DISCONNECTED SKELETONS FOR SHAPE RECOGNITION

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

BY

ÇAĞRI ASLAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MAY 2005

Approval of the Graduate School of Natural and Applied Sciences

____

Prof. Dr. Canan Özgen
Director

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

____

Prof. Dr. 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 Master of Science.

____

Assoc. Prof. Dr. Sibel Tarı
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Cem Bozşahin (METU,CENG) ________________________

Assoc. Prof. Dr. Sibel Tarı (METU,CENG) ________________________

Assoc. Prof. Dr. Volkan Atalay (METU,CENG) ________________________

Dr. Ayşenur Birtürk (METU,CENG) ________________________

Dr. Mine Özkar (METU,ARCH) ________________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Çağrı Aslan

Signature :

ABSTRACT

DISCONNECTED SKELETONS FOR SHAPE RECOGNITION

Aslan, Çağrı
M. Sc., Department of Computer Engineering
Supervisor: Assoc. Prof. Sibel Tarı

May 2005, 97 pages

This study presents a new shape representation scheme based on disconnected symmetry axes along with a matching framework to address the problem of generic shape recognition. The main idea is to define the relative spatial arrangement of local symmetry axes in a shape centered coordinate frame. The resulting descriptions are invariant to scale, rotation, small changes in viewpoint and articulations. Symmetry points are extracted from a surface whose level curves roughly mimic the motion by curvature. By increasing the amount of smoothing on the evolving curve, only those symmetry axes that correspond to the most prominent parts of a shape are extracted. The representation does not suffer from the common instability problems of the traditional connected skeletons. It captures the perceptual properties of shapes well. Therefore, finding the similarities and the differences among shapes becomes easier. The matching process is able to find the correct correspondence of parts under various visual transformations. Highly successful classification results are obtained on a moderate sized 2D shape database.

Keywords: Shape representation, shape recognition, skeletons.

ÖZ

ŞEKİL TANIMA İÇİN BAĞLANTISIZ İSKELET

Aslan, Çağrı
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Sibel Tarı

Mayıs 2005, 97 sayfa

Bu çalışma şekil tanıma probleminin çözümü için bağlantısız simetri eksenlerine dayanan yeni bir şekil betimleme yöntemi sunmaktadır. Ana fikir yerel simetri eksenlerinin uzamsal düzenleşiminin şekil merkezli bir koordinat ekseninde tanımlanmasına dayanmaktadır. Betimlemeler ölçek ve yönelim farklarına, bakış açısındaki ufak değişikliklere ve eklemlemeye karşı değişimsizdir. Simetri noktaları düzey eğrileri kavislenmeye dayalı hareketi taklit eden bir yüzeyden çıkarılmaktadır. Gelişimdeki eğri üzerindeki düzleştirmenin arttırılması sayesinde şeklin sadece en belirgin parçalarına karşılık gelen simetri eksenleri çıkarılmaktadır. Yöntem geleneksel bağlantılı iskeletlerin kararsızlık problemlerine yakalanmamaktadır. Şekillerin algısal özelliklerini iyi bir şekilde yansıtmaktadır. Bu yüzden, şekiller arası benzerlik ve farklılıkların bulunması daha kolay olmaktadır. Eşleme işlemi çeşitli görsel dönüşümler altında şekillerin parçalarının karşılıklarını doğru bir şekilde bulmaktadır. Çeşitli şekillerden oluşan orta büyüklükteki bir veri tabanında yüksek oranda tanıma performansı elde edilmiştir.

Anahtar kelimeler: Şekil betimleme, şekil tanıma, iskelet

To my family

ACKNOWLEDGMENT

I would like to thank my supervisor Assoc. Prof. Dr. Sibel Tarı for her valuable guidance and insight. I enjoyed it a lot to work together with her throughout the research. I would also like to thank my wife, Simge, for all her love, support and technical assistance.

TABLE OF CONTENTS

PLAGIARISM…………………………………………………………………………… iii

ABSTRACT……………………………………………………………………………. iv

ÖZ……………………………………………………………………………………… v

DEDICATION…………………………………………………………………………. vi

ACKNOWLEDGMENT………………………………………………………………. vii

TABLE OF CONTENTS……………………………………………………………… viii

LIST OF TABLES…………………………………………………………………….. x

LIST OF FIGURES…………………………………………………………………… xi

CHAPTER

1. INTRODUCTION…………………………………………………………………… 1

1.1 Generic Shape Recognition………………………………………………………. 2

2. AXIS-BASED REPRESENTATIONS……………………………………………… 8

2.1 The symmetric axis transform (SAT) [2]…………………………………………… 8

2.2 Smoothed Local Symmetries [3], PISA [12] and Symmetry Set [4]……………….. 14

2.3 Voronoi Skeletons [17]…………………………………………………………… 16

2.4 The Work of Leymarie and Levine [11]..…………………………………………... 17

2.5 The Work of Rom and Medioni [20]…...…………………………………………… 19

2.6 Cores [5]…………………………………………………………………………….. 20

2.7 FORMS [27]………………………………………………………………………… 22

2.8 Shape Axis Tree [6]………………………………………………………………… 24

2.9 Shock Grammar and Shock Graphs [10,21,24,25]………………………………….. 25

2.10 The Method of Tari, Shah and Pien (TSP) [26]…………………………………… 29

2.11 Discussion………………………………………………………………………… 33

3.USE OF AXIS-BASED REPRESENTATIONS FOR RECOGNITION…………………. 36

3.1 FORMS [27]………………………………………………………………………… 36

3.2 Shock Tree [25]……………………………………………………………………… 37

3.3 Shape Axis Tree [6,13]……………………………………………………………… 39

3.4 Shock Graphs [21]…………………………………………………………………... 40

3.5 Discussion…………………………………………………………………………… 42

4. DETECTION OF SYMMETRY BRANCHES…………………………………………… 44

4.1 Detection of sym-points…………………………………………………………….... 44

4.1.1 New Surface Computation…………………………………………………………48

4.2 Computation of sym-branches…………………………………………………… 53
5. REPRESENTING SPATIAL RELATIONS……………………………………………56
5.1 The Canonical Coordinate Frame………………………………………………… 58
5.2 Spatial Organization of Symmetry Branches…………………………………….. 62
6. MATCHING AND RECOGNITION……………………………………………………65
6.1 Data Structure for Shape Matching……………………………………………..65
6.2 Shape Similarity………………………………………………………………….... 66
6.2.1 Score Computation………………………………………………………… 67
6.3 Matching Process……………………………………………………………….… 70
6.4 Experimental Results…………………………………………………………….. 71
6.4.1 Matching Examples……………………………………………………….. 71
6.4.2 Recognition Examples…………………………………………………….. 78
7. DISCUSSION AND FUTURE WORK………………………………………………… 84
REFERENCES.. 89

APPENDICES

A. IMPLEMENTATION DETAILS.. 91
A.1. Surface Computation…………………………………………………………… 91
A.2. Detection of Symmetry Points..…………………………………………………… 92
A.3. Detection of Symmetry Branches……………………………………………….. 93
A.4. Setting Up The Canonical Coordinate Frame ...…………………………………… 94
A.5. Describing Spatial Relations and Measurable Properties………………………….. 95
A.6. The Matching Process……………………………………………………...……. 96
A.7. Performance Results……………………………………………………………….. 96

LIST OF TABLES

Table 6-1 Data structure for shape matching 66

LIST OF FIGURES

Figure 1-1 Some silhouettes that are easily recognized. 3
Figure 1-2 Two shapes that represent the same entity.......... 4
Figure 1-3 Symmetric axes [2] of the shapes in Figure 1-2. 5
Figure 1-4 An example of occlusion 6
Figure 2-1 The propagation of the fire front on a rectangle. 9
Figure 2-2 The distance map of a rectangle. 9
Figure 2-3 (Taken from [2]) The sym-transform 10
Figure 2-4 Types of sym-points. 11
Figure 2-5 (Taken from [2]) Flexure operation. 12
Figure 2-6 (Taken from [2]) A first level of shape classification based on sym-ax connectivity.......... 12
Figure 2-7 (Taken from [2]) Equivalent shapes with same directed sym-ax topology.......... 13
Figure 2-8 The instability problem of Blum's symmetric axis formulation.......... 14
Figure 2-9 (Taken from [9]) Location and participation choices for recording sym-points.......... 15
Figure 2-10 (a) (Taken from [20]) SLS (b) (Taken from [12]) PISA.......... 15
Figure 2-11 The hierarchical skeleton extracted from a leaf shape. 17
Figure 2-12 (Taken from [11]) Example of a potential surface.......... 18
Figure 2-13 Example of active contour evolution on a rectangle.......... 18
Figure 2-14 (Taken from [20]) SLS 19
Figure 2-15 Decomposition of a shape into parts. 19
Figure 2-16 (Taken from [5]) Boundaryness detectors and their connections. 21
Figure 2-17 "Cores" computed from a rectangle shape with one saw-tooth edge.......... 21
Figure 2-18 (Taken from [5]) A shape and its cores at largest scale 22
Figure 2-19 Primitives and mid-grained parts.......... 23
Figure 2-20 The range angle function computation 23
Figure 2-21 A dog figure and its parts. 24
Figure 2-22 The shape-axis tree computation for different values of jump cost. 25
Figure 2-23 (Taken from [10]) The effect of curvature dependent motion on noise elimination.......... 26
Figure 2-24 (Taken from [10]) The reaction-diffusion space for a human shape 27
Figure 2-25 (Taken from [25]) Four types of sym-points used in [25] 28
Figure 2-26 (Taken from [24]) The effect of diffusion on the extracted sym-branches.......... 28
Figure 2-27 The distance surface and the function 1-v for a rectangle 30
Figure 2-28 The motion from one level curve to the next.......... 30
Figure 2-29 (a) Surface v near an elliptic point (b) Surface v near an hyperbolic point 32
Figure 2-30 Sym-points of the duck shape with level curves superimposed.......... 33
Figure 3-1 The skeleton operators from top to bottom: cut, merge, shift and concatenate 37
Figure 3-2 (Taken from [21]) (a) An example shape (b) Its shock tree description.......... 38
Figure 3-3 (Taken from [13]) Two shapes and their corresponding shape axis trees.......... 39
Figure 3-4 (Taken from [21]) (a) An example shape and (b) its shock graph representation 40
Figure 3-5 (Taken from [21]) A low cost is assigned to the deformation that transforms A to B.......... 41
Figure 3-6 (Taken from [21]) Some matching results. 41
Figure 3-7 (Taken from [21]) Matching results for some query shapes. 42
Figure 3-8 (Taken from [21]) Two shapes with the same topological structure. 43
Figure 4-1 A rectangle shape and its minimizing surface 45
Figure 4-2 A vase shape and the results of sym-point detection using TSP.......... 45
Figure 4-3 A second vase and the results of sym-point detection using TSP.......... 46
Figure 4-4 Sym-point detection using TSP formulation on a segmented MRI image.......... 47
Figure 4-5 The sym-points of the segmented MRI image.......... 48
Figure 4-6 A duck shape and the results of initial diffusion step 49
Figure 4-7 The results of the final diffusion step and the sym-points of the duck shape 50

Figure 4-8 The level curves after initial and final diffusion and the computed sym-points 51
Figure 4-9 Sym-point computation of a dog-bone shape 51
Figure 4-10 Sym-points of two hand shapes 52
Figure 4-11 Full sym-points of some shapes with significant necks 53
Figure 4-12 Positive an d negative sym-points of a turtle shape. 54
Figure 4-13 Colored sym-branches of some sample shapes after pruning. 55
Figure 5-1 (Taken from [16]) The hierarchy of shape information in Marr and Nishihara's model.... 57
Figure 5-2 The major sym-branches of some example shapes. 58
Figure 5-3 The states of the hand shape at the times minor sym-branches terminate. 59
Figure 5-4 Four possible reference axes of the hand shape 60
Figure 5-5 The major sym-branches of (a) the hand shape and (b) a human shape. 60
Figure 5-6 The instabilities associated with reference axes that can lead to matching failures 61
Figure 5-7 The possible coordinate frames of a shape whose symmetry is more than two-fold. 62
Figure 5-8 Possible reference axes for a dog-bone shape. 62
Figure 5-9 Sym-branches of the hand shape after the cut operation. 63
Figure 5-10 The reference axes and the position vectors of the sym-branches of the hand shape. 63
Figure 6-1 (a) Square window function (b) Similarity function inside the window 68
Figure 6-2 The probability of similarity based on the normalized length feature. 69
Figure 6-3 Robustness under scale difference. 71
Figure 6-4 Robustness under rotation. 72
Figure 6-5 Examples of matching under articulation. 73
Figure 6-6 Robustness under small boundary perturbations. 74
Figure 6-7 Matching in the case of missing parts 75
Figure 6-8 Correspondences between shape pairs belonging to different categories. 76
Figure 6-9 Unintuitive correspondences of the matching scheme 77
Figure 6-10 The silhouette database used in recognition experiments 78
Figure 6-11 Some query results. 79
Figure 6-12 Some query results. 80
Figure 6-13 Some query results. 81
Figure 6-14 Some query results. 82
Figure 7-1 A stroke shape, a shape with a hole and their full symmetry axes 85
Figure 7-2 A rectangle and its sym-points 86
Figure 7-3 Two recognition tasks 87
Figure 7-4 The concept of class skeleton. 88
Figure A-1 The effect of the size of the image on the time it takes to compute the descriptions 96

CHAPTER 1

INTRODUCTION

Our visual system relies on many visual cues to interpret the world it sees. Prior experience is also believed to be involved in the process of inferring complex shape properties and deriving the most probable interpretation of the world in terms of the descriptions stored in memory. Shape has been considered different from other visual cues. It is sufficient for recognition most of the time. We are able to recognize thousands of objects from their silhouettes which contain only shape information. Other visual cues such as color, texture and motion are attributes of an image whereas shape is an attribute of a region and it requires a segmentation step before computing it. These properties of shape indicate that it is very important for recognition and it is the basis of internal descriptions which probably influence the processes of early vision. Because of the key role of shape information in the overall perception process, shape representation and recognition have been central research topics in computer vision.

There have been innumerous works on the representation and recognition of shapes. From the heavily numerical schemes to the structural ones, many different representations have been proposed. They differ in the aspects of shapes that they make explicit and those aspects that they push into the background. Most of these representation schemes are suited for narrow shape domains where there is limited and predictable variability of input data. They have their own associated recognition algorithms and their own set of results, independent of other schemes [18]. Generic object recognition demands representations that can capture the large degree of variability as a result of changes in illumination, viewpoint direction, rotation, scale, articulation etc. Many researchers have tackled with generic object recognition problem and tried to identify the requirements of shape representation schemes that can be used for generic object recognition. In particular, the idea of decomposing the shape into primitives and building up a description of the shape by a frame that expresses the relations among these primitives was first made explicit by Marr & Nishihara [15] and has been one of the most promising guidelines for recognition. Representation schemes based on symmetry axes have been considered in this respect because of their ability to capture the perceptual properties of shapes.

Axis-based representations entered the literature with the symmetric axis transform (SAT) of Blum [2] and have been well studied since then. Only a limited amount of information about a shape is captured by these representations but that information is very useful for recognition. The idea of an axis-based representation is appealing, but it has been a major issue to obtain accurate and stable descriptions of shapes using this kind of representation. Over the years, although many different techniques have been suggested, no stable axis-based representation scheme has been devised. There are some recognition studies based on axis-based representations but the inherent instabilities of the representations prevent them from being used in practical recognition tasks.

The goal of this thesis is to design a new axis-based representation scheme which provides stable descriptions of shapes that can be used in a recognition framework. The main contributions are:

- A new symmetry point detection scheme which extracts the most prominent symmetry axes of a shape is proposed;
- A new method to set up a canonical coordinate frame, relative to which the metric properties are measured, is offered;
- Unlike the conventional approach of using connected symmetry axes, a representation that is based on disconnected symmetry axes is proposed.

The organization of the thesis is as follows: In the remaining of this chapter, an overview of the generic shape recognition problem is given and some requirements of shape representations are derived which guide the design of the representation scheme proposed in this study. Chapter 2 and 3 cover the important axis-based representation schemes and matching frameworks presented mostly in the last decade. The result of the review highlights the common approach taken, the problematic areas in the implementation of ideas and reveals the necessary steps towards better representation schemes. The representation scheme proposed in this work is presented in chapter 4 through 6. Specifically, chapter 4 is on the extraction of local symmetry axes of a shape. The very important issue of axis regularization is explained there. In chapter 5, we explain our method to describe the spatial relations of local symmetry axes of a shape using an object centered reference frame. The design of the representation is completed in this chapter. In chapter 6, the matching framework based on the new representation scheme is explained, and several matching and recognition examples are presented. Finally, chapter 7 discusses the presented work and highlights the issues remaining to be solved in future.

1.1 Generic Shape Recognition

The purpose of this section is to give common definitions that will be used throughout this thesis and to present an overview of the addressed problem. In particular, we give the requirements of shape representation schemes and discuss why the axis-based representation idea is promising.

A *representation* for a shape is a formal scheme for describing that shape or some aspects of it together with rules that specify how the scheme is applied to any particular shape [14]. A *description* of a shape is the result of using a representation scheme on that shape [14]. For instance, if the moments of a shape are used as features, the formula to compute the moments is part of the representation whereas the scalar value obtained from the shape using the formula is its description. *Shape Matching* refers to the problem of determining the correspondence between two shapes. *Shape Recognition* refers to the problem of determining the category of an unknown shape. Generic shape recognition implies that there is no restriction on the variability and nature of shapes in the application domain.

Our visual system is faced with the problem of choosing the most probable interpretation of the world among the many possible. It is believed that our visual system makes use of many different visual cues such as color, shading, shape, motion, texture and prior experience to arrive at a useful interpretation. When faced with such an extremely complex problem which is thought to be solved by using different visual cues, it is natural to try to divide this problem and study its basic parts separately. In shape recognition problem, our goal is to understand how objects are recognized based solely on shape information and to create programs that can carry out such a task. The motivation behind this problem is that it is believed that shape information plays a crucial role in the recognition process. As Figure 1-1 shows, we are quite good at recognizing these objects from their silhouettes or primitive drawings. Since there is no color, shading, motion or texture information that can be extracted from these images, the recognition problem can only solved by relying on shape information.

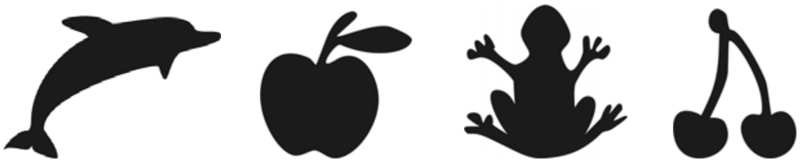


Figure 1-1 Some silhouettes that are easily recognized.

There are some general requirements of generic shape representation schemes. First, the shape domain has unconstrained and unpredictable variability. A representation for generic shape recognition should be able to capture this large degree of variability in the shape domain. It should provide similar descriptions under visual transformations such as translation, scale, rotation, viewpoint direction, articulation etc. Second, perceptually similar objects should have similar descriptions in the representation. This is known as the stability requirement. Third, the descriptions should be unique. Otherwise, different shapes may have similar descriptions. In order to satisfy the opposing requirements of stability and uniqueness, hierarchical schemes such as scale-space representations are considered in which a shape is described at different levels of detail [9,10,16,19,25].

Existing shape representation schemes may be classified according to many different criteria. The paper by Loncaric [13] presents a good overview and a classification of existing representations. Most of the representation schemes do not satisfy the stability requirement and are not robust under visual transformations. Consider a representation task for describing the shapes in Figure 1-2. We perceive these shapes as representing the same entity, therefore, a representation scheme is expected to capture their equivalence.

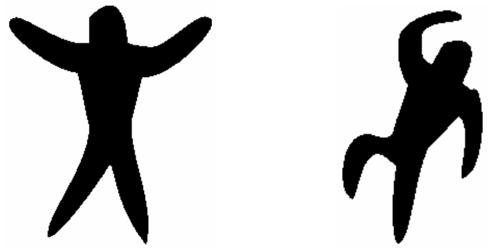


Figure 1-2 Two shapes that represent the same entity

If shapes are represented by global descriptors such as moment or Fourier transform based approaches, articulated shapes belonging to the same category have usually different descriptions. Likewise, when the boundary of a shape is transformed to a one-dimensional function (e.g. Fourier descriptors), the signature of the boundary remains the same under simple transformations such as rotation and scale, but not in the case of articulation and other non-linear transformations. Moreover, the important two-dimensional properties of the shape such as region information, symmetry, object parts are pushed into the background. Local features of the boundary such as curvature have also been used to describe shapes. In this type of representation, global information such as the relations of shape parts is not expressed. If we used curvature information to describe the shapes in Figure 1-2, their description would be different because their local features are different (the curvatures at the points where the arms and legs connect to the torso are different).

The main advantage of using axis-based representations is that rather than the geometric properties they capture the perceptual properties of shapes. Consider Figure 1-3 where the symmetry axes of the human shapes in Figure 1-2 are shown. Each of the parts of the human shapes is made explicit in the symmetry axis description. The descriptions are similar since the organization of the symmetry axis segments is similar. Another advantage is that axis-based representations capture both region and boundary information. If the original SAT of Blum is used, the shape boundary is precisely obtained. Also, Leyton [11] proved that there is a symmetric axis for each curvature extrema along the shape boundary. The region information can be retrieved from the fact that width information is stored in symmetry points.

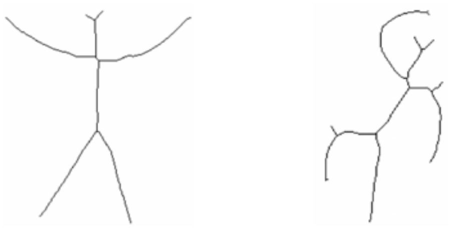


Figure 1-3 Symmetric axes [2] of the shapes in Figure 1-2.

The main difficulty with the shape representation research is the lack of appropriate criteria to evaluate the quality of representations. The difficulty with devising such a common criteria for shape representation is that the shape recognition problem, although a subproblem of vision, is very broad and complex. We do not have a detailed answer to the question "What a shape recognizer is supposed to do?". David Marr was very influential to show the correct style of approach to problems of vision. According to Marr [14], analysis of a problem should include a clear understanding of what is to be computed, why it is computed and how the computation is to be done and the physical assumptions on which the method is based. The style of approach he suggested, focused on defining the problem at hand precisely rather than on the design of algorithms. Without a precise definition of what needs to solved, we:

- Create ad hoc solutions rather than generic ones,
- Fail in deriving criteria to judge the effectiveness of solutions,
- Fail in making accurate comparisons of solutions.

"Recognizing shapes as well as humans do" is neither a precise definition of the problem nor a useful one. It does not define the sub problems that need to be solved and it does not provide us with appropriate criteria to evaluate the effectiveness of methods: Unless an algorithm performs comparably to humans, it should be considered ineffective. The number of shapes that can be recognized by a system also cannot be a criterion. Unless the number of shapes recognized by the system is very large, we cannot determine if the method is ad hoc or not. We can not know if the method will perform better when additional information such as texture, color is incorporated into a recognition system. Therefore, we have to know precisely what is computed and why. The answers to these questions must be a set of constraints and requirements that collectively define the problem precisely. According to Marr's computational point theory point of view, we should decide what is computed and why.

In this thesis, as in many other representation studies, it is assumed that the shape boundary has been previous extracted. Since segmentation is an issue which remains largely unsolved, it may be argued that segmentation must be dealt with before shape representation and recognition problem is considered. This argument inherently assumes a strictly bottom-up processing model in the visual system. If this is the case, then research in shape analysis has to wait for the developments in segmentation to develop a desired vision system. If the computation in a visual system includes top-down processing, then findings of shape analysis research must be used in segmentation research. The question of whether the visual system uses a bottom-up or top-down processing model remains unclear. It is mostly believed that both kinds of information transfer occur in a visual system.

The second restriction on the shape domain is that no occlusions of shape parts are permitted. Therefore, we assume that all the parts of shapes are visible in their normal proportions. Figure 1-4 shows an example of a shape which the system designed in this thesis is not expected to handle.

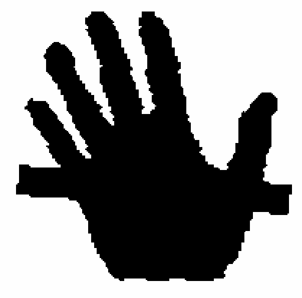


Figure 1-4 An example of occlusion

We believe that the information needed to resolve the difficulty encountered in the case of occlusions must be supplied by other visual processes. One motivation for using silhouettes is that it allows us to divide the vision problem by studying individual perceptual processes. Many issues need to be taken into account when we perform such a division and work on a constrained space. We cannot just assume that each part of a system works alone and in isolation from the rest of the system. We must precisely know the connection between the examined subsystem and the rest of the system. In particular, we must identify what kind of information that the subsystem receives and what kind of information that the subsystem outputs. Failing to identify such information leads to:

- Methods that rely on the properties of narrow shape domains and that fail to work on large shape domains (ad hoc methods);

- Methods that try to derive information which should be supplied from the outside by using only the tools available to the subsystem.

In Figure 1-4, the contours between the hand shape and the stick are not available. In fact, we do not know which shape is closer to us. Other perceptual clues such as T and Y junctions may provide us this information. The reason we are interpreting this image as consisting of a hand shape and a stick rather than an unknown single shape is prior experience. It is a common approach in literature to obtain descriptions of occluded shapes without relying on any previous information. Such methods may work on some instances but they are far from being generic approaches.

The same kind of argument is applicable for large viewpoint variations. A shape representation scheme that does not rely on prior experience or external information should not be expected to give similar descriptions when a shape is viewed from different directions. The loss of information in the depth dimension makes it impossible to derive the correct interpretation of an object from only its silhouette.

CHAPTER 2

AXIS-BASED REPRESENTATIONS

In the second part of the previous chapter, shape recognition problem was analyzed and some requirements and restrictions were derived which guide the design of the representation scheme developed in this thesis. Since the invention of the symmetric axis transform by Blum, it was realized that axis-based representations are better suited for the generic recognition task than the other types of representations. They capture the perceptual properties of shapes, which is necessary for recognition. Many generic shape representation schemes have been proposed that involve directly or indirectly an axis about which the shape is symmetric. In this chapter, some of these important representations are reviewed. First, the symmetric axis transform idea of Blum is explained in detail to set up the terminology and to identify the main issues related to this shape representation. The implementation of Blum's idea on a computer has been very difficult and much of the earlier research is focused on this issue. Matching and recognition applications based on axis-based representations entered the literature only in the last decade. Therefore, the remaining of the review is mainly focused on these more recent studies. At the end of the section, the advantages and disadvantages of each work are summarized and compared. Although the axis-based representations reviewed in this section were mainly introduced for recognition, only some of them have been used in recognition frameworks. Those works are reviewed separately in the next section.

2.1 The symmetric axis transform (SAT) [2]

Blum introduced the first axis-based representation of 2D shapes called the *medial axis* or *symmetric axis*. In this representation, the description is shifted from the boundary to the interior of a shape by using as the primitive, a disc or a growth of a point. His model can best be understood by the grassfire analogy (also called prairie fire model) in which the interior of a shape is thought to be filled with dry grass and a fire is started simultaneously on every point on the boundary. The fire front propagates with constant speed, and at any time, all the points on the fire front are equidistant from the boundary. The symmetry points of the shape are the singularities (shocks) of the advancing fire front and the locus of these points constitutes the symmetric axis of the shape. By keeping track of the time when these singularities are formed, it is possible to obtain the shape boundary precisely by inverting the process. Figure 2-1 illustrates the propagation of the fire front that is started on the boundary of a rectangle. The four sides of this rectangle can be considered as four distinct fire fronts. The symmetric axis is the collection of points where two or more fire fronts meet (shown by the thick line).

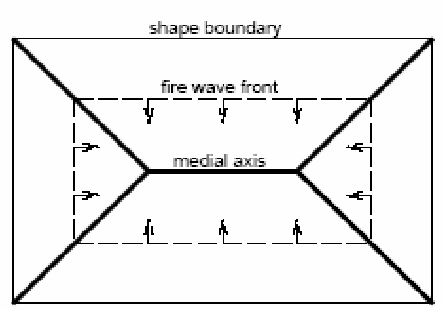


Figure 2-1 The propagation of the fire front on a rectangle.

The space-time plot of the fire front propagation for the rectangle above can be constructed by considering the rectangle surface as the input plane and the time when the advancing fire front passes through a point as the orthogonal z coordinate. The surface obtained in this way is called the *distance map*. In the case of a rectangular shape, it is a pyramid ending with a ridge at the top. The level curves of the distance map describe the propagation of the fire front and the projection of the singular locus of the surface forms the symmetric axis. Figure 2-2 shows the distance map of a rectangle. Notice that only the distance values of the interior points are plotted.

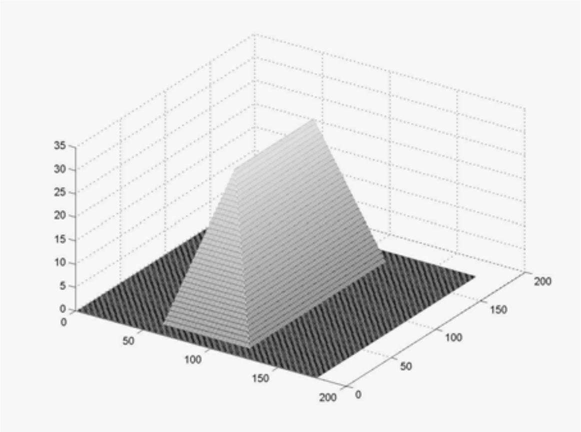
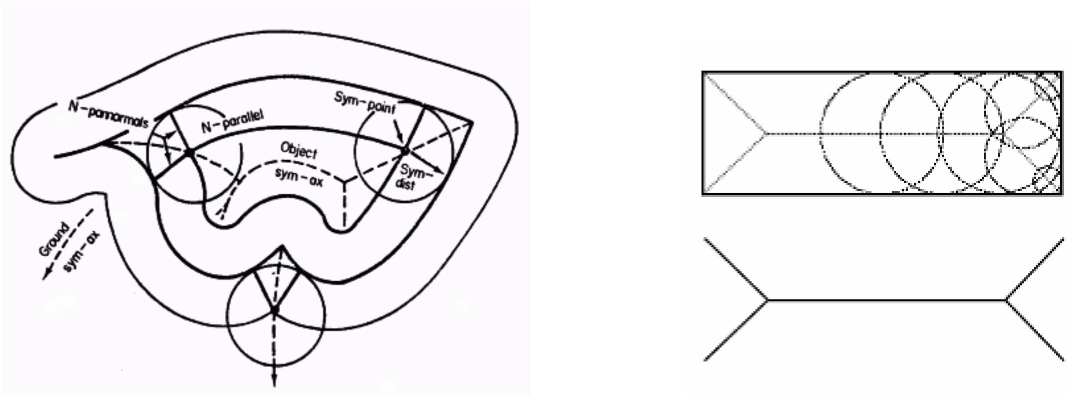


Figure 2-2 The distance map of a rectangle.

In precise mathematical terms, the symmetric axis can be defined using the maximal disc formulation (Figure 2-3). A *symmetry point* (sym-point) is the intersection of two or more pannormals with equal length (a pannormal is the shortest line drawn from a point to the boundary) or the center of a maximal disc that touches the shape boundary in more than one point. The locus of sym-points constitutes the *symmetric axis* (sym-ax). The shortest distance from a sym-point to the shape boundary (pannormal length) is the *symmetric point distance* (sym-dist). Sym-dist corresponds to the time value in the grassfire model and the radius of the maximal disc inscribed in the interior of a shape. The sym-ax with associated sym-dist at each point is the *symmetric axis function* (sym-function). In literature, the terms symmetric axis function, medial axis transform (MAT) and symmetric axis transform (SAT) are used interchangeably. In Blum's model, a symmetry axis can also exist in the ground. Because the interior sym-ax of an object completely describes that object, the ground sym-ax is ignored throughout this work. The *object* is the union of discs of sym-dist radius on each sym-point of the sym-ax.



(a) **(b)**

Figure 2-3 (a) (Taken from [2]) The sym-transform. (b) The visualization of the maximal disc formulation on a rectangle.

Since its introduction, the implementation of SAT in a computer has been a challenge. Much of the research has been focused on accurate extraction of the symmetric axis in the discrete domain. Some methods are based on the maximal disc formulation, whereas some others simulate the fire propagation. Probably due to difficulties associated with a robust implementation of SAT, more crucial ideas presented in Blum's 80 pages long paper are forgotten. The symmetric axis transform does not only provide a static, stick figure view of a shape. Blum proposed a way to derive the properties of a shape directly from its symmetric axis by using the properties of sym-points.

Symmetry points or shocks in the grassfire model can be classified by the properties of their pannormals or their maximal disc. The classification of sym-points is made based on the number of intervals and points their maximal disc touches the shape boundary and their sym-dist. If a sym-point touches the shape boundary in n distinct locations or intervals, it is categorized as an n-sym-point. The categorization based on sym-dist is made by observing whether the sym-dist of a sym-point is a strict maximum, a strict minimum or constant when compared to those of neighboring sym-points. In the grassfire analogy, this denotes whether the fire burns into the points, away from the points or burns at the same time in the neighborhood. Sym-points that are a strict maximum in their neighborhood are *bulb* points. A 1-sym-point with strict minimum is a *sprout* point. A 2-sym-point with strict minimum is called a *pinch* point (a neck point). Points at which the sym-dist is constant are called *worm* points. Sym-points at which the sym-dist increases along some axes and decreases along others are called *fork* points (a bifurcation of the sym-ax). The sym-points whose sym-dist is neither minimum nor maximum form a differentiable curve. These points are called *smooth sym-points*. Blum did not propose a special name for the collection of smooth sym-points although the differentiable curves formed by such points are important elements in skeletal descriptions. These curves will be called symmetry branches or *sym-branches* throughout this study. The smooth sym-points are the only type of sym-points that can exist over an interval, all others being isolated discrete points. The change of the sym-dist along a sym-ax allows the notion of direction to be assigned to a sym-ax. A *directed sym-ax* is defined in the direction of increasing sym-dist. These different types of sym-points are illustrated in Figure 2-4.

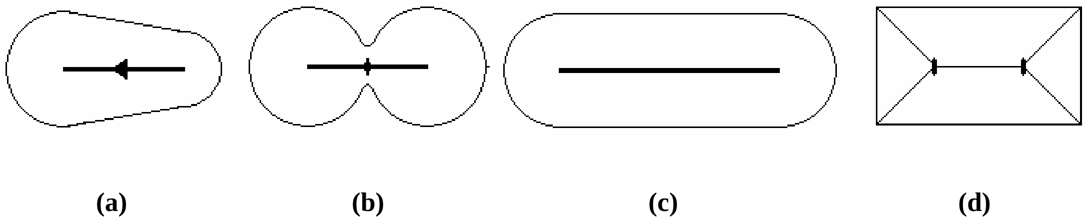


Figure 2-4 (a) A directed sym-ax. This sym-ax consists of a collection of 2-sym-points and two 1-sym-points (a bulb on the left end of the sym-ax and a sprout at the right end of the sym-ax). These points collectively form a differentiable curve. (b) A pinch point (marked at the center of the smooth sym-points). Note that there are also two bulb points on the sym-ax (c) Worm points. (d) Fork points (marked).

Curvature is traditionally defined by a change in tangent or normal angle with respect to an arc length. The axis curvature or asymmetric curvature is defined using a sym-ax. A projected boundary curvature is the angular rate of change of boundary normal with respect to sym-ax arc length. Since a smooth sym-point has two contact points on the boundary, two boundary curvatures may be defined for such a point. The *axis curvature* is the average of the projected boundary curvatures along its normal. It is, therefore, the average amount the projection of both boundaries curve in the same direction. The *symmetric curvature* or *width curvature* is the difference between both projected boundary curvatures and corresponds to the rate at which the object angle is changing symmetrically about the sym-ax. The axis curvature and symmetric curvature terms are used to explain the symmetry and the flexure of a shape. A symmetric object is one whose axis curvature is zero at all smooth points. Flexure (Figure 2-5) is an operation which changes axis or asymmetric curvature while maintaining the sym-dist and consequently the symmetric curvature. Blum showed that flexure has the properties that area, perimeter and integral boundary curvature are invariants of the operation.

This is a very important result which shows that the symmetric axis description remains the same under articulation.

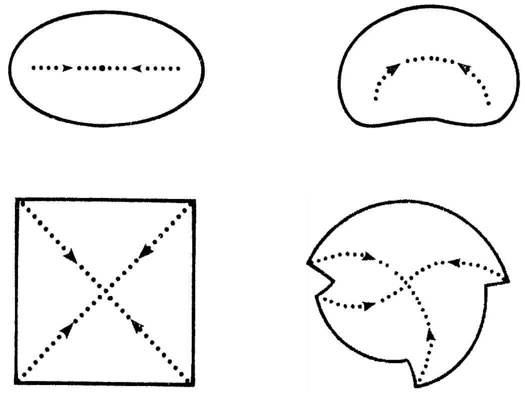


Figure 2-5 (Taken from [2]) Flexure operation.

The main motivation of Blum was to devise a representation that captures the “softer”, perceptually important properties of shapes rather than the traditional geometric properties such as congruence, area, perimeter etc. Using the properties of points of the sym-ax, Blum proposed a new category of shape properties which are collectively called *axis morphology.* The main idea is based on setting up equivalence classes in the shape domain. If two objects are in the same class, they are considered as equivalent. The classes can also be divided into subclasses to obtain a hierarchical grouping of shapes.

The sym-points on the sym-ax are classified as smooth and non-smooth points. By distinguishing non-smooth points on the sym-ax and disconnecting the sym-ax at non-smooth points, an undirected graph is obtained. Finite connected graph pieces represent finite connected parts. A loop in the graph represents a hole in the object and a tree represents an object without holes. Therefore, the topology of the object is given in sym-ax connectivity and it can be used to define a first level of shape equivalence classes. At this step an ellipse and a shape with a pinch point are considered as equivalent (Figure 2-6).

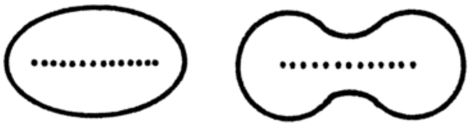


Figure 2-6 (Taken from [2]) In a first level of shape classification based only on the sym-ax connectivity, an ellipse and an object with a pinch point are considered as equivalent.

The additional properties derived from the symmetric axis can be used to discriminate the shapes in the same class and to define equivalence subclasses. For instance, the direction of the sym-ax can be used to construct a directed graph description of shapes. The ellipse and the shape with the pinch point are now considered different. The sym-branches of the ellipse flow towards the shape center whereas in the shape with the pinch point, sym-branches flow away from the shape center (pinch point). Some example shapes that are in the same equivalence classes based on the directed graph representation are shown Figure 2-7.

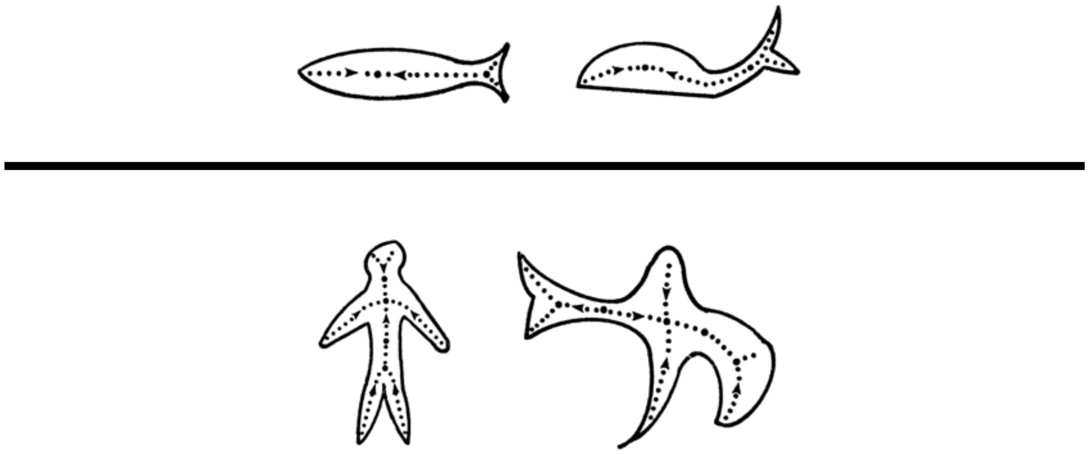


Figure 2-7 (Taken from [2]) Equivalent shapes with same directed sym-ax topology.

The ideas presented by Blum laid the foundation of a new approach to shape description and recognition. Along with the new shape representation scheme, he proposed methods for recognition of shapes based on their symmetric axis descriptions. His representation scheme makes explicit the perceptually important properties of shapes and pushes into the background the traditional geometric properties. The traditional numerical schemes nearly do the opposite: Traditional geometrical properties are made explicit and used. This is why the symmetric axis transform has been the most popular and most studied global space-domain method [13].

The implementation of Blum's ideas by a computer was not straightforward. There are innumerous studies on the subject. For many years, the main tool for the computation of medial axis in the discrete domain was morphological thinning. In morphological thinning, the object boundary is eroded successively and uniformly until its skeleton is retrieved. Special procedures are employed in the process to ensure that the removal of an object pixel does not alter the topology of the thinned shape. Analytic computation of sym-points based on polygonal approximation of shape boundaries has been also proposed. Another earlier class of algorithms is based on ridge following on the distance map. These earlier approaches will not be investigated in this study because they are superseded by newer algorithms (explained in this section) that offer more accurate and efficient computation.

Apart from the difficulty of its implementation, the symmetric axis transform has a major drawback for recognition: A small change in the shape boundary may yield a large change in its symmetric axis. This violates the stability requirement of representations. The classical example of a rectangle illustrates this phenomenon well (Figure 2-8(a)). Another common instability results from a small change in the location of an object part (Figure 2-8(b)). This small change leads to a change in the location of fork points and alters the topology of the shape significantly.

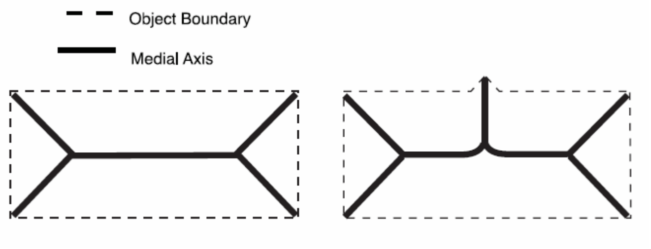


Figure 2-8 The instability problem of Blum's symmetric axis formulation. A small bump on the boundary of the rectangle causes a major change in its sym-ax.

The crucial point to make about the instability problem of the symmetric axis transform is that it is an information preserving representation. The sym-ax and the sym-dist at each sym-point allow the reconstruction of the exact object boundary. This means that even the smallest boundary details are preserved in the representation. An information preserving representation is not necessary for recognition. In fact, rich descriptions prevent loose grouping of shapes into equivalence classes. A variety of techniques have been suggested to overcome the instability problems of the symmetric axis transform: new definitions of symmetry have been proposed, sym-ax is pruned to discard those branches that correspond to unimportant details, shape boundary is smoothed to discard noise etc.

2.2 Smoothed Local Symmetries [3], PISA [11] and Symmetry Set [4]

After Blum's symmetric axis transform, several other definitions of symmetry entered the literature such as SLS [3], PISA [11] and Symmetry Set [4]. The motivation behind these attempts is to obtain a symmetry representation that has more desirable features than the symmetric axis such as stability. The paper by Jenkinson and Brady [8] presents a good overview of these different definitions of symmetry. The difference mainly results from two choices: location and multiplicity. Location is the position chosen to record a sym-point. Multiplicity refers to how many times a segment of the boundary can participate in forming symmetries. As Figure 2-9(a) shows, the position of symmetry point can be recorded at the center of the maximal disc (U), at the midpoint of the chord (Q), or at the midpoint of the arc (P). For multiplicity, there are two options: single participation and multiple participation. Figure 2-9(b) illustrates the notion of multiplicity. In the first shape, the boundary point B participates only once in forming symmetries. Single participation imposes that all bitangent circles are inscribed. In the multiple participation case (shape on the right), the boundary point B is used more than once for sym-point detection. Notice that the larger disc is not inscribed in the shape.

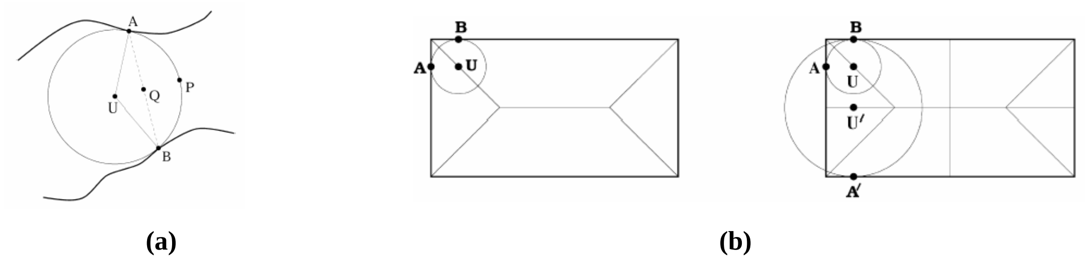


Figure 2-9 (Taken from [8]) (a) Three locations to record a sym-point: U – center, Q – mid-chord, P – mid-arc (b) Single and multiple participation

According to location and multiplicity choices, the different definitions of symmetry can be classified as follows: The Symmetric Axis Transform [2] of Blum uses center point and single participation. Smoothed Local Symmetries [3] of Brady and Asada uses mid-chord point and multiple participation. Process Inferring Symmetric Axis [11] of Leyton uses mid-arc point and single participation. The Symmetry Set [4] of Bruce and Giblin uses center point and multiple participation. It can be seen that the symmetric axis is a subset of the symmetry set.

The motivation behind recording sym-points on different locations of the disc is to eliminate the excessively long branches corresponding to the small perturbations of the shape boundary in the original SAT. Using multiple participation, additional important symmetries may be captured but many other redundant ones are also generated. Among these symmetry definitions, we cannot assert that one of them is significantly better than the others. Figure 2-10(a) shows the full SLS axes of an example shape. The representation hides, rather than makes explicit, the prominent branches of the shape. An example of PISA [11] of Leyton is shown in Figure 2-10(b). Using single participation and mid-arc point, PISA is a disconnected symmetric axis.

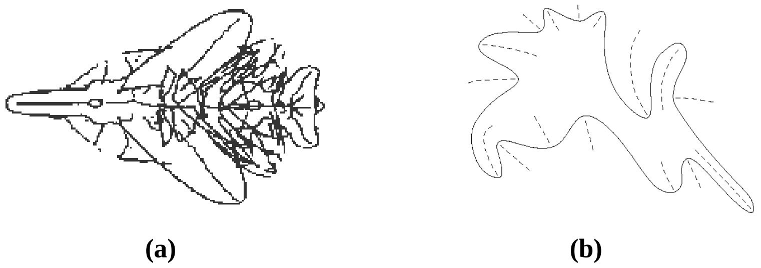


Figure 2-10 (a) (Taken from [19]) Full SLS axes of a shape (b) (Taken from [11]) PISA of a shape

2.3 Voronoi Skeletons [16]

Ogniewicz [16] used the concept of Voronoi diagram to extract the symmetric axis in a geometrically correct way in the discrete domain and investigated various pruning methods to regularize it. The discrete analogue of the SAT is to approximate the shape boundary by a set of points and let each boundary point initiate a circular fire front. The locus where the fronts meet is the Voronoi diagram of the shape. Excluding the Voronoi edges between neighboring boundary points, a good approximation to the symmetric axis is computed which is called the *discrete Voronoi medial axis* (DVMA) or Voronoi skeleton. As the boundary samples increase, the DVMA becomes a more accurate approximation of the original medial axis. Since it is a good approximation, a Voronoi skeleton shares the instability problems of the medial axis. Without any regularization, the skeleton contains many branches due to noise. Traditionally, pruning of the symmetric axis has been mostly used to regularize it. Pruning methods define a saliency measure for axis points and discard those axis points whose significance is below a predefined threshold. Variability in significance measures is the major source for variability in pruning methods. Among the significance measures used are axis length, propagation velocity of the symmetric axis in the prairie fire model, maximal thickness of an axis, the length ratio of the axis and the boundary it unfolds. The paper by Shaked and Bruckstein [21] reviews and classifies the different pruning methods used in the literature.

In [16], Ogniewicz and Kubler offered three saliency measures: the length of the boundary segment between the two points of the maximal disc (potential residual), the ratio of this boundary segment and the perimeter of the maximal disc (circularity residual) and the difference between the length of the boundary segment and its replacement (chord residual). However, the pruning procedures are not sufficient for eliminating the spurious branches when the shape boundary is jagged. These branches cannot be removed by increasing the threshold because other prominent branches may be eliminated in the process as well. As reported in [16], the main reason is that no distinction can be made between sym-branches that are due to globally salient shape features and those that are due to small boundary details. The hierarchy of skeleton branches (skeleton pyramid) is established by using an algorithm that assigns a rank order measure to skeleton branches. By generating different hierarchy levels of the skeleton branches, a scale-space representation is obtained. In Figure 2-11 the hierarchical skeleton obtained from a leaf shape is shown. The first order skeleton represents the highest scale.

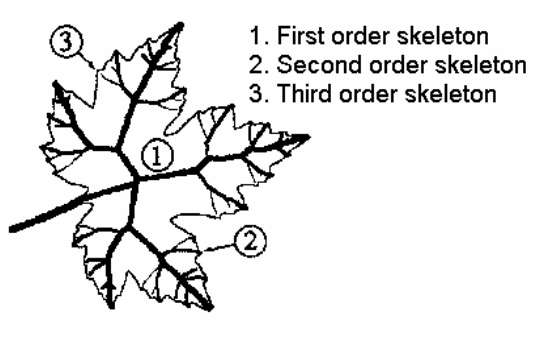


Figure 2-11 The hierarchical skeleton extracted from a leaf shape.

Two main issues need to be considered for Voronoi skeletons. First, the complexity and accuracy of the representation depends on the discretization of the boundary curve. If the resolution provided by the image grid is used, an enormous number of boundary points are obtained. On the other hand, increasing the sampling interval leads to a loss of accuracy especially if the shape boundary is jagged. The second issue is pruning. In [21] and [16], in-depth analysis of pruning methods are given. The pruning problem remains largely unsolved. The main problem is that the computed saliency values of axis branches do not reflect the perceptual prominence of parts of shapes. Moreover, some pruning approaches cannot be used because of the danger of disconnection of the sym-ax. For instance, Blum proposed the propagation velocity of the symmetry axis in the grassfire model as a saliency measure. It is not used in practice because pruning based on this significance measure leads to disconnected axes. To capture the topology of a shape, it has been forced that the sym-ax of a shape must be connected.

2.4 The Work of Leymarie and Levine [10]

Among the works we review, this study is the first example of a symmetric axis extraction method based on simulating the prairie fire model. The researchers used an active contour to model the fire front. In active contour model (also called snakes), a curve is approximated by a collection of points. According to an energy function, the points on the curve move to minimize the defined energy. First, an active contour is initialized on the shape boundary. The inverse of the surface obtained from a shape's distance map is used as a potential surface. The points on the active contour move on this potential surface trying to minimize their potential energies i.e. heights. Figure 2-12 shows an example of a potential surface for a rectangle shape. This surface is obtained by inverting the surface obtained from the distance transform of a rectangle.

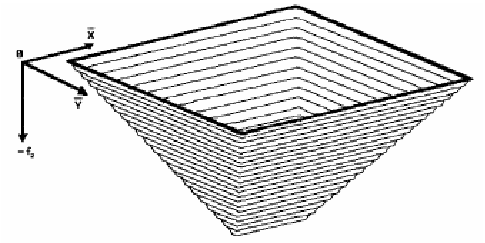


Figure 2-12 (Taken from [10]) Example of a potential surface.

The points on the active contour normally move toward the local minima of the potential surface. During its evolution, the active contour folds into thin lines at singularity points. When such conditions are detected, these special points are not allowed to move any further. To implement the idea, it is necessary to compute the curvature extrema along the shape boundary and attach active contour points at these curvature extrema. If the active contour is not fixed at these critical points, it falls down along the ridge. Figure 2-13 shows an example of grassfire propagation. Four curvature extrema are detected for the rectangle shape. The points where the active contour folds into thin lines are apparent. The evolution stops when the active contour reaches a steady state. Special precautions are taken for detecting symmetry branches whose ends do not lie on the shape boundary.

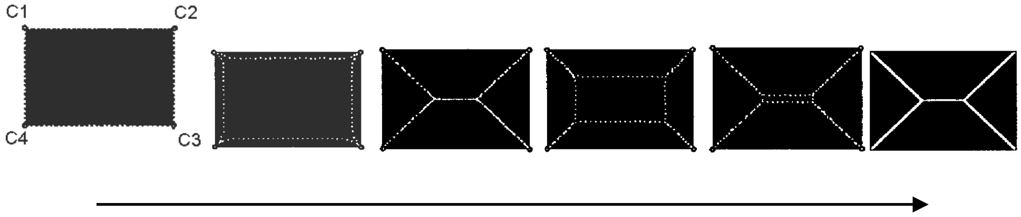


Figure 2-13 Example of active contour evolution on a rectangle. The arrow indicates increasing time.

The critical points where the active contour is fixed are also used to obtain a graph representation of shape. The boundary segments between these critical points represent fire fronts. Bifurcation points are detected from the fact that at such points there will be active contour points from the three different fire fronts. The active contour is disconnected at such points to obtain an undirected graph. The boundary information in the form of curvature extrema provides axis regularization and a scale space representation. By defining saliency measures for the curvatures e.g. maximum relative curvature amplitude, region of support, the number of boundary segments is changed. The parts of shapes where the active contour is not initially fixed are not represented by the medial axis.

The use of a local property (curvature) raises questions on the robustness of the representation. The main reason which makes the medial representations popular is that purely local shape analysis doesn't give robust information about the properties of a shape. The curvature maximum of a shape part is not the only property that determines its saliency. Therefore, the active contour may be fixed to the end-points of unimportant protrusions. Globally more important protrusions may be lost in the representation.

2.5 The Work of Rom and Medioni [19]

Rom and Medioni combined various ideas into a unified scheme to devise a representation that produces a decomposition of a shape into parts together with axis-based descriptions of these parts. The method is another example of a shape representation study that uses both global and local boundary information in obtaining shape descriptions. The axis-based descriptions of parts are derived using SLS. As mentioned previously, SLS computation results in many axes that are ambiguous or irrelevant. To resolve this ambiguity and recover the correct branches, the parts of the shape are determined by identifying negative curvature minima on the shape boundary [7]. The parts of the boundary segmented by such points are potential parts. According to the Symmetry-Curvature Duality Theorem of Leyton[11], the correct SLS axes explaining every part can be recovered by finding axes emerging from local curvature maxima. By using the symmetry and curvature information, the parts of the shape and the SLS representing those parts are identified (Figure 2-14).

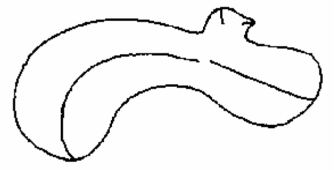


Figure 2-14 (Taken from [19]) The recovered SLS axes of a shape

After the identification of parts, the smallest parts are removed at each step of a recursive procedure to produce a hierarchical decomposition. Each recursive step represents a level in the hierarchy. A small part is defined to be a part with an axis that is no longer than 1.5 times the shortest axis. Figure 2-15 shows the resulting descriptions of an example shape. The original shape is on the left. At every node, the parts that are removed for that level of the description are shaded. The remaining shape is shown in white. Local sym-branches of the shape are shown on the right.

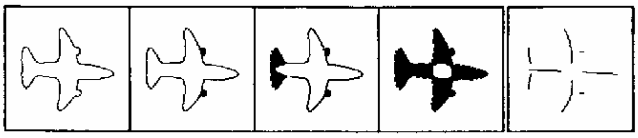
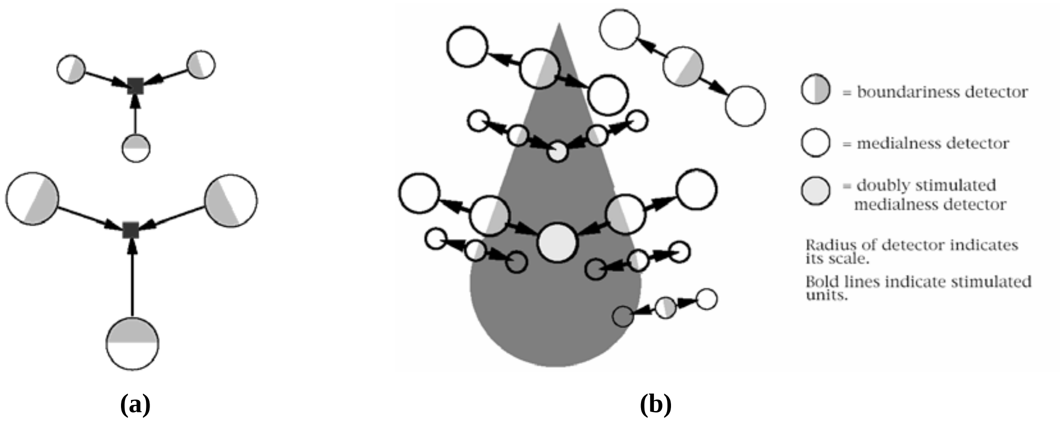


Figure 2-15 Decomposition of a shape into parts.

Unlike the study of Leymarie and Levine [10], the use of curvature information does not present problems for accurate extraction of the skeletal structure because the curvature information is not used as a saliency feature. There is not any recognition system in literature based on the representation scheme proposed in this study. To be used for recognition some issues must be considered. First, the hierarchical decomposition idea may lead to instabilities. Because the removal of a shape part is determined using a threshold, a small change in the length of an axis (or in the length of the shortest axis) may cause the removal of that axis in one instance, and not in another similar instance. The same situation may occur if a small branch is added to the shape description due to a small perturbation of the boundary. These phenomena cause instability at all levels since the decomposition at a level proceeds from the output of the previous level. Second, additional information about the relations of the protrusions of a shape must be stored in the descriptions. For instance, if the axes were connected, the relation of parts might be captured in a graph structure and graph matching methods would be used for recognition.

2.6 Cores [5]

As opposed to most studies on shape representation which assume that shape boundaries are already identified, "cores" representation by Burbeck and Pizer offers a combined approach to shape segregation and representation. The basic tool of the representation is the boundaryness detector which signals a degree of stimulation of boundaryness. The edge information is obtained by the collection of responses from such detectors. The segregation of a region from its surroundings is based on the idea that a region's boundaries must be linked across the region itself. Therefore, a detector on the shape boundary must be paired with another detector on another part of the boundary to satisfy this requirement. The boundaryness detectors have also scale (the area over which boundary information is computed). Small scale boundaryness detectors are connected to another over short distances whereas large scale boundaryness detectors are connected to another over large distances. Figure 2-16 illustrates boundaryness detectors and their connection. The radius of a detector indicates its scale. Only detectors of the same scale interact, and they connect along directions perpendicular to the orientation of each detector. Each boundaryness detector signals the presence of a boundary for a width proportional to its scale. Those detectors that catch the correct orientation and location of edges show high degree of stimulation. Note that in Figure 2-16(b) a large scale detector is not able to capture the small parts of the teardrop shape (also a small scale detector is not able to capture a large part of the shape since its signaling distance is too small to connect with another detector at other parts of the boundary).


= boundariness detector
= medialness detector
= doubly stimulated medialness detector
Radius of detector indicates its scale.
Bold lines indicate stimulated units.

(a) (b)

Figure 2-16 (Taken from [5]) (a) Boundaryness detectors (shown with circles) and their connections. (b) The boundary detectors retrieve the boundary of a teardrop shape.

A boundaryness detector excites a region at a distance proportional to its scale. Adding up the excitation of different detectors, a three-dimensional plot can be obtained. Since an excitation is at the middle point of two detectors, it can be thought of as a medialness detector. The width information is inherently stored within each boundaryness detector; therefore, a symmetric axis representation is obtained. A "core" is a trace in the three dimensional plot representing the regions with high medialness. Figure 2-17 shows the "cores" of a rectangle shape with one saw-tooth edge.

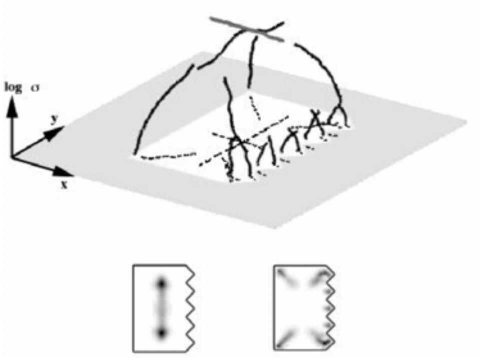


Figure 2-17 "Cores" computed from a rectangle shape with one saw-tooth edge.

When the plot of the rectangle shape is examined, it can be seen that the projection of "cores" on the (x,y) plane gives the axes at which the shape is locally symmetric. The excitation value of a branch provides a saliency measure and therefore leads to a scale-space representation. The symmetry axes of the rectangle shape at two different scales are shown in Figure 2-17.

There are two issues to be solved in this study. The first one is shape boundary extraction from an image; a core could be created for every pair of roughly parallel boundaries. These parallel boundaries may not belong to the same object. This leads to a strong possibility that neighboring objects interfere each other's representation which is a very serious issue for accurate segregation.

The second issue is illustrated in Figure 2-18. In this figure, the cores shown at the center of the shape represent the object at the highest scale. These cores are due to the parallel boundaries shown with arrows. The distance between these boundaries is the largest when the other boundary pairs of the shape are considered; therefore, the cores representing them are largest in scale. If this shape is non-rigid, its parts may articulate. In that case, these parallel boundaries may no longer be parallel and no cores would be created between these boundaries, which causes a significant change in shape description.

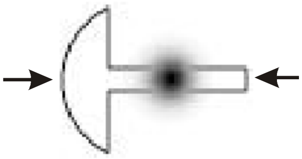


Figure 2-18 (Taken from [5]) A shape and its cores at largest scale. The medialness detected at the center of the shape is mainly due to the parallel boundaries shown with arrows.

2.7 FORMS [26]

Zhu and Yuille presented a shape representation scheme and a recognition system based on this representation. The system is mainly designed to describe and recognize animate objects which do not usually have sharp features on their boundary. The parts of a shape are described by the deformable primitives and the skeleton of the shape determines how these parts are arranged in a hierarchy to form the description of the shape.

Animate shapes are modeled in three stages. At the lowest level there are two shape primitives. The first is the *worm*, which is a rectangle with joint circles at both ends and the second is the *circle*. In the next stage, mid-grained parts are obtained which are deformed versions of the primitives. Deformed worms are used to describe elongated parts and deformed circles are used to describe short parts. Figure 2-19 shows the shape primitives and their deformations (mid grained parts).

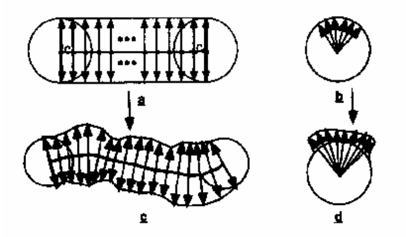


Figure 2-19 (a) Worm primitive (b) Circle primitive (c) The mid-grained part resulting from the deformation of the worm primitive (d) The mid-grained part resulting from the deformation of the circle primitive.

The last stage is the shape grammar. It allows one to define an animate object as a hierarchical collection of mid-grain parts whose axes form the skeleton structure. A mid-grained part is able to describe simple shapes but complex shapes are described using the shape grammar.

In order to compute the descriptions of shapes from images, a skeletonization algorithm is used to recover the shape grammar. The relations of mid-grained parts are determined by the grammatical structure. The points where the skeleton bifurcates form the points where new mid-grained parts are formed. The skeleton is extracted from shapes by two processes. The first process determines the sym-points. A circle is defined and deformed to form a maximal disc inside a shape. The center of the minimizing circle forms a sym-point. The second process determines the bifurcation points of skeleton. A range-angle function is used to determine whether the skeleton should bifurcate. This function gives the distance from a symmetry point to the shape boundary in a particular direction (Figure 2-20). The peak values of this function are thresholded to determine if the protrusion in a particular direction is due to noise or a real skeleton branch.

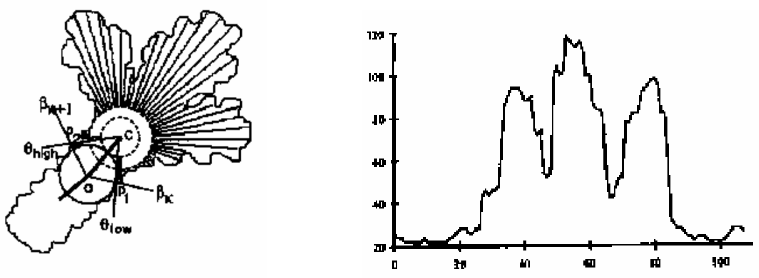


Figure 2-20 The range angle function computation at a point.

Figure 2-21 illustrates how a dog figure is partitioned into seven mid-grained parts according to retrieved skeleton structure.

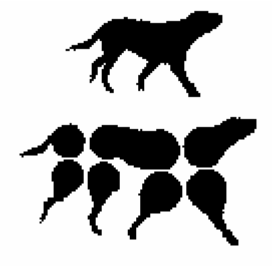


Figure 2-21 A dog figure and its computed parts.

The primitives of the representation are adequate for describing the shapes for which they are designed. New primitives may be added to describe a wider shape domain. The stability of the representation depends on the skeletonization algorithm. The effectiveness of the range angle function, which is the primary method used to overcome the instability of skeletons, may be questioned. For a limb of a shape to be captured, the range angle function should give a high value at a point in the direction of the limb. If the axis curvature of this limb is high, that is, if this limb is highly bent, then the range angle function would give a small distance value. A prominent limb with high axis curvature may be lost. The recognition system presented with the representation scheme is explained in detail in the next chapter.

2.8 Shape Axis Tree [6]

This study presents a variational approach to symmetry computation. The power of a variational formulation is that the desired features of the resulting descriptions such as stability, robustness to noise, scale space etc. can be included as constraints in the variational framework. Once a variational framework is constructed, the features of the descriptions can be "tuned" using the weights of the constraints.

The description computed from a shape is called the *shape-axis tree*. The matching of two parameterizations of a shape boundary is formulated as a variational problem. A powerful feature of the system is that the boundary curve need not be connected so symmetry points can be computed from shapes with open contours. First, a starting point on the boundary curve is selected, and the curve is traced forwards and backwards from this starting point to generate two parameterizations. According to mirror symmetry, distance and parallelism constraints, the two parameterizations of the shape boundary are matched.

The solution to this variational problem is a function which maps one curve to the other. Matches are piecewise continuous and monotone. The order along a boundary is necessary for perceptually correct matchings. The shape axis is the locus of midpoints that connects two matched points on the boundary. The discontinuities on the binary matching function correspond to the bifurcations of the shape axis. When a boundary point is matched to an identical point in the other parameterization, this point is designated as a leaf of the shape axis. Such points are the curvature extrema along the boundary curve.

The factor that determines the scale of the descriptions and that allows the stability of the shape axis is the penalty term in the variational formulation for discontinuities in the resulting binary match function. This penalty term is called jump cost and penalizes discontinuities (the number and the degree of bifurcations) in the match function. Figure 2-22 shows how the points on a shape boundary are matched and local sym-branches are extracted. The first shape-axis tree is computed using a higher jump cost than the second shape-axis tree computation. It can be seen that the opposing goals of stable and sensitive descriptions can be obtained easily by changing the jump cost in the variational formulation.

As to the computational details, the optimal matching is found using dynamic programming. The algorithm runs in polynomial time in the number of points used to sample the shape boundary. However, the number of points used is large for accurate extraction of the shape axis tree making the curve matching step computationally expensive. The matching algorithm presented in this study is reviewed in the next section.

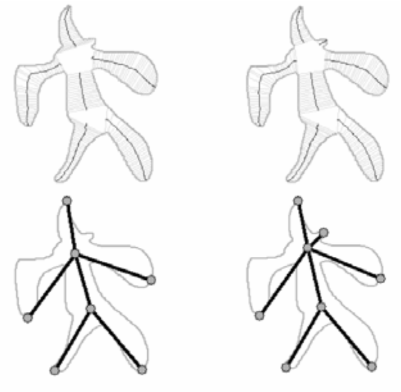


Figure 2-22 The shape-axis tree computation for different values of jump cost.

2.9 Shock Grammar and Shock Graphs [9,20,23,24]

Curve evolution based shape analysis was introduced to the computer vision literature by Kimia et al.[9]. The level set formulation of Osher and Sethian [17] provided an accurate scheme to implement front propagation and Kimia et al. used it to extract shape skeletons by simulating the fire front propagation.

In curve evolution, the main idea is to deform a curve by moving its points in the direction of inward or outward normal according to a prescribed velocity. Kimia et al. considered evolution of the curve with velocity that depends on two components: a constant component corresponding to morphology and a smoothing component proportional to curvature.

Consider a shape represented by the simple closed curve Γ. Let $C(s,t)$ be the evolving family of curves where t represents time and $C(s,t=0)=\Gamma$. The evolution equation of this curve moving under constant and curvature dependent motion along the inward normal direction is given by:

$$\frac{\partial C}{\partial t}=(\beta_0-\beta_1\kappa)\vec{N}$$

where β_0 and β_1 are constants controlling the relative weight of each component, κ is curvature and $\vec{N}$ is the inward normal.

The standard way to implement this equation is to use Osher and Sethian's level set formulation in which the initial curve is embedded as the zero level set of the signed distance function (the distance map) and let all level curves of this function evolve according to the equation. The method handles the topological changes during the evolution of the curve well. If the weight of the curvature component is set to zero, a morphological evolution is obtained which is identical with the fire front motion in the grassfire model of Blum. The singularities (shocks) that develop during the process correspond to sym-points of the symmetric axis. They must be detected using a shock-capturing scheme.

While the constant motion of the shape boundary provides an accurate implementation of Blum's SAT in the discrete domain, the curvature dependent motion serves to alleviate its instabilities. The curvature deformation process mainly removes noise and regularizes the evolving shape boundary. Figure 2-23 shows how the curvature deformation process eliminates the noise and reveals the fundamental structure of the shapes. The first row shows the original shapes and the other rows show the states of shapes at increasing times during their evolution.

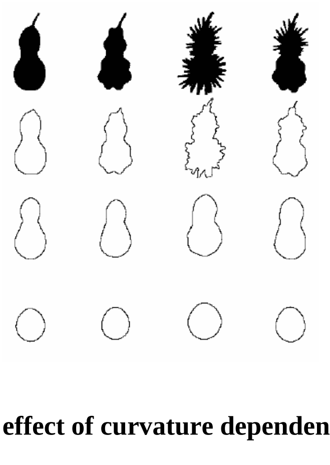


Figure 2-23 (Taken from [9]) The effect of curvature dependent motion on noise elimination.

The ratio of β_0 and β_1 determines the detail of the evolving shape boundary and suggests a scale space representation. The representation of a shape at all possible time and all possible ratios β_0 / β_1 $(\beta_1 < 0)$ [9] is the *reaction-diffusion space* for that shape. Figure 2-24 illustrated the concept for a human shape. In the case of pure reaction, the deformation of the shape is rigid and the shape breaks into pieces during the process. Under pure diffusion, all the protrusions and indentations are smoothed out and the shape converges to a circle and finally to a single point.

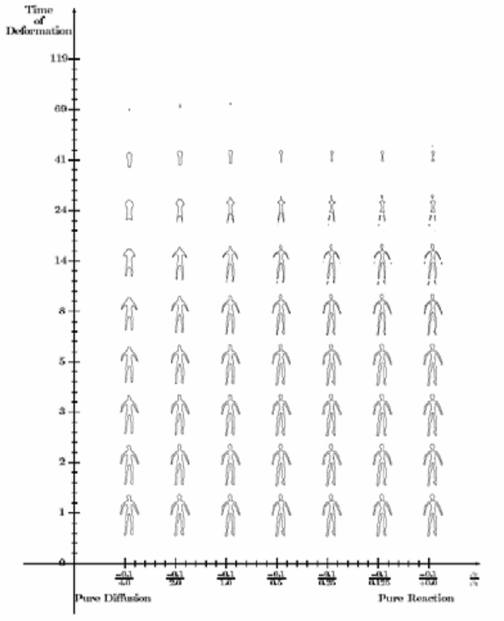


Figure 2-24 (Taken from [9]) The reaction-diffusion space for a human shape

The classification of sym-points is based on the differential properties of the surface in which the shape boundary is embedded. Smooth sym-points, which they called *first-order shocks*, are detected by the high curvature on the level curve and non-vanishing gradient on the surface. Since diffusion is incorporated in the process, corners are smoothed out and singularities must be detected using a curvature threshold. Pinch (or neck) points (called *second-order shocks*) correspond to hyperbolic points on the surface and are identified by isolated vanishing gradient and principal curvatures of different signs. Worm points (called *third-order shocks*) lead to a group of parabolic points on the surface. The gradient of the surface and the product of principal curvatures at such points are zero. Finally, *fourth-order shocks* represent the points where the evolution of the curve comes to rest. These points are the elliptic points of the surface. Blum termed these points as bulb points. The curve evolution based shape analysis provides the location and types of sym-points. Grouping them into sym-branches to reach the eventual goal of a graph representation is not a trivial task. In particular, the sym-branches corresponding to the collection of smooth sym-points or the collection of worm points must be identified. The shock grammar proposed by Siddiqi and Kimia [23] is essentially a rule system to retrieve the correct representation of the shape by discarding false sym-branches. Figure 2-25 summarizes the four types of sym-points used in the shock grammar representation.

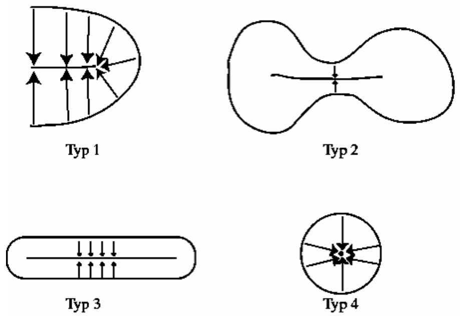


Figure 2-25 (Taken from [24]) Four types of sym-points used in [24]: A group of smooth sym-points, a pinch point, a group of worm points, a bulb point

Curve evolution based shape analysis is one of the neatest formulations in computer vision. Moreover, the interesting ideas on axis morphology presented by Blum in late sixties finally found their successful implementation through the work of Siddiqi and Kimia.

Siddiqi and Kimia considered only the constant motion in the evolution equation to obtain the skeletal descriptions of shapes. This leads to classical symmetric axis of Blum. The symmetric axis provides no means to separate prominent branches from the unimportant ones. Even a small protrusion on the shape boundary may produce a long skeletal branch. When curvature dependent motion is introduced, the significance of a sym-branch is proportional to its survival over scales (reaction-diffusion space). Siddiqi and Kimia suggest to use this fact to assign a level of significance to each sym-branch obtained under pure reaction. The effect of diffusion on the sym-point detection process is shown in Figure 2-26. The leftmost shape represents the sym-ax obtained under pure reaction. From left to right, the weight of the curvature dependent motion is increased. As the diffusion is increased, the sym-branches corresponding to unimportant branches are annihilated.

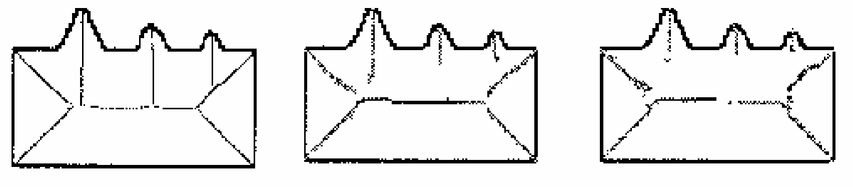


Figure 2-26 (Taken from [23]) The effect of diffusion on the extracted sym-branches

Although the idea of a combined framework (axis regularization and medial axis extraction) is appealing, it has not been used in practice for obtaining stable axis-based descriptions for recognition. Siddiqi and Kimia used only those axis-based descriptions obtained by morphological evolution

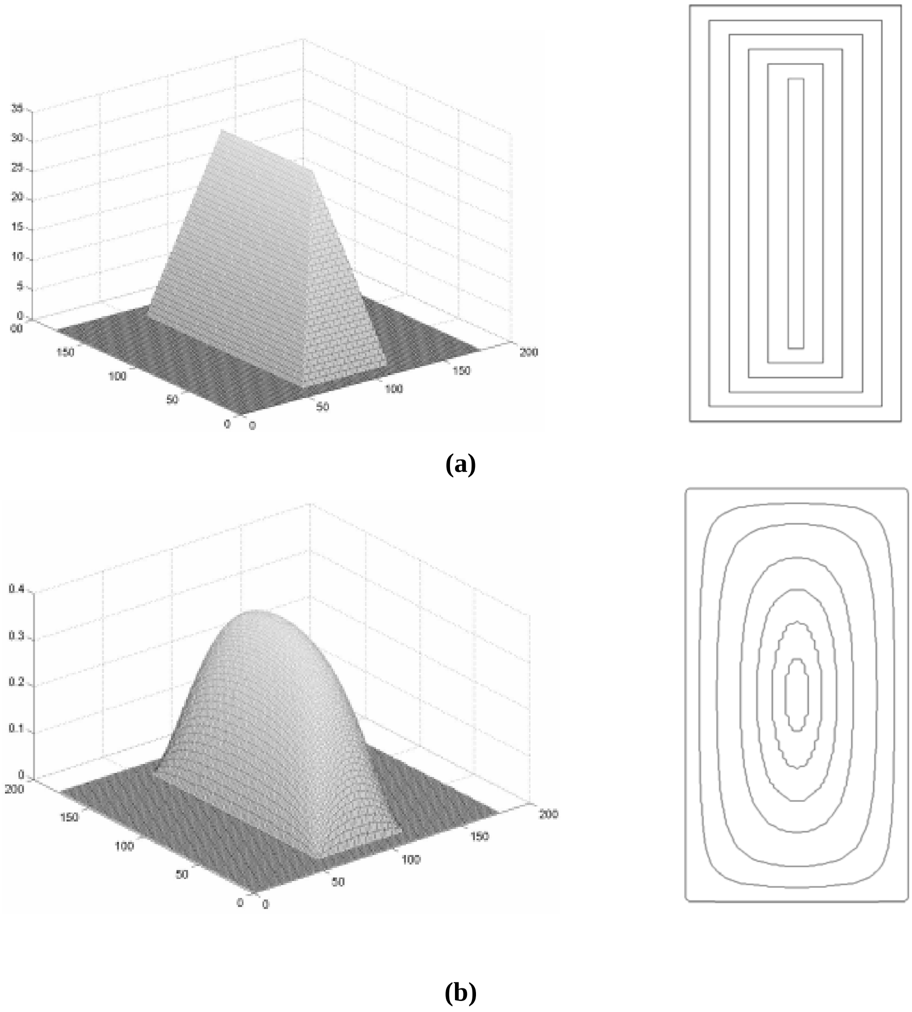
[20,24]. This may be due to two facts. First, when diffusion is introduced, detection of *first order shocks,* which are the local curvature maxima of the evolving curve, becomes difficult. Second, even a small amount of diffusion leads to a disconnected skeleton, as shown in Figure 2-26. This is not an artifact of computation. Symmetry points measure the deviation of the evolving boundary from a circle. Hence, when a curve locally gets rid of a protrusion or an indentation -under the influence of diffusion- the symmetry branch tracking it terminates. Deriving a hierarchical representation from a disconnected skeleton is a difficult problem. Two shock graph representations emerged from the shock grammar of Siddiqi and Kimia: The shock tree of Siddiqi et al.[24] and the shock graph of Kimia et al. [20]. These representations are reviewed in the next chapter where recognition systems using these representations are examined.

2.10 The Method of Tari, Shah and Pien (TSP) [25]

Tari, Shah and Pien's method (TSP) offers an alternative implementation of curve evolution for shape analysis. Its basic tool is the function v whose level curves are interpreted as a family of evolving curves under the influence of constant and curvature motions. It has significant computational advantage over the work of Siddiqi and Kimia. It does not require the specification of shape boundary and can be applied directly to raw images. Yet, the computation is further simplified when the shape boundary is known. Specifically, the function v is the unique minimizer of $\frac{1}{2}\iint\left(\rho\|\nabla v\|_2^2+\frac{v^2}{\rho}\right)dxdy$ subject to $v\,|_\Gamma = 1$ where Γ denotes the shape boundary. It is computed by solving the following the Euler Lagrange equation which can be implemented using standard finite difference approximations:

$$\nabla^2 v - \frac{v}{\rho^2} = 0, v\,|_\Gamma = 1$$

The function v is the smoothed analogue of the surface obtained from the distance transform of the shape (Figure 2-27). It equals 1 along the object boundary and decays rapidly away from the boundary. ρ is the parameter that controls the smoothing. As $\rho \to 0$, $v \to 0$ everywhere except along Γ. As ρ increases, the inner level curves of the surface become smoother.



(a)

(b)

Figure 2-27 (a) The distance surface and the level curves in morphological evolution. (b) The function 1-v and the level curves in TSP formulation.

To understand the behaviour of the level curves of v better, imagine moving from one level to another as visualized in Figure 2-28.

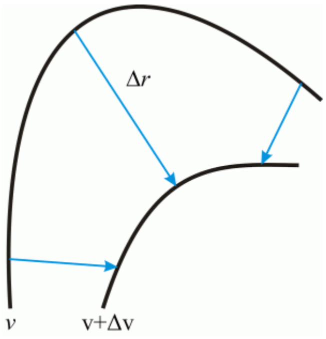


Figure 2-28 The motion from one level curve to the next. r is the arclength along the gradient lines.

As discussed in [25],

$$\Delta r \approx -\frac{\rho}{v}\left(1+\frac{\rho\kappa}{2}\right)\Delta v$$

Defining an artificial time variable as a monotonic function of v such that $\Delta t = -\frac{\rho}{v}\Delta v$ and letting $\Delta t \to 0$

$$\frac{dr}{dt} = 1 + \frac{\rho}{2}\kappa$$

Just like the standard curve evolution, the points on the level curves move in the inward normal direction (direction of decreasing v) with velocity consisting of a constant component corresponding to morphology and a smoothing component proportional to curvature. ρ is equivalent to ${}^{\beta_0}\!/_{\beta_1}$ ratio used to generate reaction-diffusion space for shapes in Kimia et al.'s formulation.

Sym-point detection and classification is based on the local geometry of the minimizing surface v. When smoothing is introduced, singularities cannot develop as corners, because they are rounded out. They become points of maximum curvature along a level curve. An important observation is the inverse proportionality of the curvature extrema to the gradient extrema. Along a level curve, the points of maximum curvature correspond approximately to the points where $|\nabla v|$ is minimum. This property of v enables a robust computation of sym-points as it replaces the computation of the second order derivative (curvature) by a computation of a first order derivative (gradient). The sym-points of a shape are defined by the set of zero-crossings of $\frac{d|\nabla v|}{ds}$, where s denotes the arc-length along the level curves. At positive sym-points $\frac{d^2|\nabla v|}{ds^2}$ is positive and at negative sym-points $\frac{d^2|\nabla v|}{ds^2}$ is negative.

In terms of global coordinates x and y:

$$\frac{d|\nabla v|}{ds} = \frac{\left\{\left(v_y^2 - v_x^2\right)v_{xy} - v_x v_y\left(v_{yy} - v_{xx}\right)\right\}}{|\nabla v|^2}$$

$$\frac{d^2|\nabla v|}{ds^2} = v_{\eta\xi\xi} + \frac{v_{\xi\xi}\left(v_{\xi\xi} - v_{\eta\eta}\right)}{|\nabla v|}$$

where

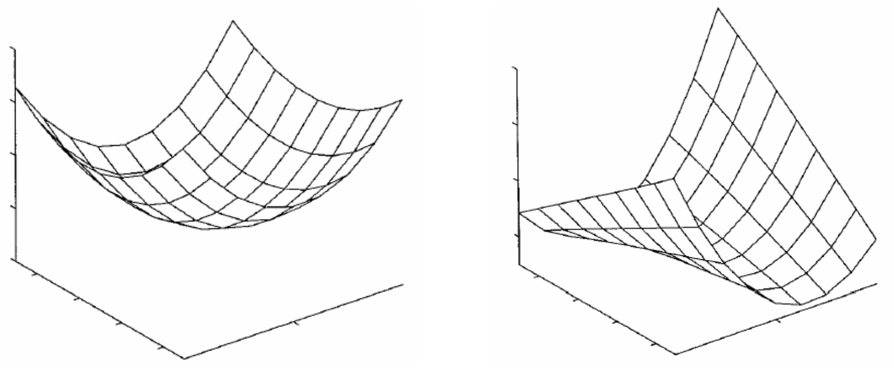
$$v_{\xi\xi} = \frac{\left\{ v_y^2 v_{xx} - 2v_x v_y v_{xy} + v_x^2 v_{yy} \right\}}{\left|\nabla v\right|^2}$$

$$v_{\eta\eta} = \frac{\left\{ v_x^2 v_{xx} + 2v_x v_y v_{xy} + v_y^2 v_{yy} \right\}}{\left|\nabla v\right|^2}$$

$$v_{\eta\xi\xi} = \frac{1}{\left|\nabla v\right|^3} \left\{ v_x v_y^2 v_{xxx} + v_y \left(v_y^2 - 2v_x^2 \right) v_{xxy} + v_x \left(v_x^2 - 2v_y^2 \right) v_{xyy} + v_x^2 v_y v_{yyy} \right\}$$

During the course of evolution, positive sym-points track the evolution of the protrusions of a shape while the negative sym-points track the evolution of its indentations. A positive sym-branch may merge with a negative one terminating both branches. This happens when the protrusion represented by the positive sym-branch is completely smoothed out. More complicated merges between positive and negative sym-branches may occur. If a branch does not terminate at such a junction, it comes to rest at a surface extremum.

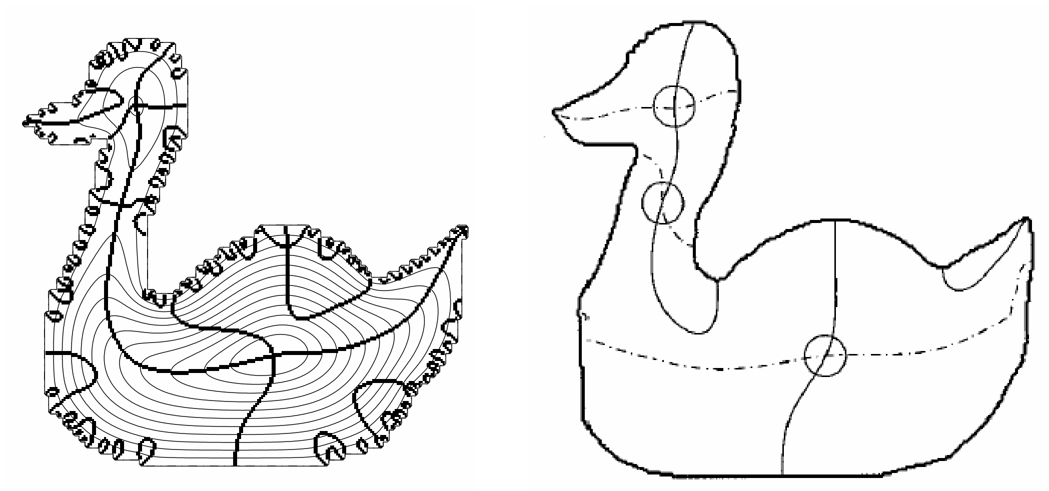
The points on the surface where the gradient vanishes $\left|\nabla v\right| = 0$ indicate special sym-points such as bulb and pinch points. Elliptic points are the center points where the shape shrinks into a point and they are indicated by the positive determinant of the Hessian (Figure 2-29(a)). Hyperbolic points correspond to pinch points in Blum's formulation and second order shocks in Kimia et al.'s terminology. The determinant of Hessian at these points is negative (Figure 2-29(b)). In TSP formulation, worm points are not encountered in practice. Due to the interaction between opposite boundaries at the neck points, the boundaries tend to form crosses rather than be tangent to each other.



(a) **(b)**

Figure 2-29 (a) Surface v near an elliptic point (b) Surface v near an hyperbolic point

Figure 2-30(a) shows the sym-points of a duck shape with the level curves superimposed. Notice how a symmetry branch tracking a protrusion merges with a symmetry branch tracking an indentation and terminates. There are three special points of the surface corresponding to the neck and the centers of the head and the body (Figure 2-30(b)). These points are detected by the simultaneous zero crossings of v_x and v_y.



(a) (b)

Figure 2-30 (a) Full sym-points of the duck shape with level curves superimposed (b) The zero crossings of v_x, v_y and the special sym-points (See Appendix A.1, A.2).

TSP formulation provides essentially the same shape analysis provided by the method of curve evolution. However, it does not extend to the case of morphological evolution; that is ρ cannot be too small. Therefore, the traditional morphological skeleton cannot be obtained. The formulation provides the locations and types of sym-points of a shape but no method to derive the sym-branches is given.

Our sym-point detection method is a modification of TSP. Prior to our work TSP is not pushed towards recognition. It is true that TSP gets rid of instabilities of morphological skeleton but it runs into other instabilities such as the saddle point problem discussed in Chapter 4.

2.11 Discussion

Literature on axis-based shape representation is vast. Since its introduction, the SAT of Blum has been studied extensively. All the representation schemes investigated in this chapter are considered important pieces of research on axis-based shape representation. They have fostered new ideas and guided research. They are reviewed rather in a detailed way in order to determine the common approach taken, problems encountered and to establish what they are actually trying to accomplish.

The ideas presented by each study can be analyzed in four different topics. These are:

- Detection of symmetry
- Regularization
- Scale space (or hierarchical) representation
- Describing relations of primitives

These topics also are the steps that should be taken when devising an axis-based representation scheme. It should be pointed out that no study reviewed in this section offers a solution for all of these steps.

Detection of symmetry is the first step of every axis-based representation scheme. Although the earlier research on this topic is not examined here, some issues were identified with the symmetry extraction step of some of the recent works [10,16,26], which indicates the severity of the problem. Curve evolution based approaches [23,25] currently offer the best solution since the symmetry axis is accurately extracted and sym-points are classified.

Regularization, which is performed for obtaining stable descriptions, can be incorporated in the symmetry detection process [6,10,23,25] or can be performed after the symmetry axis is obtained [16,19]. All regularization approaches try to separate important features from unimportant ones because the salient characteristics of shapes do not change often. Descriptions based on these characteristics are less likely to have instability problems. The ineffectiveness of the regularization methods that work on the symmetric axis is due to the fact that symmetric axis does not provide any means to determine the importance of parts. Curve evolution based approaches alleviate this problem since the survival time of a branch indicates its relative importance.

Scale space representations of shapes have been considered essential for recognition and most of the studies reviewed generate coarse to fine descriptions of shapes. However, this idea has not been applicable for a number of reasons. First, the scales generated are not “absolute”. The selection of the same regularization parameters (e.g. pruning threshold, diffusion of the boundary) for different shapes does not guarantee that these shapes will be represented at the same level of detail. This is because the regularization parameters consider local properties of primitives. For instance, the survival time of a symmetry branch used in [23] is a local property which depends on the curvature of nearby protrusions and indentations. In [10], curvature information is used which is again a local property. Second, most methods require symmetry axis to be connected so that the relations among branches can be expressed easily. If a symmetry branch doesn’t connect to the main symmetry axis, it is discarded. The transition from one scale to the other may be accompanied by substantial changes in the axis structure. Because of this large change, the task to determine the correspondences between symmetry branches at different levels of detail becomes a difficult problem.

An important aspect of a representation for recognition is the structure of primitives. When the similarities of two shapes are to be determined, the organization of primitives must be used to enforce coherence of the correspondences. All the methods offering solution for this issue [2,6,10,23] depend on the topology of the symmetry axis. In order to capture the topology, connectivity must be guaranteed in the symmetry axis extraction step. The organization of symmetry branches in those representations is described in a graph or a tree structure. For the representations that produce disconnected symmetric axis, determining topology is a difficult problem and no solution is offered in such representation schemes for describing relations of primitives.

CHAPTER 3

USE OF AXIS-BASED REPRESENTATIONS FOR RECOGNITION

In Chapter 2, important representation schemes designed for generic shape recognition problem were reviewed. Some of these schemes have been used in recognition frameworks which are reviewed in this chapter.

Shape matching problem is the basis for recognition. In shape matching, the best correspondence between two shapes is determined. There are two issues to be considered when devising a matching algorithm based on primitive based descriptions of shapes. First, a similarity or a distance measure for the individual comparison of primitives must be defined. Second, the best correspondence of two shapes, which is determined by the collection of pairs of matched primitives and their similarities, must be found. Coherence on the collection of matched primitive pairs must also be enforced so that the correspondence found is perceptually accurate. Specifically, the organization of primitives must be stored in the descriptions to enforce coherence on the matching of two shapes. When the features of individual primitives are not sufficient to provide organization information, it is explicitly stored in the description. The traditional approach has been to capture the organization of primitives by storing them in a graph or a tree structure. The nature of how this information is stored i.e. whether it's a graph or a tree, affects the correctness and complexity of matching process. In this chapter, we discuss how some matching schemes approach to aforementioned issues and give their performance results.

3.1 FORMS [26]

The primitives of description are "mid-grained parts". The individual similarities among mid-grained parts are determined by their geometrical similarities. The presence and absence of one part in the object is assumed independent of the matching of the others. Not all matchings between mid-grained parts are considered equivalent. A matching contributes to the whole similarity score of two shapes depending on its importance. Matching of a large shape part affects the whole similarity of shapes more than the matching of a small part. The matching process is a branch-and-bound algorithm that searches over all possible matches between two shapes to find the matching that produces maximum similarity score. An interesting idea not found in other studies reviewed here is the top-down verification process incorporated in the matching algorithm. The skeleton extraction process is considered a bottom-up process where no a priori information is used. When a matching residual between a model shape in the database and an input shape is detected; that is, when the skeleton structures are different the top-down verification process is employed. Using four operators, this process generates a number of possible skeletons from the skeleton of the input shape (Figure 3-1).

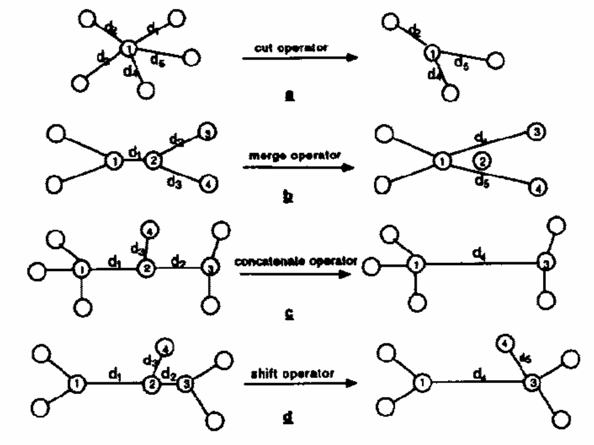


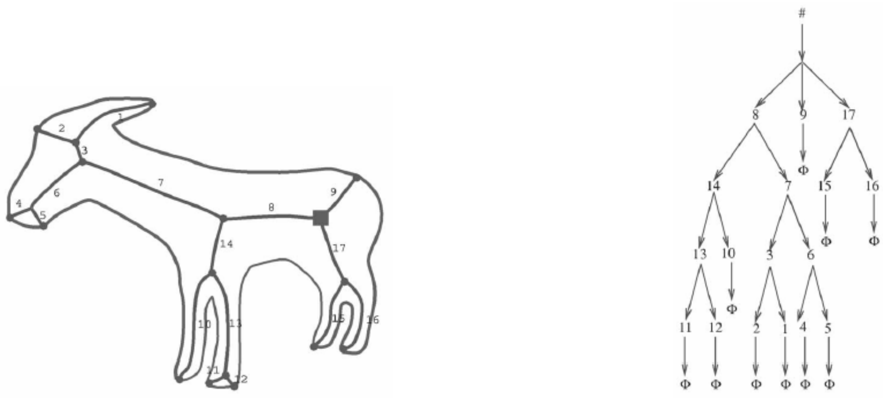
Figure 3-1 The skeleton operators from top to bottom: cut, merge, shift and concatenate

The motivation behind this approach is to make use of the partial matching between the model shape and the input shape and to adjust the skeleton of the input shape to compensate for the errors in the skeleton extraction process. A skeleton adjustment means that a new description of a shape is obtained. Perceptually different shapes may have the same skeleton structure because of this adjustment. This is handled by including costs for applying skeleton operators and by the fact that the mid-grained parts of these shapes would be different.

The recognition performance is tested on a database consisting of 17 categories with two shapes in each category. The ability of the matching process under changes in viewpoint, articulation or scale is not reported.

3.2 Shock Tree [24]

In the shock tree of Siddiqi et al. shock types label each vertex of a graph structure and the shock formation times provide the directed links between vertices. Since first and third order shocks are neighbored by other shocks of the same type, the group of first order shocks and third order shocks constitute a vertex of the graph. By representing shapes as attributed graphs, the matching problem becomes the well-studied attributed graph matching problem. The graph structure derived from shocks is converted to a tree structure (the shock tree) in which the oldest fourth order shock is designated as the root. This is mainly to increase computational efficiency. The graph matching problem is NP- hard whereas polynomial time algorithms exist for tree structures. The hierarchy of skeleton elements is captured well in a tree representation but a small change in the shape may cause the oldest shock to change leading to a significant change in object topology. Moreover, the shock tree structure doesn't capture the planar order of its primitives which may result in false matches. The shock tree representation is illustrated in Figure 3-2.



(a) **(b)**

Figure 3-2 (Taken from [20]) (a) An example shape (b) Its shock tree description

In [24], the similarity measure between two vertices depends on two components: a topological similarity measure and a geometrical similarity measure. Only the shocks of the same type are matched. Since the relations of shape primitives are captured in a tree structure, each node can be considered as a root of a subtree. The topological measure depends on the structure of a subtree rooted at a node. An eigenvalue characterization of tree structures is used to compute the topological similarity measure. The geometrical similarity measure compares the curves formed by the group of first order shocks and third order shocks geometrically.

The matching of two shock trees is formulated as a subtree isomorphism problem. A simple tree matching may be insufficient because the tree computed from the image may represent the entire scene and not just the object that we want to classify.

This study is a very good example of the use of tree structures for axis-based representations. The most important advantage of using a tree structure is the ability to capture its structure using eigenvalue characterization. Without too much loss of uniqueness, the topological comparison of two trees is simply carried out by finding the distance between two vectors in space. The same researchers, in another study, [22], implemented this eigenvalue characterization successfully for indexing on a silhouette database of 60 shapes (the number of categories in the database is not reported). The second important advantage is the modularity provided by tree structures. Each vertex of the shock tree is itself a tree. This makes it possible to use alternative matching schemes. Rather than comparing the whole query tree with a model tree, the parts of the query tree may be identified first. From this identified parts, the nature of the object may be determined. This approach makes use of the alternate access path to model shapes as proposed by Marr and Nishihara [15].

The major issue of shock tree approach of Siddiqi et al. is that the underlying shock representation shares the same instabilities of Blum's symmetric axis. A small change in the shape structure usually leads to large changes in tree topology. In addition, all the boundary details of the shape boundary are stored in the symmetric axis. The complex skeleton structure leads to a tree structure with many vertices which is a major issue for efficiency. Another practical issue is that the computed distance measures are not normalized. When a query shape is matched against the model shapes in the database, the most similar model shape can be determined, but it is not known whether the most similar model shape is indeed similar. Finally, the planar order of shape primitives is not captured in the representation which may lead to perceptually unnatural matchings.

As to the performance results, the matching algorithm results in correct matchings most of the time on a small database of 25 shapes with nine different categories.

3.3 Shape Axis Tree [6,12]

As opposed to the shock tree representation of Siddiqi et al., in the shape axis tree every edge represents an object substructure. Therefore, the primitives of the description are the tree edges not the vertices. The similarity or distance of edges is determined by comparing their respective boundary segments. This comparison is carried out by computing the cost to deform one boundary segment to another. The cost is based on the bending and stretching necessary to match two boundary segments. For instance in Figure 3-3, two shapes are shown along with their shape axis trees. The cost of matching the edge u_2-u_5 in the first axis tree with the edge v_1-v_4 of the second axis tree is the cost of deforming the boundary segment B-C-D in the first shape to the boundary segment A-B-C in the second shape.

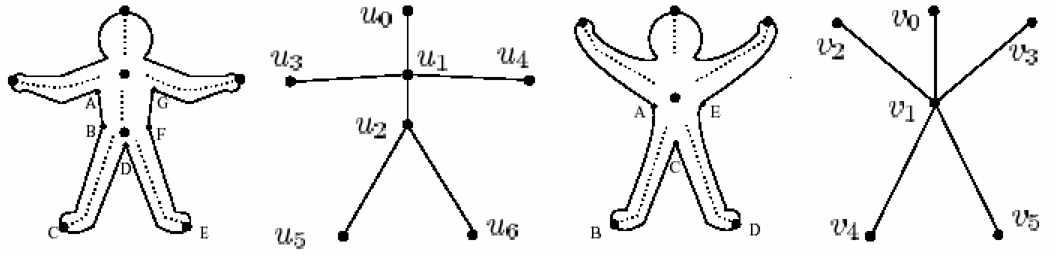


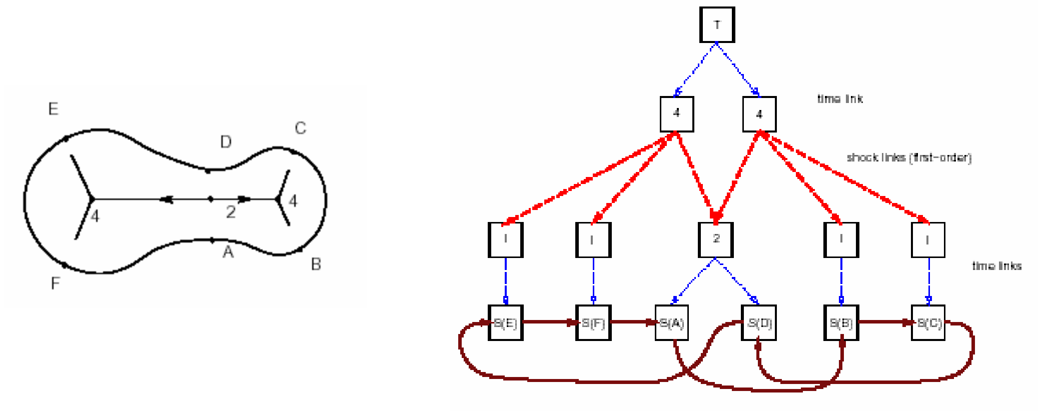
Figure 3-3 (Taken from [12]) Two shapes and their corresponding shape axis trees.

Although the tree structure obtained from the variational framework is robust to small boundary perturbations, instabilities can occur. In Figure 3-3, it can be seen that the shape axis tree of two similar human shapes are significantly different. To overcome this difficulty, the researchers consider not only node-to-node correspondences but also node-to-path correspondences. Some tree matching operators similar to skeleton operators used in FORMS [26] are employed to convert a path in the tree to a single edge. This newly formed single edge is matched with an edge in the other tree so that a node-to-path correspondence is formed. For instance, u_2 node in the first axis tree is merged with u_1 so that the path u_1 to u_5 is matched to the edge v_1-v_4.

Although the shape axis tree representation has been applied to shape matching, no recognition experiments have been reported.

3.4 Shock Graphs [20]

Kimia et al. proposes a graph representation based on shocks. The group of first-order shocks and third order shocks form the edges of this graph structure where second and fourth order shocks, bifurcation points, end-points of sym-branches constitute its vertices. The shock graph representation is illustrated in Figure 3-4.



(a)

(b)

Figure 3-4 (Taken from [20]) (a) An example shape and (b) its shock graph representation

When compared to other axis-based representations in the literature, the shock graph of Kimia et al. provides more information e.g. type of sym-points, direction of sym-branches. Nevertheless, morphological skeleton is computed from shapes. Prior to [20], no method has been proposed to account for the instabilities of the shock graph.

The main idea in this study is to overcome the difficulties of the representation in the matching process. The distance of two shapes is determined to be the sum of the cost of deformations that transforms one shape to the other. A low cost is assigned to the deformations that relate shapes on different sides of an instability. For instance, the shock graph topology of the two shapes in Figure

3-5 are significantly different. The deformation cost that transforms shape A to shape B is low so that the distance between these shapes are small. Likewise, a low cost is assigned for the removal of a skeleton branch which is mainly to overcome the instabilities resulting from boundary perturbations.

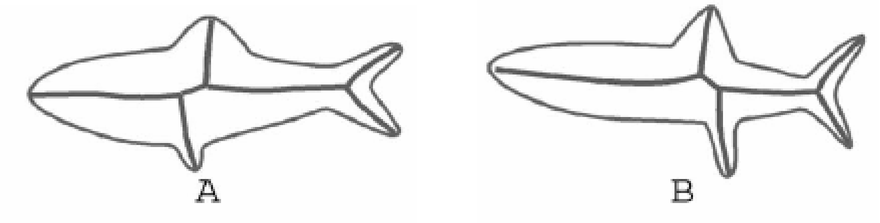


Figure 3-5 (Taken from [20]) A low cost is assigned to the deformation that transforms A to B.

The main issue with the instabilities of the symmetric axis is their detection. For instance, it is difficult to determine if a sym-branch is due to a major protrusion or a minor boundary perturbation. By assigning a low cost to the splice operation, which removes a sym-branch, it may be possible to determine the similarity of two similar shapes which have different shock graphs due to the instabilities. On the other hand, perceptually different shapes can be considered similar. For this reason, when sym-branches are compared, the boundary segments that they represent are also compared. To compare sym-branches and boundary curves, a curve matching algorithm is used that finds the optimal alignment of two curves and then determines the minimum deformation of one curve to another, where the cost is defined as the sum of stretching and bending energies.

For the matching of two shapes, an edit-distance algorithm is employed that finds the optimal deformation of one shape to another. Figure 3-6 shows some matching results. The sym-branches colored with gray indicate shape parts that are not matched. One thing to note is that the shock graph topologies of very similar shapes are different.

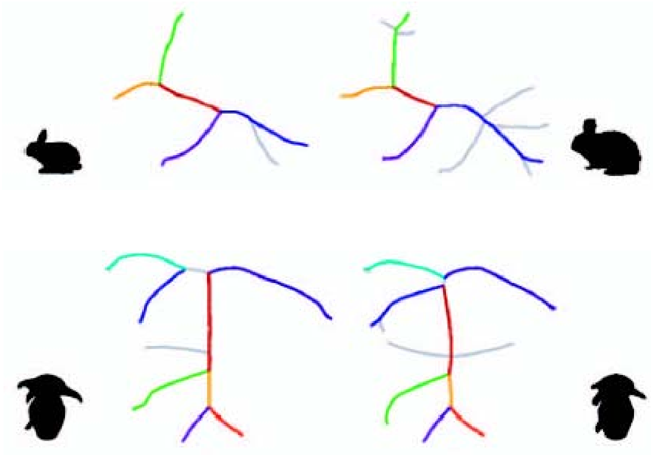


Figure 3-6 (Taken from [20]) Some matching results.

The matching algorithm is tested on a database of 216 shapes. While the number of shapes in the database is far more than that of other studies, there are only 18 categories. Moreover, the within category variability in terms of orientation, size, articulation of parts is not large. In most cases, the matching is algorithm finds the nearest neighbors of a query shape.

The reported matching results show that an absolute measure of similarity is not obtained (Figure 3-7). For instance, the distance between the first query shape and its nearest neighbor is 705. On the other hand, for the last query shape, this value is 186. A threshold must be employed to determine if two shapes are indeed from the same category. If this threshold is based on a value around 200 (considering the similarity of last query shape) then the first query shape would not be classified since its distance to its nearest neighbor in the database is much higher. If, on the other hand, this threshold were based on a value around 700, then the last hammer shape would be considered the same as a bone.

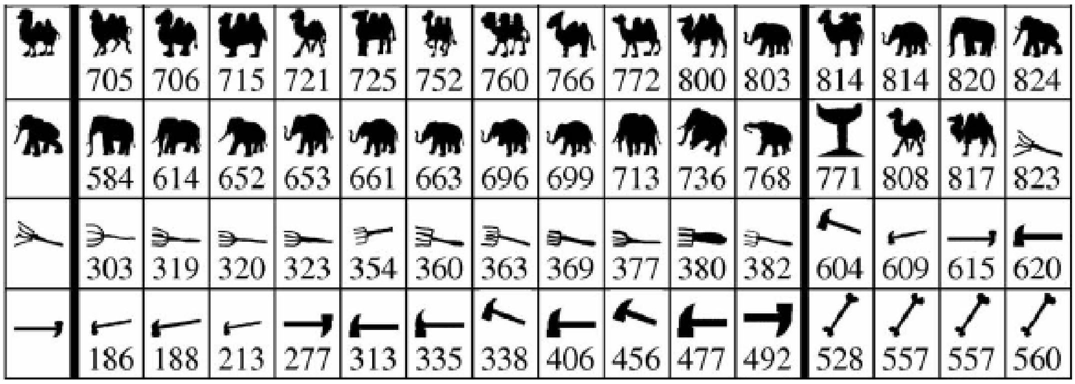


Figure 3-7 (Taken from [20]) Matching results for some query shapes.

3.5 Discussion

In all of the matching and recognition frameworks reviewed here, the organization of primitives (axes) is represented by graph or tree structures. The tree structure offers low computational complexity since polynomial time algorithms exist for tree matching problem. However, the representations reviewed in this chapter that use tree structures do not capture the planar order of primitives. For instance, the shapes in Figure 3-8 are considered equivalent in those representation schemes. This information may be necessary in various situations e.g. finding the correspondences of fingers between two hand shapes. The shock graph stores this information. The algorithms that match shock graphs and skeletons in Zhu and Yuille’s study are computationally more expensive since no polynomial time algorithms exist to match graph structures.

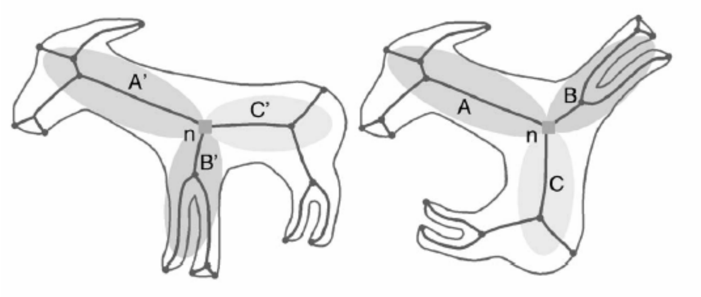


Figure 3-8 (Taken from [20]) Two shapes with the same topological structure.

The geometrical comparison of sym-branches or the boundary segments that they represent is used in [12,20,24]. This method is against the main motivation behind axis-based representations. Blum proposed SAT to push into background those traditional geometric properties. When the branches are scaled and rotated versions of each other, the geometrical similarity is high but in the case of articulation of a part, the geometrical similarity measure fails to signal similarity.

The matching approaches reviewed have been reported to perform well on small databases of shapes where the number of different categories does not exceed 20. Their performance on larger databases has not been examined.

CHAPTER 4

DETECTION OF SYMMETRY BRANCHES

The first step in an axis-based representation scheme is the accurate extraction of symmetry branches from images. In Chapter 2, it was argued that methods based on the prairie fire model have advantages over other methods. In particular, the shock grammar [23] and the method of Tari, Shah and Pien (TSP) [25] offer a combined framework for sym-point detection and axis regularization. When we compare these two methods, we see that the former one allows extraction of the traditional morphological skeleton, which the latter cannot. On the other hand, in TSP it is much easier to implement the boundary motion and axis regularization. The review of the recognition frameworks in the previous chapter showed that the problem of shape representation is most important. If the instabilities of the representations are common, complex measures have to be taken in matching and recognition to compensate for the deficiencies of the representation schemes. Therefore, the focus of this chapter is on axis regularization. A method like TSP removes the instabilities due to small boundary perturbations however there are other instabilities associated with the use of smoothing parameter. The goal of this chapter is to investigate those instabilities and to modify TSP to handle them.

In addition, whether the method of curve evolution or the method of TSP is used, these methods only provide the detection of sym-points of a shape. The grouping of sym-points into appropriate sym-branches is not a trivial task and is addressed in this chapter.

4.1 Detection of sym-points

TSP is composed of two steps:

- Computation of the v function (distance surface) whose level curves mimic the curve evolution, and
- Extraction of sym-points using the differential properties of the v function.

Using four illustrative cases, we discuss the major instabilities of the surface computation:

Case 1: Consider the rectangle and the surface $1-v$ computed from this rectangle in Figure 4-1.

The surface is computed using ρ = *8,* which is a reasonable value that is suitable for many shapes. It can be seen that most of the interior of the rectangle has remained intact (surface has the initial value of zero at these points). Because of the insufficient diffusion, the differential properties of the surface cannot be computed accurately and further shape analysis cannot be carried out.

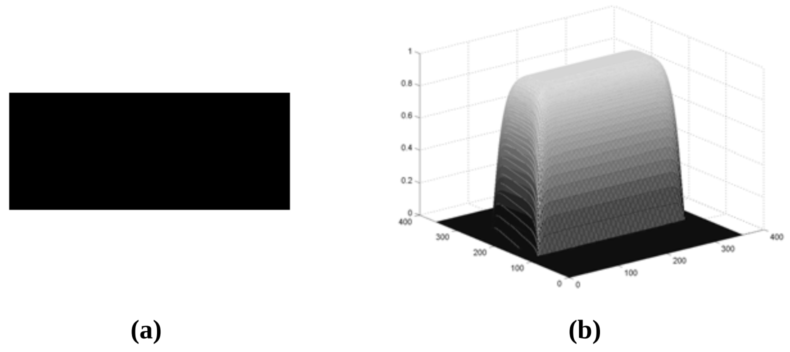


Figure 4-1 (a) A rectangle shape (b) Surface $1 - v$ ***(*** ρ ***= 8)***

This problem of insufficient diffusion occurs because the speed of a point on the curve depends on the interaction between nearby points. For the relatively thin parts of a shape, the opposite boundaries affect each other sooner than the broad regions and the speed of the level curves in these thin parts is increased. On the contrary, for broad regions in a shape, the opposite boundaries start to affect each other much later. If a smoothing parameter whose width is negligible compared to the width of the broad regions, the diffusion equation reaches a steady state without much affecting the inner parts of the shape. With insufficient smoothing, the surface v doesn't provide us with the shape analysis we require. It can be suggested that some sort of scaling and resizing may be performed before computing the function v. This would likely to fail, because the amount of smoothing required depends on the thickness of the limbs of a shape, rather than its overall size.

Case 2: Consider the vase shape in Figure 4-2 and the results of its surface computation and sym-point detection process.

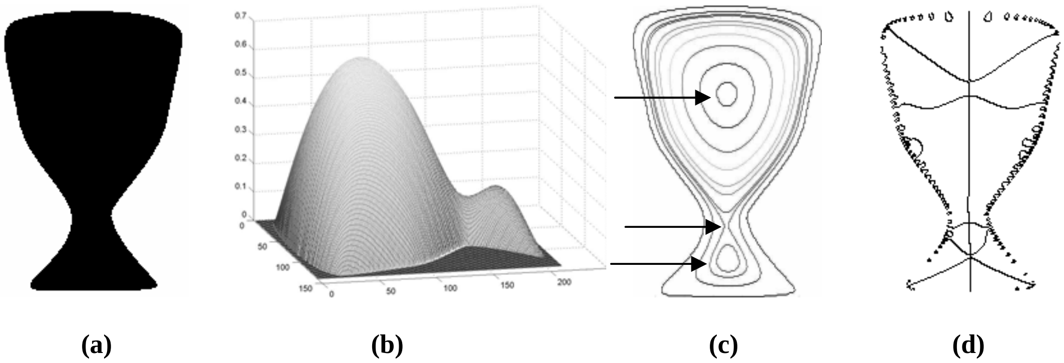
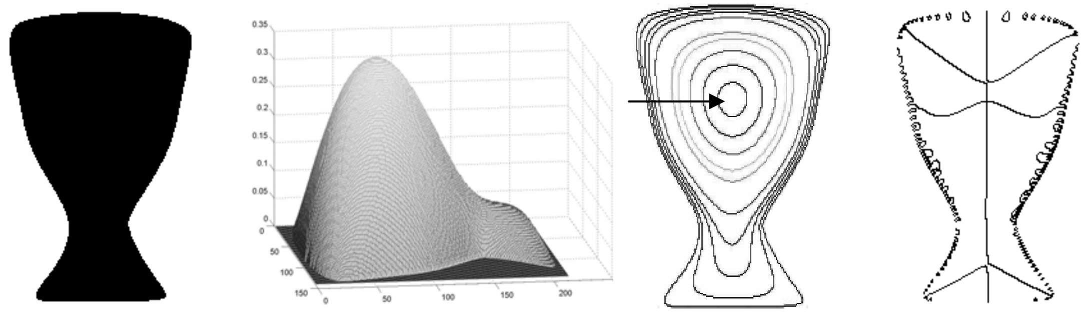


Figure 4-2 (a) The first vase shape to be examined (b) Surface 1-v obtained (*ρ =32)* (c) Level curves and the locations of special sym-points (d) Full sym-points

It can be seen from the surface 1-v that the shape contains two elliptic points and a hyperbolic point. The level curves of the function break at the hyperbolic point and continue towards the two elliptic points. As shown in Figure 4-3, the second shape whose symmetry points are to be determined is another vase shape which is identical with the first vase shape except a thicker neck.



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

Figure 4-3 The second vase shape (b) Surface 1-v obtained (*ρ =32)* (c) Full sym-points

Now, the topology of the shape changed significantly although the two shapes are very much alike. The second surface has only one elliptic point. This situation may frequently happen on the neck points and different descriptions for similar objects can easily be obtained.

Case 3: The effect of ρ on the symmetry axis and on the evolving level curves in TSP formulation is visible in Figure 4-4. As ρ gets larger, the points of high curvature on the level curves move faster, the protrusions are smoothed out earlier and less important sym-branches shrink. Also, the length of a sym-branch becomes an accurate measure of its importance as ρ gets larger. In morphological evolution, a curvature extremum and therefore the sym-branch tracking it survive until the end of the evolution (ignoring bifurcations). This is why the length of a sym-branch in a morphological skeleton does not reflect its prominence.

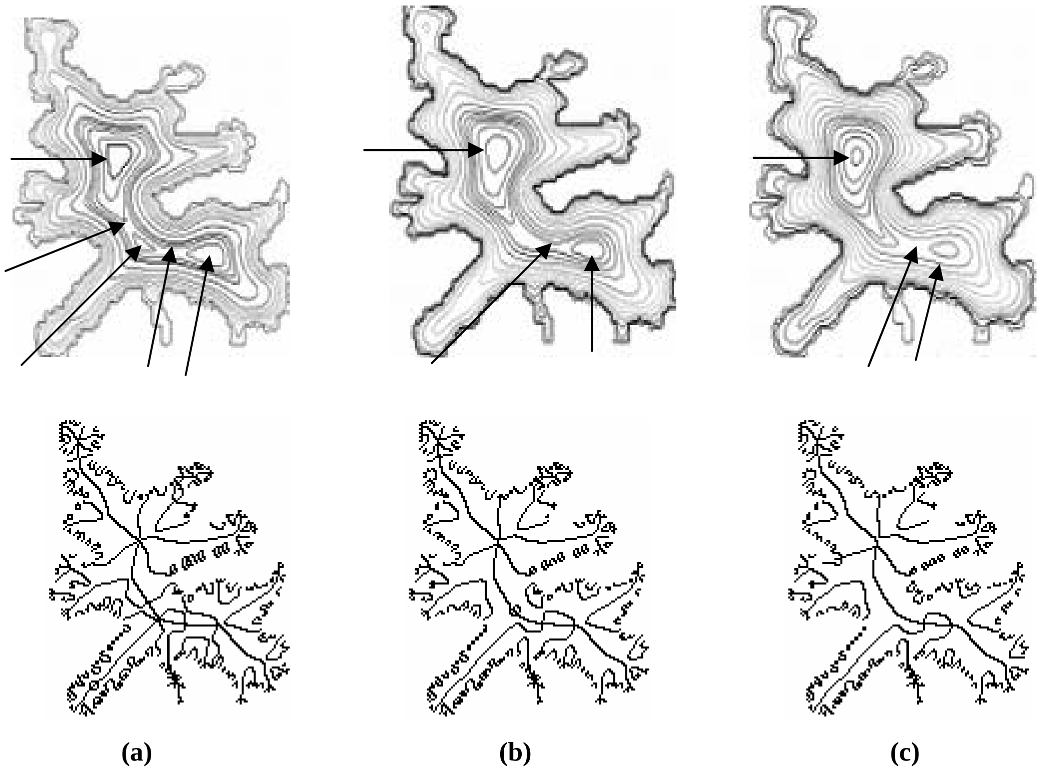
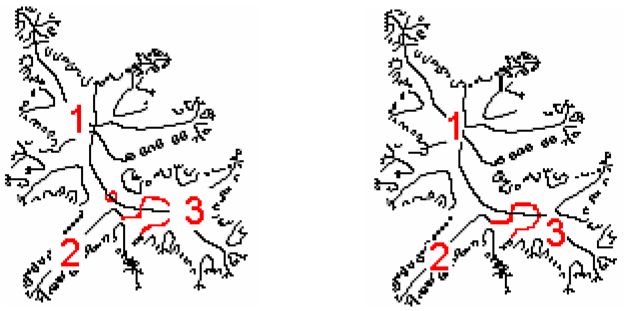


Figure 4-4 Sym-point detection using TSP formulation on a segmented MRI image. The smoothing values are (a) ***ρ = 4 (b) ρ = 8 (c) ρ = 16***

The level curve images in Figure 4-4 indicate the fact that in the case of ρ = 4, the evolving shape boundary splits into three curves and therefore shrinks into three different elliptic points. For ρ = 8 and ρ = 16 there are two elliptic points. The sym-points computed with ρ = 8 and ρ = 16 are examples of a pathological situation that can frequently occur with the TSP framework. These images contain some sym-branches that do not correspond to any protrusion or indentation of a shape. In Figure 4-5 these branches are colored with red. The reason for these computation artifacts is that the diffusion is stopped at such a critical time that the shape is between two different interpretations which differ in topological structure. In Figure 4-5(a), the computation is stopped when the shape was transforming from a shape with three major blobs to a shape with two major blobs. The circular sym-branch colored with red is due to the interaction of the elliptic point of the part numbered two and the neck point between part one and part two which can be seen in Figure 4-4(a). As Figure 4-5(b) shows, increasing the amount of diffusion caused this sym-branch to disappear since the topological change is complete. This time, the shape is between the state with two blobs (part 1 and part 2 together, and part 3) and the state with one blob. The red sym-branch is due to the interaction of the elliptic point of part three and the neck point between part two and part three.



(a) (b)

Figure 4-5 The sym-points of the segmented MRI image when (a) ρ = 8 (b) ρ = 16

4.1.1 New Surface Computation

If the shape representation scheme is to be used on a broad shape domain where a great variability on the thickness, length, width and size is expected, the level of smoothing required for each shape should be determined. This level, which is necessary to obtain stable descriptions, varies from shape to shape. The computation time increases as the amount of diffusion is increased. Then, the main issue is when to stop diffusion.

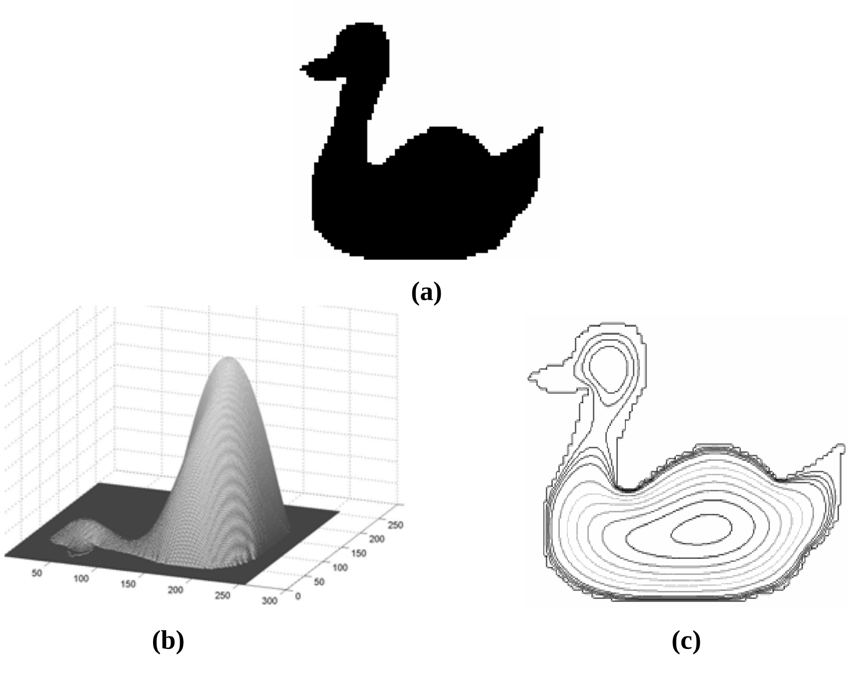
To overcome the topological instabilities illustrated in cases 2 and 3, the strategy employed in this thesis is *to select a small smoothing value and increase it until a function with a single extremum point is obtained which means that all the shapes are shrinked into a single point.* This calls for an iterative method and the following linear diffusion with Dirichlet conditions on the shape boundary is considered:

$$\frac{\delta v}{\delta \tau} = \nabla^2 v, \ v\big|_{\Gamma} = 1$$

This equation is derived from the Euler-Lagrange equation used in TSP. Letting $\rho \to \infty$ drops the second term of the TSP ($\frac{v}{\rho^2} \cong 0$). The solution of this equation is a function that is equal to one everywhere. The trivial solution of this formulation is not considered. The surface v is obtained at a critical time T during its evolution towards ones everywhere. Sufficiently evolved surface has a single elliptic point corresponding to a single shape center. Pure smoothing shrinks every shape to a single "round" point and it may be argued that it is perceptually unnatural to reduce shapes which deviate a great deal from being a circle. It should be pointed out that even under pure diffusion, the prominent parts of a shape are retained in its description. *The important thing is not the final state of the shape but how it evolved to that state.*

There is one practical difficulty associated with some dog-bone or dumbbell-like shapes where the two main parts of the shape have nearly the same prominence. It takes a significant amount of computation time to reduce these kinds of shapes to a single point. Therefore, it is logical to retain their dumbbell-like topology in the final description. Having two types of descriptions may lead to an instability when some shapes that are between these two types are encountered. This is a trade-off between computational efficiency versus accuracy. The situation with dumbbell-like shapes can also occur with shapes that have more than two nearly identical blobs connected with thin necks. These shapes are not considered in this thesis, because it is not essential to the ideas proposed here, and these kinds of shapes can be handled easily by incorporating additional checks. The representation scheme is able to describe many shapes, and shapes with more than two identical blobs are seldom encountered in practice.

Two things must be determined in the surface computation stage: The shape topology and the time T at which diffusion is stopped. The determination of a shape's topology is done after an initial diffusion stage. This is mainly to obtain a smooth surface that permits shape analysis (computation of sym-points). The initial diffusion step is stopped when it affects all parts of a shape. This can be controlled by checking the lowest value of the surface. After the initial diffusion, the ratio of the values of the surface at elliptical points is computed. Using a simple threshold, the type of the shape is determined. If this ratio is small, it means the two parts connected by the neck are not identical and it won't take long to obtain a surface with single local minimum. On the other hand, if these main blobs of the shape are nearly identical, the shape is considered a dumbbell-like shape. Consider the duck shape and the state of the surface after the initial diffusion step in Figure 4-6.



(a)

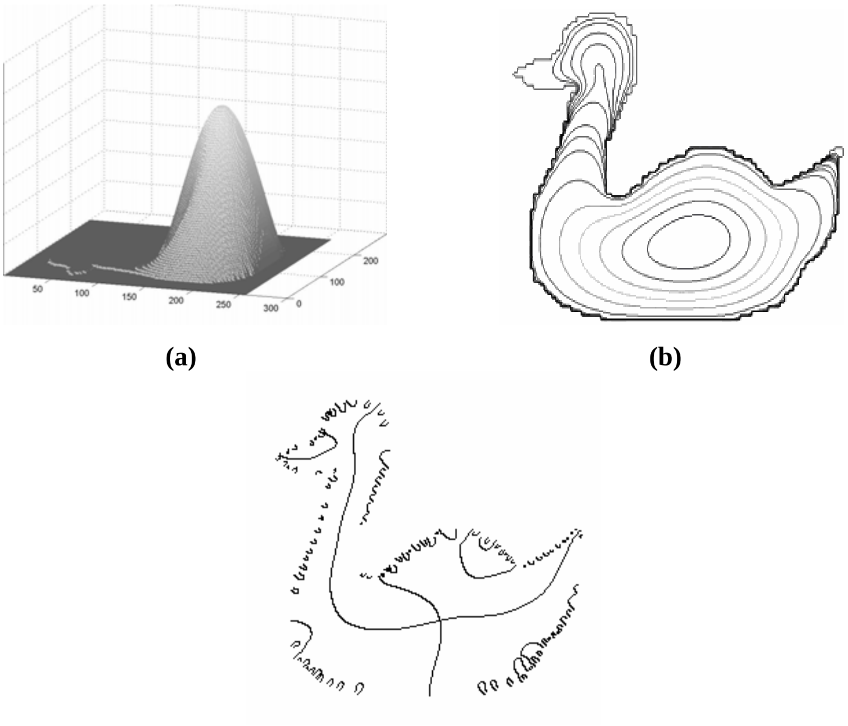
(b)

(c)

Figure 4-6 (a) A duck shape (b) Surface computed from the duck shape at the initial diffusion step (c) The level curves of the surface

After the initial diffusion step, the surface computed from the duck shape has two elliptic points. As the level curves indicate, the evolving shape boundary breaks at the neck of the duck and splits into two closed curves. These two curves evolve independently and shrink into points at the projection of the elliptic points of the surface on the image plane.

The surface values at these points are compared. Since there is great difference in this case, the duck shape is considered a single-blob shape. To obtain a surface with a single minimum the diffusion process is continued. After the smoothing becomes sufficiently large, the level curves of the surface simulate the shrinking of the duck shape into a single point. At this point, the diffusion is stopped and sym-points are computed from this surface. Figure 4-7 (a)-(b) shows the state of v and its level curves after the final diffusion. Now, the level curves do not break during the evolution and the elliptic point at the head of the duck shape is no longer present.



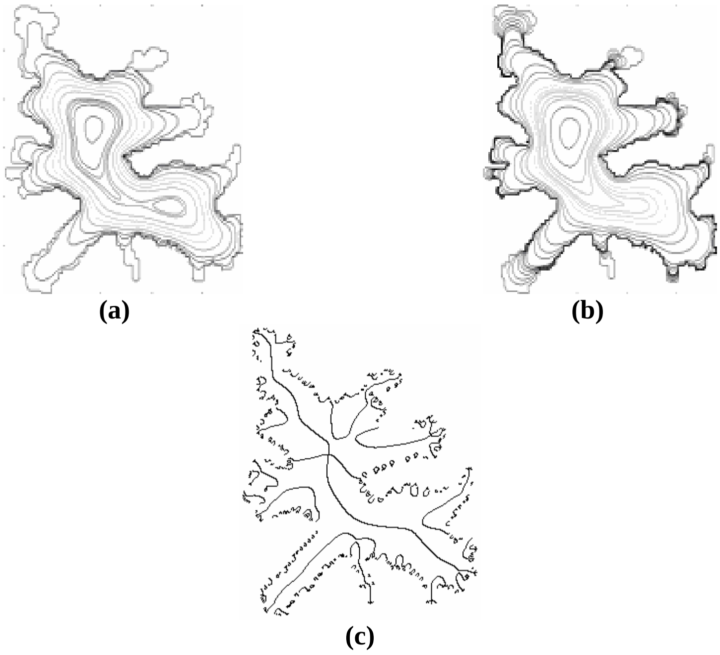
(a) **(b)**

(c)

Figure 4-7 (a) Surface 1-v after the final diffusion (b) The level curves of the duck shape (c) Sym-points of the duck shape. Arrows show the neck, which is not captured.

Despite the large amount of diffusion, the significant protrusions of the duck shape are represented by sym-branches. The termination of a branch tracking a protrusion depends on its prominence relative to nearby protrusions. This prominence is determined by its relative area since diffusion affects large areas later than the thin areas of the shape.

The diffusion of the surface until a single extremum is obtained prevents these pathological situations of the TSP formulation since there is no chance the computation is stopped in the process of a topological change. Figure 4-8(a) shows the level curves of the MRI image after the initial diffusion step. The level curves break at the neck point between parts two and three. After the final diffusion (Figure 4-8(b)), there is only one elliptical point and the resulting sym-point detection process from this surface contains no computation artifacts or unintuitive sym-branches (Figure 4-8(c)).

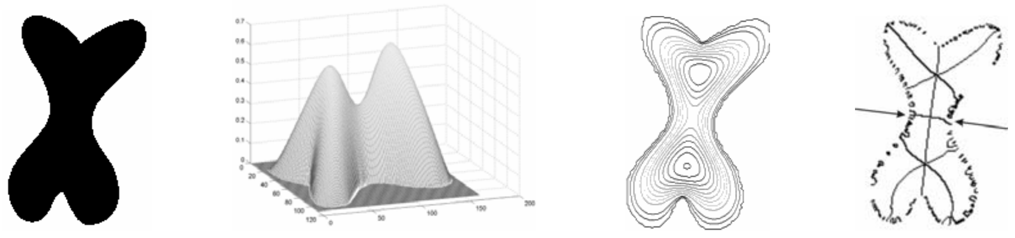


(a) (b)

(c)

Figure 4-8 (a) The level curves after initial diffusion (b) The level curves after final diffusion (c) Full symmetry points

An example of surface computation in the case of a dog-bone shape is shown in Figure 4-9. After the initial diffusion step, the topology of the shape is determined to be dumbbell-like as the two parts of the shape have nearly the same prominence.



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

Figure 4-9 (a) A dog-bone shape (b) The surface 1-v after the initial diffusion (c) The level curves of the surface. (d) Full sym-points. Arrows indicate sym-branches that capture the neck of the shape.

The idea of diffusing the surface v until a single minimum is obtained is not sufficient for stable axis-based descriptions. The surface of a shape with no neck points permits shape analysis after the initial diffusion step. Due to scale difference, descriptions of single-blob shapes may differ in detail. The solution to this problem is to diffuse the surface v until its minimum is above a predefined threshold. This approach ensures that the most major parts of different shapes are affected the same by the diffusion process. Figure 4-10 illustrates the scale invariance of the sym-point detection process.

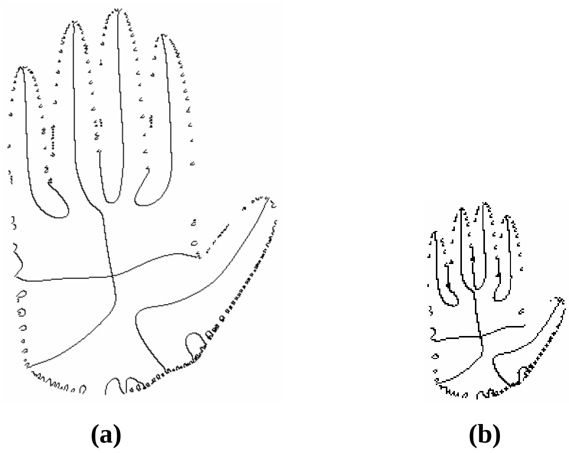


Figure 4-10 Sym-points of two hand shapes. Image dimensions are (a) 300x425 pixels (b) 100x141 pixels

To sum up the main ideas of the sym-point detection process, an iterative sym-point detection process is proposed. The diffusion is stopped when the surface v satisfies the conditions that it has a single minimum point and this minimum point is above a predefined threshold. Enforcing this constraint on the surface computation step offers the following advantages:

- A small change in the shape boundary does not cause a great change in its description. The length of a sym-branch is a true indicator of its prominence;
- The topological instabilities are greatly reduced since most shapes are considered to belong to only one category;
- Scale invariance is obtained. Only the most important branches are captured.

The computational details of the process is given in Appendix A.1 and A.2. Figure 4-11 shows the results of sym-point computation for some example shapes. Despite their obvious necks, the representation interprets them as a single blob.

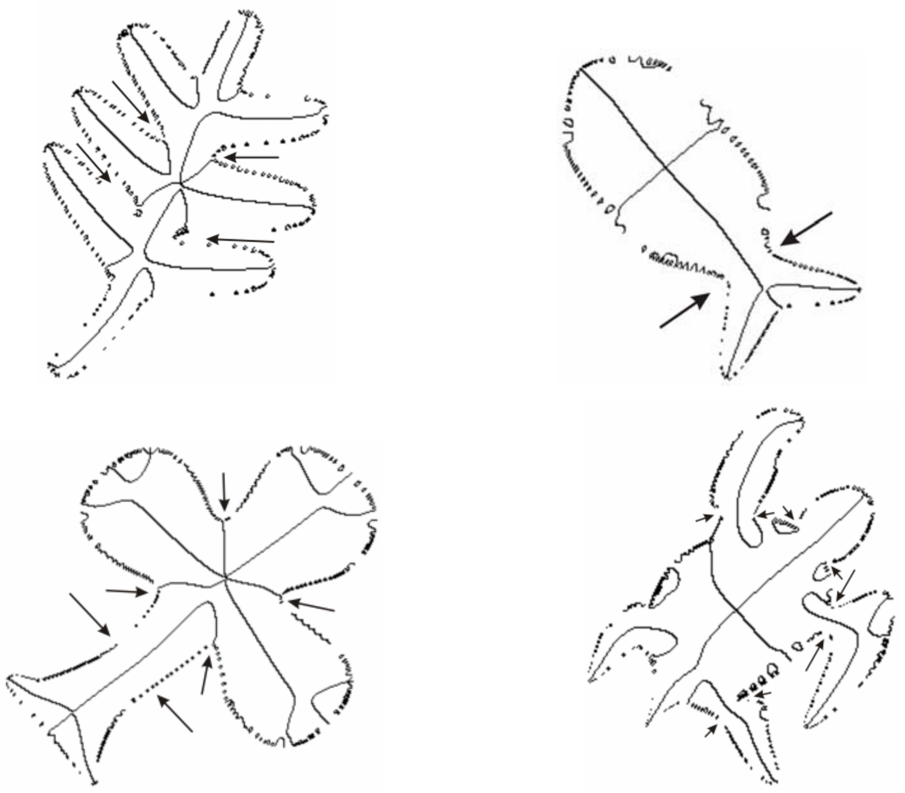


Figure 4-11 Full sym-points of some shapes with significant necks. The necks of these shapes are indicated with arrows.

4.2 Computation of sym-branches

A sym-branch is defined to be the differentiable curve which is a collection of smooth sym-points. In Blum's axis morphology and in some other skeletal representations, these branches constitute the edges of the graph representation. The result of curve evolution based shape analysis or TSP formulation is the set of sym-points. The next step in deriving the skeletal representation is to group sym-points into sym-branches. This is not a trivial task. The representation schemes that use the symmetric axis determine the symmetry branches by checking the connectedness of symmetry points. In [23], a set of rules to group skeletal points are presented. These rules define the valid groupings and transitions of sym-points, so that false sym-branches are discarded and correct sym-branches are extracted. In this study, the grouping is roughly based on the connectedness.

Determining sym-point connectedness is trivial. The symmetry axis can be extracted from a binary image just by considering sym-point connectedness. The main difficulty is to identify sym-branches and to get rid of the noise branches on the shape boundary. It must be ensured that sym-points that should not have been in the same branch should not be grouped into one branch. In addition, two sym-points that should have been in the same branch should not be grouped in different branches.

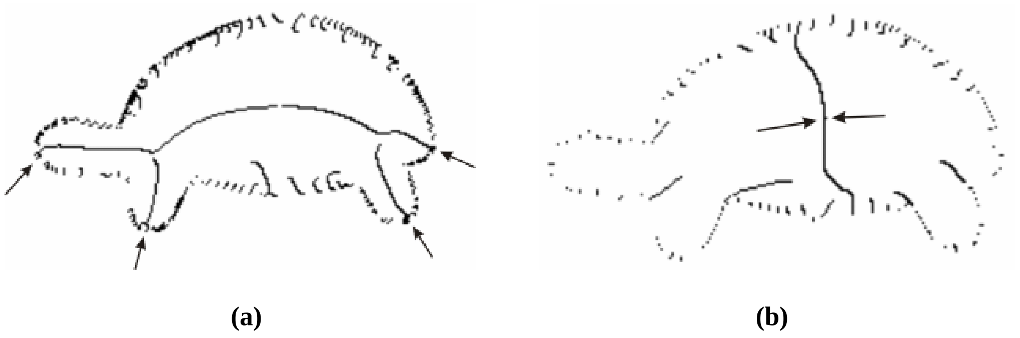


Figure 4-12 (a) Positive sym-points of a turtle shape. Arrows indicate the locations where major sym-branches connect with noise branches on the shape boundary. (b) Negative sym-points of the turtle shape. Arrows indicate the elliptic point (shape center).

The major positive branches in Figure 4-12(a) connect with small positive branches on the shape boundary (shown with arrows). An extraction procedure based only on connectedness will fail to capture only the major branches. Since the surface v changes monotonously, along a symmetry branch the value of the surface also changes monotonously. The major positive branches are separated from the small branches on the boundary using this property. The major branches in Figure 4-12(b) that reach the center point of the shape do not have any gap between them. Sym-branch computation process detects the special sym-point (shown with arrows) and disconnects the sym-ax to obtain two sym-branches. Of course, the positive and negative sym-points are grouped with points of the same type.

Even though smoothing is introduced by the curvature based motion, pruning methods are still necessary. In Figure 4-12 many noisy branches on the boundary can be seen since diffusion process regularizes the evolving boundary, not the exact shape boundary. For a stable shape representation, these small branches must be discarded. The traditional pruning methods on morphological skeletons employ complex measures to determine part saliency since the morphological skeleton does not provide any. In our work, the pruning is only used to remove the small noise branches. The saliency measure used is the length of a sym-branch. As explained previously, this measure reflects a part’s true prominence and it is easy to compute. Pruning is performed in two stages. In the first stage, which is right after the detection of the sym-branches, the smallest branches in the shape are removed to decrease computation time for the later stages. For instance, the sym-branch detection process detects more than 100 branches in the turtle shape in Figure 4-12. However, only about 15 sym-branches should be considered for later processing. The branches whose length is shorter than a predefined threshold value are discarded. This threshold value is very small so it is not possible to lose a significant branch of a shape. The second pruning stage is performed after it is determined which sym-branch connects where. In TSP formulation every sym-branch terminates either at a junction with another branch of different type or at an extremum point. The branches that do not satisfy this requirement are simply the ones due to noise and which were detected by the branch detection process erroneously. The second pruning step discards those branches.

In Appendix A.3, the algorithms for sym-branch computation are given. The sym-branches of some shapes after pruning are shown in Figure 4-13.

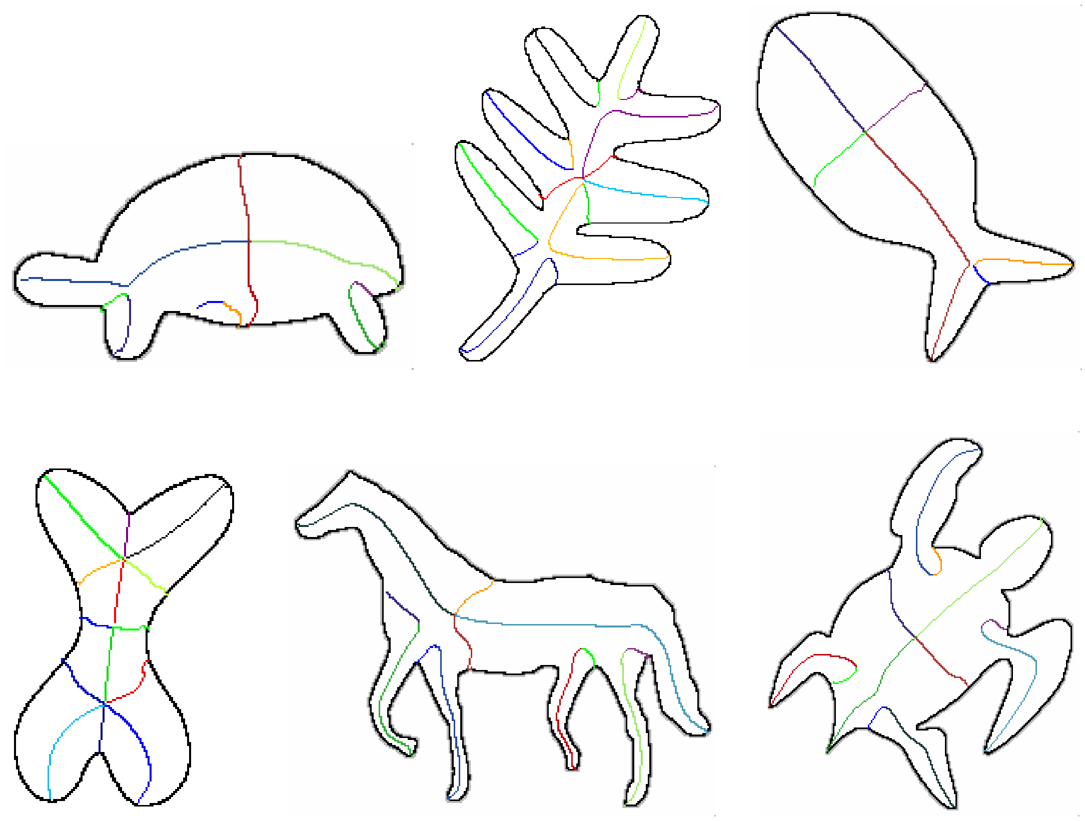


Figure 4-13 Colored sym-branches of some sample shapes after pruning.

CHAPTER 5

REPRESENTING SPATIAL RELATIONS

In the previous chapter, the problem of sym-branch extraction is addressed and a method to detect the local sym-branches which correspond to the prominent protrusions and indentations of a shape is proposed. It is shown that these local sym-branches are associated with stable properties of shapes. Defining primitives based on the local symmetry branches and describing the relations among primitives is the topic of this chapter.

In shape matching where we find the best correspondence between two shapes, the organization of primitives are used to enforce coherence of the matched pair of sym-branches. In traditional shape skeletons, the organization of sym-branches is described in a graph or a tree structure. In order to capture the relations of sym-branches in such structures, connectivity must be guaranteed in the sym-ax extraction step. Otherwise, it would be much difficult to determine which sym-branch connects where.

Our method of describing the relations of shape primitives is quite similar to the one proposed in the 3D model representation of Marr and Nishihara [15]. The ideas proposed by Marr and Nishihara, have formed the most important guidelines for generic shape recognition. Since our axis-based representation is along the same direction, it is appropriate to describe the approach by considering it in the context of Marr and Nishihara's and representation scheme.

According to Marr [14], three general choices in the design of a representation are:

- The description elements,
- The coordinate system used,
- The organization of the information in descriptions imposed by the representation.

In the 3D model representation, the description elements are the generalized cones [1], which are based on the natural axes of an object. The locations of shape primitives are described in a coordinate system defined by the viewed object. This coordinate system is formed by designating one of the prominent axes of a shape as the principal axis. The locations of other axes are described relative to this principal axis. In order to satisfy the opposing goals of stability and sensitivity, the information that captures the more general and less varying properties of a shape must be decoupled from information that is sensitive to the finer properties of the shape. This leads to a hierarchical organization of information. Marr and Nishihara proposed to form this hierarchy based on the axes of a shape. Figure 5-1, taken from [15], illustrates the idea well:

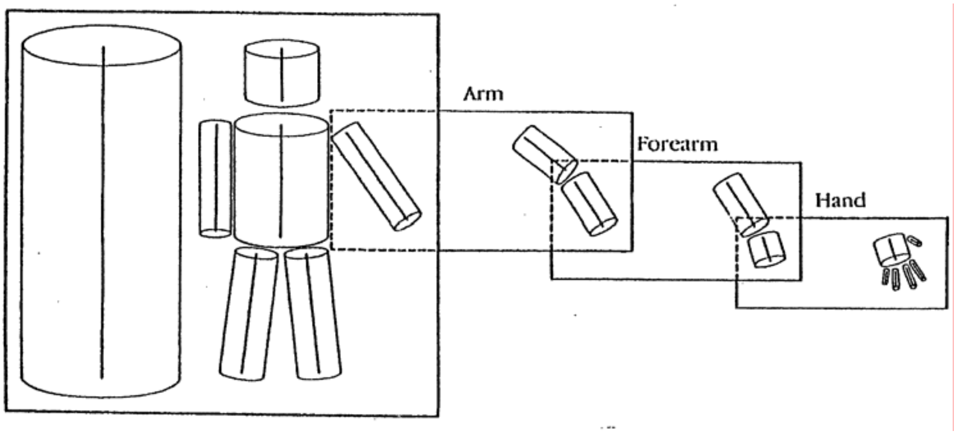


Figure 5-1 (Taken from [15]) The hierarchy of shape information in Marr and Nishihara's model.

Each box in Figure 5-1 describes a 3D model. In each 3D model there is a model axis (shown on the left side of each box), which is a single axis defining the extent of the 3D model. It provides coarse information about properties such as size and orientation about the overall part described. For the 3D models describing the shape parts, the natural axes of a shape can be used as model axes for shape parts. The coordinate frame of the shape part can be defined using its model axis, or one of its major axes. This axis is called the model's principle axis. The spatial arrangement and measurable properties of other axes contained within the spatial context specified by the model axis are defined relative to the principal axis. For the top level, the symmetry branches of the shape form the component axes and the model axis must be determined explicitly.

The implementation of this idea is not trivial. It should be pointed out that it is impossible to derive the 3D model description from images without ambiguity. There is always information loss in the depth dimension because of the projection of the 3D world to the 2D image plane. Marr and Nishihara proposed ways to determine the principal axis of objects from images, but they pointed out the difficulties when unconventional views of objects are encountered. Because of the impossible task of deriving the 3D model representation directly –without an intermediate description- this model seems to be more suited for the *internal* descriptions of shapes. These internal descriptions can be formed through experience. Only after the object is viewed from many directions to resolve any ambiguity or uncertainty, its 3D model description can be precise. The description obtained from a single 2D image of an object can serve as an intermediate description and the other issues will be considered only for silhouettes of objects.

5.1 The Canonical Coordinate Frame

The model axis for the 3D model serves two purposes: It determines the canonical coordinate frame of the shape and provides coarse information about the shape such as size and orientation. This axis has to be determined by the most salient geometrical characteristics of shape so that the changes in viewpoint, articulation of the parts of a shape do not affect the chosen axis.

Our approach is to determine a *reference axis* that is only used to define the object centered reference frame. It depends on the fact that if a sym-branch survives long enough, it comes to rest at a shape center or a neck point. There are always at least two positive and two negative sym-branches that flow into a shape center (elliptic point) [26]. These branches represent the most prominent features of a shape and they will be called *major sym-branches* (Figure 5-2).

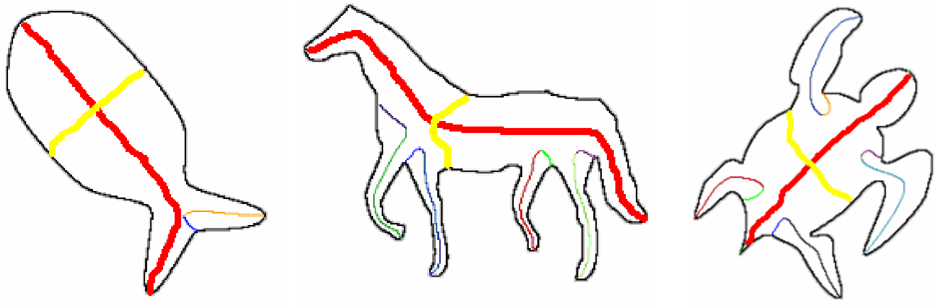
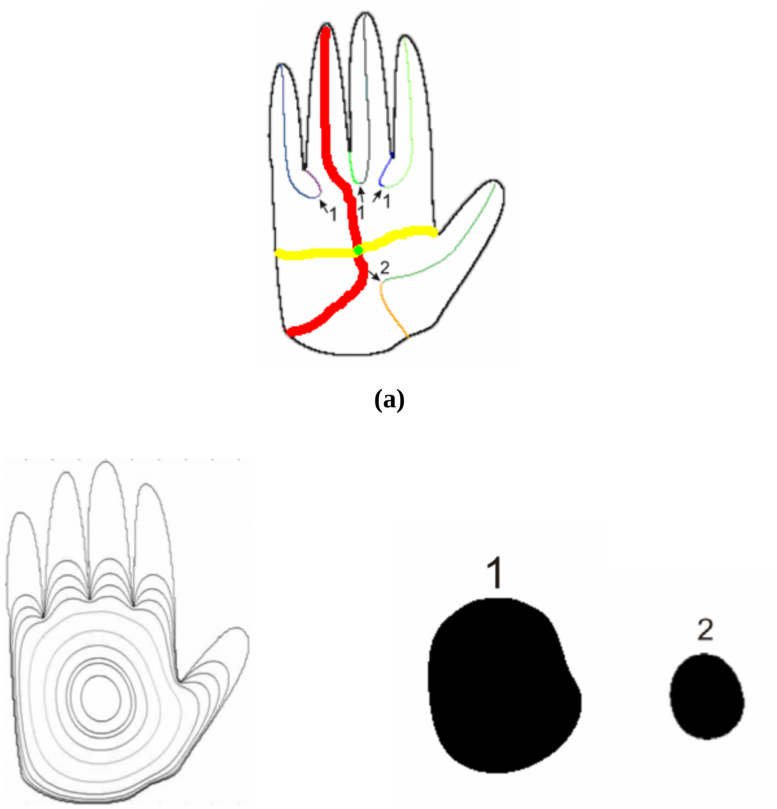


Figure 5-2 The major sym-branches of some example shapes. The branches colored with red are positive sym-branches whereas the ones colored with yellow are negative ones.

During the evolution, when all minor branches have terminated at junction points, the resulting shape includes only the most significant branches and it can be considered as the coarsest description of the original shape. A shape may undergo changes in scale, rotation, and viewpoint. It may also undergo non-rigid transformations such as articulation and boundary perturbations. However, the coarsest structure will remain almost the same. Figure 5-3(a) shows the sym-branches of a hand shape. The shape center is indicated with a green dot. The sym-branches marked with red are the major positive sym-branches whereas the ones marked with yellow are the major negative sym-branches. During the evolution of the boundary curve of the hand shape, some positive sym-branches merge with negative ones and terminate. The branch termination points are indicated with arrows. The level curves of the evolving shape boundary and states of the shape after these branches terminate is shown in Figure 5-3(b). The first blob represents the state of the hand shape after three sym-branches corresponding to three fingers terminate. Notice that the protrusion corresponding to the thumb is still present. The sym-branch corresponding to the thumb is more prominent and it terminates later in the process. The second shape represents the state of the shape when all the branches except the major ones terminate. The shape becomes, in its coarsest form, an ellipse. From that point only the two positive sym-branches and two negative sym-branches continue towards the shape center and survive until the end of the evolution. The three positive sym-branches corresponding to the three fingers of the hand shape terminate almost at the same time.



(a)

(b)

Figure 5-3 (a) The sym-branches of a hand shape. (b) The level curves of the evolving shape boundary and the states of the hand shape at the times the sym-branches indicated by the numbers in (a) terminate.

The center point and one of the major sym-branches allows one to set up a canonical coordinate frame (Figure 5-4). Any one of the major sym-branches can be selected. No matter which major sym-branch is designated as a reference axis, the same branch must be chosen for similar shapes. Since there are two major sym-branches of the same type, there is an ambiguity in the process. If the descriptions of two similar shapes depend on different coordinate frames, the matching algorithm will be unable to determine the similarities of shapes. This situation necessitates creating at least two descriptions since there is no exact method to select the same major sym-branch among the two possible. The method used in this thesis, is to divide the shape into two halves and represent these halves in their own coordinate frame. This does not solve the ambiguity problem and is only considered for decreasing the computational complexity of the matching process, which will be explained in the next chapter. For each shape, two different descriptions are stored in which the starting reference axis is different. In each description, the shape consists of two subshapes.

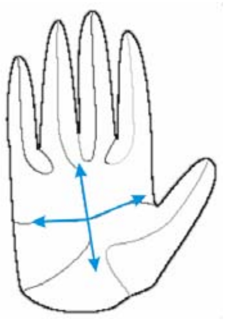
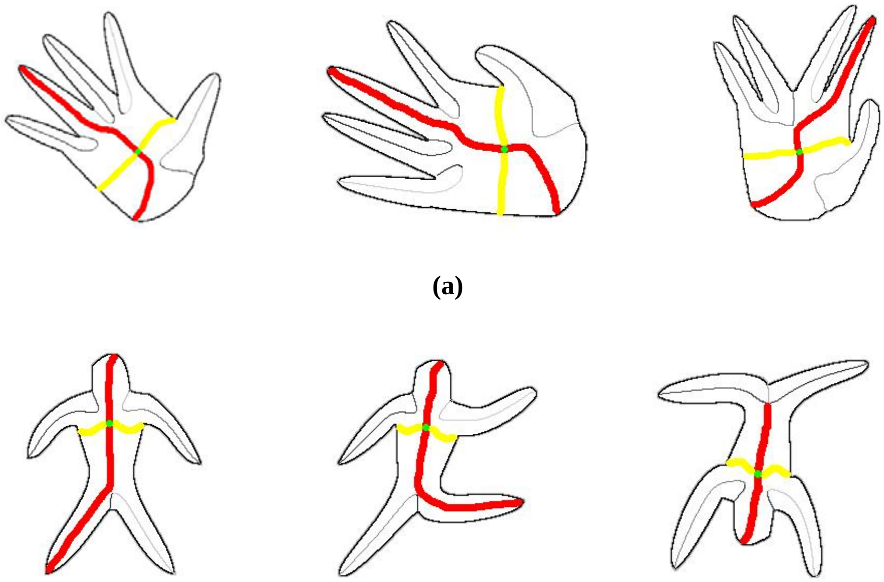


Figure 5-4 Four possible reference axes of the hand shape

Since the model axis of a shape must be determined explicitly, the major positive sym-branches of shapes seem to be suitable for that purpose. However, as we move away from the ellipse representing the coarsest form, these major positive sym-branches may bend or even bifurcate, hence, stable properties from the whole sym-branch cannot be obtained. Figure 5-5 shows the difference in major sym-branches among similar shapes. This difference frequently occurs on positive sym-branches because of articulation. The reference axis is defined as the line connecting the shape center to a nearby point on the selected major sym-branch. This point is chosen within the ellipse representing the coarsest form. The coarse properties of the shape provided by the model axis in 3D model representation can be obtained by other means. For instance, size information can be retrieved from the sum of the lengths of all sym-branches of the shape.



(a)

(b)

Figure 5-5 The major sym-branches of (a) the hand shape and (b) a human shape.

When defining an object centered reference frame, the main consideration is that it must be stable. The coordinate frame is defined based on the most salient shape properties, which we expect to be invariant under most transformations. Indeed, unless a shape undergoes a transformation that includes a change in the depth dimension, the coarse structure of the shape remains nearly the same. A large change in viewpoint direction and an occlusion of a part are examples of changes that have effects in the depth dimension. These situations must be handled by other means because we cannot rely on information that we cannot measure accurately.

The major sym-branches of a shape may be organized in ways that are more complicated. More than two branches of the same type may reach the center of the shape. Moreover, when a sym-branch reaches very near the shape center (apart from the two major sym-branches that flow into the shape center), a small change in the shape may result in this minor branch becoming a major branch. These situations indicate that the symmetry of the shape is more than two-fold. If the reference frame changes when a shape undergoes transformations, the resulting description will be useless. Consider the shapes in Figure 5-6. In Figure 5-6(a) three negative sym-branches flow into the shape center. A straightforward solution may be using the three major negative sym-branches to describe one-third of the shape but this solution does not handle the instability that is shown in Figure 5-6(b). The reference axes of these two shapes belonging to the same class are different.

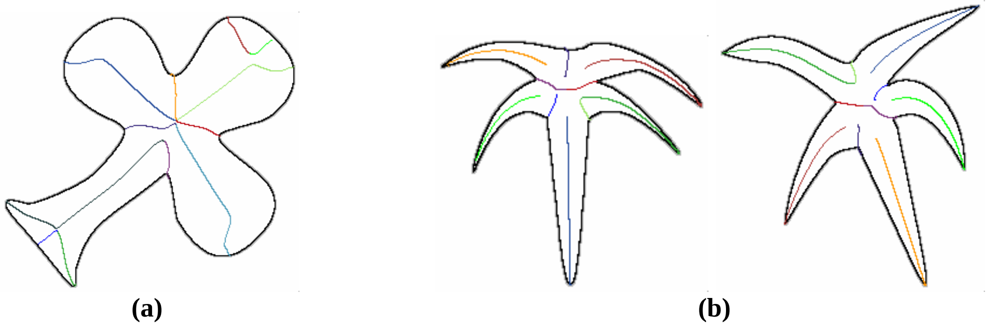


Figure 5-6 The instabilities associated with reference axes that can lead to matching failures.

Similar to the idea in [26], a simple solution is to interpret these situations as the ambiguities of the representation and to generate a number of possible descriptions of shapes. If there are n major axes that reach the shape center, all the two permutations of n major axes are selected to generate possible descriptions (Figure 5-7).

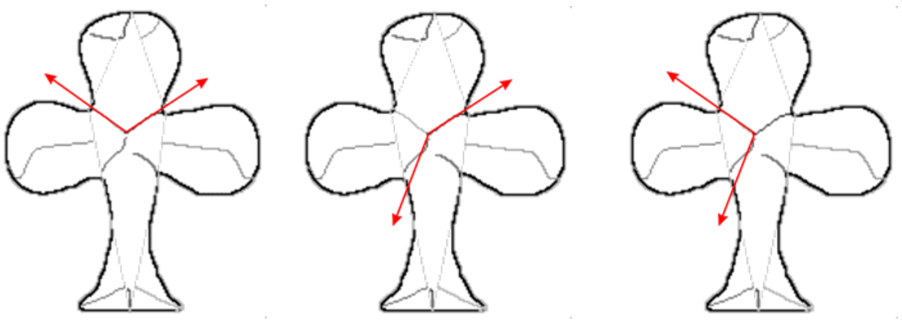


Figure 5-7 The reference axes of possible coordinate frames of a shape whose symmetry is more than two-fold (See Appendix A.4).

For a dog-bone or dumbbell-like shape, one of the three extrema may be chosen as the center of the shape. One solution is to generate three descriptions of the shape based on setting up three different coordinate frames centered on these three extrema points. Shapes with more complicated topological structure can be described using this approach. However, the fact that each hyperbolic point of the surface has at least two positive sym-branches with negative curvature [26] removes this ambiguity (Figure 5-8). These positive sym-branches are designated as reference axes, and the rest of obtaining the description of a shape is the same as in the case of single-blob shapes.

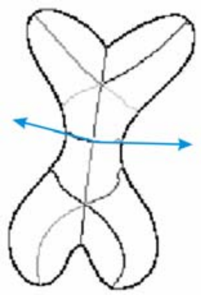


Figure 5-8 Possible reference axes for a dog-bone shape (See Appendix A.4).

5.2 Spatial Organization of Symmetry Branches

Once the coordinate frame is set up, it is easy to describe the relative placement of sym-branches. The locations of sym-branches can be defined by a vector in the chosen coordinate frame. For each sym-branch, the distance of the branch from the shape center, and the angle between its position vector and the reference axis are stored. This gives the position of the symmetry axis in polar coordinates. The termination points are used to describe the location of a sym-branch because the points where they connect to the main part of the shape tend to remain the same when the shape's limb articulate. A minor issue here is to describe the location of the major positive sym-branches that reach the shape center. These sym-branches do not give accurate information about their location. It is not known where the protrusions, which these major branches represent, connect to the main part of the shape. This issue is solved based on the fact that each positive axis has one negative axis to its left, and one negative axis to its right along the shape boundary. The points where the protrusions represented by the major positive sym-branches connect to the main part are determined by finding the intersection of the positive sym-branch with the line drawn between its neighboring negative sym-branches. The resulting sym-branches of the hand shape are shown in Figure 5-9.

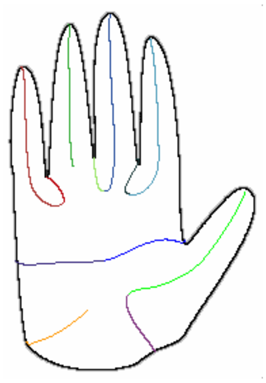


Figure 5-9 Sym-branches of the hand shape after the cut operation.

Figure 5-10 shows the reference axes in the hand shape and (r,θ) pairs for its symmetry branches. The locations of the sym-branches at the top half of the shape are described using the reference axis on the right, and the locations of the branches at the other half are described respect to the reference axis on the left.

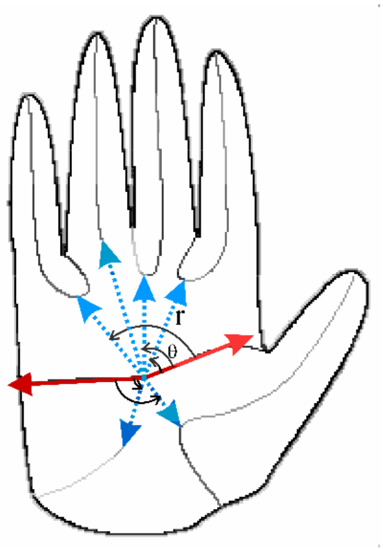


Figure 5-10 The reference axes (red) and the position vectors (blue) of the sym-branches of the hand shape.

Respecting the planar order of shape primitives is important for perceptually correct matchings. By storing the locations of shape primitives, the order information is captured. On the other hand, the location is not the only feature. The length of a sym-branch, which is a true indicator of its prominence, is also used. Therefore, the location feature by itself is not sufficient to prevent incoherent matchings. For that purpose, an order feature is employed. Sym-branches are added to a subshape in a counter clockwise direction, hence, the array of symmetry branches is sorted in ascending order of their angle with reference axes. Representing the order of the sym-branches along the shape boundary allows one to sort out impossible correspondences in the matching process. If A, B, and C are the three consecutive symmetry branches in the first shape, and A', B', and C' are the three consecutive symmetry branches in the second shape, the order constraint prevents the mapping of C to B' if A is mapped to A' and B is mapped to C'. This results in perceptually more plausible matchings and reduces the computation time of the matching algorithm. The algorithms for describing the location and spatial order of branches are given in Appendix A.5.

In order to cope with scale changes length feature should be normalized. We use the total branch length as a reference. Hence, the ratio of the length of the branch to the total length is used as the normalized length. The normalized length provides a scale invariant feature while the total branch length provides information to discriminate scale.

CHAPTER 6

MATCHING AND RECOGNITION

In the previous two chapters, the design of our axis-based representation scheme is explained. In this chapter, matching and recognitions problems are addressed by developing a matching framework based on the new representation scheme. Several matching examples are presented which show the robustness of the representation under visual transformations such as scale, rotation, and articulation. Finally, the recognition performance of the matching process on a moderate sized shape database is shown.

6.1 Data Structure for Shape Matching

The information stored in a shape's description is shown in Table 6-1. The primitives used in the matching step are the local symmetry branches. The features of a branch used in comparing it with another one are its type (whether it is positive or negative), its location (r,θ) and its relative length. The sym-branches that are designated as reference axes are compared based only on their relative length. The sym-points of a branch are stored in its description since additional properties can be derived from the differentiable curve they form. For each sym-branch, links to its neighboring branches are kept so that the planar order is captured in the description of a shape.

Apart from the information about the individual sym-branches, some properties of the shape that provide overall information about the shape are stored. In Marr and Nishihara's 3D model, the model axis provided the coarse information about a shape such as size and orientation. Here, the total length of branches indicates the total size of the shape. The relative lengths of sym-branches are computed from this information.

Table 6-1 Data structure for shape matching

Description element	Information Stored
Shape	Center Point (x_0, y_0) Total Length of Branches Orientation of Reference Axis $\{m_0, m_1\}$
Local Symmetry Branch	Type (Positive, Negative) Location (r, θ) Normalized Length Reference Axis (Yes, No) Sym-points Next Symmetry Axis Previous Symmetry Axis

An important thing to consider when designing a representation scheme is that variant descriptors should be incorporated in the descriptions. The invariance to changes in scale and rotation are desirable but there are situations in which transformation variant descriptors must be used, e.g. discriminating '6' from '9'. For that purpose, those properties necessary for deriving the variant properties of the shape are stored in its description. Specifically, the location of the shape center in extrinsic coordinate frame can be used for translation variance. The total length of branches can be used to differentiate scale, while the orientation of the reference axis in extrinsic coordinates provides the orientation information of a shape.

6.2 Shape Similarity

When the descriptions of shapes consist of collections of primitives, it is usual to determine the similarity of two shapes by comparing their primitives and summing the similarity scores of their matched pairs of primitives to arrive at a total similarity score. To compare sym-branches individually, four features are used:

- Type (Positive or Negative)
- r

- θ
- Normalized Length

The problem is how to compare individual features and how to arrive at a similarity score based on these features. For each feature, a normalized scale that is between zero and one is used. The type information can be separated from the other features since the comparison of a positive and a negative branch is semantically wrong: One corresponds to a protrusion and the other to an indentation. If the types of two branches are different, their similarity should be zero.

The main issue when determining the similarity based on location and length features is the precision with which they are distinguished. In 3D model representation, Marr proposes to store location information with varying degrees of precision. For instance, the location that the arm component connects to the torso may be specified precisely whereas its orientation may be defined with little restriction. One possible way to do that, as proposed by Marr, is to associate location features with angular and linear ranges. This is accomplished by dividing the feature space into bins so that the tolerance or precision is determined by the range of the bin. In a hierarchical representation scheme, precision can be increased as the level of detail increases.

In our representation scheme, the restrictions are on the location and length features. Rather than storing this information in a branch's description, it is enforced during comparison.

6.2.1 Score Computation

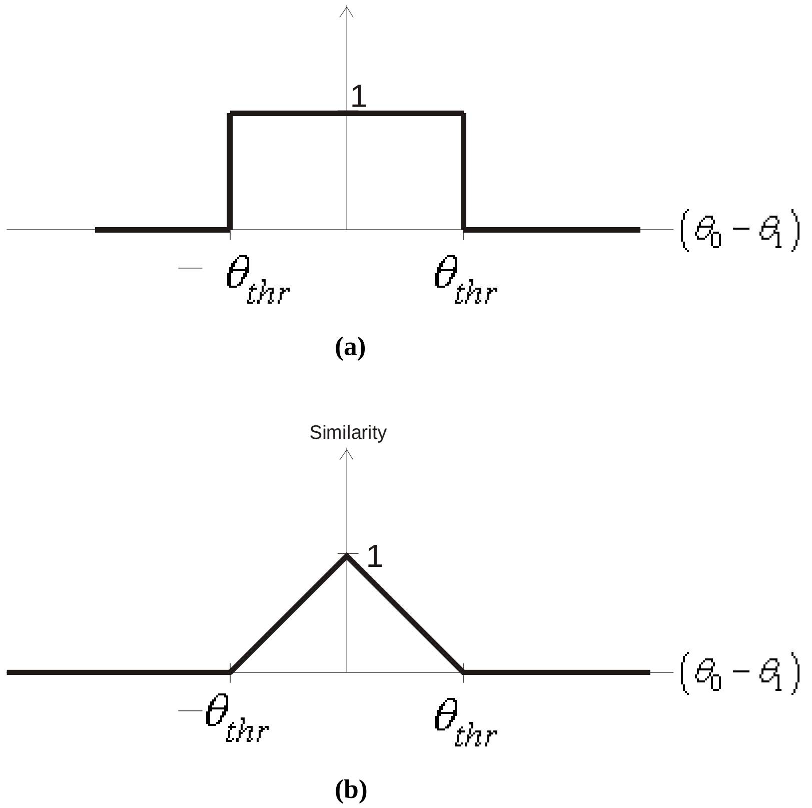
An obvious choice for similarity is the absolute difference. A threshold may be defined so that the similarity score is set to zero when the absolute difference exceeds the threshold. The similarity score for the location features r, θ are:

$$r_{sim}\left(r_0, r_1\right) = \max\left(0, 1 - \left(\frac{\left|r_0 - r_1\right|}{r_{thr}}\right)\right)$$

$$\theta_{sim}\left(\theta_0, \theta_1\right) = \max\left(0, 1 - \left(\frac{\left|\theta_0 - \theta_1\right|}{\theta_{thr}}\right)\right)$$

While the idea is easy to implement, it has shortcomings. The threshold must be carefully selected. If the threshold is adjusted so that the tolerance on the maximum differences is small, similar branches may not be matched which causes a low similarity score for the matching of two shapes belonging to the same class. On the other hand, if the tolerance is large, then discrimination capability is reduced. Using tolerance ranges in the descriptions as in 3D model representation of Marr is also subject to similar problems. A slight change in the location of a limb may cause its description to change significantly. The higher levels would not have this instability problem because of wide ranges but the instabilities at the lower levels must be handled using the information from the higher levels. In order to combine the similarities of different features, a weighted sum of these features can be computed. One important thing to ensure is that if the similarity of any feature is zero then the total similarity score should be zero. This is because the weighted averaging can produce a high similarity score for two branches that are similar in some respects but very different in others.

Notice that the above idea uses a square window function demonstrated in Figure 6-1(a). Inside the window the similarity score is a linear function of the absolute difference and the score is zero outside the window Figure 6-1(b).



(a)

(b)

Figure 6-1 (a) Square window function (b) Similarity function inside the window

As a remedy, we propose to replace the sharp cut-off with a softer transition. Gaussian distribution function is a good candidate:

$$\theta_{sim}\left(\theta_0, \theta_1\right) = \exp\left(-\left(\frac{\left(\theta_0 - \theta_1\right)^2}{2\sigma_\theta^2}\right)\right)$$

The idea is to consider a Gaussian distribution whose mean is the feature value of the first branch and whose standard deviation is provided externally. The similarity of two branches is simply the probability of the feature value of the second branch in the Gaussian distribution. Figure 6-2 illustrates the idea. The mean of the distribution is the normalized length of the first branch (0.176). The probability of 0.202 (the normalized length of the second branch) in this distribution is 0.83. This value is used as the similarity score.

Sym-Branch 1	Sym-Branch 2
Normalized Length: 0.176	Normalized Length: 0.202
r: 0.027	r: 0.028
θ: 2.85	θ: 2.14

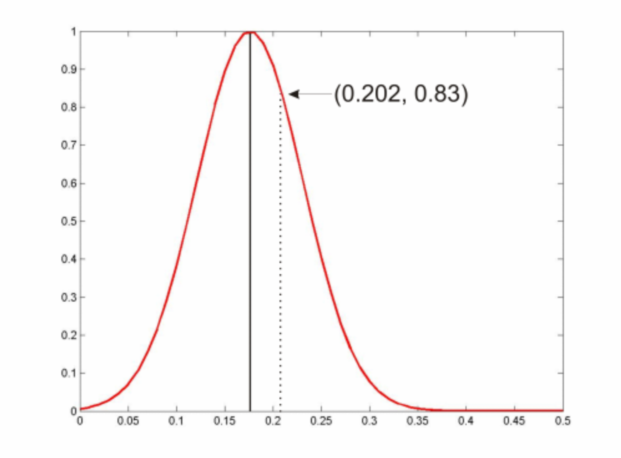


Figure 6-2 The probability of similarity based on the normalized length feature.

There are two alternative computations for branch similarity. A weighted averaging of the probabilities derived from using univariate Gaussian distribution for each feature can be used hence; different weights may be assigned to different features. The second option is to use a multivariate Gaussian distribution. The similarity score can be computed in a single step using the formula:

$$sim\left(l_0, l_1, r_0, r_1, \theta_0, \theta_1\right) = \exp\left((-0.5) \bullet \begin{bmatrix} l_0 - l_1 & r_0 - r_1 & \theta_0 - \theta_1 \end{bmatrix} \bullet \begin{bmatrix} \frac{1}{\sigma_l} & & \\ & \frac{1}{\sigma_r} & \\ & & \frac{1}{\sigma_\theta} \end{bmatrix} \bullet \begin{bmatrix} l_0 - l_1 \\ r_0 - r_1 \\ \theta_0 - \theta_1 \end{bmatrix} \right)$$

$$= \exp\left((-0.5) \bullet \left(\frac{\left(l_0 - l_1\right)^2}{\sigma_l} + \frac{\left(r_0 - r_1\right)^2}{\sigma_r} + \frac{\left(\theta_0 - \theta_1\right)^2}{\sigma_\theta} \right) \right)$$

In a recognition process, where a query shape is compared with all the shapes in a catalogue, determining the most similar shape is not enough. To make a distinction between known shapes and shapes that are encountered for the first time, the question of whether the two shapes compared are similar must be answered. For that purpose, the similarity scores determined by the matching process must be *absolute*.

In our representation scheme, a shape can be considered as a collection of protrusions and indentations. Because of the excessive smoothing of the evolving shape boundary, the relative importance of protrusions and indentations are found accurately. Therefore, if we define the total similarity of two shapes as the weighted sum of matched pairs of primitives, where the weights are determined by the significance of branches, then an accurate probabilistic measure of similarity is obtained. The main reason such similarity measures have not been obtained in other studies is that no method has been devised to determine the significance of a description primitive correctly.

Shape parts that are not matched contribute a similarity value of zero to the overall similarity score. The shapes to be compared may have different number of branches. All of the branches of the shape with smaller number of branches may be matched. In order to capture the fact that some of the branches of the other shape are not matched and to retain symmetry, the total similarity value is calculated both using the weights of the first shape's parts and the second shape's parts. The lower one of these two similarity values is selected.

6.3 Matching Process

The matching process finds the best subset of the matched pairs of symmetry branches by maximizing the overall similarity score. The primitives of description are not stored in a graph or tree structure, therefore graph or tree specific algorithms cannot be used. A bipartite graph can be formed using the sym-branches as its vertices. The matching problem can be formulated as a bipartite matching problem. There are efficient algorithms to solve the bipartite matching problem. However, the matching framework has an additional constraint: The boundary order of the symmetry branches of a shape should be preserved in the matching. As explained in chapter 3, the order constraint is necessary for perceptually correct matchings. Because of this additional constraint, our matching algorithm is a branch and bound algorithm that searches over all possible matchings of two shapes.

Although the worst-case complexity of a branch and bound algorithm is high, in practice the matching process is very fast. The representation scheme produces coarse level descriptions of shapes. The number of symmetry branches in these coarse descriptions is small so the search over all possible matchings is not computationally expensive. In addition, a number of measures are employed to reduce the computation time. First, shapes are represented as consisting of two halves. The problem is transformed from the matching of two whole shapes to matching of the two subshapes of these shapes. A drastic decrease in computation time is obtained, since the number of permutations that have to be generated by the matching process decreases greatly. Second, those matchings that would violate the order constraint are not tested. Finally, the generation of a permutation is stopped when it is determined that the current branch of computation will not be able to produce a higher similarity value than the current maximum. Additional details are given in Appendix A.6.

6.4 Experimental Results

In this section, we present some matching and recognition examples. The performance results are given in Appendix A.7.

6.4.1 Matching Examples

The matching results are shown on some example shape pairs. Each figure shows the robustness of the representation scheme under a different visual transformation. Figure 6-3 illustrates the robustness under scale difference. The slight difference between the shapes is due to discretization.

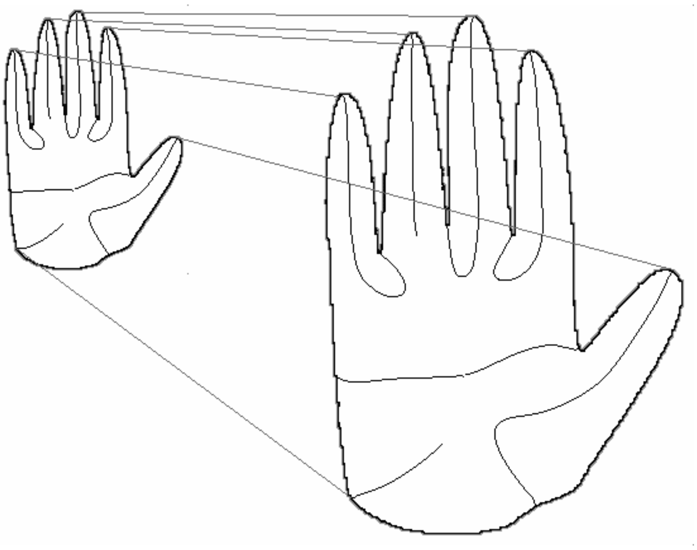
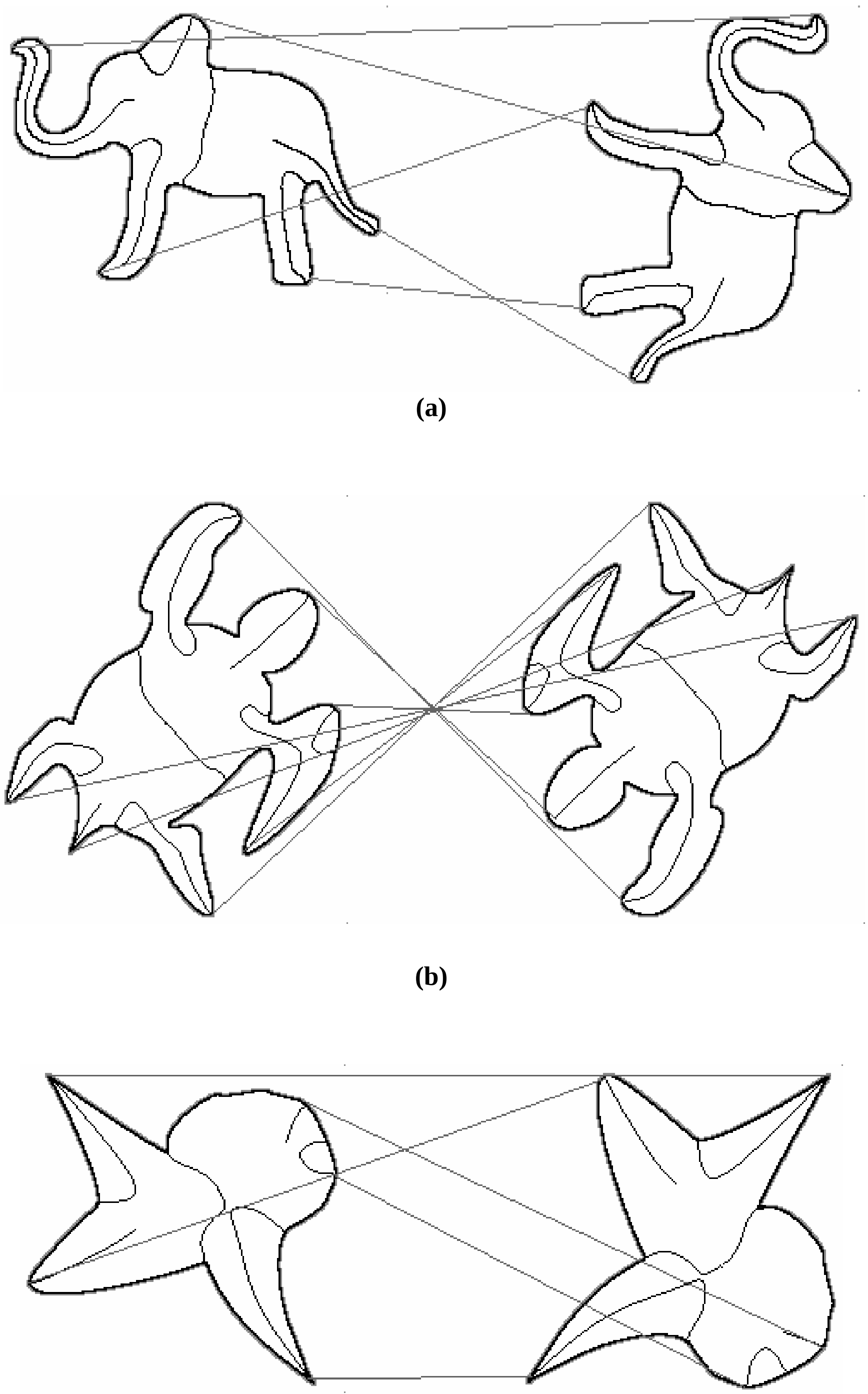


Figure 6-3 Robustness under scale difference. The similarity score is 0.968.

In Figures 6-4 and 6-5, the matching process is able to find the perceptually correct correspondences when a shape undergoes rotation and articulation, respectively. As in the case of scale difference, the similarity values are not exactly 1 in the case of rotation.

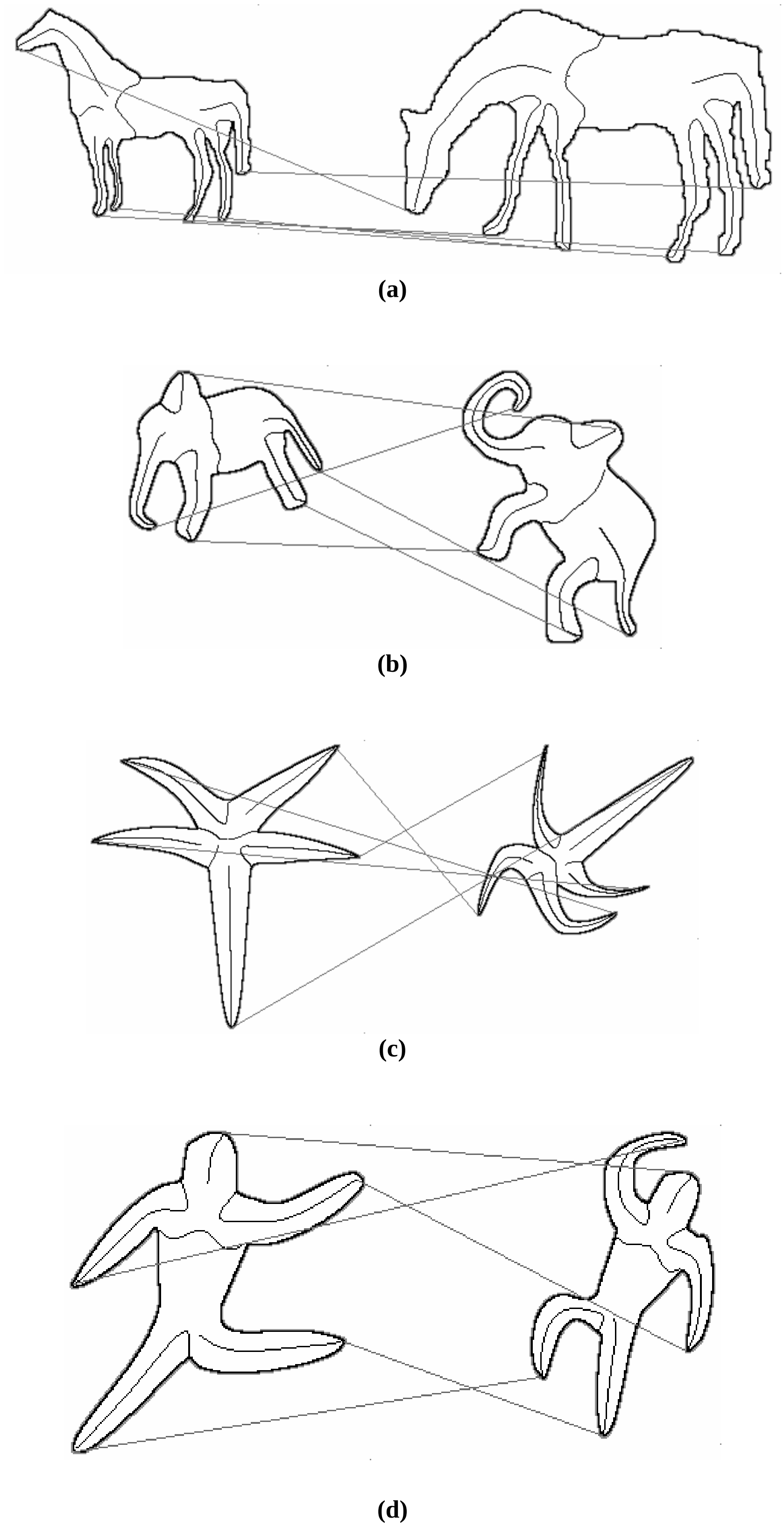


(a)

(b)

(c)

Figure 6-4 Robustness under rotation. The similarity scores are (a) 0.968 (b) 0.988 (c) 0.923.



(a)

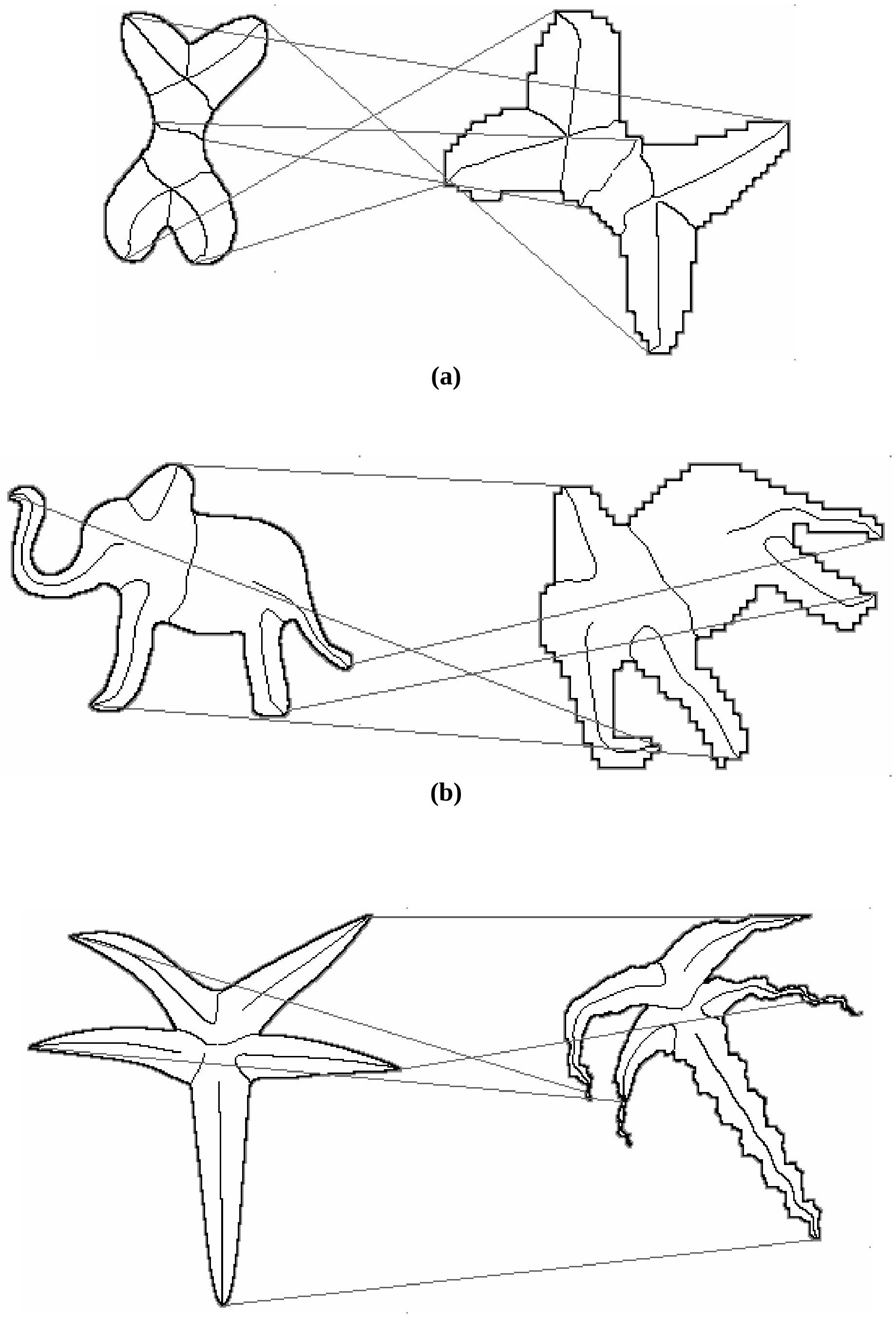
(b)

(c)

(d)

Figure 6-5 Examples of matching under articulation. Similarity values are (a) 0.816 (b) 0.82 (c) 0.89 (d) 0.926.

In Figure 6-6, the matching process finds the perceptually correct correspondences under small boundary perturbations. Notice that although the shape boundaries are different, the structure of the symmetry branches is the same. The small protrusions and indentations of the boundary are not retained in the descriptions.



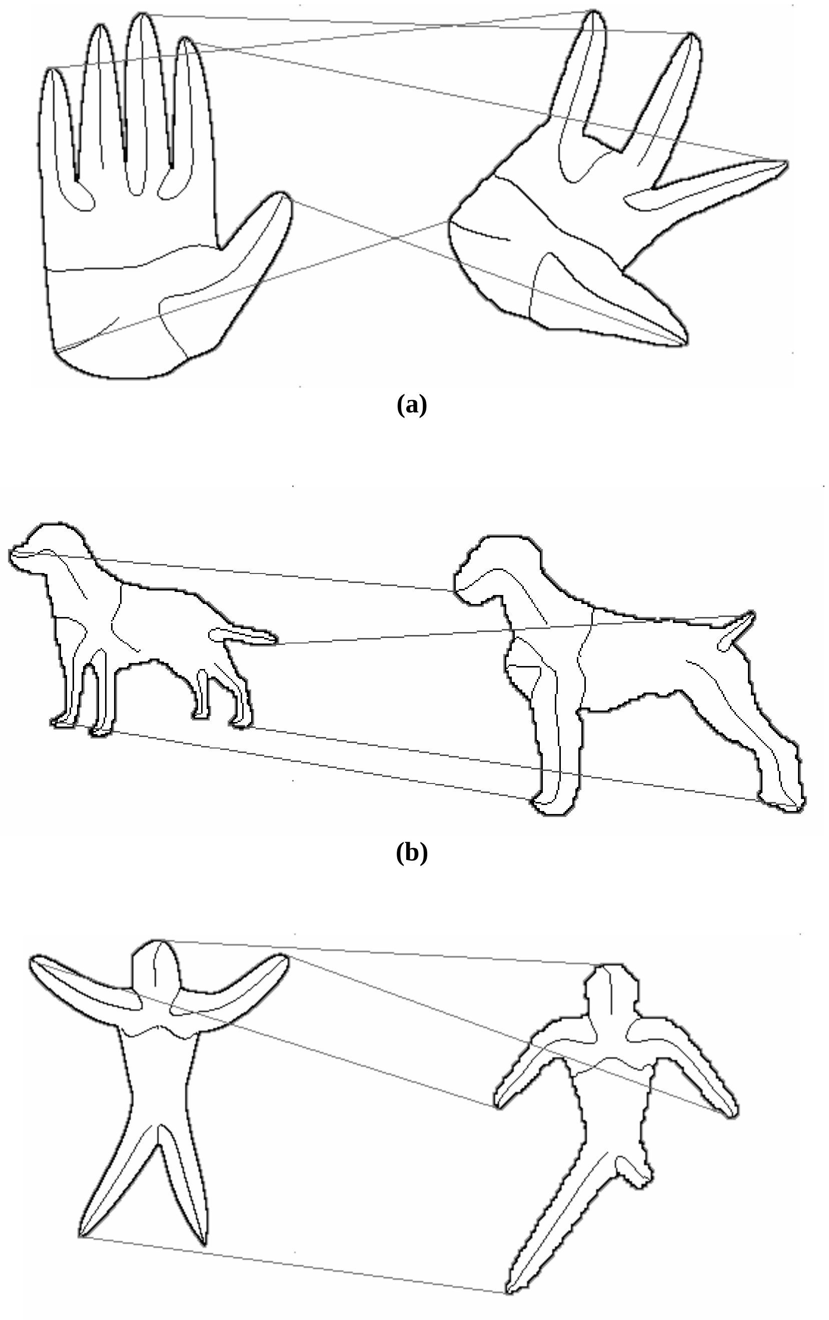
(a)

(b)

(c)

Figure 6-6 Robustness under small boundary perturbations. The similarity values are (a) 0.754 (b) 0.894 (c) 0.832

In the case of missing parts (Figure 6-7), the perceptually correct correspondences are found since the spatial organization of the symmetry branches are stored in the descriptions. The unmatched parts lower the similarity scores significantly. For instance, in Figure 6-7(b) there is a large viewpoint change which leads to new shape parts to emerge. The matching process finds the most probable correspondence of parts based on their location but the similarity score is low.



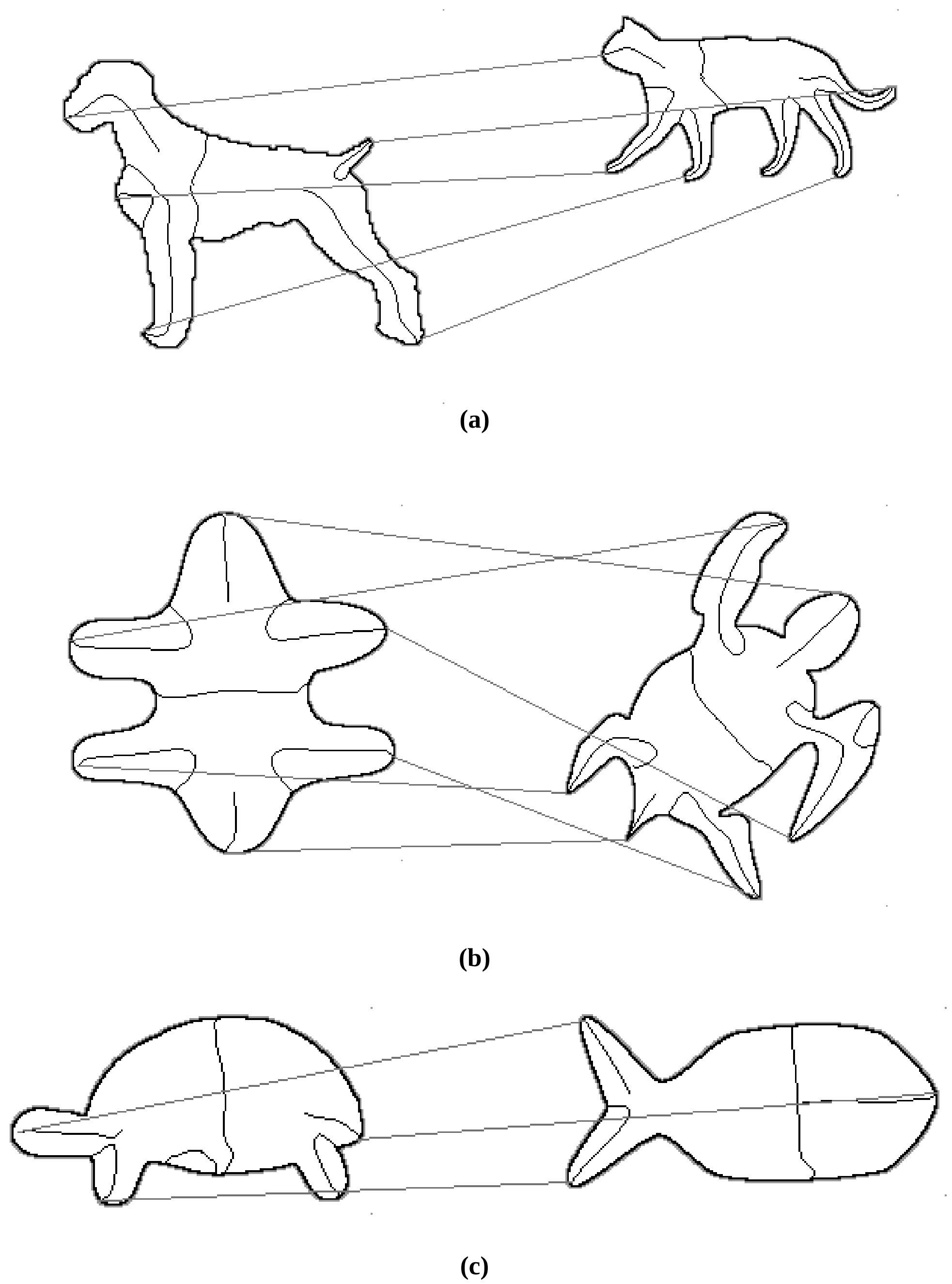
(a)

(b)

(c)

Figure 6-7 Matching in the case of missing parts. Calculated similarity values are (a) 0.734 (b)0.48 (c) 0.75.

Figure 6-8 shows the correspondences found between shapes belonging to different categories. The crucial thing to notice here is that the matching process finds perceptually most plausible matchings. The low results of Figure 6-8(a) and Figure 6-8(c) are due to branches that are not matched. In Figure 6-8(b) a high similarity value is detected due to the coarse descriptions generated by the representation scheme. In a sense, these two shapes may be considered similar since both have the same number of limbs at similar locations.



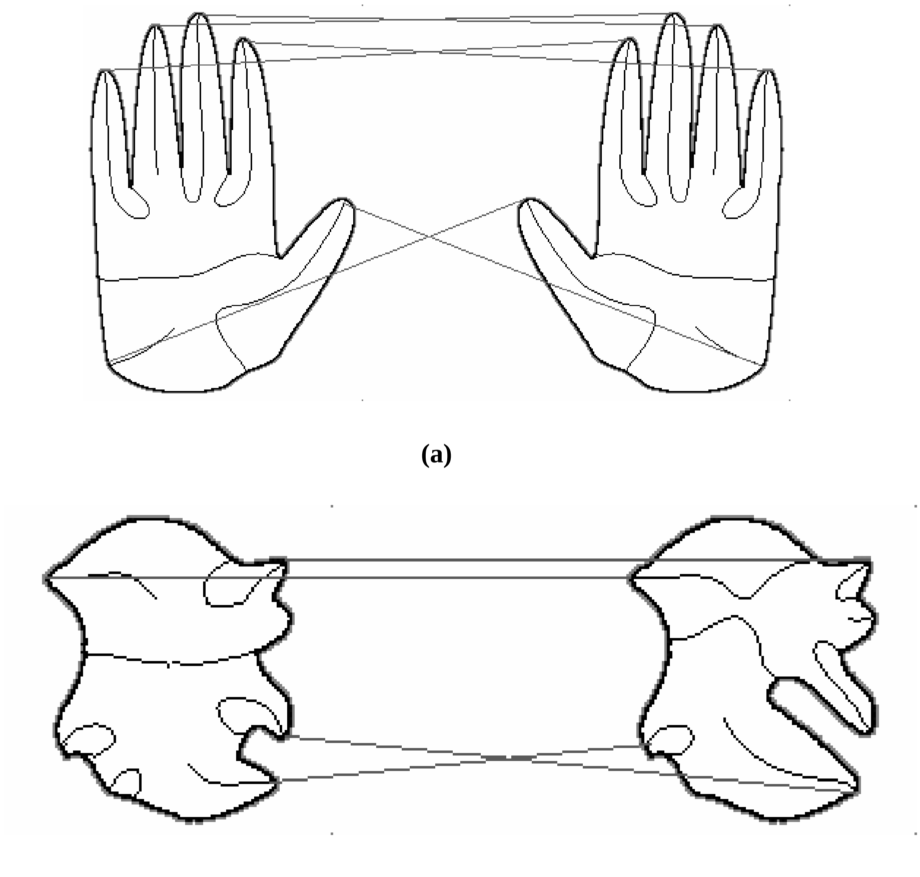
(a)

(b)

(c)

Figure 6-8 Correspondences between shape pairs belonging to different categories. The similarity scores detected are (a) 0.28 (b) 0.81 (c) 0.32.

Figure 6-9 shows two examples of unintuitive correspondences found by the matching scheme. The shapes in Figure 6-9(a) are normally considered identical for a human observer. The matching algorithm fails to match the thumbs because of the order constraint enforced by the matching algorithm. Additional checks can easily be incorporated into the matching algorithm to handle this mirror symmetry situation. However, we prefer to enforce the order constraint in matchings and ignore to handle these kinds of situations because by enforcing the order constraint perceptually better results are obtained. The difference between these two shapes may be interpreted as a large degree of viewpoint variation (a 180° rotation around y-axis). In addition, in a recognition system, this problem can be solved by storing multiple views of a model shape. The similarity value is 0.71 which indicates a large degree of similarity since the major fingers of these shapes are matched. In Figure 6-9(b) the coarse shape structure changes which leads to a change in the reference axis. When the reference axis changes, it is impossible to find the perceptually correct correspondences. On thing to note is that such changes would occur only in the case a change in the globally prominent characteristics of shape.



(a)

(b)

Figure 6-9 Unintuitive correspondences of the matching scheme. The similarity scores are (a) 0.71 (b) 0.50

6.4.2 Recognition Examples

In order to evaluate the classification performance of our matching process, we have constructed a diverse silhouette database. The database, which is shown in Figure 6-10, consists of 180 shapes with 30 categories. Among the shapes within the same category, there are differences in orientation, scale, articulation and small boundary details. This is mainly to evaluate the performance of the matching process under visual transformations.

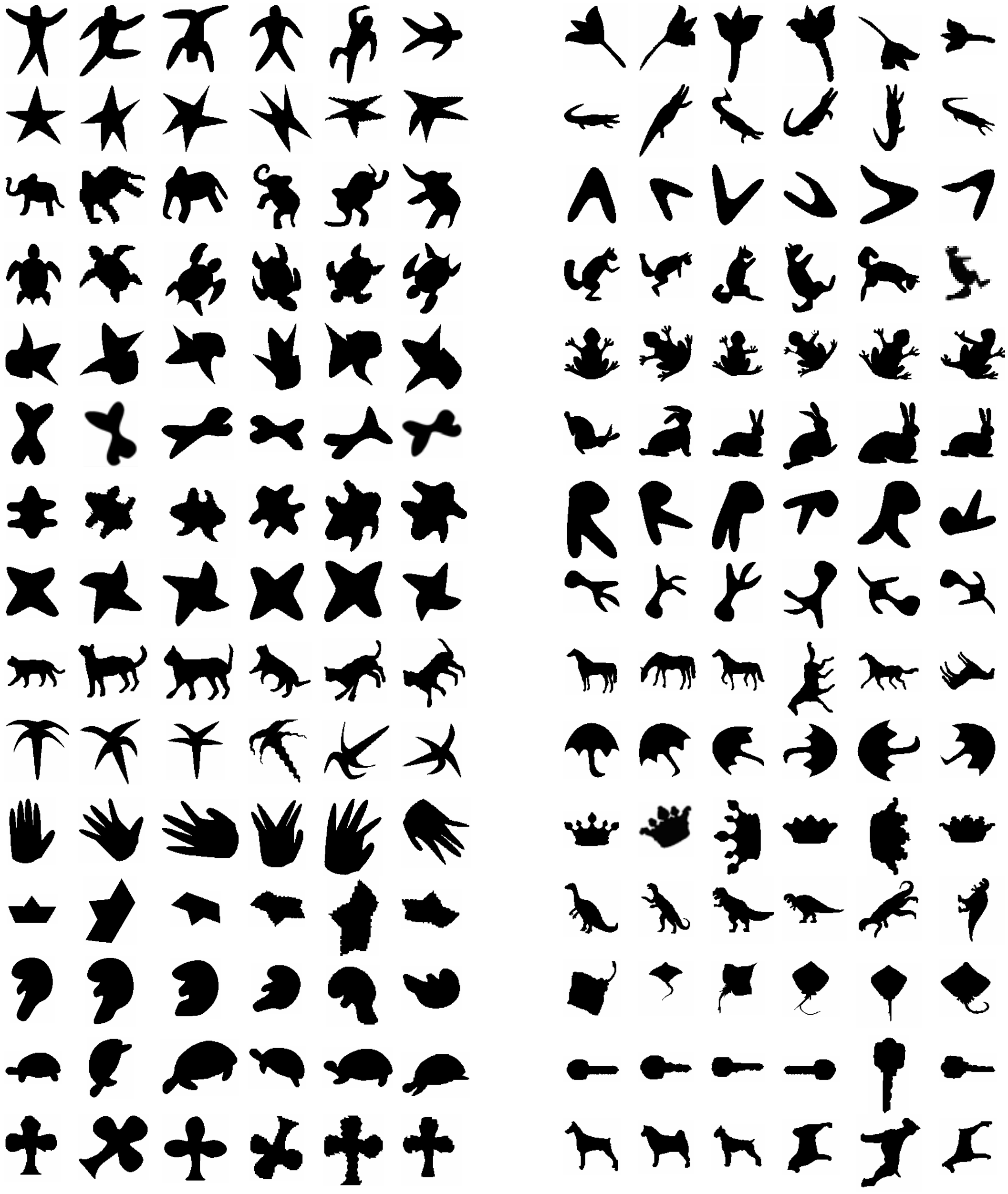
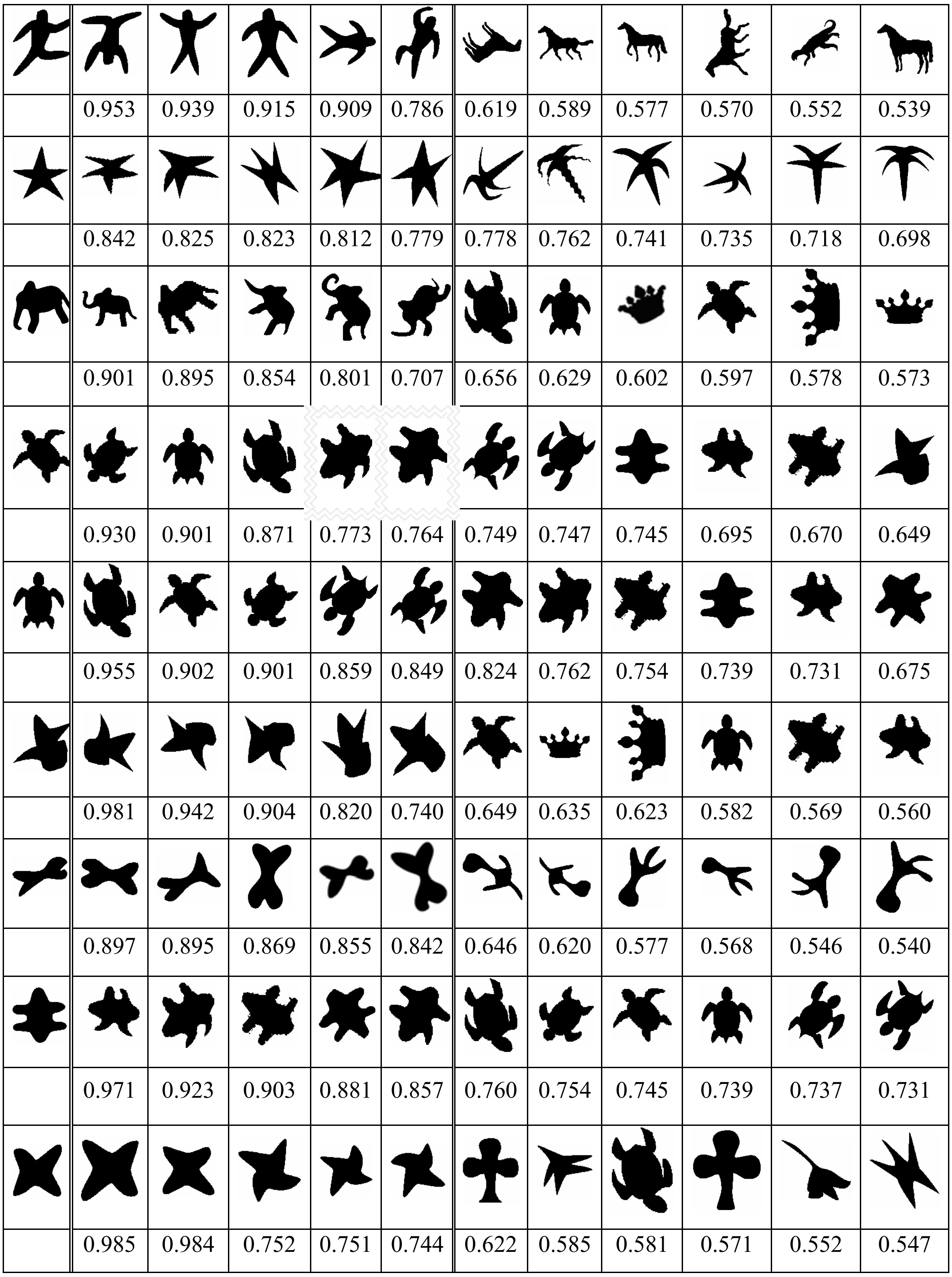


Figure 6-10 The silhouette database used in recognition experiments

The following four figures show the results of 39 queries. The query shapes are selected from the database. In a successful classification, within-category similarities should be highest; that is, the most similar five shapes must be from the same category (we ignore the query shape itself).



	0.953	0.939	0.915	0.909	0.786	0.619	0.589	0.577	0.570	0.552	0.539
	0.842	0.825	0.823	0.812	0.779	0.778	0.762	0.741	0.735	0.718	0.698
	0.901	0.895	0.854	0.801	0.707	0.656	0.629	0.602	0.597	0.578	0.573
	0.930	0.901	0.871	0.773	0.764	0.749	0.747	0.745	0.695	0.670	0.649
	0.955	0.902	0.901	0.859	0.849	0.824	0.762	0.754	0.739	0.731	0.675
	0.981	0.942	0.904	0.820	0.740	0.649	0.635	0.623	0.582	0.569	0.560
	0.897	0.895	0.869	0.855	0.842	0.646	0.620	0.577	0.568	0.546	0.540
	0.971	0.923	0.903	0.881	0.857	0.760	0.754	0.745	0.739	0.737	0.731
	0.985	0.984	0.752	0.751	0.744	0.622	0.585	0.581	0.571	0.552	0.547

Figure 6-11 Some query results.

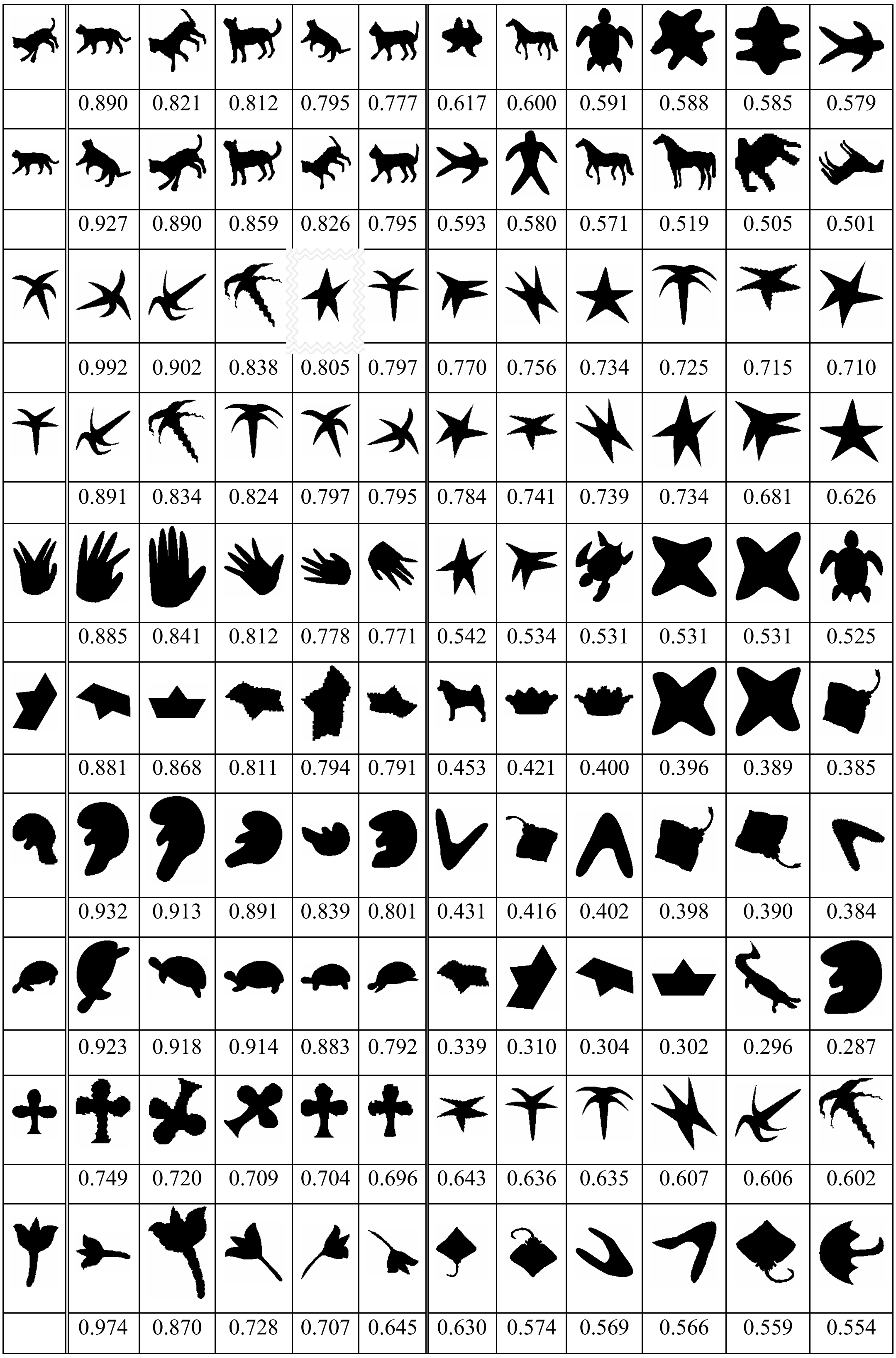


Figure 6-12 Some query results.

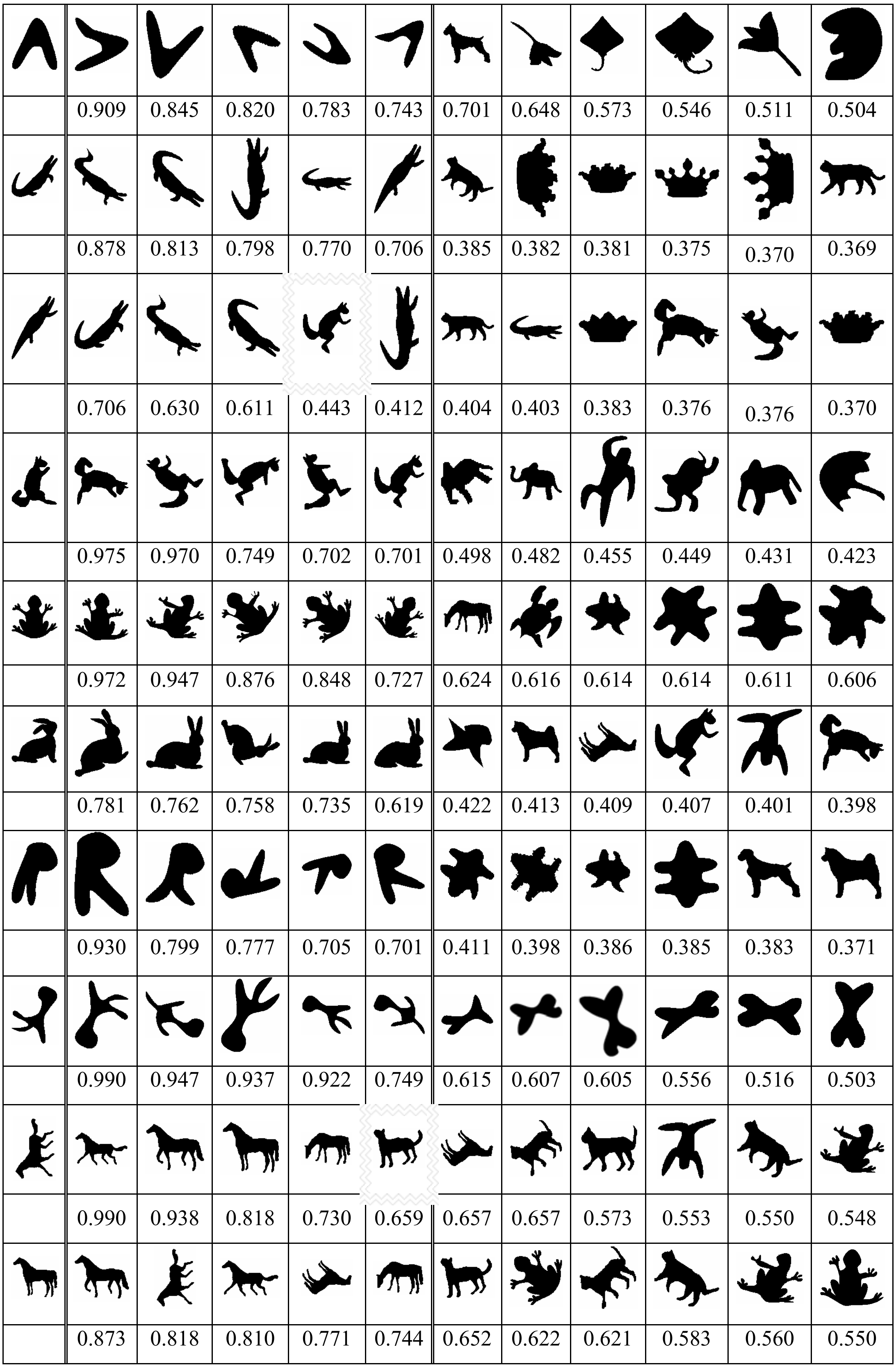


Figure 6-13 Some query results.

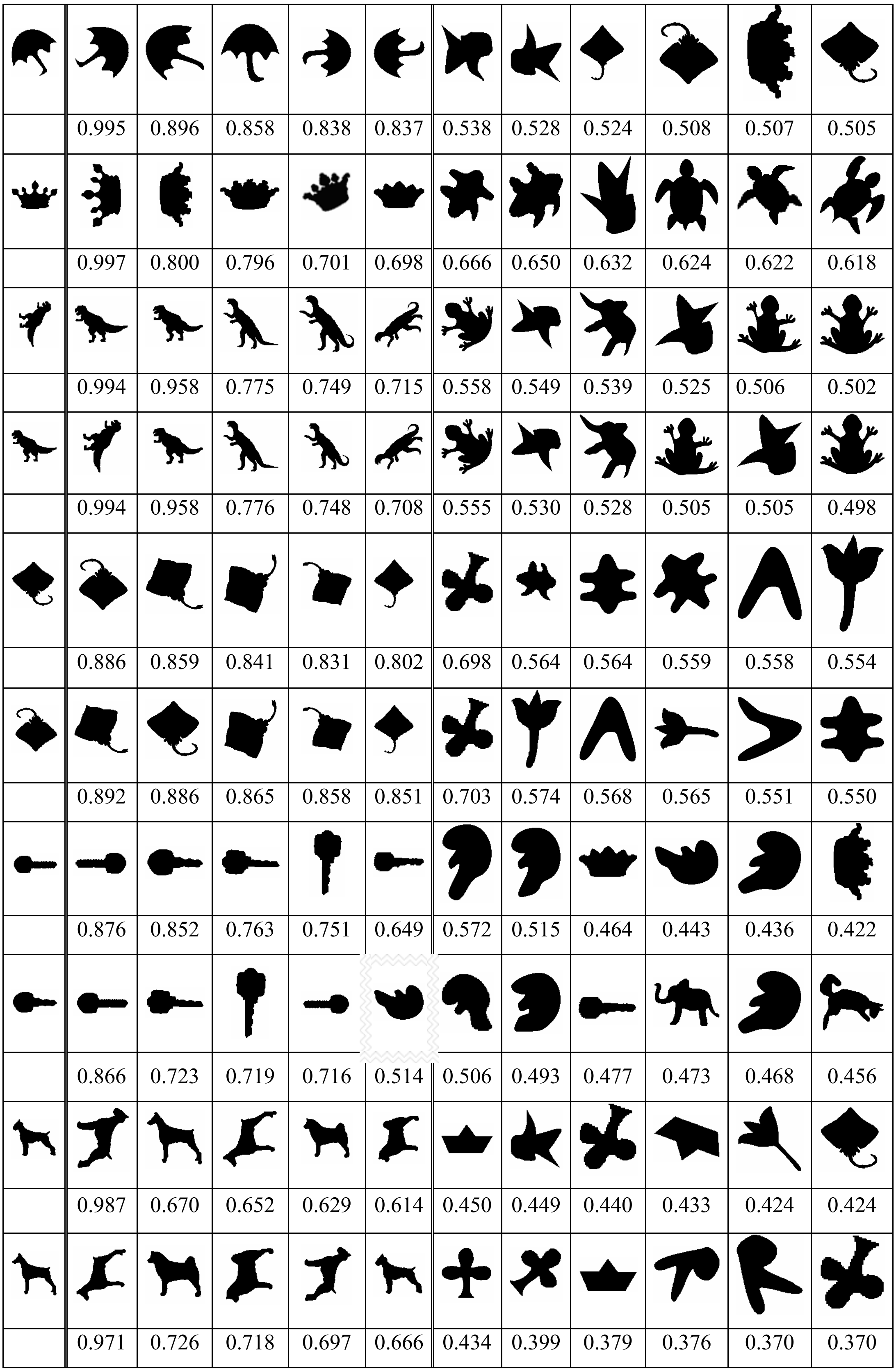


Figure 6-14 Some query results.

The example queries show that the classification is successful most of the time. Even the coarse descriptions of shapes are useful for discriminating them. The coarse descriptions are also the main reasons for some classification errors. The performance of the matching process depends on how the shapes in the shape domain differ from each other. Therefore, in a shape database that includes very similar shapes the classification performance of the matching process would not be so high.

CHAPTER 7

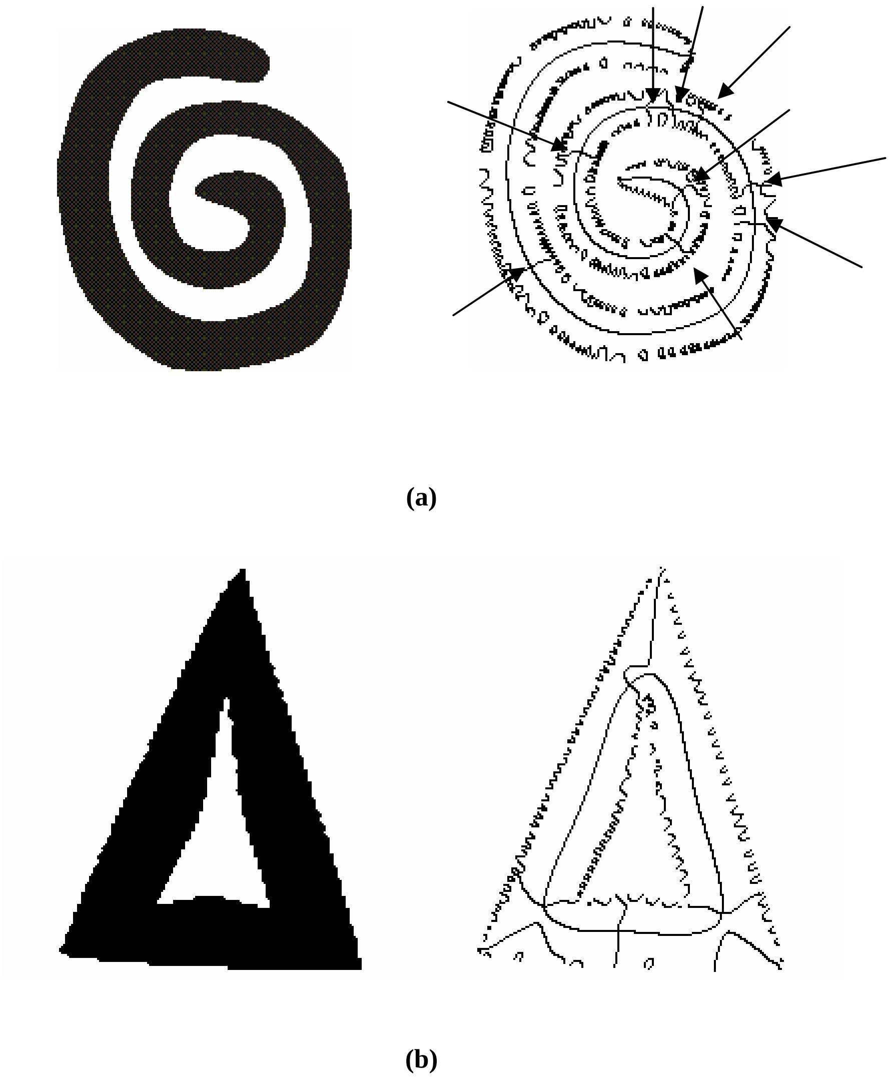
DISCUSSION AND FUTURE WORK

In this thesis, an unconventional approach to shape recognition using disconnected symmetry branches is presented. Unlike the common skeletal representations, the symmetry axes in our framework are disconnected. It is precisely the disconnected nature of the branches that enables us to measure the prominence and metric properties of shape parts accurately. It was known that curvature dependent motion of the shape boundary provides an accurate prominence measure of symmetry branches. Because it has been long considered that connectedness of symmetry branches must be ensured, this idea was dismissed and never used in any recognition framework. If we look at the problem from a computation theory point of view, and try to determine why this connectedness constraint is necessary, we see that the topology of the symmetry axis (which demands connected symmetry branches) provides the organization information of the sym-branches. If this organization can be expressed by other means, then it is clear that the connectedness constraint is not necessary. Describing the spatial relations of sym-branches in a canonical coordinate frame enabled us to use local sym-branch based descriptions in a recognition framework.

The representation scheme and the recognition results showed that the representation issue is most crucial. In the design of the representation scheme, the main goal is stability. For that purpose, the axis regularization parameter is selected very large. Resulting descriptions of shapes remained stable under various visual transformations. That stability allowed us to use a very fast and simple matching algorithm. When the complexity of matching process developed in this thesis is compared with those of the matching frameworks in Chapter 3, the difference is significant.

It may be argued that the designed representation scheme does not satisfy the sensitivity requirement. Using coarse descriptions of shapes makes it impossible to discriminate the subtle differences between shapes. The answer to this argument is that a multiple level representation scheme seems to be necessary to satisfy the opposing goals of sensitivity and stability. It is better to start out with a stable scheme rather than a sensitive one. A stable scheme is associated with globally more important properties of shapes while a sensitive scheme is also related to subtle and local details. Deriving the global properties from local details is a difficult problem. Going from the coarse to detailed descriptions would be easier because the globally important sym-branches have been identified and can be separated from those branches representing local detail.

There are many issues that need to be considered in future work. First of all, the symmetry point detection scheme needs some improvements to be applicable to a wider class of shapes. Currently, the scheme is not applicable to shapes that have more than two equally prominent main parts and to shapes with holes. Figure 7-1(a) shows a stroke shape and its full symmetry axis. Although a large amount of diffusion is applied, there are many elliptical and hyperbolic points on the surface (shown with arrows). The reason of this phenomenon is that the width of the shape is nearly constant everywhere. No part of it is more significant than other parts. Figure 7-1(b) shows a shape with a hole. The representation interprets it as a shape consisting of three main parts. The multiple description idea can handle this situation but more complicated situations may lead to more descriptions. It may be prohibitively expensive to compare shapes with many descriptions.



(a)

(b)

Figure 7-1 (a) A stroke shape and its full symmetry axis (b) A shape with a hole and its full symmetry axis

The second problem with the symmetry point detection may occur when the shape has a region with constant width. The worm points are not encountered in the symmetry point detection scheme because the evolving opposite boundaries form crosses at their intersection. Although the full symmetry axis is extracted accurately in these situations, sometimes the branch type information may be inaccurate. Figure 7-2 shows a rectangle and its positive and negative symmetry branches. Some sym-points that should be classified as positive are determined as negative. This situation can be handled depending on the fact that two positive sym-branches should reach the shape center. Incorrectly classified branches can be detected using this constraint. Other similar situations can be handled by incorporating semantic checks in the symmetry point detection procedure.

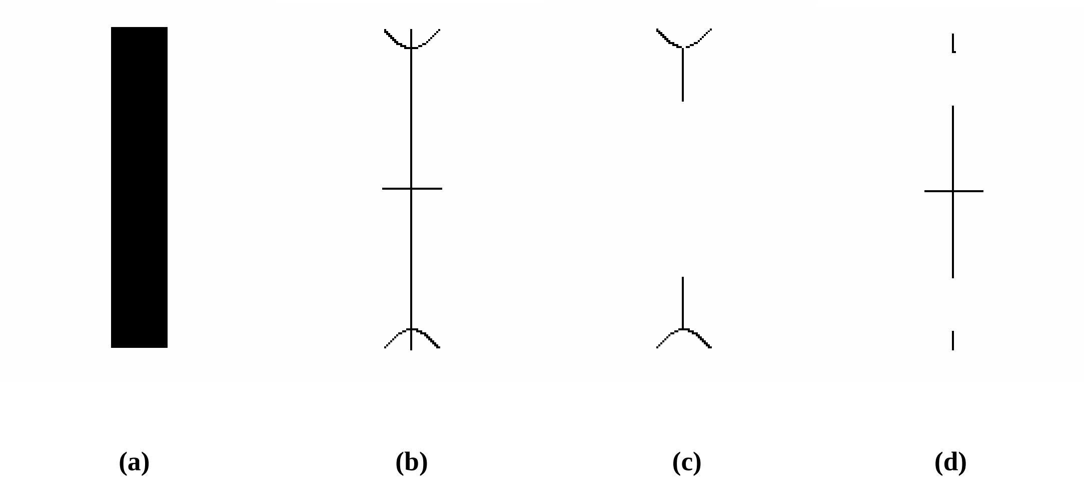


Figure 7-2 (a) A rectangle and its (b) full sym-points (c) positive sym-points (d) negative sym-points

The practical difficulties associated with scale space representations are explained and the problem is not addressed in the thesis. It remains as a major research topic to be investigated. In other skeletal representations which use connected symmetry branches, the transitions between consecutive scales may include big changes in object structure. This is because branches are totally removed from the axis between scale transitions. The disconnected nature of symmetry axis may provide smooth transitions between scales. The amount of smoothing may be the main tool for obtaining descriptions of shapes at different scales.

The problem of recognition under occlusion and large viewpoint changes is also not addressed in this thesis. Although these issues seem to require directly measurable information rather than semantic information, there are still situations where the semantic information is necessary. Figure 7-3 illustrates one of them.

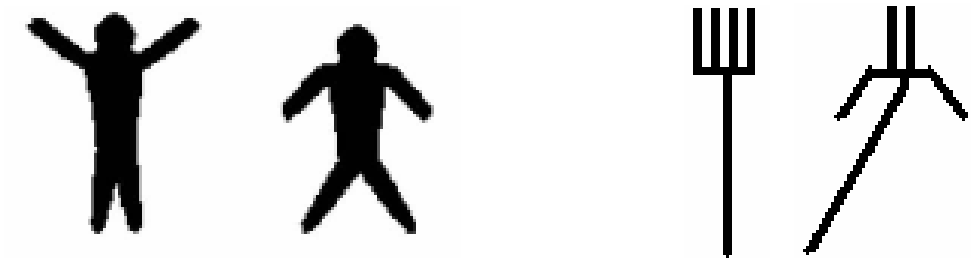


Figure 7-3 Two recognition tasks

We can easily recognize the two figures on the left and to us, these figures represent the same entity. Using a shape descriptor that is robust to articulation of parts, the similarity of these shapes is captured. The two figures on the right also appear very close using that the same kind of descriptor because the second figure is obtained from articulation of the parts of the first figure. We can identify the first figure as a fork, but the second figure does not resemble any object we know. We do not recognize it as a fork because since it is a rigid object, most of us probably have not seen its parts articulated in such a way. The only way to discriminate the two situations is to use semantic information in the form of prior experience. This information can be incorporated into the description by specifying the degree of variation of a feature as in Marr and Nishihara's 3D model representation. There may be great variability on the orientation of an arm in a human figure whereas the rigidity of the fork shape may be captured by setting the variability on the local orientation of its parts to very low values.

Describing the articulation information is easy; the main question is how to learn that information. One possible way to do that is to train a recognition system with labeled samples which show many possible views of shapes. The degree of variability for each feature of each shape part may be learned by fitting probability models to feature values.

The training of the recognition framework may provide a concept of *class skeleton* to be implemented. In Figure 7-4, a number of different fish are shown at the top. While their species are different, we can loosely group them into the class fish. The crucial question is "What are the characteristics of these shapes that allow us to group them into the class fish?" The class skeleton is an axis-based description which includes only the discriminative properties of a class. Since the tails and the front parts are common in the fish shapes in Figure 7-4, the class skeleton for a fish class may include only axes corresponding to these parts. The biggest advantage of such an approach is that when an unknown shape is encountered, first it can be matched to class skeletons rather than all the shapes in the database.

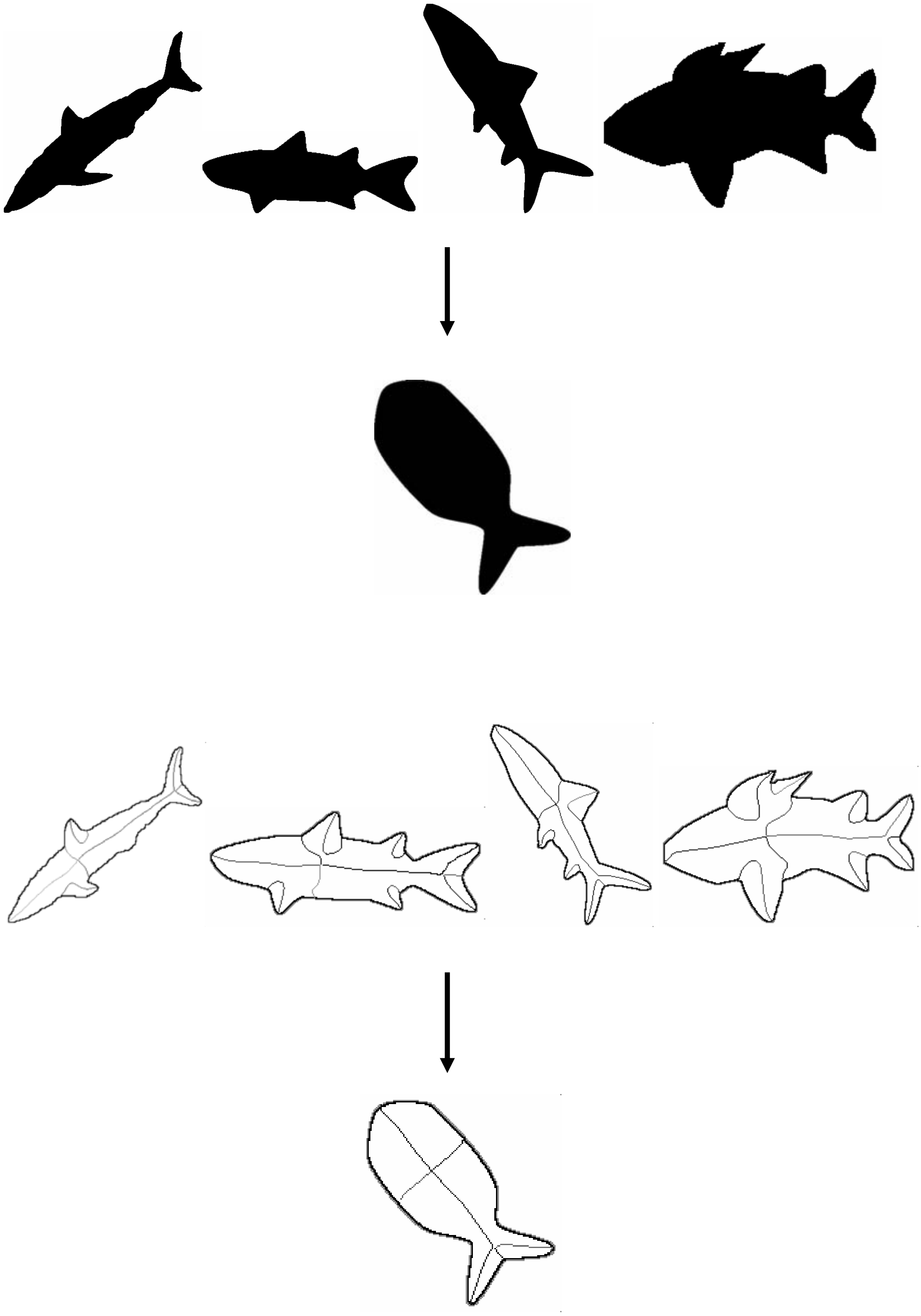


Figure 7-4 The concept of class skeleton.

REFERENCES

[1] Binford T.O., "Visual Perception by Computer", *Proc. IEEE Conf. Systems and Control*, vol. 1971.

[2] Blum H., "Biological shape and visual science", *J. Theor. Biol.*, vol. 38, no. 2, pp. 205-287, 1973.

[3] Brady M. and Asada H., "Smoothed Local Symmetries and their implementation", *Int. J. Robot. Res*, vol. 3, no. 3, pp. 36-41, 1984 .

[4] Bruce J.W., Giblin P.J., and Gibson C.G., "Symmetry Sets", *Proc. R. Soc. Edinb. Sect. A-Math*, vol. 101, pp. 163-186, 1985.

[5] Burbeck C.A. and Pizer S.M., "Object Representation by Cores - Identifying and representing primitive spatial regions", *Vision Res.*, vol. 35, no. 13, pp. 1917-1930, 1995.

[6] Geiger D., Liu T. L., and Kohn R. V., "Representation and self-similarity of shapes", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 86-99, 2003.

[7] Hoffman D.D and Singh M., "Salience of visual parts", *Cognition*, vol. 63, no. pp. 29-78, 1997.

[8] Jenkinson M. and Brady M., "A saliency-based hierarchy for local symmetries", *Image Vis. Comput.*, vol. 20, pp. 85-101, 2002 .

[9] Kimia B. B., Tannenbaum A. R., and Zucker S. W., "Shapes, Shocks, and deformations .1. The components of 2-dimensional shape and the reaction-diffusion space", *Int. J. Comput. Vis.*, vol. 15, no. 3, pp. 189-224, 1995.

[10] Leymarie F. and Levine M.D, "Simulating the grassfire transform using an active contour model", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 1, pp. 56-75, 1992.

[11] Leyton M., "A Process-Grammar for shape", *Artif. Intell.*, vol. 34, no. 2, pp. 213-247, 1988.

[12] Liu T. L. and Geiger D., "Approximate Tree Matching and Shape Similarity", *Proc. Seventh IEEE Int'l Conf. Computer Vision*, pp. 1129-1135, 1998.

[13] Loncaric S., "A survey of shape analysis techniques", *Pattern Recognit.*, vol. 31, no. 8, pp. 983-1001, 1998.

[14] Marr D., "Vision", *Freeman*, 1982.

[15] Marr D. and Nishihara H.K., "Representation and recognition of spatial-organization of 3-dimensional shapes", *Proc. R. Soc. Lond. Ser. B-Biol. Sci.*, vol. 200, no. 1140, pp. 269-294, 1978.

[16] Ogniewicz R.L. and Kubler O., "Hierarchical Voronoi Skeletons", *Pattern Recognit.*, vol. 28, no. 3, pp. 343-359, 1995.

[17] Osher S. and Sethian J., "Fronts propagating with curvature dependent speed: Algorithms based on the Hamilton-Jacobi formulation", *J. Comp. Phys.* vol. 79, no. 1, pp. 12-49, 1988.

[18] Ponce J., Bajcsy R., Metaxas D., Binford T.O., Forsyth D.A., Hebert M., Ikeuchi K., Kak A.C., Shapiro L., Sclaroff S., Pentland A., and Stockman G.C., "Object representation for object recognition", *in Procedings, IEEE Conference on Computer Vision and Pattern Recognition*, vol. pp. 147-152, 1994.

[19] Rom H. and Medioni G., "Hierarchical decomposition and axial shape-description", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 10, pp. 973-981, 1993.

[20] Sebastian T.B., Klein P. N., and Kimia B. B., "Recognition of shapes by editing their shock graphs", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 550-571, 2004.

[21] Shaked D. and Bruckstein A.M., "Pruning medial axes", *Comput. Vis. Image Underst.*, vol. 69, no. 2, pp. 156-169, 1998.

[22] Shokoufandeh A., Dickinsing S. J., Siddiqi K., and Zucker S.W., "Indexing using a spectral encoding of topological structure", *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, no. pp. 491-497, 1999.

[23] Siddiqi K. and Kimia B. B., "A shock grammar for recognition,", *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 507-513, 1996.

[24] Siddiqi K., Shokoufandeh A., Dickinson S. J., and Zucker S. W., "Shock graphs and shape matching", *Int. J. Comput. Vis.*, vol. 35, no. 1, pp. 13-32, 1999.

[25] Tari Z.S.G., Shah J., and Pien H., "Extraction of shape skeletons from grayscale images", *Comput. Vis. Image Underst.*, vol. 66, no. 2, pp. 133-146, 1997.

[26] Zhu S.C. and Yuille A.L., "FORMS: A flexible object recognition and modeling system", *Int. J. Comput. Vis.*, vol. 20, no. 3, pp. 187-212, 1996.

APPENDIX A

IMPLEMENTATION DETAILS

A.1 Surface Computation

The diffusion process is governed by the following linear diffusion equation:

$$\frac{\delta v}{\delta \tau} = \nabla^2 v,\ v\big|_{\Gamma} = 1$$

This equation is solved using standard finite difference approximations. Central differences are used for space derivatives and forward difference is used for time derivative. Discretizing this equation leads to:

$$\frac{v_{i,j}^{k+1} - v_{i,j}^{k}}{\Delta t} = v_{i+1,j}^{k} + v_{i,j+1}^{k} + v_{i-1,j}^{k} + v_{i,j-1}^{k} - 4 * v_{i,j}^{k}$$

where i, j denotes space coordinates and k denotes iteration step. The initial v function is the input image. The time step value should be smaller than 0.25 for the scheme to converge. The boundary condition $v\big|_{\Gamma} = 1$ is enforced by not diffusing at those image points whose values are equal to 1. Symmetry conditions are assumed on the image boundary:

$$\begin{aligned} v_{0,j} &= v_{1,j} & \qquad v_{n+1,j} &= v_{n,j} \\ v_{i,0} &= v_{i,1} & \qquad v_{i,n+1} &= v_{i,n} \end{aligned}$$

In order to detect elliptic points, we first determine the points at which the gradient vanishes. It is numerically difficult to detect the condition $|\nabla v| = 0$. Therefore, we use an alternative computation by considering the simultaneous zero-crossings of v_x and v_y (Figure 2-30). The vanishing of the gradient indicate either an elliptic point or a hyperbolic point. Final classification is done by considering the determinant of the Hessian at such points. If it is positive, the point is elliptic. If it is negative, the point is hyperbolic. The algorithm for surface computation step is as follows:

Algorithm 1 The Surface Computation

```
  i.  Initial Diffusion: Diffuse the image until the minimum point of the
      surface reaches 0.1

 ii.  Compute the elliptic points of the surface

iii.  IF number of elliptic points > 2
           Further diffuse the surface until number of elliptic points <= 2

 iv.  IF number of elliptic points = 1
           Topology of the shape is single-blob
      ELSE IF number of elliptic points == 2
           IF the elliptic point values are not close
                  Topology of the shape is single-blob
      ELSE
           Topology of the shape is dumbell-like

  v.  Final diffusion:
           IF topology is single-blob
                  Diffuse the surface until the elliptic point count = 1
                  and the minimum of the surface > 0.5
```

A.2 Detection of Symmetry Points

The symmetry point detection method uses the method of TSP given in page 31 on the surface computed in the previous step. Again, central difference approximations are used for space derivatives and forward difference approximation is used for time derivative. The algorithm for symmetry point detection is as follows:

Algorithm 2 Symmetry Point Detection

```
  i.  For each point of the surface, compute $\frac{d|\nabla v|}{ds}$, $\frac{d^2|\nabla v|}{ds^2}$

 ii.  Compute the zero crossings of $\frac{d|\nabla v|}{ds}$ as Full Symmetry Points

iii.  For each symmetry point
      IF $\frac{d^2|\nabla v|}{ds^2}$>0
             Designate the symmetry point as a positive symmetry point
      ELSE
             Designate the symmetry point as a negative symmetry point
```

A.3 Detection of Symmetry Branches

The grouping of symmetry points into symmetry branches are carried out in a number of procedures which are given in Algorithm 3. First, pixel connectedness is used to determine the branch segments Then, the first pruning step discards the detected small branches that are due to noise to decrease the complexity of the further computations. The symmetry branches are disconnected at the junction points and, finally, the branches that are insignificant and that are not paired with another branch of opposite type are discarded.

Algorithm 3 Symmetry Branches Detection

```
  i.  Find Branches:
      FOR each symmetry point
         FOR each previously detected symmetry branches
                IF the symmetry point is sufficiently close to the branch
                then append this point to the branch

                If the symmetry point is not appended to any branch, start
                a new branch

      REPEAT
         FOR each symmetry branch
                Find the distance to every other branch.
                IF a sufficiently close branch is found, combine them into
         one

      UNTIL no branches are combined into one

 ii.  Prune Branches:
      FOR each symmetry branch
            IF branch length is smaller than the threshold value
                  Discard branch
            ELSE
                  Update total branch length value

iii.  Split Branches
      a) Compute zero crossings of $\frac{d\left|\nabla v\right|}{ds}$, $\frac{d^2\left|\nabla v\right|}{ds^2}$

      b) Designate the points where $\frac{d\left|\nabla v\right|}{ds}$ and $\frac{d^2\left|\nabla v\right|}{ds^2}$ cross zero
         simultaneously as junction points.

      c) FOR each symmetry branch
            FOR each junction point
                  Find the nearest point on the branch to the junction
                  point

                  IF this nearest point is not too close to the end of
                  the branch and IF the nearest point is sufficiently
                  close to the junction point
                        Split Branch

 iv.  Prune Branches
```

```
    FOR each positive symmetry branch
          IF (Branch Length / Total Branch Length) is lower than the
          threshold
                Discard Branch
                Update total branch length value

 v. Discard Unpaired Branches

    FOR each symmetry branch
          Find the nearest branch of opposite type
          IF this nearest branch is not sufficiently close
                Discard Branch
```

A.4 Setting Up The Canonical Coordinate Frame

The reference axes of a shape are those branches that reach to shape center. Depending on the topology of the shape, the type of branches used as reference axis differs. The algorithm for coordinate frame generation simply finds the branches of required type that flow into the shape center. Then, for every pair of reference axis, a new coordinate frame is formed:

Algorithm 4 Coordinate frame generation

```
 i. IF topology is single-blob
          The types of branches to be searched is negative
    ELSE
          The branches to be searched are of type positive with negative
          curvature

ii. FOR each branch of the required type
          IF it is sufficiently close to the shape center
                Designate this branch as a reference axis

    FOR each reference axis
          FOR each other reference axis
                Generate a coordinate frame
```

A.5 Describing Spatial Relations and Measurable Properties

The angle between the reference axis and the vector drawn from the center point to the branch termination point can be computed using:

$$\cos^{-1}\frac{(a*b)}{|a|*|b|}$$

The counter clockwise angle can be determined from the fact that the direction of the angle is specified by a vector C perpendicular to these two vectors. This vector is either parallel or anti-parallel to the vector cross product AxB. If C points anti-parallel to AxB then the counter clockwise angle is just the angle calculated above. If C points parallel to AxB then the counter clockwise angle is 360 - the angle. The computation is simply:

$$z = a_x * b_y - b_x * a_y$$

$$\text{if } z < 0 \rightarrow \text{angle} = 360^\circ - \text{angle}$$

Determining the order of branches along the boundary is difficult. Measuring the angle between two position vectors of symmetry branches is not sufficient to determine the neighbors of a symmetry axis. Because of the bending of branches we cannot simply select the next branch as the branch with minimum counter clockwise angle difference. The solution preferred in this study is to determine which branches can connect which other branches. A line drawn between two neighboring branches should not intersect any other symmetry branch. Using this fact, we constrain the number of possible neighbors of a branch. Most of the time there are only two possible neighbors for a branch (the one to its left and the one to its right). The axis order computation procedure tries all possibilities and selects the order that gives the minimum total angle difference (Algorithm 5).

Algorithm 5 Axis order computation

```
NextBranch (
  Input: index → Index of the current branch
  oind → The order index (the place in the data structure where the new
  axis is to be placed)
  totalangle → The total angle difference between ordered branches

   i.  FOR each branch that is not added
             IF two branches can be connected by a straight line without
             intersecting another branch, select the branch
  ii.        Compute ANGLE as the counter clockwise angle between them
 iii.        IF the branch is the starting branch (which means the cycle is
             complete)
                  IF total number of ordered branches is maximum OR (total
                  number of ordered branches equals the current maximum AND
                  totalangle is lower than the angle of the maximum order)
                          Designate this order as optimal order
             ELSE
                  CALL NextBranch (branchindex, oind+1, totalangle + angle)
```

A.6 The Matching Process

The matching algorithm simply tries all possible matchings of branches. If a correspondence will violate the order of matching, it is not tested. The permutations in which one or multiple branches are not matched are also tested. The total similarity score is simply computed using:

$$sim\left(shape_0, shape_1\right) = \sum_{1}^{n_{shape_0}} l_0 * sim\left(l_0, l_1, r_0, r_1, \theta_0, \theta_1\right)$$

The normalized length of a branch is also used as a weight in the similarity computation. The similarity score may differ depending on which shape is chosen as shape$_0$ and which shape is chosen as shape$_1$. To make similarity score computation symmetric, we simply calculate:

$$sim2\left(shape_0, shape_1\right) = \min\left(sim\left(shape_0, shape_1\right), sim\left(shape_1, shape_0\right)\right)$$

A.7 Performance Results

To determine the time it takes to compute descriptions and to match shapes experiments are carried out on a Pentium 4 2 GHz CPU. The time required to compute the descriptions from binary images depends mainly on the diffusion step. The size of the image significantly affects the duration of the diffusion step. The axis detection step is also affected because as the size of the image increases the number of pixels representing the sym-points increase. As an example, when the size of the shape in Figure A-1 is 100x113 pixels, the diffusion takes 1 sec. When the image size is doubled, the diffusion takes 6 seconds. The overall computation takes two seconds in the smaller shape whereas it takes 8 seconds in the larger shape.

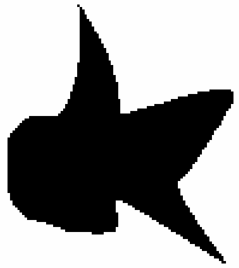


Figure A-1 The effect of the size of the image on the time it takes to compute the descriptions. When the image size is 100x 113: Diffusion length: 1 sec. Axis Detection Step: 1 sec. When the image size is 198x225: Diffusion Length: 6 sec Axis Detection Step: 2 sec

Since the representation scheme produces coarse descriptions of shapes, the number of branches and the number of descriptions are small. Therefore, even the matching of the most complex shapes in the database takes approximately 1 second. However, when a shape is compared to all the shapes in the database, the number of descriptions of the query shape affects the computation time. For instance, while it takes 15 seconds to classify a shape with two descriptions, it takes 25 seconds to classify a shape with six descriptions. Because of the fixed complexity of retrieving shapes from storage, the change in computation time is not as high as expected.