is no vertex with a lower relevance measure than the threshold. In the second one, the number of iteration determines the stop criteria. The evolution continues until a predefined number of vertex remains. This criteria allows the same size of feature vector to be extracted. Note that the above mentioned stop criteria requires a context dependent threshold value, which may be obtained only trial and error.

In our experiments, we use the iteration criteria to stop the evolution process. Various sizes of feature vectors are extracted from the BAS functions and tested on MPEG 7 Part B data. The results are shown in table 7.10. The table indicates that lower sizes of feature vectors give higher similarity rates than the ones in row BAS function and piecewise constant approximation method.

Table 7.10: Similarity rates of Polyline Approximation for MPEG 7 CE Part B.

| Similarity Alg. | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| DWP | 69.08 | 78.01 | 80 | 81.67 | 82.35 |

### 7.2.5 Fourier Descriptors

In this set of experiments, we use Fourier Descriptors, extracted from BAS functions in order to avoid the complexity of similarity measurement algorithms. Since the phase information is ignored, there is no need to use OCS, DWWP or CSS algorithms. Therefore, the similarity is simply measured by the Euclidean distance. This approach is indeed very fast and simple with the considerable tradeoff in the retrieval rate as indicated in Table 7.11. The decrease in the retrieval rates is expected due to the loss of the phase information. However, it is observed that the BAS function FD's outperforms the centroid distance function FD‘s, which is reported as the best FD method in [145]. Note that the BAS FD‘s give considerably better results even only the first order BAS is used. In order to obtain the same number of Fourier coefficients for the query and database objects, the BAS functions are re-sampled to $N$ samples. In the experiments, we re-sample the BAS functions to 256, so that the Fourier transform can be performed efficiently using Fast Fourier Transformation (FFT) algorithm. Each moment function is represented as a complex number by giving the value 0 to the complex part. The spectral coefficients of BAS moment functions are, then, concatenated in order to form the feature vector of corresponding BAS function. As a result, a feature vector of size $MT$ is constructed, where $M$ is the number of moment functions and $T$ is the number of coefficients used to represent each moment function.

Table 7.11: Similarity rates for MPEG 7 CE Part B using Fourier Descriptors.

| Similarity Measurement | 8 | 16 | 32 | 64 |
|---|---|---|---|---|
| Centroid Distance Function FDs | 65.46 | 66.61 | 66.72 | 66.73 |
| 1st+2nd+3rd Moment BAS FDs | 71.56 | 73.02 | 73.17 | 73.18 |

### 7.2.6 Sampling by Inverse Fourier Transformations

Fourier theorems provide a convenient tool to represent the BAS functions in a more compact form. In this set of experiments, the Fourier transform of BAS functions are used to find an optimal sampling rate. In the transform domain, the Fourier spectrum is truncated after the $T$ coefficients. The descriptor is, then, defined by using the $T$ components of the inverse transform. It is observed that, for the MPEG-7 database, when $T \geq 64$, no further improvement is achieved. The reason for this is the rapid decay of the spectral coefficients which becomes zero after $T = 60$. Table 7.12 indicates the results of using inverse Fourier transform as a shape descriptor for various $T$ values. Note that even for $T = 32$ the descriptor yields better results than the other MPEG-7 descriptors reported in the literature.

Table 7.12: Similarity rates for MPEG 7 CE Part B using Sampling by Fourier Transforms.

| | 8 | 16 | 32 | 64 |
|---|---|---|---|---|
| BAS with Sampled by Inv. Fouirer Transform | 65.30 | 73.81 | 81.42 | 82.38 |

FFT algorithm is used to re-sample the BAS functions to 256 points. FFT and inverse FFT algorithms are applied for each BAS moment function separately. Each item of the feature vector is, then, constructed by using the moment values of each sample point. After the vector size of 64, no further improvement is achieved, indicating that the Nyquist rate for this data set has an upper bound of 64.

### 7.2.7 Approximation by Wavelet Transforms

This set of experiments use Wavelet transforms for feature extraction and perform the dimensionality reduction by wavelet approximation coefficients.

As in the case of Fourier based methods, the BAS functions are re-sampled to 256 in order to have the same number of points for each shape. The dyadic discrete wavelet transformation is applied for each moment function. The approximation coefficients at a predefined level are then used in the feature vector. The feature vector is, then, constructed by using the coefficients of each BAS moment function.

Various family of wavelets tested in the experiments are listed below:

- Haar,

- Daubechies,
- Symlets,
- Coiflets,
- Discrete Meyer,
- Biorthogonal Spline,
- Reverse Biorthogonal Spline,
- Quadratic.

The above wavelets are implemented in MATLAB and the filters are reproduced from WAVELAB developed by D.L. Donoho. The performance of each wavelet family is shown in table 7.13.

Table 7.13: Similarity rates for MPEG 7 CE Part B using Wavelet Approximation Coefficients.

| Wavelet Type | 8 | 16 | 32 | 64 |
|---|---|---|---|---|
| Haar | 64.03 | 73.66 | 81.61 | 82.03 |
| Daubechies | 64.65 | 73.58 | 81.59 | 81.99 |
| Symlets | 66.09 | 75.25 | 81.45 | 81.87 |
| Coiflets | 66.05 | 75.16 | 81.31 | 81.88 |
| DMeyer | 65.32 | 75.03 | 81.22 | 81.67 |
| Biorthogonal | **67.51** | **76.45** | **81.82** | **82.15** |
| RBiorthogonal | 67.64 | 76.33 | 81.83 | 82.16 |
| Quadratic | 67.20 | 74.42 | 81.11 | 81.69 |

In the table, it is shown that different wavelets yield similar results at each sample size. However, the biorthogonal wavelets give the highest similarity rate for small vector sizes $T \leq 16$. The reduction in vector size provides storage and computational efficiency especially for large image databases.

### 7.2.8 Piecewise Linear Approximation by Wavelet Modulus Maxima

Last experiment on the feature extraction methods is performed using the approximation of BAS moment functions by a set of minima and maxima points. The detection of maxima/minima points is performed by the Wavelet Transform Modulus Maxima (WTMM).

The first step of the algorithm detects the positive and negative transitions on the first order BAS moment function, using WTMM. The BAS function is re-sampled to 256 points in order to apply fast dyadic wavelet transform. WTMM is performed on the quadratic spline mother wavelet filter bank algorithm proposed in [88]. The second step, then, searches for the the maxima and minima points based on the in the result of WTMM. After detecting the minima and maxima on the first order BAS moment function, the first moment values together with the second and third moment values at these points are used in the feature vector.

Note that, as in the case of Polyline Approximation by evolution method, the minima and maxima points do not lie in equal intervals on the BAS function. Therefore, their relative positions on the $x$ axis of BAS function are also used in the feature vector.

Note also that the size of feature vectors differ in this method because various number of maxima and minima points can be extracted from each BAS function.

In the experiment performed on MPEG 7 Part B data set, PLAWTMM method performs 74.90 % similarity rates. The average vector size of the descriptors is 18.

### 7.2.9 Comparison of Feature Extraction Methods

In this section, the comparison of proposed feature extraction methods is given. For convenience, the vector sizes are taken in dyadic scales. The results of all the methods except the WTMM approach are given in table 7.14. The reason for disgarding WTMM approach is that this method extracts feature vectors with different sizes.

Table 7.14: Comparison of feature extraction methods.

| Feature Ext. Method | 8 | 16 | 32 | 64 |
|---|---|---|---|---|
| Row BAS | 55.20 | 69.00 | 78.70 | 82.30 |
| PCA | 63.84 | 73.67 | **81.60** | 82.39 |
| PA | 65.59 | **77.07** | **81.55** | 82.13 |
| FD | **71.56** | 73.02 | 73.17 | 73.18 |
| SFT | 64.30 | 72.81 | 80.42 | 82.38 |
| AWT (Bior) | 67.51 | 76.45 | **81.82** | 82.15 |

Table shows that Polyline Approximation method gives highest recognition rate with the vector size of 16. The second highest rate is performed with the Approximation by Biorthogonal Wavelet Transformation. Both performances are better than the available methods in the literature. Note that sample sizes higher than 32 increases the performance at a marginal rate for all the methods, excluding the Fourier Descriptors. On the other hand, Fourier Descriptor method is substantially better for lower vector size than 16. This result is expected due to the high energy compaction capability of the Fourier Descriptors. However, the improvement achieved by the higher vector sizes using other methods, is very critical in practical applications. Therefore, one can judge that FD does not yield satisfactory results in dealing with the shape similarity problems in large database, consisting of wide range of shapes. All of the feature extraction methods except FD, give satisfactory results at or above the Nyquist rate, which is around 64. Therefore, one can judge that determination of the Nyquist rate is critical for a given database.

### 7.2.10 Robustness to Occlusion and Noise

The BAS shape representation is also tested under occlusion and noise. In the first set of experiments, each query shape in MPEG 7 Part B is occluded in various percentages of the original shape. In the second set of experiments, noise is added to the query shape. Then, the similarity based retrieval performance is measured. The database objects remain the same and the number of similar images, which belong to the same class is counted in the top 40 matches, as described in section 7.1.

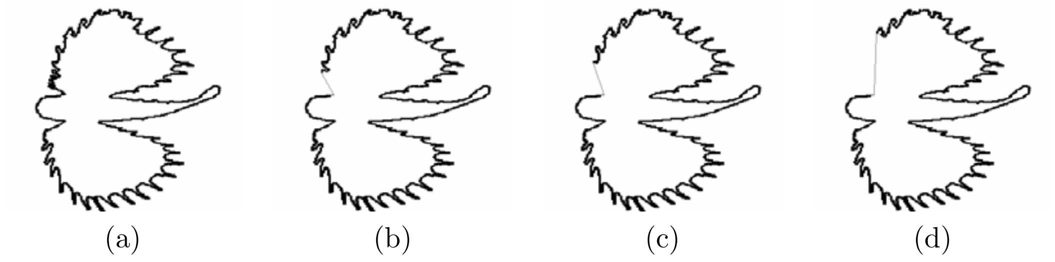


Figure 7.7: An example of occlusion test, (a) Original Shape and its occluded versions with deletion of (b) 4 % (c) 6 % (d) 10 % boundary points.

In order to generate the occluded shape, some part of the shape boundary is deleted manually. Then, the end points of the open boundary are connected by a straight line. The selection of which part of the shape to be deleted is performed randomly. The percentage of boundary points to be deleted is increased from 2 % to 10 % in each experiment (see figure 7.7). The results of similarity rates with different levels of occlusion is shown in table 7.15. The table indicates that as the percentage of occlusion increases from 0 % to 10 %, the performance of the BAS descriptor decreases almost linearly. This result indicates that BAS is quite stable under occlusion.

Table 7.15: Performance result of occlusion test by Piecewise Constant Approximation BAS feature.

| Occlusion Level | Vector Size = 20 |
|---|---|
| 0 % (Original) | 77.57 |
| 2 % | 76.69 |
| 4 % | 75.04 |
| 6 % | 73.88 |
| 8 % | 71.14 |
| 10 % | 69.66 |

For this particular experiment, the feature extraction is performed by Piecewise Constant Approximation. The decrease in the similarity rates as the occlusion level increases, is partly because of the characteristics of the similarity measurement algorithm. Since a portion of the shape boundary is deleted, the corresponding BAS moments is also removed from the representation. Therefore, full matching algorithm, utilized for the occluded shape matching is not convenient. If the similarity distance were calculated using a partial matching algorithm, the performance tests would yield better results.

In order to test the proposed shape descriptor under noise, uniform noise is added to the shape boundary. For this purpose, uniform noise is added to various percentage of shape boundary between 10-50 %. In other words, let the boundary points of the original shape is represented by $p_i = x_i, y_i{}_{i=1}^{N}$, then the noisy shape is generated by

$$p_i^* = p_i + \eta_i \tag{7.2}$$

where $\eta_i$ is a uniform noise,

$$\eta_i = \begin{cases} l & 0 \leq l \leq k, \\ 0 & elsewhere. \end{cases} \tag{7.3}$$

During the experiments $k$ is taken as 3. Noise is added to only a percentage of the points changing between 10-50 % (see figure 7.8). The results of similarity rates with various noise levels is shown in table 7.16.

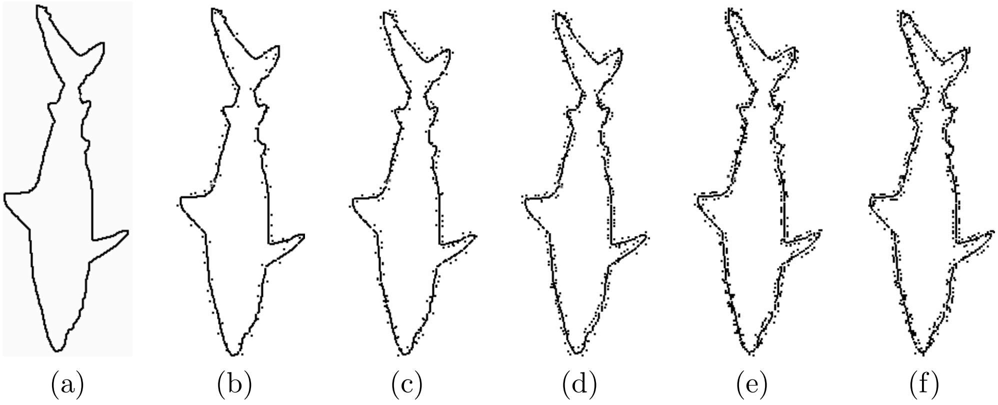


Figure 7.8: An example of noise test, (a) Original shape and its nosiy versions with changing of (b) 10 % (c) 20 % (d) 30 % (e) 40 % and (f) 50 % of the boundary points.

Table 7.16: Performance result of noise test by Piecewise Constant Approximation of BAS feature.

| Occlusion Level | Vector Size = 20 |
|---|---|
| 0 % (Original) | 77.57 |
| 10 % | 77.34 |
| 20 % | 77.16 |
| 30 % | 76.92 |
| 40 % | 76.08 |
| 50 % | 75.65 |

As it is shown in the table, the decreases in similarity rates are very little. This indicates the robustness of the proposed BAS descriptor under noise.

## 7.3 Evaluation of Circular HMM

The performance of proposed circular HMM topology is compared with the classical topologies. The features extracted from BAS representation are fed into various types of HMMs and the recognition performance of each topology is tested.

In the experiments, we use MPEG 7 Part A1, A2 and B data sets, which contain 70 classes of various shapes. In each class, there are 32 images. For each class, we train an HMM model using 10 samples. The remaining 22 samples are then used in the recognition tests.

The feature vector is constructed by using the first order BAS moment function. For this purpose, the derivative of the BAS moment values are coded between 0 and 6.

This task is performed in two steps. In the first step, 60 samples with equal distances are extracted from the BAS function. In the second step, each sample is coded with respect to its preceeding sample as shown in figure 7.9. This results in a feature vector whose size is 60 and which takes values between 0 and 6.

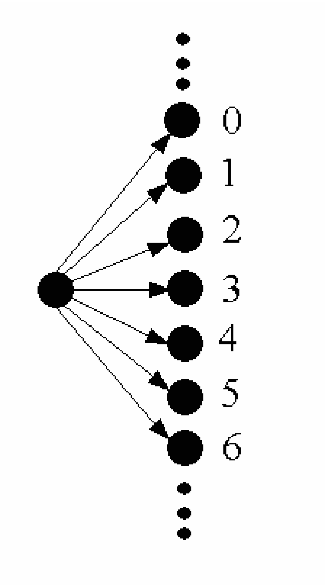


Figure 7.9: Feature extraction for Circular HMM evaluation.

The circular HMM topology is compared with left-to-right and unconstrained (ergodic) topologies. Recall that, the circular HMM is a simple modification of left-to-right HMM model, where the initial and terminal states are connected through the state transition probabilities. This connection eliminates the need to define a starting point of a closed boundary. Since the shape representation process starts with different boundary points, the left-to-right topology is not convenient to determine the exact starting point. Therefore, it is obvious that the recognition performance of left-to-right topology is supposed to be lower than the circular HMM. Ergodic topology, on the other hand, allows the revisits of each state in finite intervals and loses the sequential order of the data. However, the sequential information on the boundary is an important visual property of the shape. A change in the order of visual primitives (convex and concave parts), may drastically change the characteristic of the shape. This fact is depicted in figure 7.10, where two boundaries with the same convex/concave parts expose different shapes.

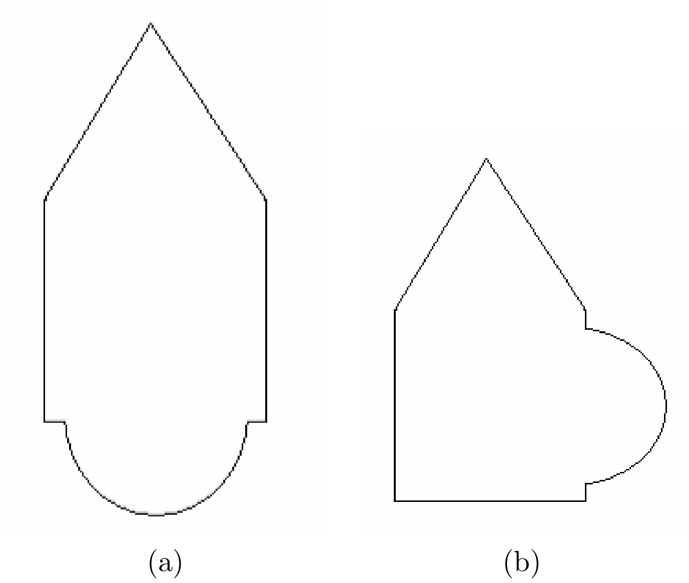

(a) (b)

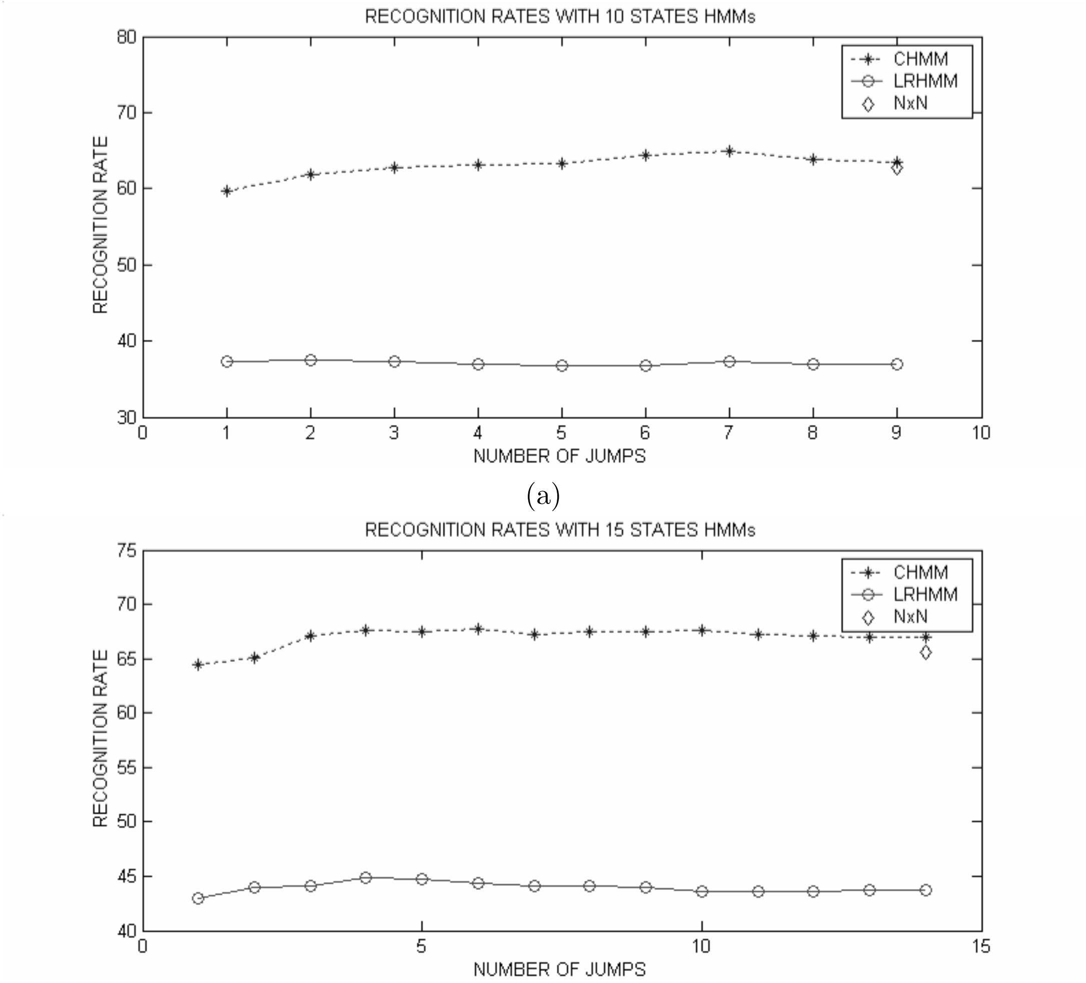
Figure 7.10: Sequential order in visual parts of shape boundary, (a) Original shape boundary, (b) Another shape with the same convex/concave parts in different order.

Since the shape boundary feature is an ordered sequence, ergodic model does not fully utilize the sequential information of the boundary data. In addition, the computational complexity of ergodic model is relatively higher than the temporal ones.

The results of experiments performed on MPEG 7 data set is shown in figures 7.11 and 7.12. In this set of experiments, the HMM topologies with different number of states are used.

In the figures, it is depicted that for this particular data set and features, the circular HMM outperforms both the left-to-right topology and ergodic topology in various number of states. Since the starting point of the boundary differs in shape boundary, the recognition result in the left-to-right topology is very low compared to the other topologies. Ergodic model, on the other hand, gives lower recognition rates than the circular topology. The difference is approximately 2 % in 10 state HMMs. As the number of states increases, the difference reaches to 5 % in 30 state HMMs.

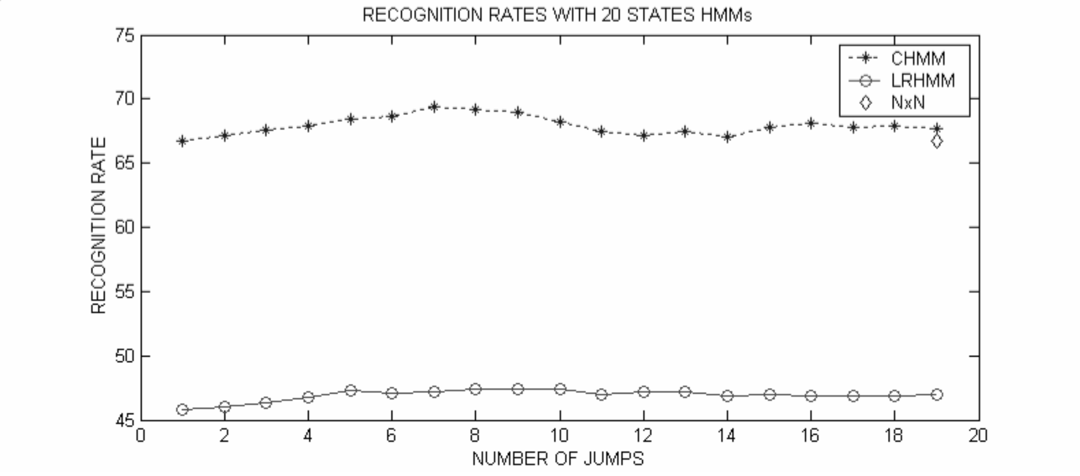
Finally, experiments are performed to find an optimal number of jumps in the circular HMM topology. If we plot the number of jumps in CHMMs which give the maximum recognition rates (see figure 7.13), it is seen that the percentage of the jumps starts 70 % in 10 state CHMM and converges to 10 % in 30 and higher number



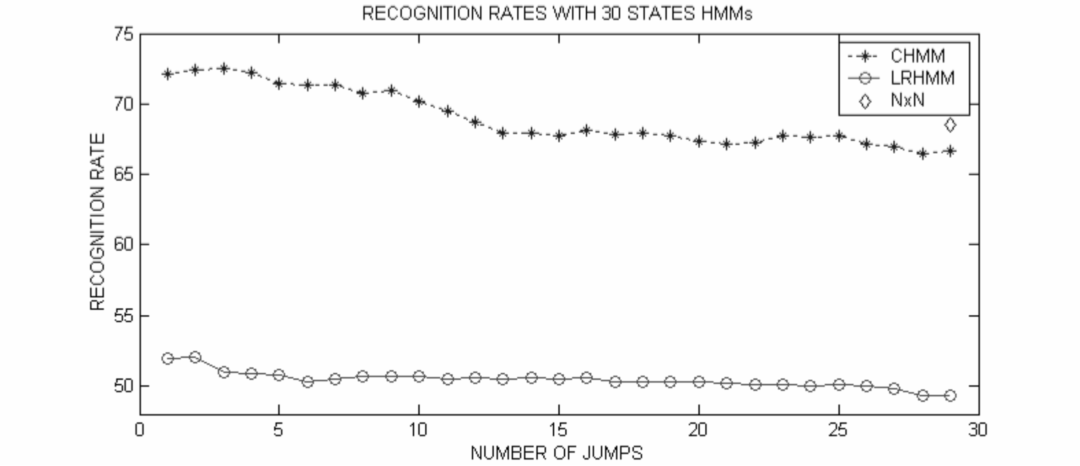
(a)

(b)

Figure 7.11: Recognition results of HMM topologies with (a) 10 states, (b) 15 states



(a)



(b)

Figure 7.12: Recognition results of HMM topologies with (a) 20 states, (b) 30 states.

of state CHMMs.

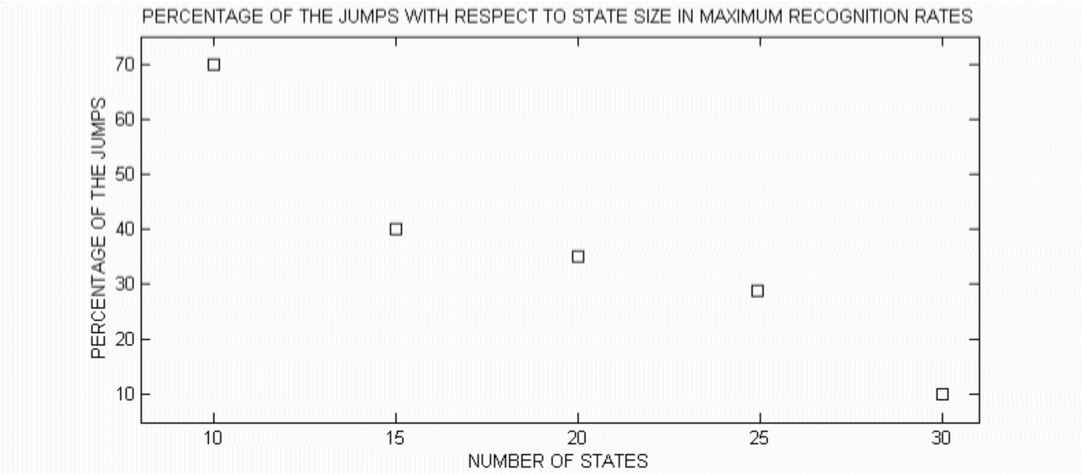


Figure 7.13: Number of jumps in CHMMs with maximum recognition rates.

The figure indicates that for small number of states, the optimal topology is a CHMM, which is close to an ergodic model. However, as the number of states increases, the CHMM jumps exponentially decrease to 10 % for 30 states. For higher number of states than 30, the recognition rates decrease due to the vector size, which is taken as 60 in the experiments. Therefore, one can judge that the optimal jump size 10 % for this data set.

# CHAPTER 8

# CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, we focus on three aspects of shape analysis problem:

- representation (and description),
- similarity measurement and
- recognition.

and propose new approaches for the problems mentioned above.

Firstly, a new shape representation based on the boundary is proposed. The two-dimensional object silhouettes are represented by one-dimensional BAS moment functions, which capture the perceptual information from the statistical information based on the beams of individual bearing points during the navigation on the boundary. At each bearing point, the angle between a pair of beams is calculated to extract the topological structure of the boundary. The representation avoids preprocessing steps, including smoothing, noise reduction etc. It also avoids the selection of a threshold value to represent the resolution of the boundary, thus eliminates the context-dependency of the representation to the data set. Therefore, rather than using a single representation of the boundary, at a predefined scale, the proposed descriptor uses the statistics of the representations at all scales. It gives globally discriminative features to each boundary point by using all other boundary points. It is simple and consistent with human perception through preserving the concave and convex parts of the shapes.

Since the size of the BAS functions are equal to the number of boundary points, the description of shapes with large number of points using row BAS function may be expensive in terms of computation and storage in the subsequent stages of shape analysis. In order to avoid the extra complexity, a more compact form of BAS functions is required. In order to reduce the size of the vector space, a set of features is extracted from the BAS representation. This is mainly performed in two different groups of approaches. The first group of approaches samples the BAS function by determining the Nyquist rate using the Fourier Analysis. This group of approaches identifies the optimal size of the vector space using the Nyquist rate, eliminating the threshold problem for compressing the BAS moment functions. Piecewise constant approximation, Fourier based methods and wavelet approximation method can be considered in this category. The second group detects the maxima and minima on the BAS function. Polyline approximation and piecewise linear approximation by WTMM use this approach. In the second group of methods, identification of the resolution of BAS representation remains to be an open question and obtained by trial and error during the experiments.

The size of shape descriptor is a critical parameter in all the proposed methods in the literature. It is determined by compromising the accuracy and complexity of the representation. Unlike the available methods, rather then empirically identifying the vector size of the feature space, we effectively identified the smallest possible vector size for the highest possible performance.

Secondly, we made some improvements in three different similarity distance methods defined on the BAS feature space. The main idea in similarity measurement is to minimize the distance between two vectors by allowing deformations. This is achieved by first solving the correspondence problem between items in feature vectors and then computing the distance as a sum of matching errors between corresponding items.

The first method is Optimal Correspondence of String Subsequences (OCS) Algorithm, which is developed for the infinite alphabet string matching. The OCS algorithm extends the finite alphabet editing error minimization matching to the infinite alphabet penalty minimization matching. It is, also, proved that the string matching derived from OCS is a metric. In this thesis, we adopt this algorithm to BAS features.

The second method used for similarity measurement is Dynamic Warping (DW), which is widely used in speech recognition. The DW algorithm finds an optimal match between two sequences, which allows stretching and compression of sections in the subsequences. However, in shape boundary representation, the expansion and compression of some part of the sequence change the visual appearance of the shape. Therefore, the expansion and compression of BAS functions should be handled with a specific care, unlikely in speech case. For this reason, we propose to give penalties for subsequence expansion and compression of the BAS function. This simple modification of the cost function provides substantial improvement on the overall performance of the DW algorithm.

The third method is cyclic sequence matching algorithm, which approximates the optimal solution. To ensure a consistent description of a shape, which is cyclic in nature, a unique starting point must be defined for each shape. Since this is impractical to achieve, the alignment computation must determine the amount of cyclic shift that has taken place in order to find the optimal solution. In the above methods, this is performed by shifting any sequence one item at a time and recomputing the sequence comparison. However, the computational cost increases during the similarity measurement, resulting impractical computational cost for large databases. In order to avoid this complexity, we propose an efficient cyclic sequence comparison algorithm.

One of the major contributions of this thesis is to propose a new HMM topology for shape recognition. The circular HMM topology is a simple modification of left-to-right HMM model, where the initial and terminal states are connected through the state transition probabilities. This connection eliminates the need to define a starting point of a closed boundary, in the description problem. The circular HMM is both temporal and ergodic. Therefore, the states can be revisited in finite time intervals, while keeping the sequential information in the string, which represents the shape. This structure enables one to decide on the optimal state order by simple experiments on the training data and requires no size normalization.

In the performance evaluation of proposed methods, we use the dataset of MPEG 7 Core Experiments Shape-1. The tests indicate that BAS descriptor is quite stable under noise and occlusion. It is insensitive to translation, rotation and scale. It is observed that the proposed shape descriptor outperforms all the methods in the literature in this particular data set. The experiments performed in order to test the proposed HMM topology, indicate that the circular HMM gives higher recognition rates than the classical topologies in boundary based shape recognition.

## 8.1 Future Directions

Future works on the proposed methods in this thesis can be explained under the following headlines:

- The BAS representation calculates the beam angle statistics of each point $p(i)$ by using a set of neighborhood system. For each neighborhood system $K$, there is only one pair of beams connecting $p(i)$ to $p(i+K)$ and $p(i-K)$. For the bearing point $p(i)$, the beam angle statistics is calculated using angles between the forward and backward beam vectors. In BAS representation, the definition of neighborhood system can be extended to all the pairs of beams.

- The BAS representation assumes that the shape boundary is extracted in an ordered manner and the resulting boundary points forms a sequence. In order to extend the use of BAS representation in object localization, the boundary information of shape can extracted from an edge detected image, where a set of points represents the object silhouette in an image. Then, the beam angle statistics of each point can be calculated using all the other points which lie over the image.

- In the proposed method, the BAS descriptor employs samples of the moment functions for the first three moments. This requires a feature vector with length of T. However, each item of the vector contains M values, where M is the number of moments used in the representation. In order to use a single value in each entry, a code book can be generated by a quantization procedure on the moment values. This process increase the computational efficiency of similarity measurement algorithms and decrease the space complexity.

- In order to decrease the sensitivity of BAS descriptors under occlusion, a partial matching algorithm can be developed in similarity measurement. Since the occluded part of the shape boundary is not represented in BAS function, the absence of this part should be taken into account in the matching process. This can be achieved by a partial matching algorithm in calculating the distance between two BAS descriptors.

# REFERENCES

[1] I. Abulhaiba and P. Ahmed. A fuzzy graph theoretic approach to recognize the totally unconstrained handwritten numerals. *Pattern Recognition*, 26(9):1335–1350, 1993.

[2] G. Agam and I. Dinstein. Geometric separation of partially overlapping nonrigid objects applied to automatic choromosome classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19:1212–1222, 1997.

[3] O. Aichholzer, H Alt, and G. Rote. Matching shapes with a reference point. *International Journal of Computational Geometry and Applications*, 7:349–363, 1997.

[4] H. Alt, B. Behrends, and J. Blomer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, pages 251–265, 1995.

[5] H. Alt, A. Efrot, G. Rote, and C. Wenk. Matching planar maps. In *14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 589–598, 2003.

[6] H Alt and M. Godeau. Computing the frechet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, pages 75–91, 1995.

[7] H. Alt, C. Knauer, and C. Wenk. Matching polygonal curves with respect to the freceht distance. In *International Symposium on Theoretical Aspects of Computer Science*, pages 63–74, 2001.

[8] L. Ambrosio and V. M. Tortorelli. On the approximation of free discontiuity problems. *Bolletino Della Unione Matematica Italiana*, 6:105–123, 1990.

[9] K. Arbter. Affine invariant fourier descriptors. *Science Publishers*, pages 153–165, 1989.

[10] E. Arkin, P. Chew, D. Huttenlocher, K. Kedem, and J. Mitchel. An efficient computable metric for comparing polygonal shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13.

[11] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8.

[12] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, pages 183–193, 1954.

[13] H. I. Avi-Itzhak, T. A. Diep, and H. Gartland. High accuracy optical character recognition using neural network with centroid dithering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(2):218–228, 1995.

[14] J. Babaud, A. Witkin, M. Baudin, and R. Duda. Uniquness of the gaussian kernel for scale space filtering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8.

[15] L. R. Bahl, P. F. Brown, P. V. Desouza, and R. L. Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. *Proc. ICASSP'86*, 1986.

[16] L. R. Bahl, P. F. Brown, P. V. Desouza, and R. L. Mercer. A new algorithm for estimation of hidden markov model parameters. *IEEE International Conference on ASSP*, 1988.

[17] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.

[18] L. E. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of markov processes. *Inequalities*, pages 1–8, 1972.

[19] A. Belaid and J.P. Haton. A syntactic approach for handwritten mathematical formula recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:105–111, 1984.

[20] S. O. Belkasim, M. Shridhar, and M. Ahmad. Pattern recognition with moment invariants: A comparative survey. *Pattern Recognition*, 24(12):1117–1138, 1991.

[21] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[22] A. Bengtsson and J. Eklundth. Shape representation by multiscale contour approximation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13.

[23] H. D. Block and F. Rosenblatt B. W. Knight. Analysis of a four layer serious coupled perceptron. *II. Rev. Modern Physics*, 34:135–152, 1962.

[24] H. Blum. A transformation for extracting new descriptors of shape. In *Models For The Perception Of Speech and Visual Forms*. MIT Press Cambridge.

[25] H. Blum and R. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10.

[26] M. Bober. Mpeg-7 visual shape descriptors. *IEEE Trans. Circuits and Systems For Video Technology*, 11(6):716–719, 2001.

[27] M. Brady and H. Asada. Smoothed local symmetries and their implementation. *International Journal of Robotics Research*, 3:36–61, 1984.

[28] H. Bunke and U. Buhler. Applications of approximate string matching to 2d shape recognition. *Pattern Recognition*, 26(12):1797–1812, 1993.

[29] K. R. Castleman. *Digital Image processing.* Prentice-Hall, NJ, 1996.

[30] M. Y. Chen, A. Kundu, and J. Zhou. Off-line handwritten word recognition using a hidden markov model type stochastic network. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16:481–496, 1994.

[31] F. H. Cheng, W. H. Hsu, and C. A. Chen. Fuzzy approach to solve the recognition problem of handwritten chinese characters. *Pattern Recognition*, 22(2):133–141, 1989.

[32] G. Chuang and C. C. Kuo. Wavelet descriptor of planar curves: Theory and applications. *IEEE Trans. Image Processing*, 5:56–70, 1996.

[33] F. S. Cohen, Z. Huang, and Z. Yang. Invariant matching and identification of curves using b-splines curve representation. *IEEE Trans. Image Processing*, 4:1–10, 1995.

[34] F. S. Cohen, Z. Huang, and Z. Yang. Invariant matching and identification of curves using b-splines curve representation. *IEEE Trans. Image Processing*, 4:1–10, 1995.

[35] F. S. Cohen and J. Y. Wang. Part i: Modeling image curves using invariant 3-d object curve models- a path to 3-d recognition and shape estimation from image contours. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16:1–12, 1994.

[36] S. Connell. *A Comparison of Hidden Markov Model Features for the Recognition of Cursive Handwriting, Master Thesis.* Michigan State University, 1996.

[37] L. F. Costa and R. M. Cesar Jr. *Shape Analysis and Classification : Theory and Practice.* CRC Press, 2001.

[38] M. Das, M. J. Paulik, and N. K. Loh. A bivariate autoregressive modeling technique for analysis and classification of planar shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(1):97–103, 1990.

[39] P. A. Devijer and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice Hall Edition, 1982.

[40] S. R. Dubois and F. H. Glanz. An autoregressive model approach to two dimensional shape classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):55–66, 1986.

[41] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley and Sons, Inc., New York, 2001.

[42] Y. Ephraim, A. Dembo, and L. R. Rabiner. A minimum discrimination information approach for hidden markov modelling. *ICASSP'87*, 1987.

[43] G. D. Forney. The viterbi algorithm. *Proc. of IEEE*, pages 268–278, 1973.

[44] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Transactions*, 10:260–268, 1961.

[45] H. Freeman and J. Saghri. Comparative analysis of line-drawing modeling schemes. *Computer Grphics Image Processing*, 12:203–223, 1980.

[46] K. Fukushima and N. Wake. Handwritten alphanumeric character recognition by neocognition. *IEEE Trans. Neural Network*, 2(3):355–365, 1991.

[47] P. D. Gader, B. Forester, M. Ganzberger, A. Gillies, B. Mitchell, M. Whalen, and T. Yocum. Recognition of handwritten digits using template and model matching. *Pattern Recognition*, 24(5):421–431, 1991.

[48] R. F. Gonzales and P. Wintz. *Digital Image processing.* Addison-Wesley, 1993.

[49] J. W. Gorman and O. R. Mitchell. Partial shape recognition using dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(2):257–266, 1988.

[50] A. Goshtasby. Description and discrimination of planar shapes using shape matrices. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7:738–743, 1985.

[51] J. Gregor and M. G. Thomason. Dynamic programming alignment of sequences representing cyclic patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(2):129–135, 1993.

[52] J. Gregor and M. G. Thomason. Efficient dynamic programming alignment of cyclic strings by shift elimination. *Pattern Recognition*, 29(7):1179–1185, 1996.

[53] M. Hagedoorn. *Pattern Matching Using Similarity Measures, PhD Thesis.* Utrecht University, 2000.

[54] M. Hagedoorn and R. C. Veltkamp. Measuring resemblance of complex patterns. *Discrete Geometry for Computer Imagery*, 1999.

[55] Y. He and A. Kundu. 2-d shape classification using hidden markov model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(11):1172–1184, 1991.

[56] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 18:65–96, 1984.

[57] G. W. Howell, D. W. Fausett, and L. V. Fausett. Quasi-circular splines : A shape preserving approximation. *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, 55:89–97, 1993.

[58] J. Hu and T. Pavlidis. A hierarchical approach to efficient curvilinear object searching. *Computer Vision and Image Understanding*, 63(2):208–220, 1996.

[59] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, 8:179–187, 1962.

[60] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[61] Y. Ikebe and S. Miyamoto. Shape design, representation and restoration with splines. In *Picture Engineering, Springer*, pages 75–95, 1982.

[62] A. K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1):4–38, 2000.

[63] A. K. Jain, J. Mao, and K. M. Mohiuddin. Artificial neural networks: A tuitorial. *Computer*, pages 31–44, 1996.

[64] A. K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(12):1386–1391, 1997.

[65] F. Jelinek and R. L. Mercer. Interpolated estimation of markov source parameters from sparse data. *Pattern Recognition in Practice*, pages 381–397, 1980.

[66] A. E. Johnson and M. Hebert. Recognizing objects by matching oriented points. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 684–689, 1997.

[67] T. Kanade. Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence*, 17.

[68] R. L. Kashyap and R. Chellappa. Stochastic models for closed boundary analysis : Representation and reconstruction. *IEEE Trans Information Theory*, 27(5):627–637, 1981.

[69] M. I. Khalil and M. M. Bayoumi. A dyadic wavelet affine invariant function for 'd shape recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(10):1152–1164, 2001.

[70] A. Khotanzan and H. Y. Hong. Invariant image recognition by zernike moments. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12:489–497, 1990.

[71] H. Y. Kim and J. H. Kim. Handwritten korean character recognition based on hierarchical random graph modeling. *Proc. Int. Workshop Frontiers in Handwriting Recognition*, pages 577–586, 1998.

[72] W. Y. Kim and Y. S. Kim. A new region based shape descriptor. In *ISO/IEC MPEG99/M5472*, pages Maui, Hawaii, 1999.

[73] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. Toward a computational theory of shapes: An overview. *Lecture Notes inComputer Science*, 427:402–407, 1990.

[74] T. Kohonen. Self organizing maps. *Springer Series in Information Sciences*, 30, 1995.

[75] A. Kornai, K. M. Mohiuddin, and S. D. Connell. An hmm-based legal amount field ocr system for checks. *IEEE Trans. Systems, Man and Cybernetics*, pages 2800–2805, 1995.

[76] K. Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–439, 1992.

[77] L. J. Latecki and R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1185–1190, 2000.

[78] L. J. Latecki, R. Lakamper, and U. Eckhardt. Shape descriptors for nonrigid shapes with a single closed contour. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.

[79] J. S. Lee, Y. N. Sun, and C. H. Chen. Multiscale corner detection by wavelet transform. *IEEE Trans. Image Processing*, 4:100–104, 1995.

[80] M. D. Levine. *Vision in Man and Machine.* McGraw-Hill, NY, 1985.

[81] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *Bell System Technical Journal*, (62):1035–1074, 1983.

[82] J. L. Lin and Y. S. Kung. Coding and comparison of dags as a novel neural structure with application to online handwritten recognition. *Computer Grphics Image Processing*, 1996.

[83] C. Y. Liou and H.C. Yang. Handprinted character recognition based on spatial topology distance measurements. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(9):941–945, 1996.

[84] H. Liu and D. Srinath. Partial shape classification using contour matching in distance transformation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12:1072–1079, 1990.

[85] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31:983–1001, 1998.

[86] S. Loncaric and A. P. Dhawan. A morphological signature transform for shape description. *Pattern Recognition*, 26:1029–1037, 1993.

[87] M. Maes. On a cyclic string-to-string correction problem. *Information processing Letters*, 35(2):73–78, 1990.

[88] S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1999.

[89] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14:710–732, 1992.

[90] J. Mao, K. M. Mohiuddin, and T. Fujisaki. A two stage multi-network ocr system with a soft pre-classifier and a network selector. *Proc. 3rd Int. Conf. On Document Analysis and Recognition*, pages 821–828, 1995.

[91] D. Marr. Early processing of visual information. *Proc. Roy. Soc. London*, pages 483–519, 1976.

[92] A. Marzal and S. Barrachina. Speeding up the computation of the edit distance for cyclic strings. In *15. International Conference Pattern Recognition*, pages 895–898, 2000.

[93] M. L. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry.* MIT Press, 1969.

[94] M. A. Mohamed and P. Gader. Generalized hidden markov models - part ii: Application to handwritten word recognition. *IEEE Trans. Fuzzy Systems*, 8(1):82–85, 2000.

[95] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. *Image Databases and Multimedia Search, A. W. M. smeulders and R. Jain ed., World Scientific Publication*, pages 51–58, 1997.

[96] F. Mokhtarian and A. Mackworth. Scale based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:2–14, 1986.

[97] R. A. Mollineda, E. Vidal, and F. Casacuberta. Cyclic sequence alignments : Approximate versus optimal techniques. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(3):291–299, 2002.

[98] S. Parui, E. Sarma, and D. Majumder. How to discriminate shapes using the shape vector. *Pattern Recognition Letters*, 4:201–204, 1986.

[99] T. Pavlidis. A review of algorithms for shape analysis. *Computer Graphics Image Processing*, 7:243–258, 1978.

[100] T. Pavlidis. *Algorithms For Graphics and Image Processing.* Springer, 1982.

[101] L. F. C. Pessoa and P. Maragos. Neural networks with hybrid morphological/rank/linear nodes: A unifying framework with applications to handwritten character recognition. *Pattern Recognition*, 33:945–960, 2000.

[102] I. Pitas and A. N. Venetsanopoulos. Morphological shape decomposition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(1):38–45, 1990.

[103] S. Pizer, W. Oliver, and S. Bloomberg. Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9.

[104] K. E. Price. Relaxation matching techniques comparison. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(5):617–623, 1985.

[105] R. J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP:Graphics Models and Image Processing*, 54(5):438–460, 1992.

[106] A. Quddus and M. M. Fahmi. Fast wavelet based corner detection technique. *IEE Electronic Letters*, 35:287–288, 1999.

[107] A. Quddus and M. Gabbouj. Wavelet-based corner detection technique using optimal scale. *Pattern Recognition Letters*, 23:215–220, 2002.

[108] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77.

[109] L.R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, 1986.

[110] N. V. S. Reddy and P. Nagabhushan. A three dimensional neural network model for unconstrained handwritten numeral recognition: A new approach. *Pattern Recognition*, 31(5):511–516, 1998.

[111] B. Ripley. Statistical aspects of neural networks. *Networks on Chaos: Statistical and Probabilistic Aspects*, 1993.

[112] H. Rom and G. Medioni. Hierarchical decomposition of axial shape description. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15.

[113] Y. Rui, A.C. She, and T.S. Huang. Modified fourier descriptors for shape representation a practical approach. In *Proc. First Int'l Workshop Image Databases and Multimedia Search, Amsterdam*, 1996.

[114] D. Sankoff and J. Kruskal. *Time Warps, String Edits and Macromolecules.* CSLI Publications, 1999.

[115] M. Shridhar and A. Badreldin. High accuracy syntactic recognition algorithm for handwritten numerals. *IEEE Trans. Systems Man and Cybernetics*, 15(1):152–158, 1985.

[116] K. Siddigi, J. K. Tresness, and B. B. Kimia. Parts of visual form : Psychophysical aspects. *Perception*, 25:399–424, 1996.

[117] R. M. K. Sinha, B. Prasada, G. F. Houle, and M. Babourin. Hybrid contextual text recognition with string matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9):915–925, 1993.

[118] S. Smith, M. Borgoin, K. Sims, and H. Voorhees. Handwritten character classification using nearest neighbor in large databases. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(9):915–919, 1994.

[119] C. Y. Suen and T. A. Mai. A generalized knowledge-based system for the recognition of unconstrained handwritten numerals. *IEEE Trans. Systems, Man and Cybernetics*, 20(4):835–848, 1990.

[120] C. C. Tappert. Cursive script recognition by elastic matching. *IBM J. Research and Development*, 26(6):765–771, 1982.

[121] S. Tari and J. Shah. Local symmetries of shapes in arbatrary dimension. In *Internatioanl Conference on Computer Vision*, pages 1123–1128, 1998.

[122] S. Tari, J. Shah, and H. Pien. Extraction of shape skeletons from grayscale images. *Computer Vision Image Understanding*, 66(2):133–146, 1997.

[123] M. Tayli and A I. Ai-Salamah. Building bilingual microcomputer. *System Communications of the ACM*, 33(5):495–504, 1990.

[124] A. Taza and C. Suen. Discrimination of planar shapes using shape matrices. *IEEE Trans. System Man Cybernetics*, 19:1281–1289, 1989.

[125] M. R. Teague. Image analysis via the general theory of moments. *Journal of Optical Society America*, 70:920–930, 1980.

[126] D. W. Thompson. Morphology and mathematics. *Transactions of the Royal Society of Edinburgh*, L, Part IV:No.27:857–895, 1915.

[127] O. D. Trier, A. K. Jain, and T. Taxt. Feature extraction method for character recognition - a survey. *Pattern Recognition*, 29(4):641–662, 1996.

[128] M. Trimeche, F.Alaya Cheikh, and M. Gabbouj. Similarity retrieval of occluded shapes using wavelet-based shape feature. In *SPIE, Boston*, 2000.

[129] D. Tubbs. A note on binary template matching. *Pattern Recognition*, 22(4):359–365, 1989.

[130] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[131] F. Ulupinar and R. Nevatia. Inferring shape from contour for curved surfaces. In *Internatioanl Conference on Pattern Recognition*, pages 147–154, 1990.

[132] C. Urdiales, A. Bandera, and F. Sandoval. Non-parametric planar shape representation based on adaptive curvature functions. *Pattern Recognition*, 35:43–53, 2002.

[133] R. C. Veltkamp. Shape matching : Similarity measures and algorithms. Technical Report UU-CS-2001-03, Universiteit Utrecht, 2001.

[134] R. C. Veltkamp and M. Hagedoorn. State of the art in shape matching. In *Principals of Visual Information Retrieval*, pages 87–119. Springer, 2001.

[135] S. Venkatasubramanian. *Geometric Shape Matching and Drug Design*. PhD thesis, Stanford University, 1999.

[136] A. J. Viterbi. Error bounds for convolution codes and an asymptotically optimal decoding algorithm. *IEEE Trans. on Information Theory*, pages 260–269, 1967.

[137] Y. P. Wang and T. Pavlidis. Optimal correspondence of string subsequences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(11):1080–1087, 1990.

[138] E. K. Warrington and A. M. Tayeor. The contribution of the right parietal lobe to object recognition. *Cortex*, pages 152–164, 1973.

[139] A. P. Witkin. Scale space filtering. In *8th International Joint Conference on Artificial Intelligence*, pages 1019–1022, 1983.

[140] S. L. Xie and M. Suk. On machine recognition of hand-printed chinese characters by feature relaxation. *Pattern Recognition*, 21(1):1–7, 1988.

[141] K. Yamamoto and A. Rosenfeld. Recognition of handprinted kanji characters by a relaxation method. In *Proc. 6th Int. Conf. on Pattern Recognition, Germany*, pages 395–398, 1982.

[142] F. T. Yarman-Vural and E. Ataman. Noise histogram and cluster validity for gaussian mixtured data. *Pattern Recognition*, 20(4):385–401, 1987.

[143] C. Zahn and R. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans Computer*, 21:269–281, 1972.

[144] B. Zhang, M. Fu, H. Yan, and M.A. Jabri. Handwritten digit recognition by adaptive -subspace self organizing map (assom). *IEEE Trans. Neural Network*, 10(4), 1999.

[145] D. Zhang and G. Lu. A comparative study of fourier descriptors for shape representation and retrieval. In *Proc. 5th asian Conference on Computer Vision, Melbourne-Australia*, 2002.

# VITA

Nafiz Arıca received his B.S. degree Turkish Naval academy in 1991. He worked for Navy as communication and combat officier for four years. In 1995, he joined Middle East Technical University (METU), where he received the MSc degree in computer engineering. His thesis was awarded the thesis of the year in 1998 at METU. He received the "Best paper Award" in Sinyal sleme Kurultayı (SIU 2002). His research interests include computer vision and pattern recognition; specifically character recognition, content based image retrieval. He is a member of IEEE Computer Society and Turkish Pattern Recognition and Image Analysis Society.

**Publications**

**Journals :**

BAS : A Perceptual Shape Descriptor Based on the Beam Angle Statistics. Nafiz Arıca, Fatoş Yarman-Vural. Pattern Recognition Letters, Vol. 24, pp.1627-1639, 2003.

Optical Character Recognition For Cursive Handwriting. Nafiz Arıca, Fatoş Yarman-Vural, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 6, pp. 801-813, 2002.

An Overview of Character Recognition Focused on Off-line Hadwriting. Nafiz Arıca, Fatoş Yarman-Vural, IEEE Trans. Systems, Man and Cybernetics, Part C: Applications and Reviews, vol.31, no.2, pp.216-232, 2001.

One Dimensional Representation of Two Dimensional Information for HMM Based Handwritten Recognition. Nafiz Arıca, Fatoş Yarman-Vural. Pattern Recognition Letters, vol.21 pp 583-592, 2000.

**Conference Proceedings :**

A Compact Shape descriptor based on the Beam Angle Statistics. Nafiz Arıca, Fatoş Yarman-Vural. International Conference on Image and Video Retrieval, (CIVR 2003), (also published in Lecture Notes in computer Science, Volume 2728 , pp.152-162.

Tıkız Şekil Betimleyicileri. Nafiz Arıca, Fatoş Yarman-Vural. Sinyal İşleme ve Uygulamaları Kurultayı (SIU'2003), Istanbul, Turkey, June 2003.

A Perceptual Shape Descriptor. Nafiz Arıca, Fatoş Yarman-Vural. International Conference on Pattern Recognition (ICPR) Quebec, Canada, 2002.

A Shape Descriptor Based on Circular Hidden Markov Model. Nafiz Arıca, Fatoş Yarman-Vural. Proc. International Conference on Pattern Recognition (ICPR) Barcelona, Spain, 2000.

A New HMM Topology for Shape Recognition. Nafiz Arıca, Fatoş Yarman-Vural. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP'99), Antalya TURKEY, pp. 162-168, 1999.

A New Scheme for Off-Line Handwritten Connected Digit Recognition . Nafiz Arıca, Fatoş Yarman-Vural. Proc. International Conference on Pattern Recognition (ICPR), Brisbane, Australia, pp.1127-1131, 1998.

One Dimensional Representation Of Two Dimensional Information For Hmm Based Handwritten Recognition. Nafiz Arıca, Fatoş Yarman-Vural. Proc. IEEE International Conference on Image Processing (ICIP) October, Chicago, U.S.A. 1998.

Off-Line Handwritten Connected Character Recognition. Nafiz Arıca, Fatoş Yarman-Vural. Proc. International Conference on Intelligent Processing Systems (ICIPS), pp.562-567, 1998.

Conceptual Data Modeling Of Multimedia Database Applications. Savaş Aygun, , Adnan Yazici, Nafiz Arıca . Proc. IEEE International Workshop on Multimedia Database Management Systems (IW-MMDBMS'98), Dayton, Ohio, pp.182-189, 1998.

Saklı Markov Model ile El Yazısı Tanımada Iki Boyutlu Bilginin Tek Boyutlu Sunumu. (in Turkish) Nafiz Arıca, Fatoş Yarman-Vural. Sinyal İşleme ve Uygulamaları Kurultayı (SIU'98), Ankara, Turkey, June pp.48-54, 1998.

Optical Character Recognition Without Segmentation. Mehmet Ali Ozdil, Fatoş Yarman-Vural, Nafiz Arıca. Lecture Notes on Computer Science, vol.1311, Springer, pp: 609-615, 1997.

HMM Based Handwritten Recognition. Nafiz Arıca, Fatoş Yarman-Vural. ISCIS XII, pp: 260-266, 1997.

Insan Optik Sisteminle Benzetilerek Gelistirilen bir El Yazisi Optik Karakter Tanima Sistemi. (in Turkish) Nafiz Arıca, Fatoş Yarman-Vural. Sinyal İşleme ve Uygulamaları Kurultayı (SIU'97), pp.810-816, 1997.