CATEGORY KNOWLEDGE, SKELETON-BASED SHAPE MATCHING AND SHAPE CLASSIFICATION

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

BY

İBRAHİM AYKUT ERDEM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

OCTOBER 2008

Approval of the thesis:

CATEGORY KNOWLEDGE, SKELETON-BASED SHAPE MATCHING AND SHAPE CLASSIFICATION

submitted by **İBRAHİM AYKUT ERDEM** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Müslim Bozyiğit
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Sibel Tarı
Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Neşe Yalabık
Computer Engineering Dept., METU

Assoc. Prof. Dr. Sibel Tarı
Computer Engineering Dept., METU

Assoc. Prof. Dr. Cem Bozşahin
Computer Engineering Dept., METU

Asst. Prof. Dr. Pınar Duygulu-Şahin
Computer Engineering Dept., Bilkent University

Asst. Prof. Dr. Mine Özkar
Architecture Dept., METU

Date:

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 : İbrahim Aykut Erdem

Signature :

ABSTRACT

CATEGORY KNOWLEDGE, SKELETON-BASED SHAPE MATCHING AND SHAPE CLASSIFICATION

Erdem, İbrahim Aykut

Ph.D., Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Sibel Tarı

October 2008, 179 pages

Skeletal shape representations, in spite of their structural instabilities, have proven themselves as effective representation schemes for recognition and classification of visual shapes. They capture part structure in a compact and natural way and provide insensitivity to visual transformations such as occlusion and articulation of parts.

In this thesis, we explore the potential use of disconnected skeleton representation for shape recognition and shape classification. Specifically, we first investigate the importance of contextual information in recognition where we extend the previously proposed disconnected skeleton based shape matching methods in different ways by incorporating category knowledge into matching process. Unlike the view in syntactic matching of shapes, our interpretation differentiates the semantic roles of the shapes in comparison in a way that a query shape is being matched with a database shape whose category is known a priori. The presence of context, *i.e.* the knowledge about the category of the database shape, influences the similarity computations, and helps us to obtain better matching performance. Next, we build upon our category-influenced matching framework in which both shapes and shape categories are represented with depth-1 skeletal trees, and develop a similarity-based shape classification method where the category trees formed for each shape category provide a reference set for learning the relationships between categories. As our classification method takes into account both within-category and between-category information, we attain high classification performance. Moreover, using the suggested classification scheme in a retrieval task improves both the efficiency and accuracy of matching by eliminating unrelated com-

parisons.

Keywords: shape matching, shape classification, disconnected skeleton, shape similarity, similarity-based pattern recognition

ÖZ

KATEGORİ BİLGİSİ, İSKELET TABANLI ŞEKİL EŞLEME VE ŞEKİL SINIFLANDIRMA

Erdem, İbrahim Aykut

Doktora, Bilgisayar Mühendisliği Bölümü Bölümü

Tez Yöneticisi: Doç. Dr. Sibel Tarı

Ekim 2008, 179 sayfa

İskelet tabanlı gösterimler, yapısal kararsızlıklarına rağmen görsel şekillerin tanınması ve sınıflandırmasında başarıları kanıtlanmış gösterimlerdir. Parça yapısını tıkız ve doğal bir şekilde yakalar ve kapatma, parçaların eklemlenmesi gibi görsel dönüşümlere karşı duyarsızdırlar.

Bu tezde bağlantısız iskelet gösteriminin şekil tanıma ve sınıflandırmadaki olası kullanımları incelenmektedir. Özellikle, ilk olarak bağlamsal bilginin tanımadaki önemi, bağlantısız iskelete dayalı daha önce önerilen şekil eşleme metodlarına kategori bilgisinin farklı biçimlerde dahil edilerek bu yöntemlerin geliştirilmesiyle araştırılmaktadır. Şekillerin sözdizimsel eşlenmelerindeki görüşün tersine, bize göre karşılaştırılan şekillerinin anlamsal rolleri birbirinden farklıdır ve buna göre sorgulanan şekil veri tabanında yer alan ve kategorisi bilinen bir şekil ile eşlenmektedir. Bağlamın yani veri tabanındaki şeklin kategorisine dair bilginin varlığı, benzerlik hesaplamasını etkilemekte ve eşleme başarımını arttırmaktadır. Sonradan kategorinin etkilediği eşleme metodumuz kullanılarak ki bu yöntemde hem şekiller hem de şekil sınıfları derinliği bir olan isleket tabanlı ağaç yapıları ile ifade edilmektedir, benzerliğe dayalı bir şekil sınıflandırma yöntemi geliştirilmiştir. Bu yaklaşımımızda şekil sınıfları için yaratılan kategori ağaçları, kategoriler arasındaki ilişkilerinin öğrenilmesi amacıyla kullanılan bir dayanak kümesi oluşturmaktadır. Sınıflandırma metodumuz, hem kategoriler içindeki hem de kategoriler arasındaki bilgiyi dikkate aldığı için yüksek sınıflandırma başarısı elde edilmektedir. Dahası, önerilen sınıflandırma yönteminin bir geri çağırma görevinde kullanılması ilgisiz kıyaslamaları engellediği için karşılaştırma işleminin verimini ve doğruluğunu arttırmaktadır.

Anahtar Kelimeler: şekil eşleme, şekil sınıflandırma, bağlantısız iskelet, şekil benzerliği, benzerliğe dayalı örüntü tanıma

ACKNOWLEDGMENTS

First of all, I would like to thank to my supervisor Sibel Tarı for not just her support and assistance in this thesis but also for sharing her views on science, life, and everything with me.

I am forever grateful to my beloved parents, Kadir Bora Erdem and Gülfem Erdem, for the opportunities they have given me throughout all these years.

My special thanks go to my best friend, my twin, my other half, Erkut Erdem. Without his support and help, this thesis work would have never been possible.

I am very grateful to my grandmother Şükran Erdem who passed away last year. Belonging to the first generations of our young republic, she was and always will be a role model for me. I really miss our occasional conversations.

Similarly, I am also very grateful to Nuran Halatçı, my other grandmother, for opening her house for Erkut and me in our undergraduate years and sharing her wisdom with us.

I would like to express my gratitude to my advisory committee members, Hakki Toroslu and Mine Ozkar for their friendship and their precious comments in our meetings.

I am very thankful to George Stiny for inviting me to Massachusetts Institute of Technology. My days there were limited but very precious.

My special thanks to the everyone I met in my department. Since this is a long list, I won't state their names here. I know each one of them will be a close friend of mine forever.

The research performed in this thesis has been supported by the research grant TUBITAK-105E154 and TUBITAK-BAYG PhD scholarship.

To the memory of my grandmother Şükran Erdem

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation

The ultimate goal of visual perception is to recognize or classify surrounding objects through images of the environment cast on retina, the light-sensitive part of the eye. As an emerging scientific discipline, *computer vision* shares the same goal, whose premise is that we will one day have computer systems with capabilities equivalent to those of the human visual system. But unlike a biological eye, input to a computer vision system is a digital image composed of pixels having discrete brightness values.

Taking its roots from the field of *artificial intelligence (AI)* in the beginning of 1960s, the problem of computer vision was first approached as a fairly simple problem that even Marvin Minksy, one of the fathers of AI, assigned this problem to an undergraduate student as a summer project [38]. Hence, in computational terms, these early approaches to computer vision soon failed to provide a clear understanding of the principles of vision. The well-founded theories were established during the next twenty years by the scientists like David Marr. As an influential figure, Marr suggested to interpret vision as an *information processing system* that should be investigated in three interrelated levels: (1) *computational theory* – what is to be computed and why?, (2) *representation and algorithm* – how the computation is performed? and (3) *hardware implementation* – how it is to be realized physically? [62]. Although the efforts of Marr and his colleagues transformed the field into a concrete science and significant progress has been made since then, we don't have yet a computer vision system that can fully compete with humans in its ability to recognize or classify visual objects.

The visual recognition and classification of objects require learning mechanisms that combine visual information with prior knowledge and experience. The primary source of

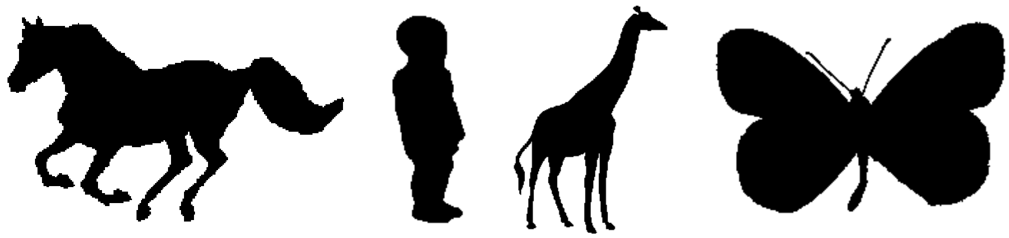


Figure 1.1: Objects can be immediately recognized and classified based on their shapes.

visual information is the *shape* knowledge since, in general, it is alone sufficient to recognize and classify a given object (Figure 1.1). The other visual clues crucial to recognition and classification are *color*, *texture*, and *spatial information*. For instance, you recognize your car in a parking lot by a search based on its color and location; or wild animals such as leopards or zebras can be classified on the basis of the color and texture of their skins. In these days, appearance-based models have gained popularity among computer vision community. At first, these studies ignore shape information and model objects by a set of image patches representing local appearance. However, there is now an increasing number of works that incorporate shape information into appearance-based approaches, *e.g.* [30, 68, 69, 95].

In this thesis, we will focus on shape-based recognition and classification of objects. The shape of objects present in nature exhibit great variability, and thus the key issue is choosing the appropriate representation scheme for both of these two problems, as is the case for all information processing systems. A shape representation should be insensitive to not only geometric similarity transformations (*i.e.* translation, rotation, and scaling) but also visual transformations such as occlusion, deformation and articulation of parts. In this regard, there is a long history of research on shape representation and recognition (For a historical discussion, see [53, 114]). Approaches to two dimensional (2D) shape representation can mainly be grouped into two broad categories: *boundary-based* (*e.g.* [9, 31, 50, 103]) and *axis-based* or *skeleton-based* (*e.g.* [3, 6, 32, 36, 91, 96, 107, 125, 127]) representation schemes. In boundary-based approaches, shapes are either represented by a set of boundary points or by a set of boundary curves. On the other hand, in skeleton-based approaches, shapes are modeled in terms of a set of axial curves explicitly representing parts of the shapes. Skeleton-based representations are superior to boundary-based ones as they naturally capture part structure and provide insensitivity to articulations and occlusion.

1.2 The Objective and Major Contributions of This Thesis

Our principle goal in this thesis is to develop efficient and reliable methods for shape matching and shape classification using the disconnected skeleton representation of Aslan and Tari [3]. In this regard, it is important to note that the proposed approaches strongly depend on our choice of representation, since some of the presented computational mechanisms becomes feasible as a consequence of the (very coarse but very stable) structure of extracted shape skeletons. The major contributions of this dissertation can be listed as follows:

1. *Enriching Disconnected Skeleton Representation*

   At the representation level, we explore the approaches to enrich the disconnected skeleton representation of Aslan and Tari [3], so that we eliminate some drawbacks of the original skeleton scheme. In particular, first, we present a way to make information regarding boundary details available for the positive skeleton branches. The information is fetched from a related smooth distance surface proposed by Tari, Shah and Pien [107], which we call TSP surface throughout the thesis, and specified in the form of a one-dimensional *radius function* representing the approximate distance to shape boundary along the branch. Second, we devise a multi-level approach to increase the level of detail in skeleton descriptions. Our approach relies on *segmenting a given shape into its parts* based on its disconnected skeleton structure and performing the skeleton analysis on the extracted parts to obtain a hierarchical representation.

2. *Incorporating Semantic Category Knowledge into Shape Matching*

   Motivated by the importance of context in human similarity judgments, we investigate a number of ways to incorporate semantic category knowledge into shape matching process. First, we present a novel extension to the tree edit distance-based shape matching algorithm of Baseski [7]. In the proposed approach, each shape in the database has a category label and the matching process of a query shape to a database shape is influenced by the additional (categorical) information provided by all the database shapes belonging to the same category. We refer to this algorithm as *category-influenced shape matching*. Building upon this formulation, we then present a coarse-to-fine strategy to incorporate *categorical boundary similarity* into shape matching by utilizing the approximate radius functions mentioned previously. Lastly, we make use of category knowledge to achieve *contextual sensitivity to articulations* in shape matching. Based on the structure of disconnected skeleton, we define a novel representation space for articulations where similar articulations lie close to each other, enabling to construct articulation priors from the members of a shape category and to make inferences about likely articulations. We incorporate this approach to the method of Aslan and Tari [3] and come up with a shape matching framework that is sensitive to unlike articulations but insensitive to likely ones.

3. *A Similarity-Based Approach for Shape Classification*
   We present a novel (supervised) shape classification method by employing a *similarity-based* approach. Having a network structure, the proposed framework first computes the distances between a given shape and existing shape categories in the database by using a variation of our category-influenced shape matching method. Then, these computed distances are embedded into a similarity space, in which support vector machine (SVM) classifiers are previously trained for each shape category, and the final decision is made according to the outputs of SVM classifiers. The similarity-based approach brings considerable improvements in terms of performance over classifying shapes based on a nearest-neighbor strategy. In this regard, it is important to note that similarity-based approaches have great importance especially for studies in structural pattern recognition as the learning and classification techniques for structural pattern recognition are not as diverse as the number of algorithms proposed in statistical pattern recognition.

1.3 Organization of The Thesis

The organization of the thesis is as follows. In Chapter 2, we give a brief review of disconnected skeleton representation of Aslan and Tari [3], and then discuss how the representation can be enriched to eliminate some of its drawbacks. In Chapter 3, we compare and contrast several skeleton-based representation schemes proposed in the literature, and discuss how they are used in generic shape recognition. In Chapter 4, we analyze two other shape matching methods, *i.e.* the method of Aslan and Tari [3] and Baseski [7], which are all based on disconnected skeleton representation of shapes. In Chapter 5, we investigate contextual effects of semantic category information on matching two shapes, where we revise and extend the matching methods described in Chapter 4 in a number of ways. In Chapter 6, we present a novel similarity-based shape classification approach based on the category-influenced shape matching method devised in the preceding chapter. In Chapter 7, we conclude the thesis with a summary of our contributions and some discussions. In Appendix A-E, we provide tables of matching and classification results obtained with the methods discussed in the thesis.

CHAPTER 2

DISCONNECTED SKELETON

In the previous chapter, we discussed visual object recognition and classification in general and compared and contrasted two generic approaches of representing objects by their shape. Among those approaches, *(local symmetry) axis-based* representations, commonly referred to as *shape skeletons*, are one of the widely used and investigated representation schemes ever since the seminal work of Blum [11]. The skeletal representations provide a compact and perceptually meaningful way of representing shape as they capture the part structure and yield a shape centered coordinate frame.

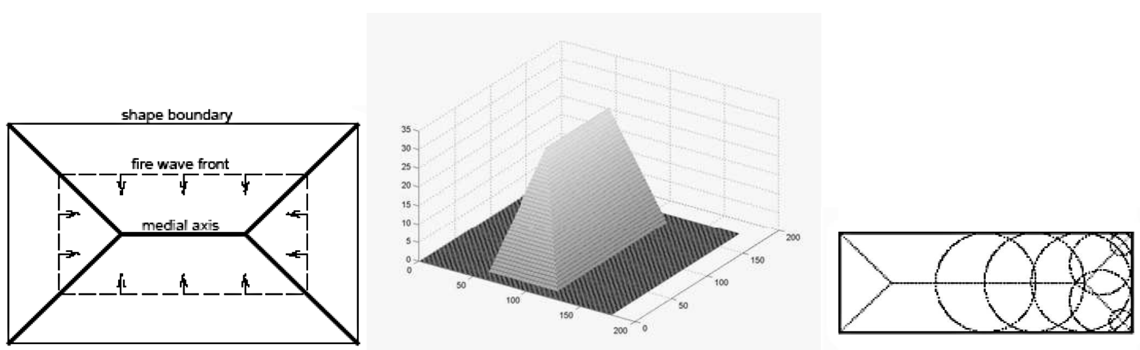
We start this chapter with a brief review of Blum's skeleton, focusing on the basic definitions. Following that, in the next section, we discuss the *Disconnected Skeleton* representation of Aslan and Tari [3] which is the underlying shape representation for the shape recognition and the shape categorization frameworks proposed in this thesis. After giving the formulation of disconnected skeleton, we will discuss its main advantages and disadvantages and then present various ways of enriching the disconnected skeleton representation in order to overcome some of its drawbacks. Finally, we conclude the chapter by summarizing the key characteristics of disconnected skeleton representation and discussing our contributions on enriching the representation.

2.1 A Short Review of Blum's Skeleton [11]

Blum's skeleton, also known as *Symmetry Axis Transform* or *Medial Axis Transform*, was introduced in [11] as an alternative shape representation where shapes are expressed in terms of local symmetries with a finite set of shape primitives in the form of axial curves. In contrast to the boundary-based descriptors, skeletal representations provide a local representation of the shape, which is insensitive to occlusion and changes in articulation of parts. As intended by Blum, the resulting representations are perceptually more meaningful. In this respect, there are also some recent supporting evidences on the psychophysical correlates (*e.g.* [43, 44]), and on the potential neurophysiological implementation of related mechanisms (*e.g.* [54]).

Blum's skeleton can be formulated based on the following three different approaches, each resulting in the same representation. The first one is the *grass fire model*. Suppose at time $t = 0$, fire fronts are initiated simultaneously at every point on the shape boundary. Letting these fire wavefronts propagate towards the center of the shape at uniform (constant) speed, as time goes by, they will meet at some interior points of the shape, thereby producing *shocks*. The skeleton of the shape is defined as the locus of these shock points (Figure 2.1(a)). Rather than looking at the dynamic picture of the process, one can also adopt a static view and interpret the fire wavefronts as the level curves of a surface, whose value at any point is the minimum distance to the shape boundary (Figure 2.1(b)). In this interpretation, skeleton is the set of points which are equidistant from at least two boundary points. Another way of constructing shape skeletons depends on the notion of *maximum inscribed circles* where each skeleton point is obtained as the center of a maximum inscribed circle that touches the shape boundary in more than one point (Figure 2.1(c)).

Although skeletons successfully capture the hierarchy of parts, a challenging issue to be resolved is the instability of skeletons that a small change in the shape might yield a significant change in its skeleton (however, as discussed in [55], the reverse is not true). In this respect, the success of any skeletonization method largely depends on how robustly skeletons are extracted in the presence of noise and changes in shape features such as protrusions,



(a) (b) (c)

Figure 2.1: Extracting the skeleton of a rectangle using (a) grass fire model, (b) distance transform, (c) maximum inscribed circles (images taken from [1]).

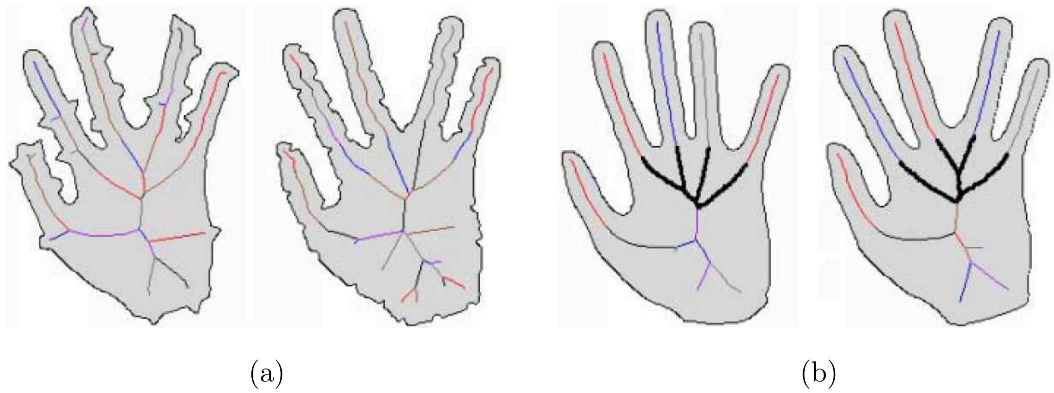


Figure 2.2: The instability of skeletons demonstrated on a collection of hand shapes. (a) The spurious branches due to boundary perturbations. (b) The topological changes in ligature regions (images taken from [115]).

indentations, necks, concavities, etc. (Figure 2.2).

Many skeleton extraction techniques exist in the literature *e.g.* [6, 32, 36, 91, 96, 107, 125, 127]. Common to all is that the skeleton branches corresponding to ribbon-like sections of shapes can always be extracted in a stable way with far less effort. On the other hand, accurate extraction of skeleton branches corresponding to noise and secondary details is a difficult process and requires more computational effort [4, 91]. The disconnected skeleton proposed in [3] differs from these approaches in the sense that the skeleton is extracted only at the locations where it can be accurately computed. As a result, the representation does not suffer from the instability of classical skeletons. In the following section, the disconnected skeleton representation of Aslan and Tari is reviewed in detail.

2.2 Disconnected Skeleton [2, 3]

Disconnected Skeleton is a very coarse but very stable skeletal shape representation. In this method, extraction of shape skeletons depends on computation of a special distance surface ϕ, which is excessively smooth version of the distance transform. Given a shape silhouette, the surface ϕ is obtained by solving the linear diffusion equation given below:

$$\begin{aligned} \frac{\partial}{\partial \sigma}\phi(x,y,\sigma) &= \left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\phi(x,y,\sigma) \\ \phi(x,y,\sigma)|_{(x,y)\in\Gamma} &= 1 \end{aligned} \tag{2.1}$$

where Γ is the original shape boundary, and σ is the artificial time parameter that can be interpreted as a scale parameter [2].

The values on the resulting surface ϕ remain in the interval (0,1], where 1-level curve of the surface correspondd to the original shape boundary Γ, and the remaining level curves approximately follow the evolution of the shape boundary towards a circle. The surface ϕ takes its root from the TSP surface proposed by Tari, Shah and Pien [106, 107], which is computed as the steady-state solution of the following linear diffusion equation with an additional term:

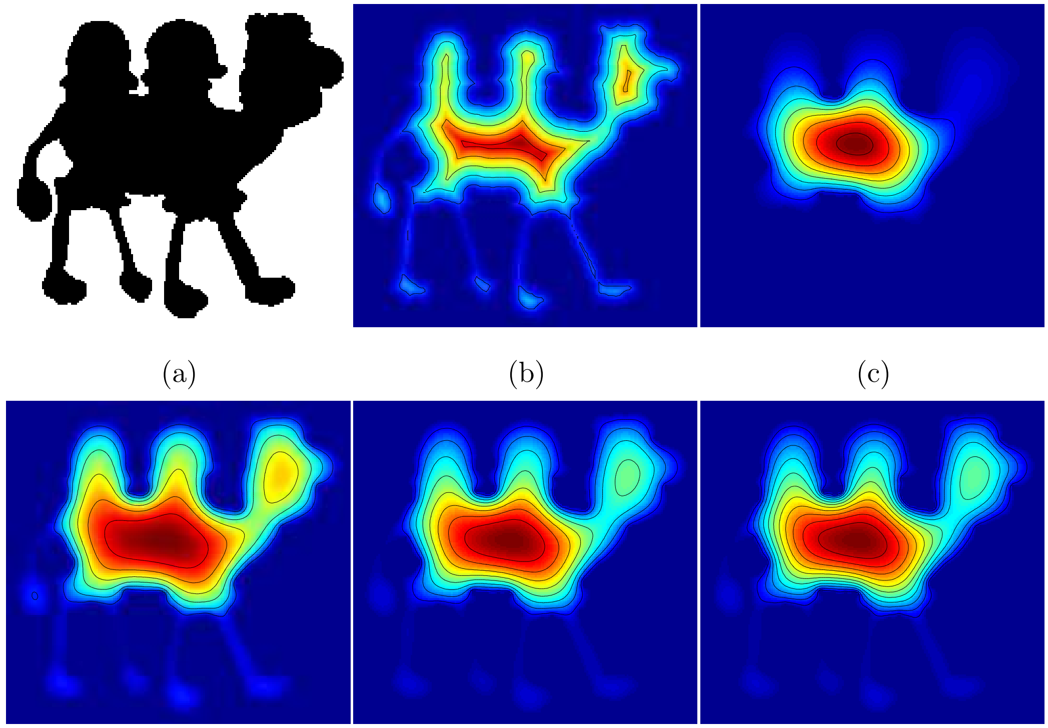
$$\frac{\partial}{\partial \sigma} v(x,y,\sigma) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) v(x,y,\sigma) - \frac{v(x,y,\sigma)}{\rho^2}$$
$$v(x,y,\sigma)|_{(x,y)\in\Gamma} = 1 \tag{2.2}$$

where Γ is the original shape boundary, σ is the artificial time parameter, and ρ is a parameter that controls the level of smoothing.

In this regard, the surface ϕ can be interpreted as the limit case of the TSP surface when we let the level of smoothing (ρ) tend to infinity. However, notice that the steady-state solution of Equation 2.1 results in a totally flat surface which is 1 everywhere. Clearly, this flat surface is not meaningful for shape analysis, and hence the diffusion is stopped a critical moment where a single extremum is reached. Note that this critical time is determined automatically by the shape itself. As expressed in [3], when the shape has two equally prominent parts, reaching a distance surface with a single extremum is computationally very time consuming. For this reason, the authors decide to preserve the dumbbell-like topology of these kind of shapes in the computation of corresponding ϕ surfaces.

To illustrate the behavior of ϕ, in Figure 2.3, we present a sample camel shape and several surface representations describing it. Given the camel shape in Figure 2.3(a), Figure 2.3(b) and (c) respectively shows the result of the Euclidean distance transform and the corresponding ϕ surface. Compare these surfaces with the TSP surfaces obtained with $\rho = 16$, $\rho = 64$ and $\rho = 256$, which are given in Figure 2.3(d)-(f), respectively. The excessive amount of regularization in computing ϕ has important consequences: First, the level curves tend to evolve to a blob-like representation of the initial shape boundary. Hence, the surface ϕ has only a single extremum point, capturing the center of this blob-like representation while the TSP surfaces might have many such extremum points. For instance, the TSP surfaces given in Figure 2.3(d)-(f) have two elliptic points corresponding to the centers of the head and body sections of the camel shape.

In [106, 107], Tari, Shah and Pien devised a simple procedure to detect skeleton points of a shape from a corresponding TSP surface. The authors simply observe the link between the curvature extrema of the evolving level curves and the differential properties of TSP



(a) (b) (c)

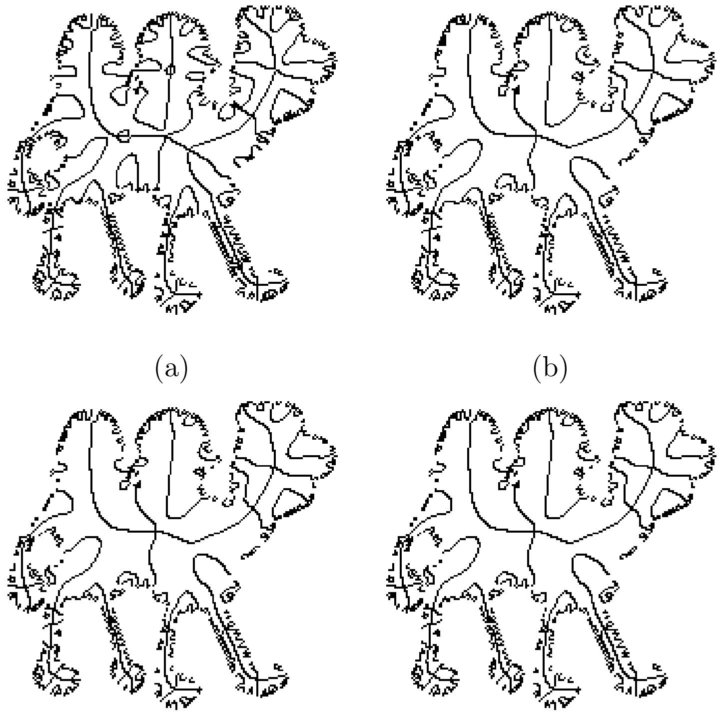
(d) (e) (f)

Figure 2.3: (a) Silhouette of a camel. The level curves of (b) Euclidean distance transform (c) $1-\phi$, (d) $1-v$, computed with $\rho = 16$, (e) $1-v$, computed with $\rho = 64$, (f) $1-v$, computed with $\rho = 256$.

surface, and define the skeleton as the closure of the set of zero-crossings of $\frac{d|\nabla v|}{ds}$, where s is the arclength in the direction of the level curves and $\frac{d|\nabla v|}{ds}$ is computed using:

$$\frac{d|\nabla v|}{ds} = \frac{\left(\left(v_y^2 - v_x^2\right) v_{xy} - v_x v_y \left(v_{yy} - v_{xx}\right)\right)}{|\nabla v|^2} \tag{2.3}$$

The skeleton points detected as zero-crossings of $\frac{d|\nabla v|}{ds}$ are always connected for each branch (until the branch gets terminated) and the skeleton branches can be classified into two sets as *positive* and *negative* (See Algorithm 2 in the Appendix of [2]). The branches that originate from a positive curvature maxima of the boundary are classified as positive whereas the ones that originate from a negative curvature minima or a positive curvature minima are classified as negative. As the value of parameter ρ denotes the level of smoothing, when ρ gets larger, the protrusions are smoothed out earlier, less important symmetry branches shrink, and the length of a branch becomes an accurate measure of its importance. This phenomenon can be observed in Figure 2.4(a)-(d), showing the skeleton extracted from the corresponding TSP surfaces computed with $\rho = 4$, $\rho = 16$, $\rho = 64$ and $\rho = 256$, respectively.

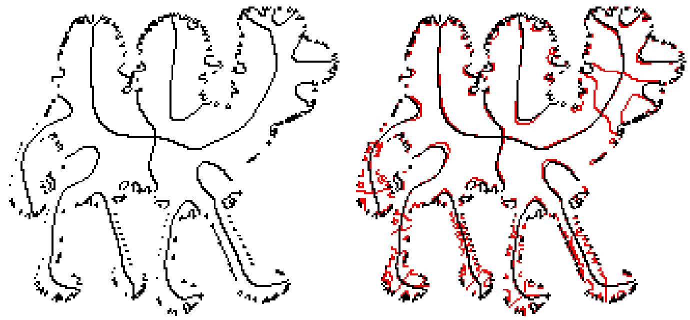


(a) (b) (c) (d)

Figure 2.4: Skeletons of the camel shape in Figure 2.3(a) extracted from the corresponding TSP surfaces, computed with (a) $\rho = 4$, (b) $\rho = 16$, (c) $\rho = 64$. and (d) $\rho = 256$.

The same skeletal analysis can be performed to extract skeleton from the surface ϕ. In this case, however, the resulting skeletons are much coarser in the sense that there exist only a small number of simple branches on which branching occurs very rarely. Moreover, in TSP skeletons, unintuitive branches might appear in the vicinity of necks due to a major pathology, which is referred to as the *saddle point instability* and is related to insufficient diffusion (See Section II.B-C of [2]). In extracting skeleton from ϕ, the saddle point instability can be avoided simply because the level of smoothing tends to infinity. The skeleton of the camel shape extracted from its ϕ surface is given in Figure 2.5(a). Compare and contrast this skeleton with the TSP skeletons of the same shape shown in Figure 2.4(a)-(d), respectively. The difference between the skeleton extracted from the TSP surface computed with $\rho = 256$ is explicitly shown in Figure 2.5(b).

The very small branches near the shape boundary appear because of the discretization and they can be easily eliminated by performing a simple pruning step. Figure 2.6 shows the resulting disconnected skeletons of some shapes after pruning. Notice that if the symmetry



(a) (b)

Figure 2.5: (a) Skeleton of the camel shape in Figure 2.3(a) extracted from the corresponding ϕ surface. (b) The difference between the skeletons extracted from the corresponding ϕ surface and the TSP surface computed with $\rho = 256$.

at the shape center is n-fold, there are n positive and n negative branches, designated as *major branches*, which meet at the shape center [3]. The remaining branches all terminate at some *disconnection point*s organized around the shape center, and hence this unconventional structure gives *disconnected skeleton* its name. At each disconnection point, a positive branch always meets with a negative one. As reported in [2, 3], one should further apply the disconnection concept (artificially) to the major positive branches in order to obtain a stable skeleton description (For a detailed analysis, see Section III.B of [2]). In Figure 2.7, the final skeleton descriptions of some shapes are illustrated.

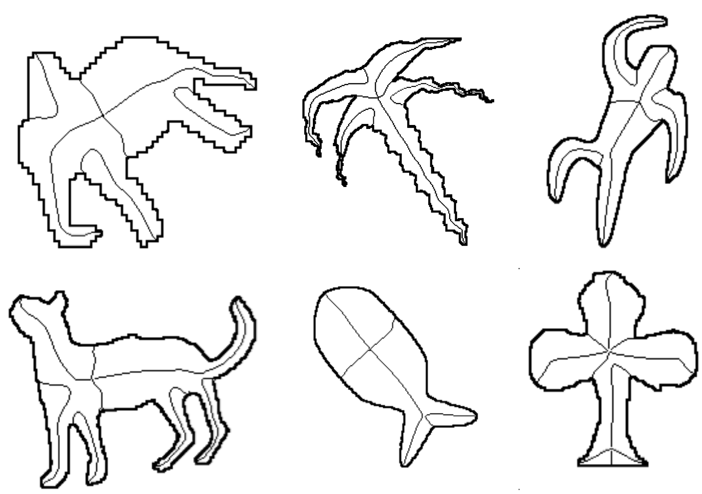


Figure 2.6: Extracted skeleton branches for some shapes after pruning (images taken from [2]).

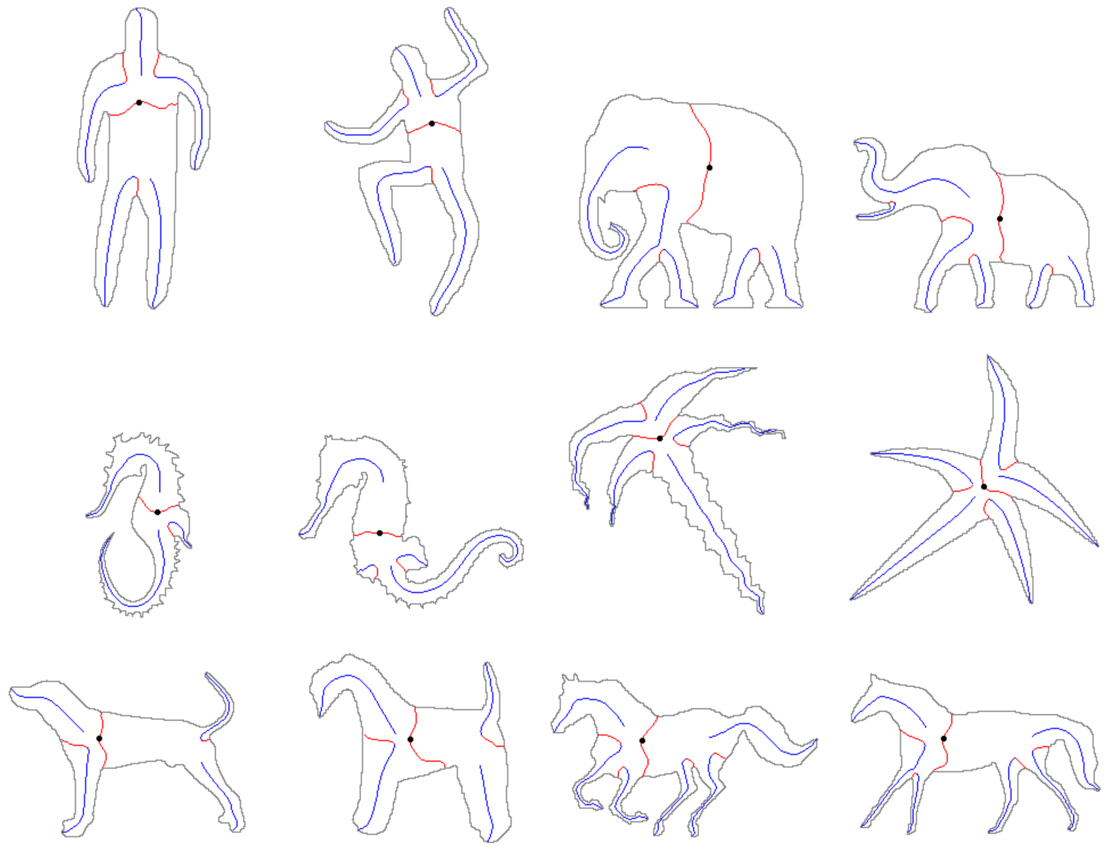


Figure 2.7: Some shapes and their disconnected skeletons. Notice that each positive branch meets with a negative branch at a disconnection point. Positive skeleton branches are shown in blue whereas the negative ones are shown in red.

In the resulting skeleton representation, the relative organization of the branches can be captured by the location of their termination points. These points can be expressed with reference to a shape dependent global coordinate frame that is constructed by any one of the negative major negative branches (Figure 2.8). This novel way of representing shapes is demonstrated to be highly robust under global transformations (*i.e.* translation, rotation, scaling) as well as articulation of parts and perturbations on the boundary [2, 3].

2.2.1 Advantages and Disadvantages of Disconnected Skeleton

Skeleton-based representations provide a compact and generic way to represent shapes in a structured manner and hence they are commonly used in visual shape recognition research, *e.g.* [5, 32, 61, 74, 90, 98, 115, 127]. Disconnected skeleton has also proven itself to be a powerful representation for shape recognition and retrieval [2, 8]. However, disconnected

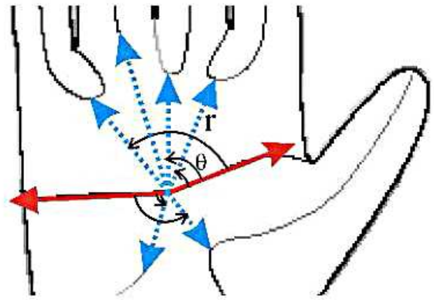


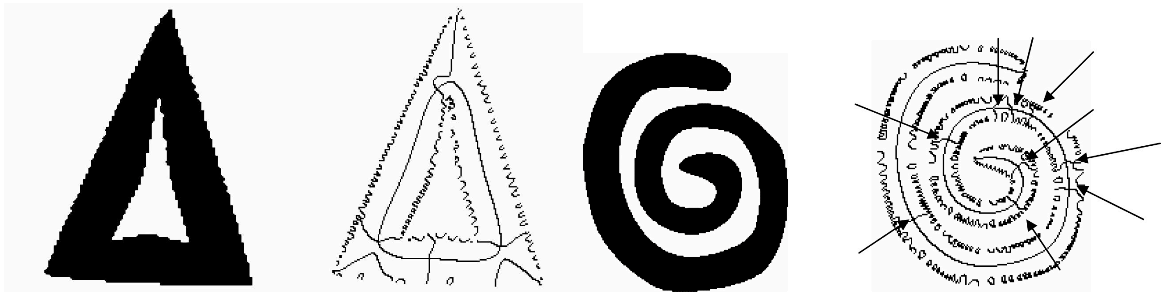
Figure 2.8: Spatial organization of extracted skeleton branches (image taken from [2]).

skeleton has its own strengths and drawbacks, as this is the case for any representation scheme.

To start with, disconnected skeleton has one remarkable advantage over the other skeletonization methods that the representation does not suffer from the instability of skeletons. This is because the skeletal analysis is performed at the coarsest scale possible and the resulting skeleton branches are unconventionally disconnected. As a consequence, no post-processing step is necessary for the skeleton prior to be used in recognition. However, one might criticize the very coarseness of disconnected skeleton descriptions. This issue is in fact about a philosophical choice of compromise between sensitivity and stability and in regards to this argument, we prefer to obtain a stable representation first than a sensitive one, and then gradually enrich the representation in a systematic way according to needs.

On the negative side, the main drawback of disconnected skeleton is the limitation that, in order to obtain disconnected skeleton description of a shape, the shape should have a closed boundary. The method is not applicable to shapes with holes (Figure 2.9(a)) or stroke shapes, *i.e.* the shapes whose width is nearly constant everywhere (Figure 2.9(b)). In these kind of situations, either the skeleton cannot be extracted accurately due to elliptical and/or hyperbolic points arise in the corresponding ϕ surface, or even if a skeleton is correctly extracted, it is not be so obvious how to define the coordinate frame in a stable way.

Another disadvantage that one might consider is the stability of the representation under occlusion or missing parts. Although the extraction of skeleton branches are not affected by these conditions since they are detected locally, the shape center might shift to a different location. When this happens, the resulting shape description might be completely different as the spatial organization of branches are expressed with reference to a global coordinate frame that is dependent to the shape center.



(a) (b)

Figure 2.9: Examples of two classes of shapes where disconnected skeleton approach do not succeed in obtaining a complete description. (a) a shape with hole and its skeleton points. (b) a stroke shape and its skeleton points (images taken from [1]).

Lastly, unlike Blum's skeleton, disconnected skeleton lacks information about boundary details in the skeleton descriptions. In Blum's original formulation [11], every skeleton branch (medial axis) is associated with a *radius function*. This radius function is a continuous, real-valued function defined on skeleton branches, whose value at each skeleton point is equal to the the distance from the skeleton branch to the closest point on the object boundary, or equivalently the radius of the associated maximal inscribed circle. By making use of the radius functions, one can reconstruct the shape exactly given the full skeleton description of the shape. Due to the excessive amount of regularization, disconnected skeleton is not information-preserving and there is no way to obtain the width of a skeleton point directly from surface ϕ. In this regard, as reported by Baseski [7], computing shape similarities merely based on the spatial organization of skeleton branches and the lengths of the branches sometimes do not reflect the perceptual similarities well (See Section 5.4. of [7]).

2.3 Enriching Disconnected Skeleton Representation

In Section 2.2.1, we have listed the drawbacks of disconnected skeleton representation. In the following sections, we will discuss various ways of enriching the representation in order to overcome some of these drawbacks. First, we will propose a simple way to obtain approximate radius functions for the extracted positive skeleton branches. Next, we will employ a multi-level approach to increase the level of detail in the descriptions.

2.3.1 Associating Approximate Radius Functions with the Positive Skeleton Branches

As we have mentioned before, the regularization employed in the formulation of the distance surface ϕ makes it practical to obtain a stable skeleton-based representation of shape. However, this stability comes at the expense of losing the information about boundary details, *i.e.* in contrast to Blum's skeleton, it is impossible to recover the distance from a skeleton point to the closest point on the shape boundary by using the corresponding ϕ surface. Hence, the radius functions of skeleton branches are absent in the resulting descriptions. In this section, we will present a straightforward way to roughly obtain this missing information by utilizing the TSP surface formulation of Tari, Shah and Pien [106, 107] where our analysis depends on one-dimensional (1D) version of the diffusion equation in Equation 2.2.

Consider a ribbon-like section of a shape illustrated in Figure 2.10. The dotted line in the figure shows the skeleton points representing the shape section. Assuming the 1D form of the Equation 2.2, the diffusion process along a 1D slice (shown in red) is given by:

$$v_{xx}(x) - \frac{v(x)}{\rho^2} = 0\ ; \quad 0 \leq x \leq 2d$$

with the boundary conditions $v(0) = 1,\ v(2d) = 1$.

The explicit solution of this equation can be easily derived as:

$$v(x) = \left(\frac{1 - e^{2d/\rho}}{e^{-2d/\rho} - e^{2d/\rho}} \right) e^{-x/\rho} - \left(\frac{1 - e^{-2d/\rho}}{e^{-2d/\rho} - e^{2d/\rho}} \right) e^{x/\rho} \tag{2.4}$$

The value of v on the skeleton point (the midpoint $x = d$) is equal to the hyperbolic cosine function $\frac{1}{cosh(d/\rho)}$, or equivalently, the distance from the skeleton point to the closest point on the boundary is given by $\rho cosh^{-1}(\frac{1}{v(d)})$. The explicit solution given in Equation 2.4 is certainly not valid for the 2D case as the interactions in the diffusion process are more complicated. However, it can be used as an approximation as follows. Let s be a skeleton

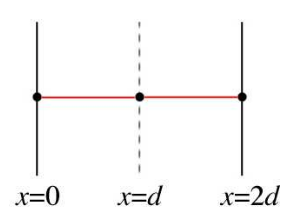
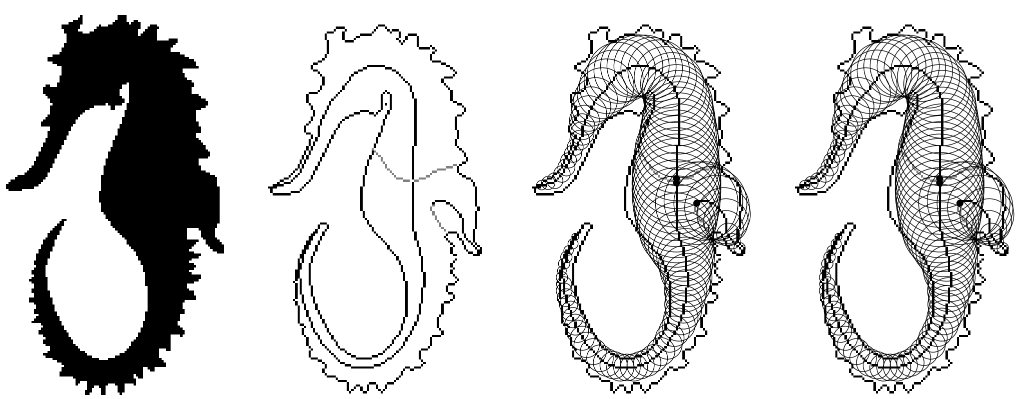


Figure 2.10: An illustration of a ribbon-like section of a shape and its skeleton description (the dotted line).

point located at (s_x, s_y) along a positive skeleton branch. Given a corresponding TSP surface v computed with a sufficiently large value of ρ, the minimum distance from s to the shape boundary, denoted by $r(s)$, can be approximated with:

$$r(s) = \rho cosh^{-1}\left(\frac{1}{v(s_x, s_y)}\right) \tag{2.5}$$

In Figure 2.11(a)-(b), a seahorse shape and its disconnected skeleton are given, respectively. Note that this is not the final description because major positive branches are not cut yet. The shape can be reconstructed by the radius functions associated with each positive skeleton branch exist in the disconnected skeleton description by drawing the corresponding maximal inscribed circles. Figure 2.11(c)-(d) shows results of shape reconstruction from disconnected skeleton using two different TSP surfaces, computed with $\rho = 128$ and $\rho = 256$, respectively. There is no observable change in the quality of reconstruction results with respect to the value of ρ. Notice that since the perturbations on the shape boundary is neglected in computing the disconnected skeleton, these small details cannot be exactly recovered. Moreover, the accuracy of reconstruction deviates from its true form at the dorsal fin of the sea horse. These are the locations where a positive branch loses its ribbon-like structure (having slowly varying width).



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

Figure 2.11: (a) A seahorse shape. (b) Its disconnected skeleton. (c)-(d) Shape reconstruction results from the disconnected skeleton description in using TSP surfaces, computed with $\rho = 128$ and $\rho = 256$, respectively. Due to demonstrative purposes, maximal circles are drawn at every third skeleton point and major positive branches are not cut.

Figure 2.12 shows some other shape reconstruction results for various shapes, using TSP surfaces computed with $\rho = 256$ (the same value of ρ is used in all of the subsequent sections). When it comes to storing the enriched disconnected skeleton descriptions, the approximate radius function of each positive branch is normalized with respect to the radius of maximum circle associated with the shape center to make the representation scale insensitive. Note that if the shape has two equally prominent parts, there will be two distinct shape centers, and in this case, the radius functions are normalized with respect to the closest center. Some shapes and their enriched disconnected skeletons with the approximate radius functions are given in Figure 2.13 through Figure 2.15.

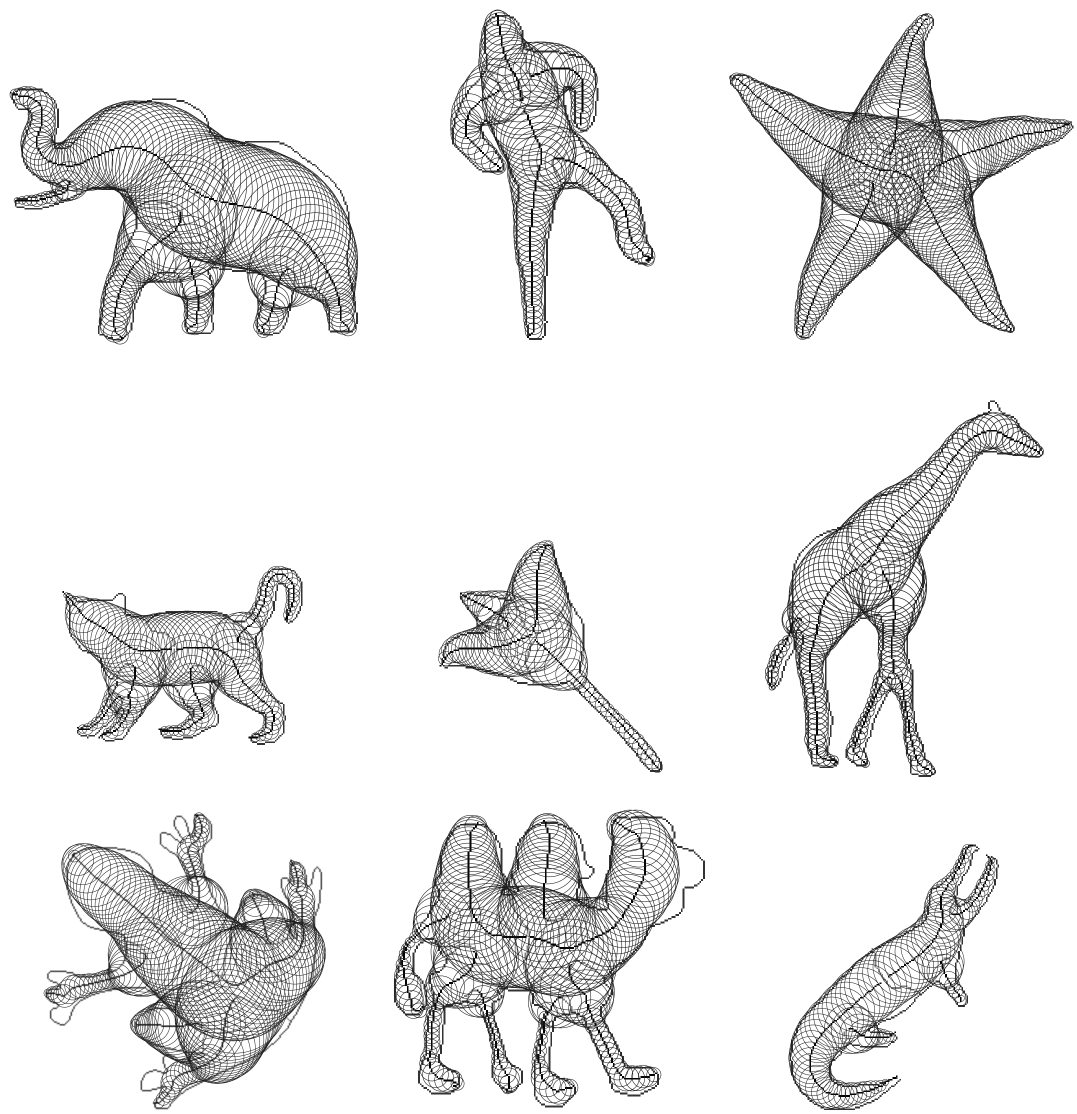


Figure 2.12: Reconstructing shapes from their disconnected skeleton descriptions using approximate radius functions obtained from the corresponding TSP surfaces.

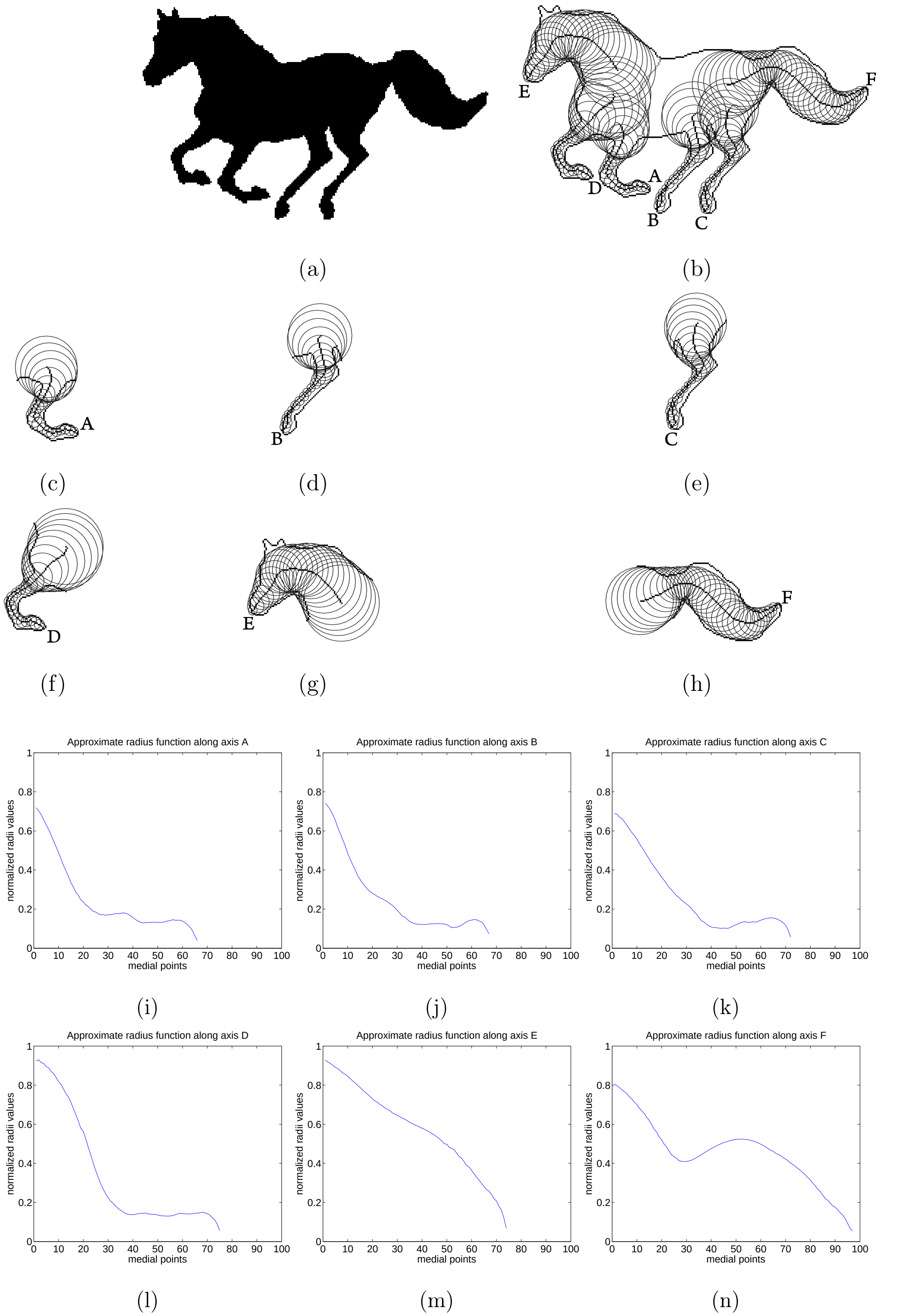


Figure 2.13: (a) A horse shape. (b) Shape reconstruction from disconnected skeleton. (c)-(h) Reconstructed shape sections associated with the positive skeleton branches A-F, respectively. (i)-(n) Approximate radius functions associated with the skeleton branches A-F, respectively.

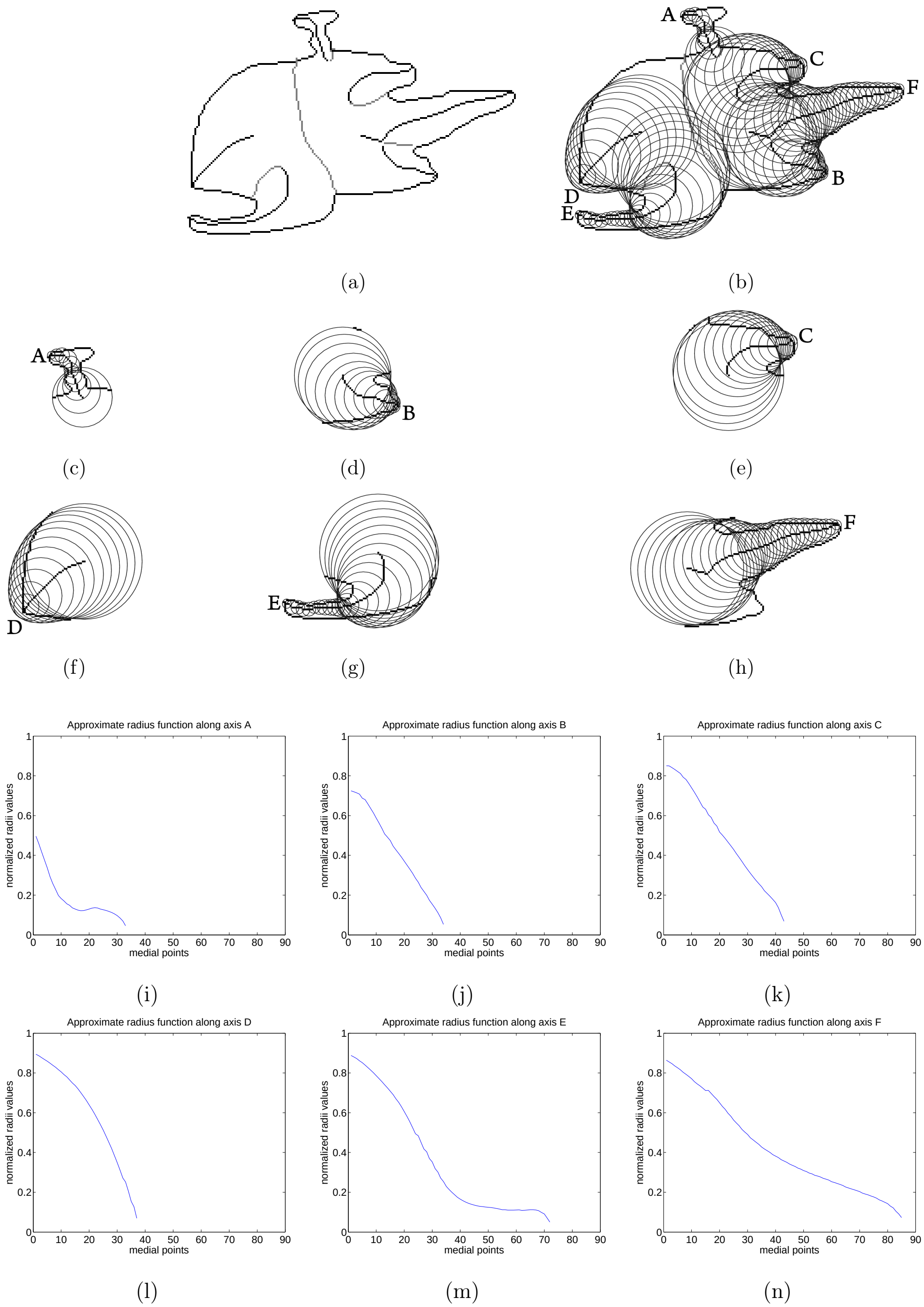


Figure 2.14: (a) A helicopter shape. (b) Shape reconstruction from disconnected skeleton. (c)-(h) Reconstructed shape sections associated with the positive skeleton branches A-F, respectively. (i)-(n) Approximate radius functions associated with the skeleton branches A-F, respectively.

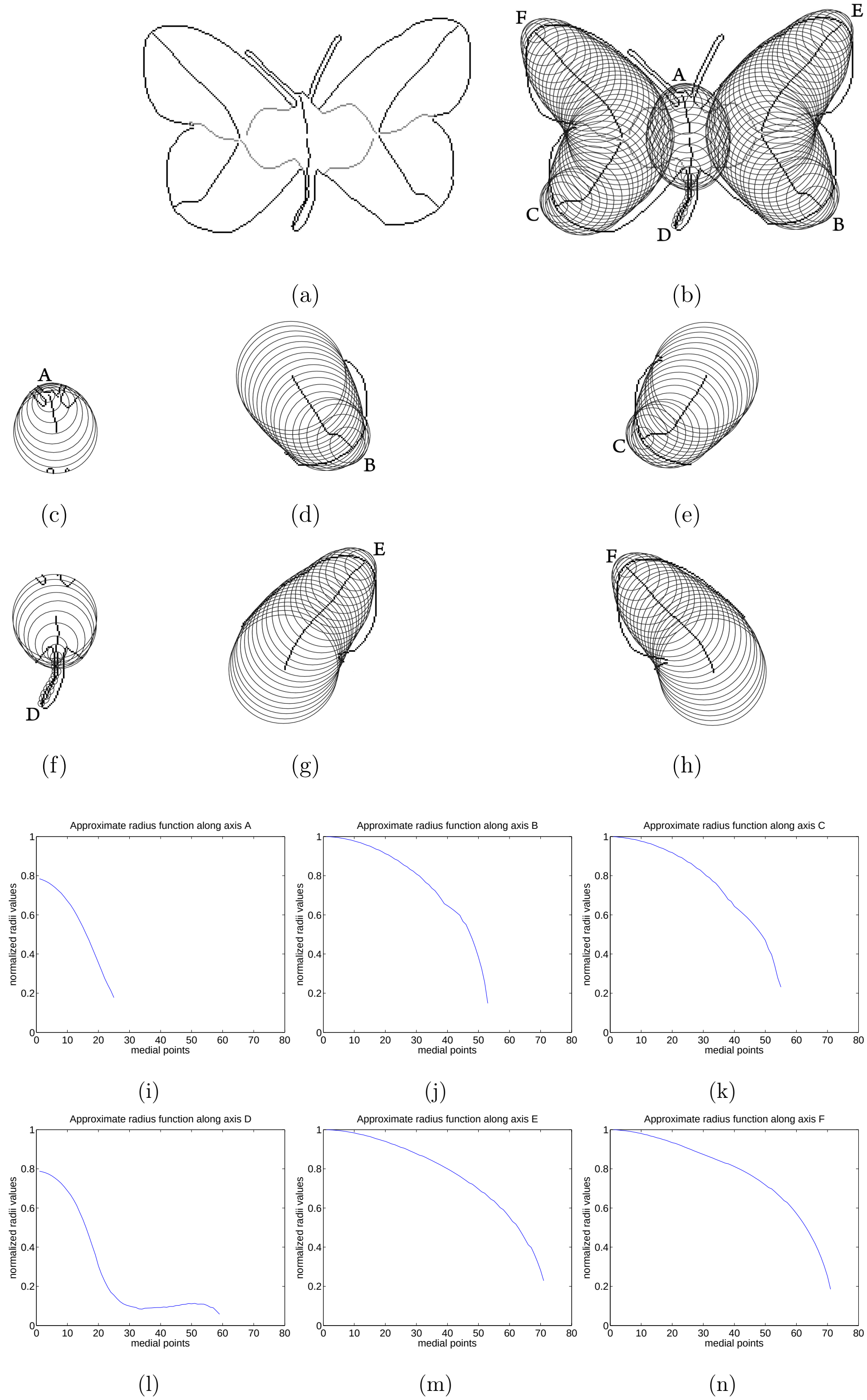


Figure 2.15: (a) A (two-centered) butterfly shape. (b) Shape reconstruction from disconnected skeleton. (c)-(h) Reconstructed shape sections associated with the positive skeleton branches A-F, respectively. (i)-(n) Approximate radius functions associated with the skeleton branches A-F, respectively.

2.3.2 A Multi-Level Hierarchical Approach To Increase The Level Of Detail In Disconnected Skeletons

The intention of Aslan and Tari in devising the disconnected skeleton is to obtain the coarsest but the most stable representation of shapes, and therefore the part structure captured by the disconnected skeleton is very simple. The extracted skeletons are in the form of a set of (unconventionally disconnected) skeleton branches, each corresponding to one of the most descriptive parts of the shape. Moreover, no branching occurs on any of the skeleton branches, meaning that the level of hierarchy is always 1 in the skeleton descriptions. Despite this coarse structure, the retrieval performance of disconnected skeleton-based shape matching algorithms of Aslan and Tari [1] and Baseski [7] were reported to be high.

Even though the discriminative power of disconnected skeleton in shape recognition is effective, one might concern about the coarseness of the representation that it lacks sensitivity. It appears that multi-level hierarchical representation schemes are required to satisfy the opposing goals of sensitivity and stability. In this regard, we propose to increase the level of detail gradually by employing a multi-level approach. Once the shape is segmented into its parts based on its disconnected skeleton description, the skeletal analysis can be performed on the extracted shape sections and finally, one can obtain a hierarchical disconnected skeleton representation.

In disconnected skeleton, each positive (negative) skeleton branch is associated with a boundary segment, which is bounded by two negative (positive) branches neighboring the positive one. To segment a shape into its parts, Baseski made use of this fact and proposed fitting cubic Bézier curves to the starting and termination points of the neighboring negative branches of each positive branch [105]. We demonstrate this approach in Figure 2.16. The cat shape shown in Figure 2.16(a) is segmented into six parts based on its disconnected skeleton given in Figure 2.16(b). The extracted parts, corresponding to the legs, head and the tail of the cat, and their disconnected skeletons are given in Figure 2.16(c)-(d), respectively. A drawback of this approach is that when the termination points of the negative branches are very far from each other, the extracted part might be perceptually less meaningful (See the tail section of the cat in Figure 2.16(c)).

An alternative approach to shape segmentation using disconnected skeleton might be computing the maximum circles at the termination points of the positive skeleton branches. To compute the radii of the maximum circles, one can employ the approach in Section 2.3.1 and utilize Equation 2.5 after computing a corresponding TSP surface. Shape parts extracted in this way resemble the shape primitive that is referred to as *circle* in FORMS [127]. The resulting segmentation of the cat shape in Figure 2.16(a), and the disconnected skeletons of the extracted parts are shown in Figure 2.16(e)-(f), respectively.

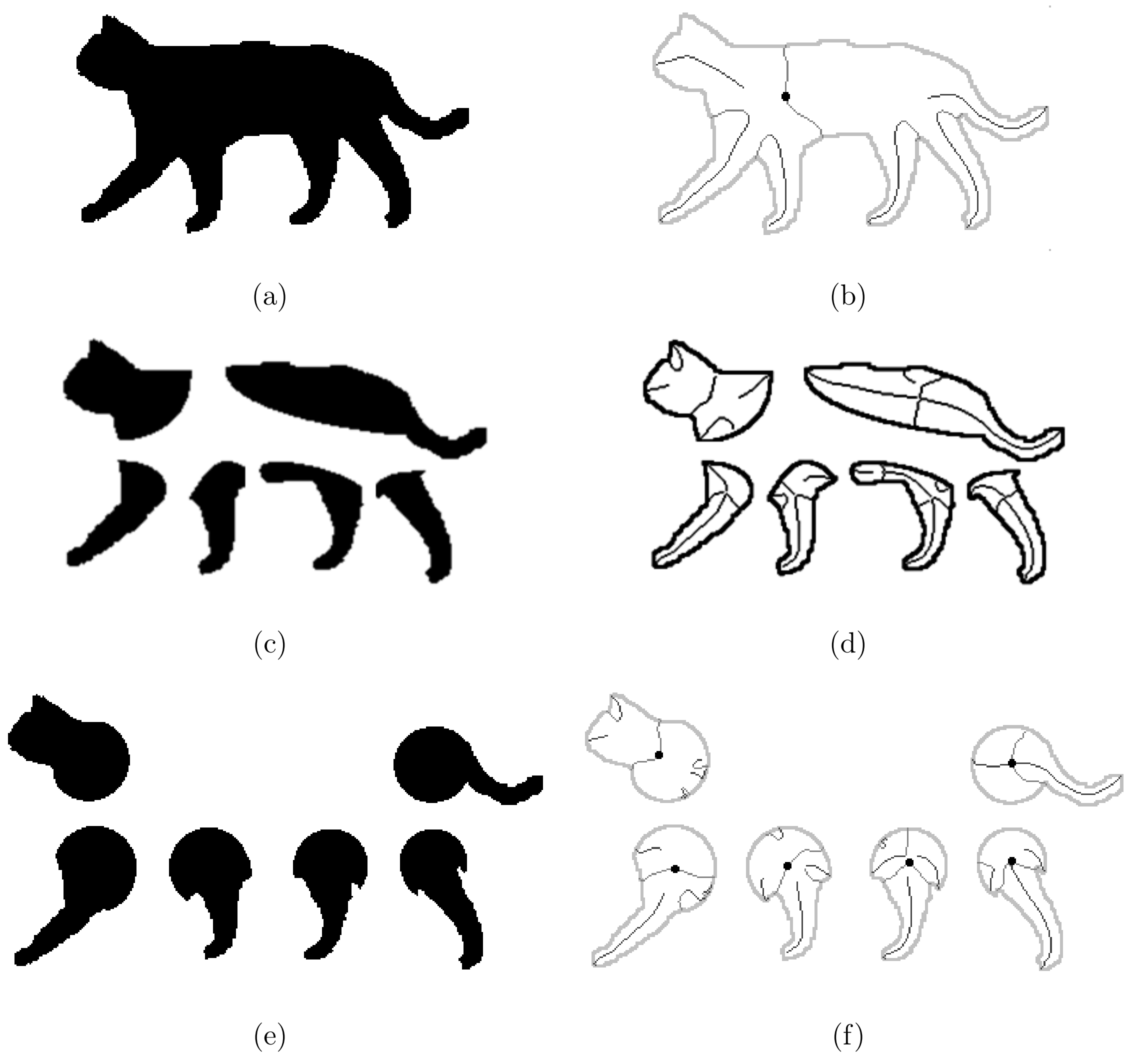


Figure 2.16: (a) A cat shape. (b) Its disconnected skeleton. (c)-(d) Its segmentation into parts by the cubic Bézier curves (images taken from [105]). (e)-(f) Its segmentation into parts by the maximum inscribed circles.

2.4 Summary and Discussion

In this chapter, disconnected skeleton representation of Aslan and Tari [2, 3] is reviewed. As a brief summary, the skeletonization process is rooted in the TSP approach of Tari, Shah and Pien [106] and relies on the computation of a special distance surface which is excessively smooth and has only a single extremum point corresponding to the center of the shape. The resulting skeletons are very coarse in the sense that no branching occurs on the skeleton branches, and besides, unlike common skeletal representations, the branches are unconventionally disconnected and terminate before reaching the shape center. Depending on the symmetry at the shape center, however, some branches meet at the shape center and these branches are used to form a shape-centered global coordinate frame. It is shown that the spatial organization of the branches captured by the location of disconnection points is a stable representation of the shape with respect to that specified coordinate frame. Moreover, due to the disconnected property of branches, extracted skeletons do not suffer from the topological instabilities that might occur near ligature regions.

Disconnected skeleton representation can be enriched in different ways. First, one can obtain the corresponding radius function of each positive branch by additionally utilizing TSP surfaces. These radius functions can be used to roughly reconstruct the shape from its disconnected skeleton and moreover, when normalized they can be used as descriptors of the boundary details. Second, a multi-level hierarchical approach to increase the level of detail in skeleton descriptions is presented. The presented approach requires segmenting the shape into its parts and performing skeleton analysis on the extracted parts in order to obtain a hierarchical disconnected skeleton representation. In this respect, two different segmentation procedures are demonstrated. While one approach fits cubic Bézier curves, the other approach we presented in this thesis uses maximum inscribed circles, and results in perceptually more meaningful segmentations.

CHAPTER 3

USE OF SKELETONS FOR SHAPE RECOGNITION

In Chapter 2, we reviewed the disconnected skeleton representation of Aslan *et al.* [2, 3] by giving details of its formulation and investigating its key characteristics. Moreover, we proposed two ways of enriching the disconnected skeleton representation to overcome some of its drawbacks. Before discussing the use of disconnected skeletons for recognition, in this chapter, we survey some of the existing skeleton-based shape matching frameworks.

Skeletal representations have been successfully used in shape recognition as they provide a compact way of expressing shapes while capturing the hierarchy of parts. In all these frameworks, a challenging issue that needs to be resolved is the instability of skeletons that two almost identical shapes might have structurally different skeletons (Figure 2.2). Hence, the success of any skeletonization method depends on how robust the extracted skeletons are in the presence of noise and shape features such as protrusions, indentations, necks, concavities. As one might expect, this instability issue can be passed over to the recognition framework, but in this case, the recognition algorithm should be devised in such a way that it includes a mechanism to handle these structural changes.

In this chapter, we review each study by pointing out how their authors attempted to solve the issues addressed above. In this respect, we mainly focus on the choice of representation scheme, *i.e.* how skeletons are extracted and their structures are expressed, in addition to the design and computational details of the underlying shape matching algorithms.

3.1 FORMS [126, 127]

FORMS was proposed by Zhu and Yuille as a categorical shape matching framework [126, 127]. Compared to other skeletal shape matching frameworks, the proposed approach is interesting in the sense that recognition is performed by a combined bottom-up/top-down approach, involving a cycle of skeleton computation and adjusting the extracted skeleton description according to the matching residual. In this way, the instabilities occurred in the skeleton extraction process can be resolved. An overview of the entire recognition process is demonstrated on a sample input shape in Figure 3.1.

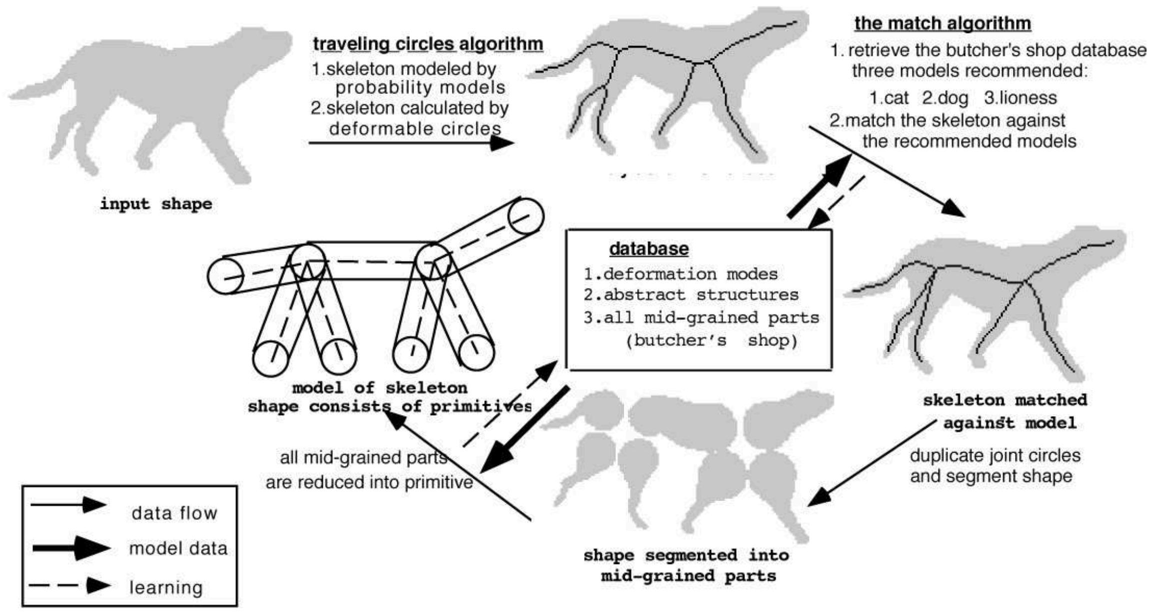


Figure 3.1: An overview of the recognition process employed in FORMS (image taken from [127]).

The skeleton of the query shape is first extracted in a pure bottom-up manner by minimizing an energy functional (Figure 3.2(a)). Based on the structure of the extracted skeleton, the input shape is segmented into parts, each of which is a deformed version of either one of the two predefined generic shape primitives, referred to as *worms* and *circles* in the text (Figure 3.2(b)), and following this the skeleton is then expressed by a graph whose nodes represent the branching and ending points of the skeleton branches (Figure 3.2(c)).

The shape database contains both the skeleton graphs of the database shapes and their segmented parts. It is important to note that the database shapes belonging to the same category share a common skeleton graph. Accordingly, each segmented part is represented in a low dimensional deformation space formed using Principle Component Analysis (PCA) based on the data collected from the category members. In addition, some other attributes

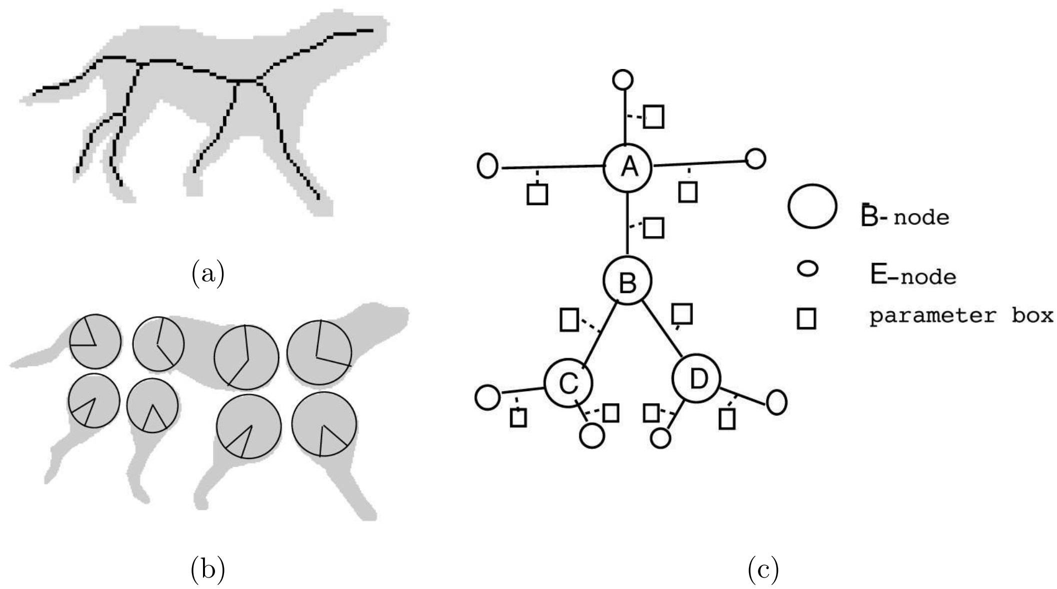


Figure 3.2: (a) The skeleton of a dog shape (b) Its segmentation of parts. (b) Skeleton graph of a human shape (images taken from [127]).

like the length (for worms), the angle specifying the angular region in which the deformations occur (for circles), the area and the radius of the maximum circles of joint points are also stored.

The proposed matching algorithm in FORMS uses a branch-and-bound strategy, returning the the partial match with the highest similarity score after searching over all possible matches between the input shape and the prototypical shape models in the database. Moreover, in contrast to other skeletal shape matching frameworks, a top-down verification process is employed as well in order to adjust the skeleton of the input shape based on the matching residual. In this respect, there exists four predefined skeleton operations (*cut*, *merge*, *concatenate* and *shift*) acting on the skeleton graphs, which can be used to obtain alternative skeleton description of the input shape(Figure 3.3). Note that each skeleton operation changes the skeleton structure, thus the segmentation into parts is different than the previous one. Hence, the similarity score is re-evaluated at each step according to changes in the description. This process is carried out until the skeleton structure of the input shape becomes equivalent to the one of the matching database shape. Although the framework is tested on a small data set, it seems the approach can deal with articulation of parts, the changes in viewpoint and occlusion.

Even though FORMS is dating back to 1995, it is quite a compelling skeletal recognition

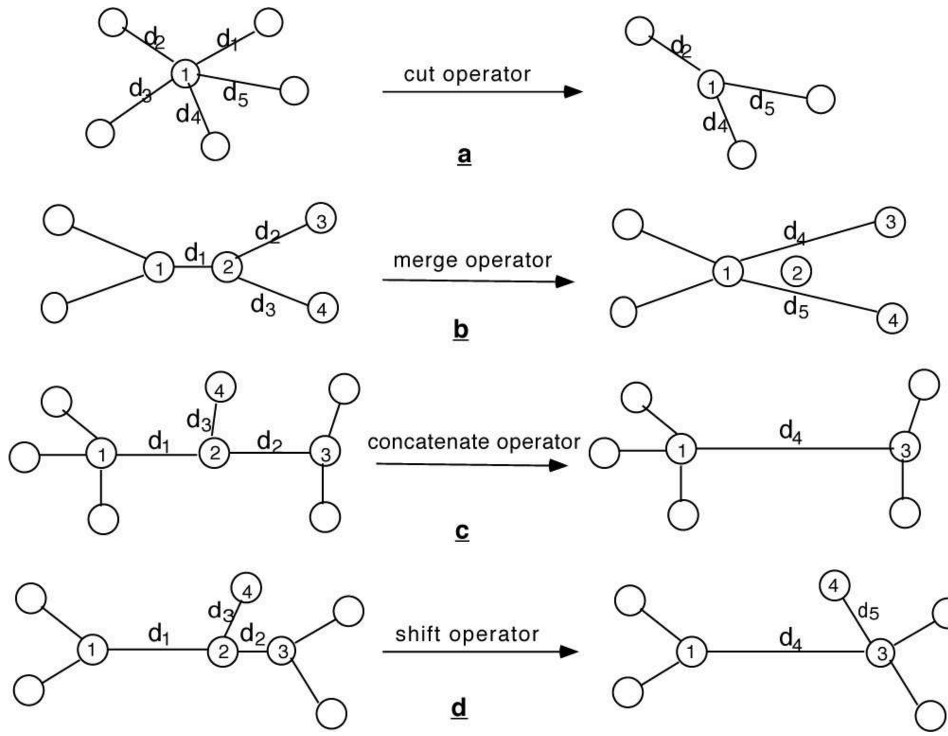


Figure 3.3: The skeleton operations defined to adjust the skeleton structure. From top to bottom are *cut*, *merge*, *concatenate* and *shift* (image taken from [127]).

framework for the reason that recognition unconventionally involves a bi-directional data flow. However, as noted by the authors themselves, the recognition success is directly related with the success of describing the input shape in terms of the specified generic shape primitives. Since the motive behind FORMS is especially dealing with the animate objects, the inanimate objects might not be described so well. Besides, introduction of a new generic shape primitive to resolve this issue is not so straightforward because this will also require a reinterpretation of the skeleton graph. As a last point, the authors addressed the issue of shape classification within FORMS framework as well, which will be discussed in a related upcoming chapter of this thesis.

3.2 Shock Trees [98] and Shock Graphs [87, 90]

Being inspired by Blum's seminal work[11], Siddiqi and Kimia devised *shock graph grammar* to classify shocks (skeleton points) formed during the propagation of the shape boundary in the skeletonization process [97]. Moreover, they utilize the grammar in eliminating the inconsistent (false) skeleton branches. Thus, by using the grammar, the hierarchy of shocks can be captured as a graph, referred to as *shock graph*, nodes of which are labeled as one of the four types of shocks. Figure 3.4 shows each of these shock types. A *first-order shock* originates from a *protrusion*, where the radius function varies monotonically. A *second-order shock* emerges at a *neck*, corresponding to a strict local minimum in the radius function. At a *third-order shock*, the radius function is approximately constant along an interval, due to *bending* of a shape section. Finally, at a *fourth-order shock* the radius function reaches to a strict local maximum, corresponding to a *seed*.

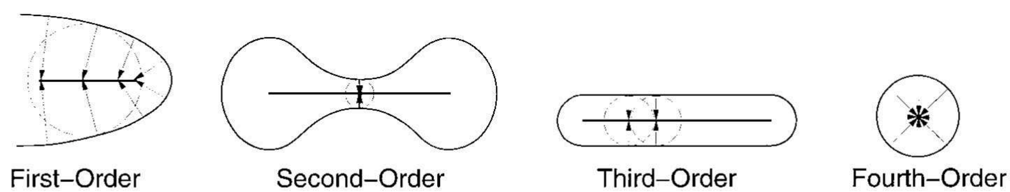
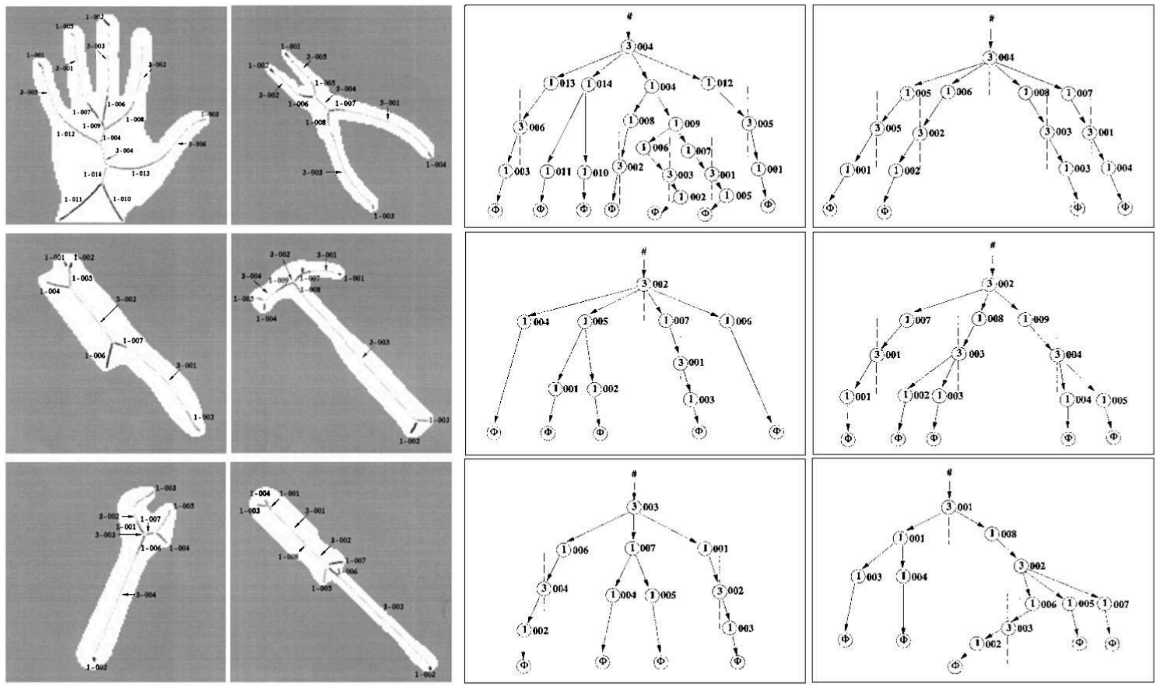


Figure 3.4: Categorization of shocks into four types (images taken from [98]). See text for the explanation.

The use of shock graphs for shape matching was first demonstrated by Siddiqi *et al.* [98]. Since inexact graph matching problem is NP-Hard, the authors first defined a mapping to reduce a *shock graph* into a unique rooted tree, which they called *shock tree*, so that polynomial time algorithms proposed for approximate tree matching can be utilized. A shock tree is in the form of directed acyclic tree and is formed by the following procedure. The oldest shock is first designated as the root of the tree while the remaining shock segments of the same type constitute the nodes of the tree. Besides, the formation times of shocks are used to define the direction of edges connecting the adjacent shock types. Shock graphs of some shapes and the corresponding shock tree representations are given in Figure 3.5(a) and (b), respectively. Several different matching methods were proposed to compute the similarity between two shock trees.

In [98], Siddiqi *et al.* presented a combined approach, involving a prior indexing mechanism and a shock tree matching method. First, a similarity between the topology of shock trees is computed, which relies on a eigenvalue characterization of shock tree's adjacency matrix. This indexing step is followed by a tree matching algorithm which takes the similarity between the shock geometry into account. Being a modification of [80], this tree matching algorithm starts from the roots of the shock trees to and continue through the



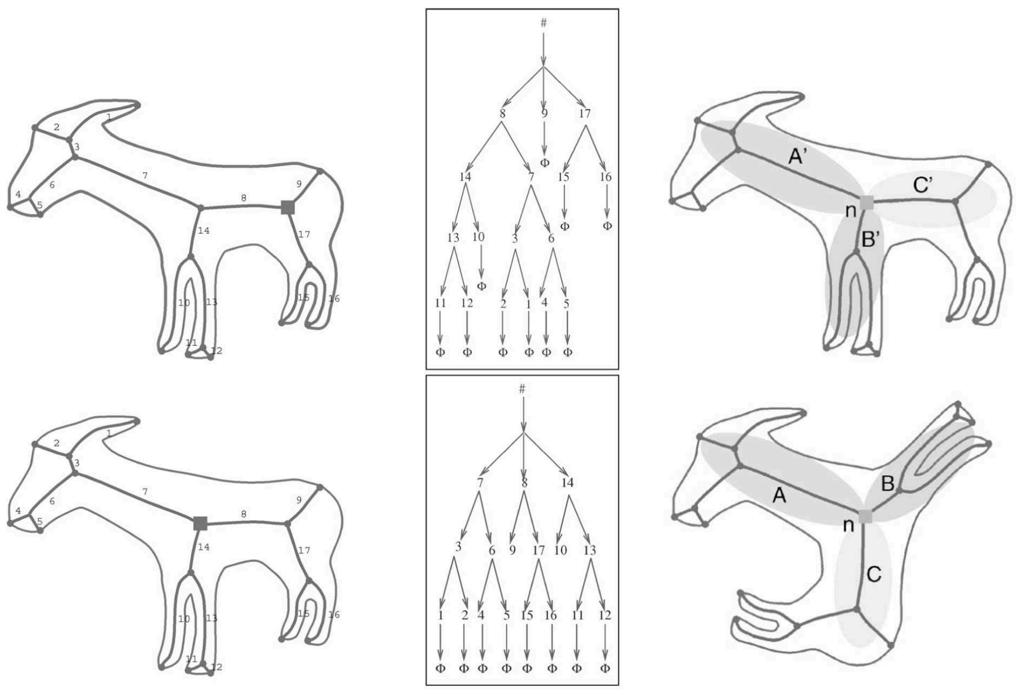
(a) (b)

Figure 3.5: (a) Shock graphs of some shapes. (b) Their shock tree descriptions (images taken from [98]).

subtrees in a depth-first fashion. The geometric similarity between two nodes is measured with respect to the aligned versions of shock segments after an affine transformation and considering the changes in scale and rotation. In [74], Pelillo *et al.* utilized the connection that any maximal subtree isomorphism between two rooted trees induces a maximal clique in the corresponding tree association graph and proposed solving the maximal clique problem in a association graph instead. Once the corresponding association tree is constructed from two shock trees, a matching between these trees is determined by computing the global maximum of a quadratic function. In this approach, the geometric similarity between two nodes is measured in terms of a linear combination of the differences in the skeletal attributes that is the lengths, radii, velocities and curvatures of the shock segments. This approach is then extended to handle many-to-many matchings in [75].

In [110], Torsello and Hancock proposed a weighted edit distance-based approximate tree matching algorithm to compute a match between two shock trees. The main idea behind this approach is the Bunke's observation in [16] that the graph edit distance and the maximal weight clique problem is equivalent under the constraint that when the cost of relabeling operation should be always higher than the sum of deleting and reinserting the nodes. In order to reflect the visual significance in calculations, the authors propose to assign each node a weight proportional to the length of the boundary segments generating the associated shocks.

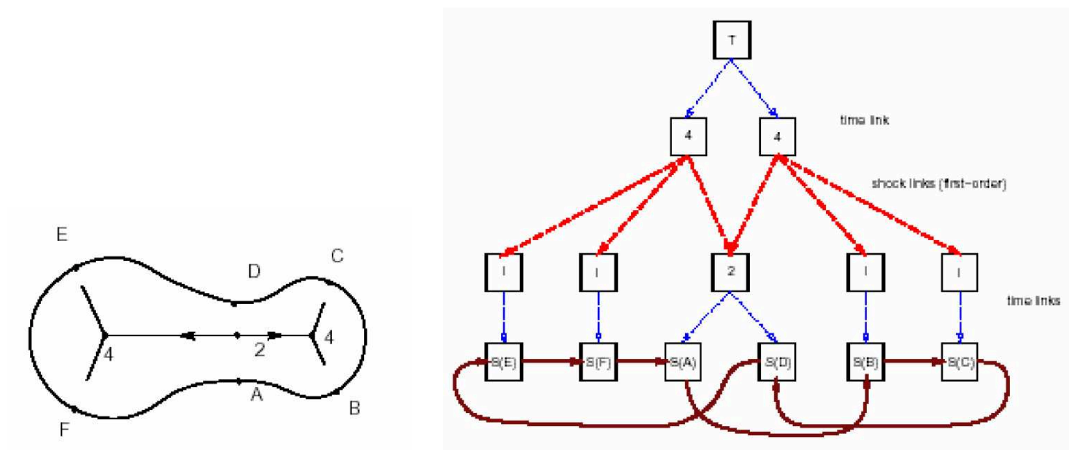
Although shock trees of Siddiqi *et al.* successfully capture the hierarchical structure of shocks, it has some drawbacks. To start with, designating the oldest shock as the root of the shock tree makes the representation unstable in the sense that a small change occurring on the shape boundary might dislocate the oldest shock, resulting in a shock tree which has a completely different topology (Figure 3.6(a)). Moreover, the planar order of skeleton branches is not explicitly stored in the nodes of shock trees. Hence, in some cases, matching two shapes with respect to their shock tree descriptions might return misleading results. For example, there is no way to distinguish between two shape where the second shape is formed from the first one by a different reordering of its branches (Figure 3.6(b)).



(a) (b)

Figure 3.6: Drawbacks of shock trees of Siddiqi *et al.*. (a) Shock graphs of two very similar shapes, together with the oldest shocks (indicated by squares) are given on the left. On the right are the corresponding shock trees. Observe that a small change in the shape might dislocate the oldest shock, causing a significant change in the topology of the shock tree representation. (b) Since the planar order of skeleton branches is not explicitly stored in the nodes of shock trees, a shape is indistinguishable in terms of its shock tree description from its another version formed by a different reordering of its branches (images taken from [90]).

In [87, 90], Sebastian *et al.* presented an alternative way of representing and matching shock graphs [97] of Siddiqi and Kimia. In this approach, shock graphs are expressed in terms of *ordered unrooted attributed trees* that bifurcation points, ending points of skeleton branches and shock segments of second-order and fourth-order are designated as the nodes of the tree whereas shock segments of first and third order constitute the edges. The skeletal attributes stored in a node are the time of formation and the direction of flow of the associated shock. Similarly, the attributes for the edges are defined by the geometry of corresponding shock segment, namely, curvature, acceleration, length and total time. A shock graph of a shape and its ordered unrooted tree representation are respectively given in Figure 3.7(a) and (b).



(a) (b)

Figure 3.7: (a) Shock graph of a shape. (b) Representing the shock graph by an ordered unrooted tree (images taken from [1]).

Moreover, Sebastian *et al.* employed an edit distance-based algorithm to determine the distance between two shock graphs represented as above. The proposed method can cope with the instabilities associated with the representation because it inherently utilizes the classification of shock graph transitions reported in [33]. Each transition is represented by any one of the four edit operations defined on the shock graph, namely, *splice*, *contract*, *merge*, and *deform*. The first three edit operations are illustrated in Figure 3.8(a)-(c), respectively. The usage of *deform* operation is to measure the dissimilarity between two matching shock segments and the boundary segments they represent. Being an extension of the measure

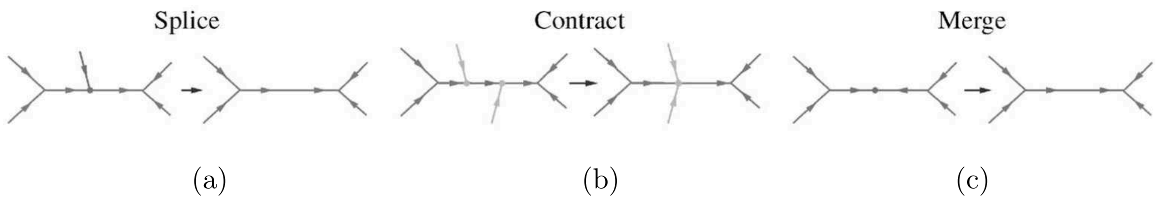


Figure 3.8: Three edit operations defined on the shock graphs. (a)-(c) *splice*, *contract* and *merge*, respectively (images taken from [90]).

in [89], deformation cost is defined by the sum of local differences in shock geometry after finding the optimal alignment between the corresponding shock segments. Additionally, the costs for other edit operations are defined by considering them as the limit cases of a deform cost.

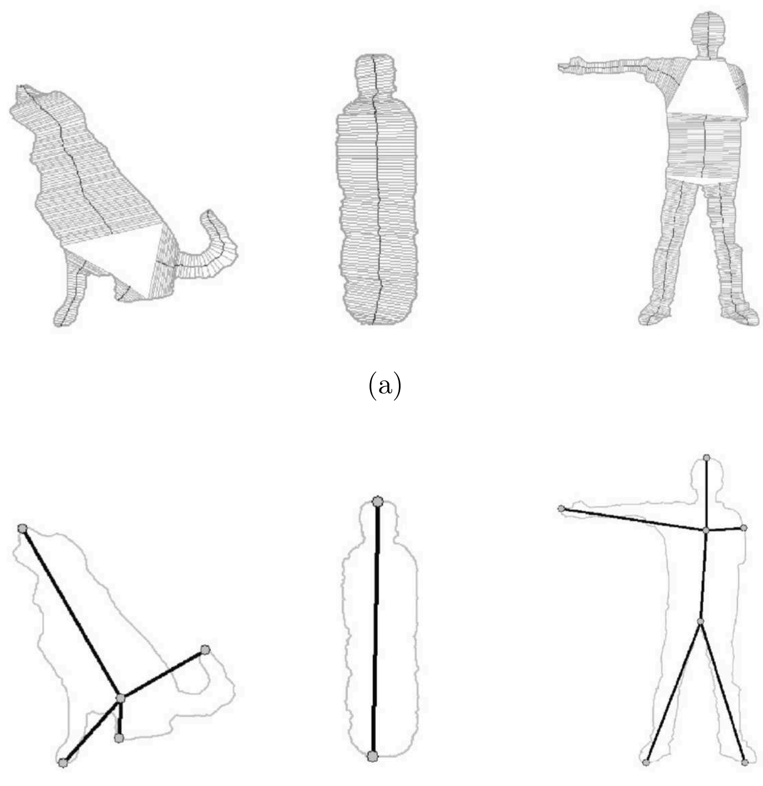
It is important to note that the skeletal shape matching framework presented in [90] does not suffer from the instability of skeleton-based representations and the experimental results show that the recognition performance is not affected much by perturbations on shape boundary, articulation of parts and reasonable viewpoint changes. On the other hand, the proposed shape matching method is computationally inefficient. The point is that although the matching method is a polynomial time algorithm with respect to the number of nodes of shock graphs, the costs of edit operations dominate the overall time complexity of the method.

3.3 Shape Axis Tree [57]

In [57], Liu *et al.* presented a compact and stable axis-based shape representation, which was referred to as *shape axis.* Formulated in a variational setting, the representation relies on a self-similarity measure which gives a set of correspondences along the shape boundary by matching two parameterizations of the shape boundary, one oriented clockwise and one oriented counterclockwise. Once an optimal matching is determined, the shape axis representation is formed by connecting the midpoints of line segments attached to each pair of matched points on the shape boundary. It is essential to note that the shape axis representation is analogous to shape skeleton, each axis representing an object substructure if the optimization criterion is based on mirror symmetry or co-circularity. Figure 3.9(a) shows shape axis descriptions of some shapes when such a criterion is utilized in determining the optimal matching.

In [57], it was demonstrated that shape axis can be expressed with a special tree structure, named as *shape axis tree*. This tree is in the form of a connected, acyclic and undirected graph where leaf nodes correspond to the ending points of shape axis whereas the remaining nodes correspond to bifurcation points, captured by the discontinuities in the set of correspondences. Note that each edge of a shape axis tree is associated with a pair of boundary segments. Figure 3.9(b) shows shape axis trees of some shapes derived from their shape axis descriptions.

Shape axis tree was first utilized for recognition by Liu *et al.* in [32, 56]. To determine the similarity between two shapes, they formulated an approximate tree matching method based on A^* search, which returns the best match between their corresponding shape axis trees. As noted before, each edge of a shape axis tree correspond to an object substructure hence the proposed matching algorithm relies on finding the correspondences between the edges of shape axis trees. The cost of matching two edges depends on how the associated boundary segment are alike and should be defined in a way that takes into account local



(a)

(b)

Figure 3.9: (a) Shape axis descriptions of some shapes. (b) The resulting shape axis trees (images taken from [56]).

deformations and regional properties.

The shape axis trees of shapes within to the same category might be structurally different due to visual transformations such as occlusion and stretching (Figure 3.10(a)). Thus, finding a one-to-one mapping between the edges is not sufficient enough to completely explain the visual correspondences. In order to cope with such structural instabilities, three additional edit operations, namely *cut*, *merge* and *merge-and-cut*, are introduced in [32, 56]. The action of each operation is demonstrated through Figure 3.10(b)-(d). In this way, many-to-many correspondences can be obtained with the help of these operations, allowing an edge to be matched with a path of two consecutive edges.

As reported in [32], a notable advantage of shock axis representation over other skeletal representations is that the proposed shape analysis can be performed on open shapes as well. However, for these kind of shapes, it is not always possible to represent the structure of their shape axis in the form of a shape axis tree. As demonstrated in Figure 3.11, the procedure defined to form shape axis trees might also result in a shape axis forest.

The matching results given in [32, 56] shows that correct correspondences can be found under challenging conditions such as articulation of parts, occlusion and missing parts. However, the recognition performance of the proposed framework was not fully investigated in

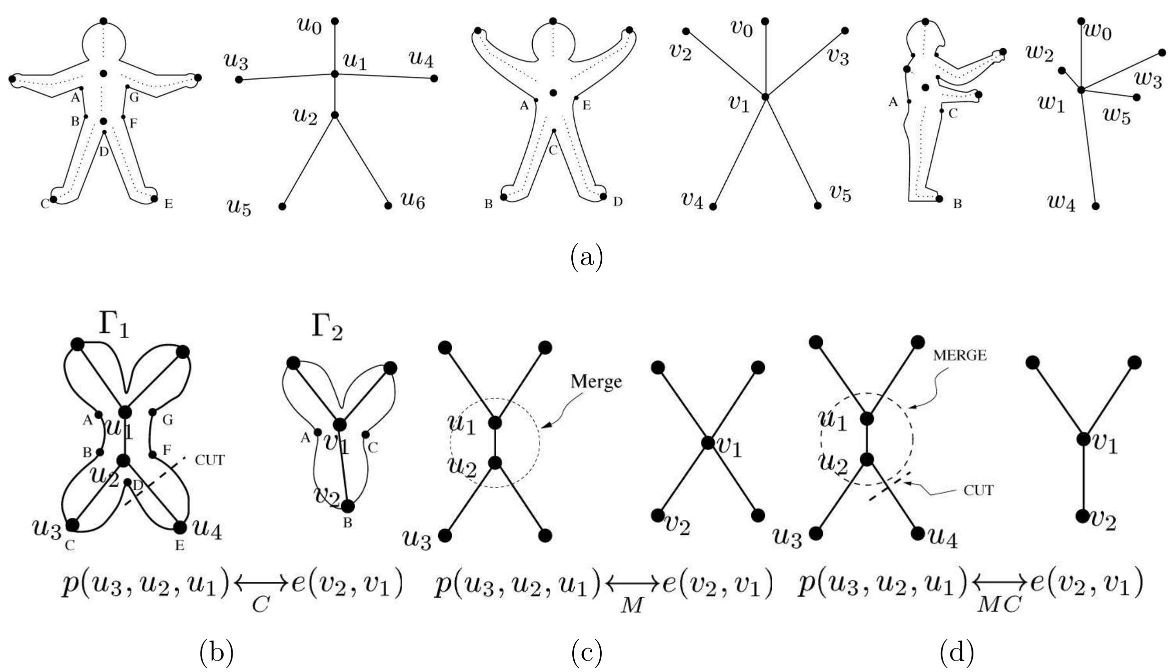


Figure 3.10: (a) Shape axis trees of some human shapes, showing some possible structural changes in the representation due to occlusion and stretching. (b)-(d) The edit operations *cut*, *merge* and *merge-and-cut*, respectively (image taken from [56]).

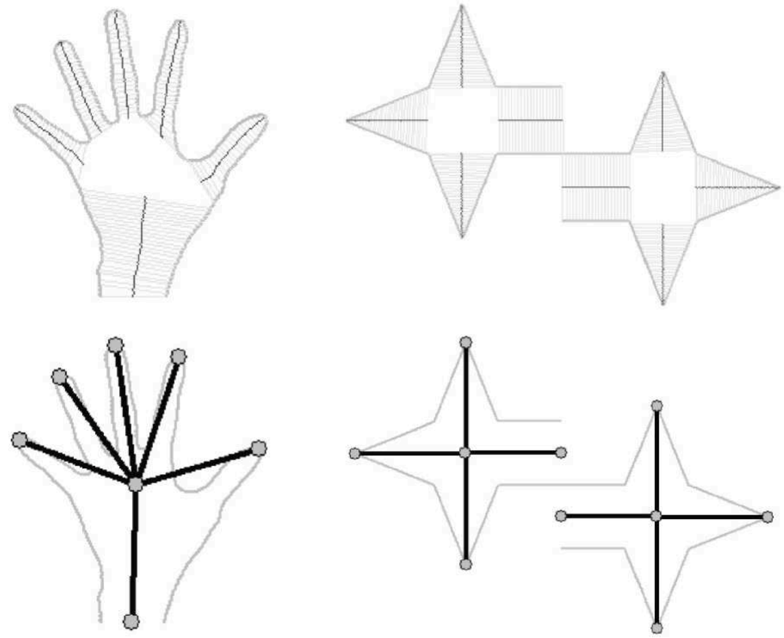


Figure 3.11: Shape axis analysis can be performed of open shapes as well, however the resulting graph structure might be a shape axis forest (images taken from [32]).

a quantitative manner. Pelillo also utilized shape axis trees to illustrate the strength of the proposed tree matching scheme in [73]. In that study, the use of shape axis trees for shape matching was tested on a very small shape data set (a total of 17 shapes, representing 6 six different shape categories), but again no recognition rate was reported. The author only stated that the proposed algorithm returned maximum subtree isomorphism in each trial.

3.4 Bone Graphs [61]

Bone graph was just recently proposed by Macrini *et al.* as a coarse skeletal representation which captures the most salient part structure of the shape [61]. In this sense, the underlying idea behind bone graphs is very similar to the one employed in the disconnected skeleton representation of Aslan and Tari [3, 2], in contrast to the fact that a bone graph is a higher level representation built on skeletons extracted by any method. The approach of Macrini *et al.* is founded on the work of August *et al.* [4] and relies on *ligature analysis* where the skeleton branches are classified as *ligature* or *non-ligature*. The term *ligature* was proposed by Blum [11] to define a proper subset of the skeleton which arises due to concave corners (Figure 3.12). Conceptually, ligature regions of skeletons are related to the attachment of parts [4] and moreover, they have little contribution to represent or reconstruct the shape [11].

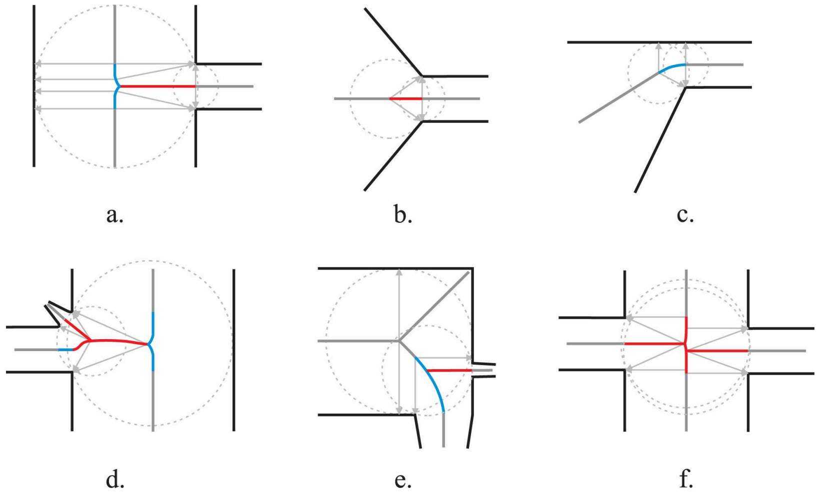
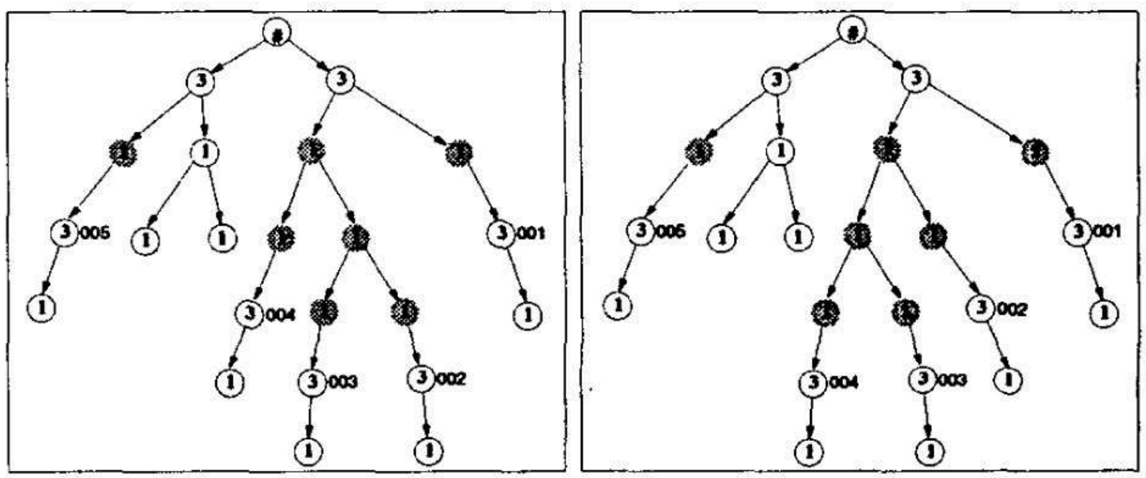


Figure 3.12: The taxonomy of ligature configurations (image taken from [61]).

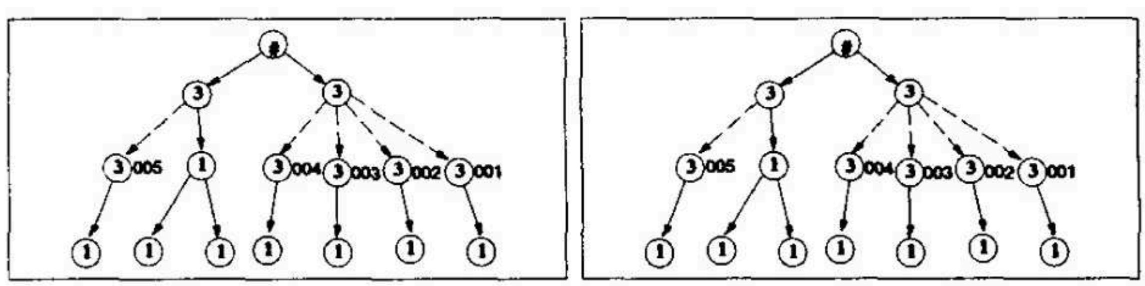
Following Blum's analysis, August *et al.* revisited the notion of ligature in [4]. After giving a formal definition of ligature based on the negative curvature minima of the boundary (a skeleton point is designated as ligature if its bitangent points fall within a fixed sized ball of the negative curvature minima), they investigated the instability of skeletons in terms of the structural changes in the ligature regions of skeletons (Figure 2.2). Accordingly, they suggested removing ligature sections of the skeletons in order to obtain robustness and they qualitatively demonstrated the use of this idea on shock trees (Figure 3.13) of some shapes. However, regarding the effect on the performance rate, the proposed approach was not analyzed quantitatively.

In [61], Macrini *et al.* utilized the *boundary-to-axis ratio* measure of Blum and Nagel [12] and formulated a more robust definition of ligature in which local scale information was considered as well (a ligature branch is defined as the branch which has at least one side whose boundary-to-axis ratio is smaller than one). Figure 3.14(a) shows the skeleton branches of a dog shape identified as either ligature (shown in green) or non-ligature (shown in black) based on this definition. However, as illustrated in the figure, some branches around junction points might be oversegmented. Hence, the initial ligature analysis is followed by a rectification step where the ligature properties around every junction point are further inspected and the problematic branches are corrected by applying the *branch fusion* operation shown in Figure 3.15 in a recursive manner. A second ligature analysis on the modified

(a)



(b)



(c)

Figure 3.13: (a) Two similar hand shapes having different skeleton structures at the ligature regions. (b) The shock trees of hand shapes where the nodes corresponding to ligature branches are shaded. (c) The resulting shock trees becomes equivalent when the ligature nodes are removed (images taken from [4]).

skeleton yields an accurate set of ligature and non-ligature branches. The modified skeleton and the corrected ligature branches of the dog shape are shown in Figure 3.14(b). The final non-ligature branches were called *bone*s and correspond to the salient parts of the shape. Figure 3.14(c) shows the bones of the dog and the reconstruction of the shape from its bones.

Based on the proposed ligature analysis, Macrini *et al.* introduced *bone graph*s as a graphical abstraction over skeleton representations, capturing the coarse part structure of a shape. Each bone graph is an unrooted tree where the nodes correspond to the non-ligature branches (bones), while the edges correspond to the ligature branches or the junctions. As shown Figure 3.14(d), the directions of the edges denote the parent-child relationship and are determined according to the relative sizes of the corresponding parts. However, in case there is uncertainty in the part hierarchy, the authors also allow undirected edges (Figure 3.16). In addition, the edges might be enriched with labels denoting the position of a part relative to its parent.

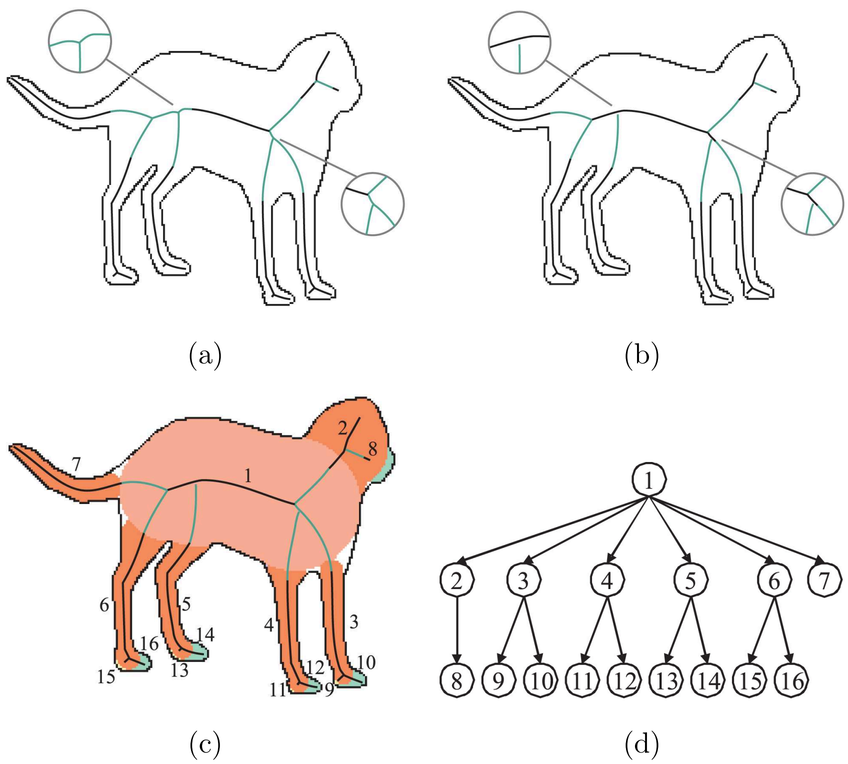


Figure 3.14: Obtaining the bone graph of a dog shape. (a) An initial classification of skeleton branches based on boundary-to-axis ratios. The ligature branches of the skeleton are shown in green whereas the non-ligature ones are shown in black. The enlarged versions of oversegmented regions are also given. (b) The final result of ligature analysis after branch fusion operations. (c) The parts and the reconstruction of the shape from the non-ligature branches. (d) The corresponding bone graph of the dog shape (images taken from [61]).

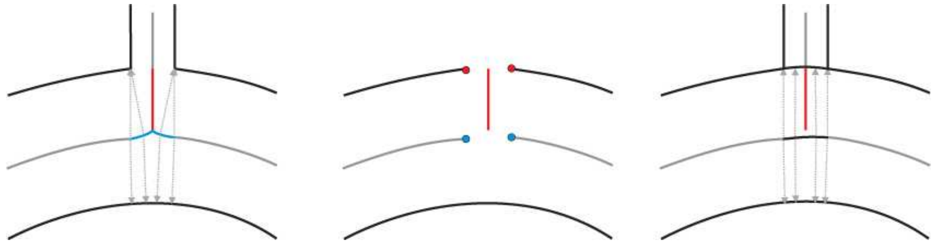


Figure 3.15: The *branch fusion operation* acting on the branch junctions (image taken from [61]).

Lastly, Macrini *et al.* compared the stability of bone graphs with the stability of shock trees of Siddiqi *et al.* [98]. The matching framework proposed for bone graphs follows the one in [98] which uses a node similarity function based on subpartitioning each bone into shock parts. The experimental results showed that substantial improvements in the recognition and the pose estimation performance were obtained since bone graphs does not suffer from the instabilities of shock trees or other skeletal representations. However, as noted by the authors, the proposed matching framework ignores *where* information, *i.e.* the spatial ordering of skeletal shape primitives, and do not use the edge attributes and labels.

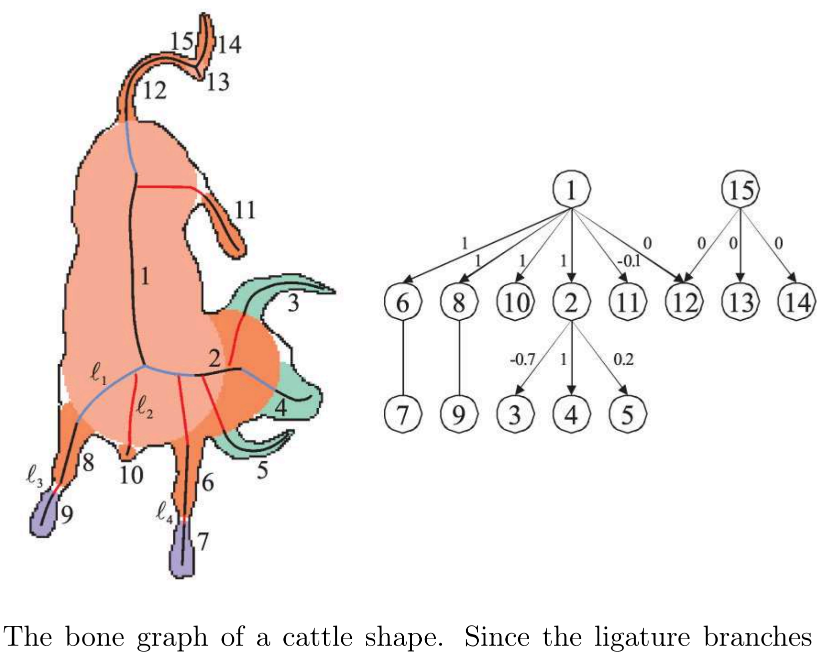
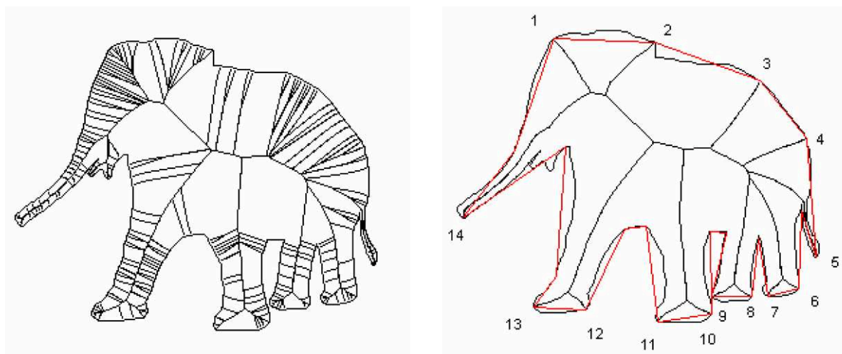


Figure 3.16: The bone graph of a cattle shape. Since the ligature branches l_3 and l_4 are associated with necks, the corresponding edges in the bone graph are undirected (image taken from [61]).

3.5 Path Similarity Skeleton Graphs [5]

In [5], Bai and Latecki presented a novel shape matching framework built upon a stable skeleton-based shape representation. In the first place, this matching method depends on an interesting skeleton pruning strategy proposed by the same authors, which is based on contour partitioning via Discrete Curve Evolution [6]. As the pruning result shown in Figure 3.17 demonstrates, the proposed pruning procedure preserves the topology of skeletons while removing redundant branches and hence, end points of skeleton branches correspond to visual parts of the shapes.



(a) (b)

Figure 3.17: Skeleton pruning by contour partitioning using discrete curve evolution. (a) Extracted skeleton branches of an elephant shape (b) Resulting skeleton after pruning (images taken from [122]).

Motivated by the pruning method in [6], Bai and Latecki employed an alternative approach to represent shape skeletons. That is, the extracted skeletons are not explicitly represented by their topological structures (with the use of graphs or trees), but they are represented with a set of geodesic paths between every pair of end points of skeleton branches instead (Figure 3.18). The resulting descriptions do not involve any junction points, and thus they do not suffer from the instability of skeletons.

In this approach, matching process of two shapes was formulated as finding the correspondences among the end points of corresponding skeleton branches. For that purpose, each skeleton is represented with a graph, in which each of its nodes refers to the end points of branches and holds the skeleton paths to all other end points. To compute the corre-

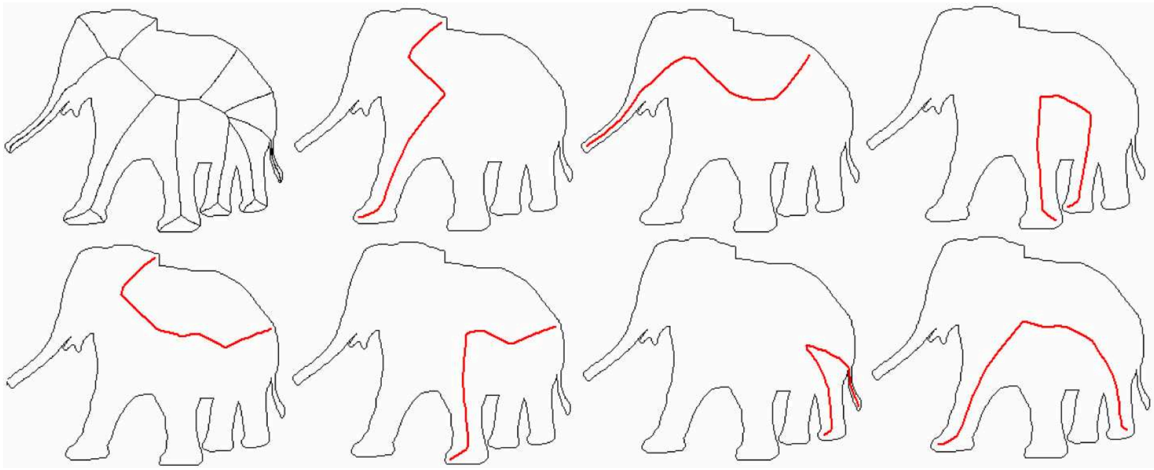


Figure 3.18: The shortest paths between the pairs of endpoints of skeleton branches (image taken from [122]).

spondences and the dissimilarity between two such graphs, the authors apply the Hungarian algorithm [48] on a matrix of dissimilarity costs between the pairs of end points, each of which is estimated based on the paths to all other end points and computed by the optimal subsequence bijection method proposed in [52]. Here, the dissimilarity between two skeleton paths depends on two terms. The first is the dissimilarity between their radius functions, and the second is the difference in their lengths.

To summarize, the method of Bai and Latecki is interesting in the way how it handles the instability of skeletons as this is the most challenging issue about the use of skeletons for shape recognition. Their approach, in contrast to other methods we reviewed in this chapter, does only depend on the similarities among the end points of skeleton branches measured in terms of the path similarities. Since the proposed approach does not require finding the correspondences among junctions points of the skeletons, it is very stable to visual transformations. However, as noted by the authors, the success of the method is limited in the presence of large protrusions.

3.6 Summary and Discussions

Skeleton-based representations are widely used in shape recognition due to their strength in capturing the part structure of shapes and their insensitivity to articulations or bendings. As we mentioned in Section 2.1, a challenging issue though with the use of shape skeletons for recognition is their structural instability in that two visually similar shapes might have topologically different skeletons. In this regard, one can either attempt to resolve this matter in the representation level and come up with a much stable representation or pass this problem to the matching algorithm which is developed in a way that it can deal with possible structural changes, or both.

Keeping the discussion above in mind, we have reviewed some popular and distinguished skeletal representation and matching schemes based on how they represent the skeleton structure and how they compare the corresponding representations in a recognition task. All these studies typically use graphical representations of skeleton structures and compute a partial match between proposed skeletal graphs or trees, returning a similarity or a dissimilarity value. In matching process, the algorithms compensate the instability of skeletons by utilizing a number of edit operations acting on either nodes or edges. Hence, the recognition performance of a method highly depends on how well the proposed edit operations model the transitions that might occur on a skeleton description of a shape. In this regard, the method of Bai and Latecki [5] is exceptional because in this method, matching of skeletons does not depend on the similarity of their topological structures, but rely on the (path) similarities among the end points of the skeleton branches.

As compared to other works reviewed here, bone graphs of Macrini *et al.* [61] also looks promising in the sense that they first seek stability in their skeletal graphs by inspecting ligature sections of the skeletons. Note that there is no need for disconnected skeletons to include such a ligature analysis due to their disconnected property of extracted branches. Moreover, the approach of Zhu and Yuille in FORMS [127] also needs further attention regardless of the limited capability of the shape primitives used in skeleton extraction in that they attempt to solve the instability of skeletons by involving a bi-directional data flow where the information passes through the matching method and the skeleton extraction procedure in both ways.

CHAPTER 4

USE OF DISCONNECTED SKELETON FOR SHAPE RECOGNITION

In the previous chapter, we discussed some skeleton-based shape recognition frameworks, each of which was built upon a different representation scheme. Our main focus was on how these skeletal representations and the related recognition algorithms could cope with the instability of skeletons.

In this chapter, we revisit the disconnected skeleton representation to discuss its use for recognition. The main motivation of Aslan and Tari in devising disconnected skeleton was to come up with the most stable representation of shape in the coarsest possible scale. Compared to other skeletal representations, disconnected skeleton appears to be an unusual approach that as its name reveals, the branches of disconnected skeletons are unconventionally disconnected. This distinctive property give rise to a very stable skeleton structure as the representation does not suffer from the instability of traditional skeletons. Moreover, the number of branches is reasonably small and furthermore no branching occurs due to the excessive smoothing involved in the extraction of skeletons. Hence, the level of hierarchy in the descriptions are always one.

In the following sections, we review two previously proposed shape matching algorithms which utilize disconnected skeleton as the underlying shape representation. In Section 4.1, we discuss the method of Aslan and Tari [3] which is based on a branch and bound approach. In Section 4.2, we investigate the tree edit distance-based shape matching algorithm proposed by Baseski [7].

4.1 The Method of Aslan and Tari [3]

When the disconnected skeleton was first introduced by Aslan and Tari [3], the authors also demonstrated the use of disconnected skeletons for shape matching. Unlike the case in most of the studies reviewed in Chapter 3, Aslan and Tari avoided representing disconnected skeletons by graphs or trees, but interpreted disconnected skeletons as unlabeled attributed point sets instead. They developed a branch and bound algorithm to compute the similarity between two shapes. Based on the data structure shown in Table 4.1, the proposed algorithm exhaustively searches over all possible matchings between branches while computing a total similarity score for each one and finally returns the optimum set of correspondences with the maximum similarity score.

In general, the shapes to be matched may have different number of branches. Hence, Aslan and Tari proposed to compute corresponding total similarity scores by the weighted sum of similarities between matched pair of skeleton branches, where the weights are determined by the normalized lengths of branches. However, as mentioned by the authors, this formulation resulted in an asymmetric measure because the similarity score changes with the choice of the reference shape. Therefore, they chose to symmetrize the measure by simply taking the minimum of the two possible similarity scores in reporting the matching results. The formal definition of the algorithm is as follows:

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ denote the two shapes to be matched and $\omega \in \Omega$ denote a set of correspondences between the skeleton branches of $\mathcal{S}_1$ and $\mathcal{S}_2$ defined in the search space Ω containing all possible matchings. Then, the total similarity between $\mathcal{S}_1$ and $\mathcal{S}_2$ is given by:

$$sim(\mathcal{S}_1, \mathcal{S}_2) = \max_{\omega \in \Omega} \left[\min \left(\sum_{(b_1,b_2)\in\omega} l_1 \times sim(b_1, b_2), \sum_{(b_1,b_2)\in\omega} l_2 \times sim(b_1, b_2) \right) \right] \tag{4.1}$$

where the similarity between attributes of two matched branches $b_1 \in \ S_1$ and $b_2 \in \mathcal{S}_2$ is determined by a multivariate Gaussian distribution:

$$sim(b_1, b_2) = \begin{cases} exp\left(-0.5\left(\frac{(l_1-l_2)^2}{\sigma_l} + \frac{(r_1-r_2)^2}{\sigma_r} + \frac{(\theta_1-\theta_2)^2}{\sigma_\theta}\right)\right) & \text{if } type_1 = type_2 \\ 0 & \text{otherwise} \end{cases}$$

where σ_l, σ_r, σ_θ respectively specify the importance of each attribute that are determined experimentally.

Table 4.1: The data structure defined to express disconnected skeleton of a shape (table taken from [2]).

Description element	Information stored
Shape	Center point (x_0, y_0)
	Total length of the axes
	Orientation of the reference axes (m_0, m_1)
Local Symmetry Branch	Type (Positive, Negative)
	Location (r, θ)
	Normalized Length
	Reference Axis (Yes, No)
	Next Symmetry Axis
	Previous Symmetry Axis

Recall that in disconnected skeleton, each branch is of a positive or negative type depending on either it corresponds to a protrusion or an indentation. Hence, setting the similarity between different types of branches to zero drastically reduces the total similarity score and practically eliminate this sort of semantically invalid correspondences.

In traversing the search space, Aslan and Tari employed a branch and bound approach to find the optimum matching in an effective way. They introduced an additional pruning strategy to discard regions of the search space that contain visually unmeaningful set of correspondences. In expanding the search tree, the order of skeleton branches of a shape should be preserved in the matching. Hence, if a correspondence violates the ordering constraint, all the related matchings are totally ignored. It is important to note that this not only reduces the computation time but improves the visual quality of matching results. Moreover, to further speed up the algorithm, Aslan and Tari divide the problem into two subproblems by defining two different coordinate frames, each of which is used to express a different half of a shape.

Figure 4.1 shows some illustrative matching results obtained by the method of Aslan and Tari. In each case, visually correct correspondences are obtained under Euclidean transformations (translation, rotation and scaling), articulation of parts and even missing features. The authors tested the performance of their matching method on the shape database shown in Figure 4.2, which contains a total of 180 shapes with 30 categories, each having 6 examples. In the experiments, each shape was used as a query shape and the most similar shapes are retrieved accordingly. Average precision-recall curve is given in Figure 4.3. The average precision was found to be around 88% at recall level 100%.

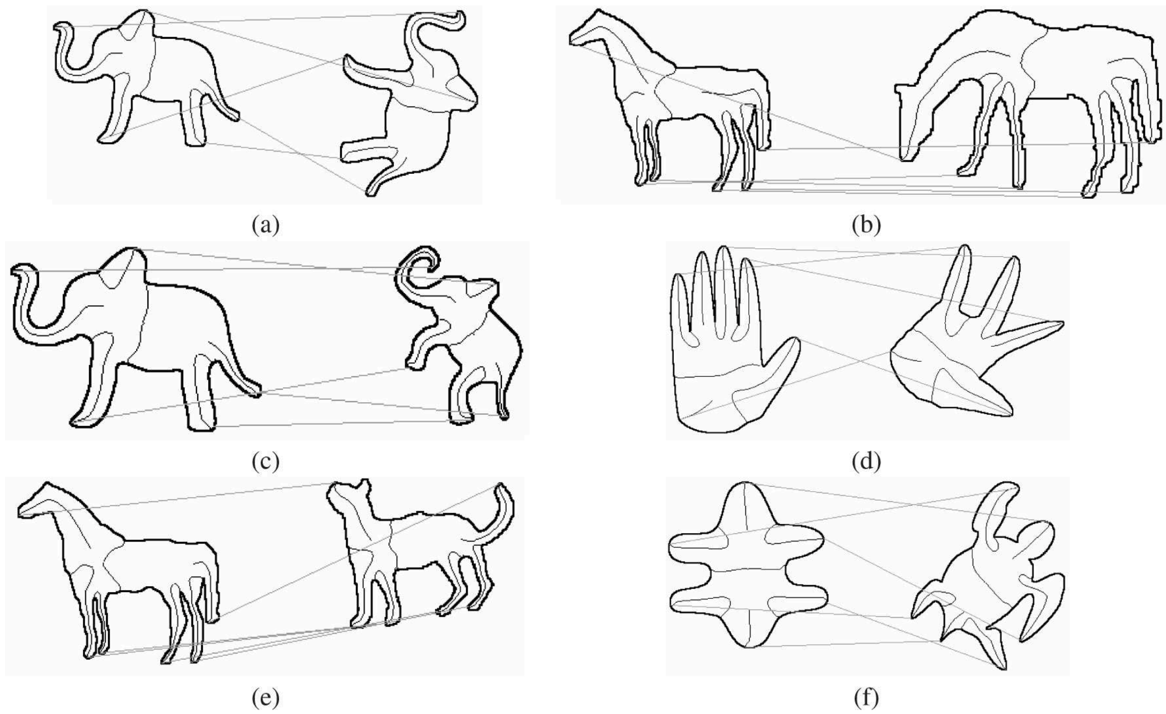


Figure 4.1: Some skeletal matching results obtained by the method of Aslan and Tari. The total similarity scores are 0.992, 0.708, 0.886, 0.652, 0.714, and 0.832, respectively (images taken from [2]).

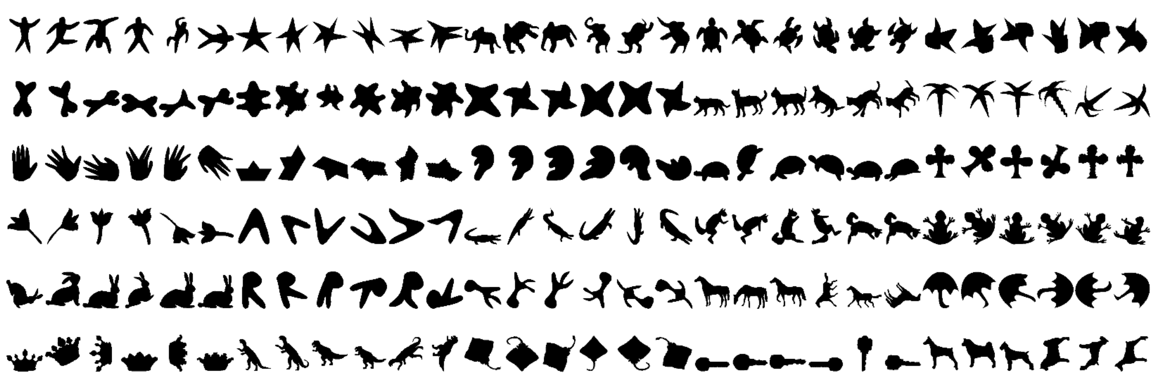


Figure 4.2: The shape database used in the experiments performed by Aslan and Tari (image taken from [2]).

4.2 The Method of Baseski [7]

In [7], Baseski employed widely used tree edit distance approach [92] and came up with an alternative shape matching method to compare shapes using their disconnected skeletons. Unlike the method of Aslan and Tari [3], their formulation depends on representing

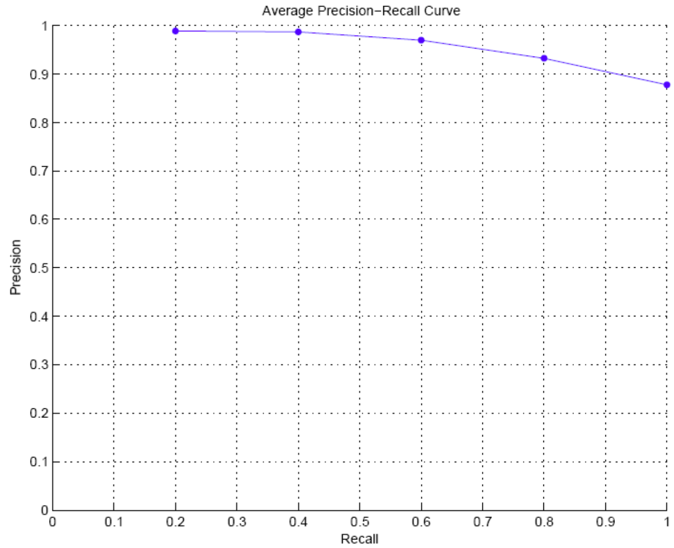


Figure 4.3: Average precision-recall graph (image taken from [2]).

disconnected skeletons by trees and for that purpose, they introduced a skeletal tree representation, which was referred to as *shape tree*. Compared to other skeleton-based graphical representations such as shock trees, the key characteristic of shape trees is that the depth of each constructed shape tree is always one since disconnected skeletons capture the most prominent part structure of shapes with only one level of hierarchy.

Representing disconnected skeletons by shape trees is quite straightforward. However, first recall that as we indicated in Section 2.2, a shape might have alternative descriptions based on the construction of the coordinate frame in its disconnected skeleton. Therefore, in [7], Baseski and Tari decided to form multiple shape trees for each alternative description of the shape. Figure 4.4 illustrates shape trees of some shapes.

Each shape tree is a rooted attributed depth-1 tree where the root node can be interpreted as the shape center but it actually holds necessary and sufficient information to construct the coordinate system. This information includes the location of the center, the directions of reference axes and a normalization factor for branch length (based on total branch length). Accordingly, each leaf node of the shape tree corresponds to one of the extracted skeleton branches and holds the following attributes:

- the disconnection location in polar coordinates (r, θ),
- the normalized length of the branch l,
- the branch type as *negative* or *positive*.

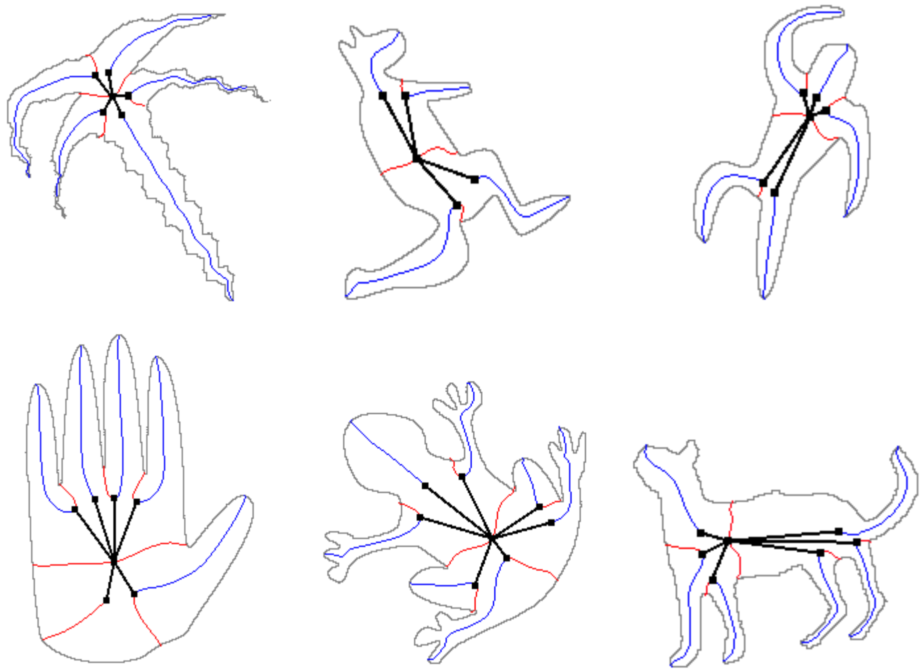


Figure 4.4: Some shape trees. Note that each disconnection point (except the pruned major branches) gives rise to two different nodes in the tree, representing the positive and negative skeleton branches meeting at that disconnection point. However, for illustration purposes, only one node is drawn.

In addition, Baseski and Tari preferred labeling each node with respect to an ordering of branches in order to devise a more efficient edit distance-based tree matching algorithm. Their choice for ordering is to start with any one of the major negative branches and hence they store alternative descriptions of the shape tree for each such possible choice (Figure 4.5). As noted before, for the shapes having n-fold symmetry, there are n major negative branches.

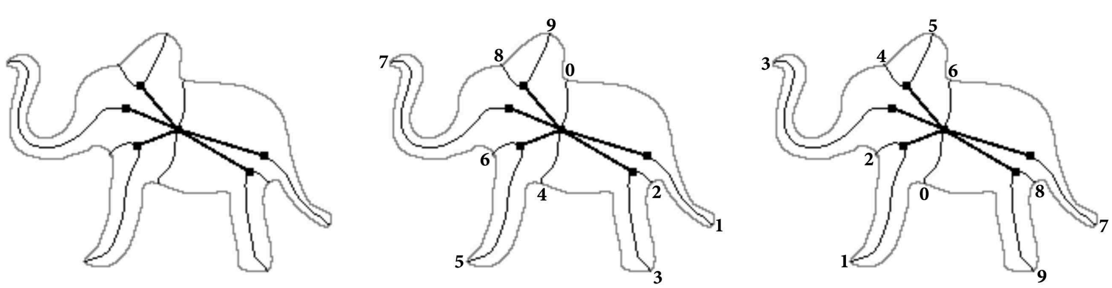


Figure 4.5: Multiple descriptions obtained with different orderings of branches.

The formal definition of the method of Baseski and Tari for skeletal tree matching is as follows: Let $\mathcal{T}_1$ and $\mathcal{T}_2$ denote the two shape trees to be matched. Since $\mathcal{T}_1$ and $\mathcal{T}_2$ are all ordered-depth-1 trees, each of them can be expressed as a list of nodes (excluding the root node):

$$\mathcal{T}_1 = \left\{ u_i = \left(u_i^r, u_i^\theta, u_i^l, u_i^{type} \right) \mid u_i \in \mathcal{N}_1 \right\}$$
$$\mathcal{T}_2 = \left\{ v_j = \left(v_j^r, v_j^\theta, v_j^l, v_j^{type} \right) \mid v_j \in \mathcal{N}_2 \right\}$$

where i, j denote the order of nodes, (r, θ) is the normalized location of the disconnection point in polar coordinates, *type* denotes the type of the branch (either positive or negative) and l is the normalized length of the corresponding skeleton branch. $\mathcal{N}_1$ and $\mathcal{N}_2$ are the set of leaf nodes of $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively.

To transform a shape tree into another, or vice versa, Baseski and Tari defined three edit operations, namely `remove`, `insert` and `change`. Let Λ denote the set of nodes removed from $\mathcal{T}_1$, Δ denote the set of nodes inserted to $\mathcal{T}_1$ from $\mathcal{T}_2$ and Ω denote the set of matched nodes. Then the the distance between $\mathcal{T}_1$ and $\mathcal{T}_2$ is given by Equation 4.2, as the cost of the sequence of edit operations $\mathcal{S}$ with minimum cost.

$$d\left(\mathcal{T}_1, \mathcal{T}_2\right) = \min_{\mathcal{S}} \left[\sum_{u \in \Lambda} \texttt{remove}\left(u\right) + \sum_{v \in \Delta} \texttt{insert}\left(v\right) + \sum_{(u,v) \in \Omega} \texttt{change}\left(u, v\right) \right] \tag{4.2}$$

The cost functions of edit operation are defined as follows. Note that each edit cost function returns a value in the range $[0, 1]$:

- `remove`. The corresponding cost function quantitatively measures how well the removed skeleton branch characterizes the shape. In this regard, the cost of removing a given node u of $\mathcal{T}_1$ is defined based on two significance measures. The first significance measure is the branch length, as argued in [1, 3, 2]. The second significance measure is the disconnection location of a branch. While the major branches do not terminate, and reach to the shape center, boundary details terminate quite early.

$$\texttt{remove}\left(u\right) = \left(\frac{u^l}{l_{max}\left(\mathcal{T}_1\right)} \right) \left(1 - u^r\right) \tag{4.3}$$

where u^l is the length of the branch, u^r is the distance from shape center and $l_{max}\left(\mathcal{T}_1\right)$ is the length of the longest branch of $\mathcal{T}_1$. See Figure 4.6.

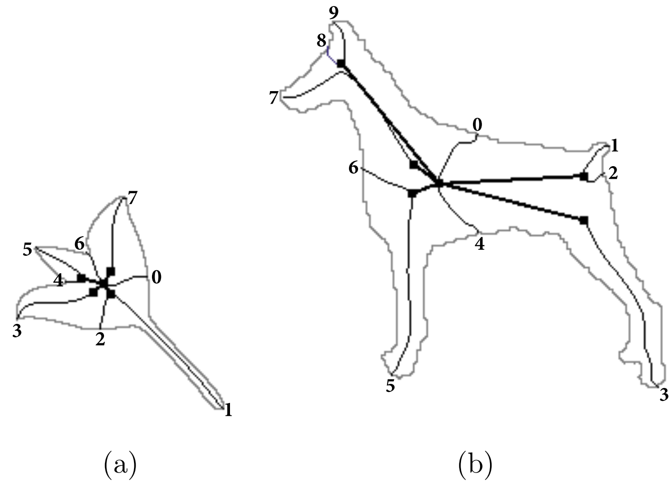


Figure 4.6: `remove` cost function. (a) Since $u_1^l \geq u_5^l$, `remove`$(u_1) \geq$`remove`(u_5). (b) Since $u_6^r \geq u_2^r$, `remove`$(u_6) \geq$`remove`(u_2).

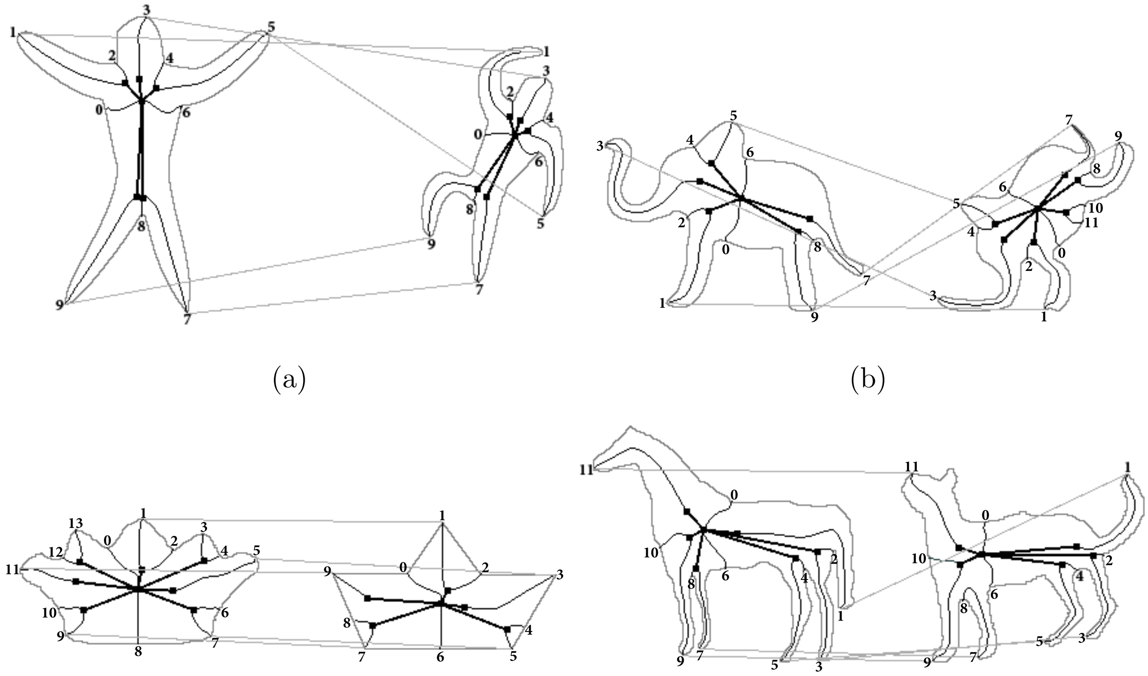
- `insert`. This operation is the dual operator of `remove`. It inserts a node from $\mathcal{T}_2$ to $\mathcal{T}_1$ (or equivalently removes the corresponding node from $\mathcal{T}_2$). Hence, the cost function given below is same with `remove` except that the length is normalized with respect to $l_{max}(\mathcal{T}_2)$:

$$\texttt{insert}\,(v) = \left(\frac{v^l}{l_{max}\,(\mathcal{T}_2)}\right)(1 - v^r) \tag{4.4}$$

- `change`. This operation computes the dissimilarity of two nodes u and v based on the differences between their attributes. The corresponding cost function resembles the one used in [74]. However, an additional constraint enforces the types of the matched branches to be identical. If they differ, the cost is set to 1.

$$\texttt{change}(u,v) = \begin{cases} 1 & \text{if } u^{type} \neq v^{type} \\ \beta_1 \frac{|u^l - v^l|}{max(u^l, v^l)} + \beta_2 \frac{|u^r - v^r|}{max(u^r, v^r)} + \beta_3 \frac{|u^\theta - v^\theta|}{max(u^\theta, v^\theta)} & \text{otherwise} \end{cases} \tag{4.5}$$

Figure 4.7 shows matching results of some illustrative shapes. In the matching process, Baseski and Tari gave more weight to the similarity of lengths by setting $\beta_1 = 0.5$ and $\beta_2 = \beta_3 = 0.25$. As these examples demonstrate, the method of Baseski and Tari is also able to obtain the correct matchings under various visual transformations. To evaluate the retrieval performance of their method, Baseski and Tari repeated the same experiments as Aslan and Tari performed in [1]. The corresponding average precision-recall curve is given in Figure 4.8. The average precision is 87% at recall level 100%. Note that this value is very close to the one in [1].



(a) (b)

(c) (d)

Figure 4.7: Some skeletal matching results obtained by the method of Baseski and Tari. Matching costs are 0.683, 1.459, 2.725 and 2.372, respectively.

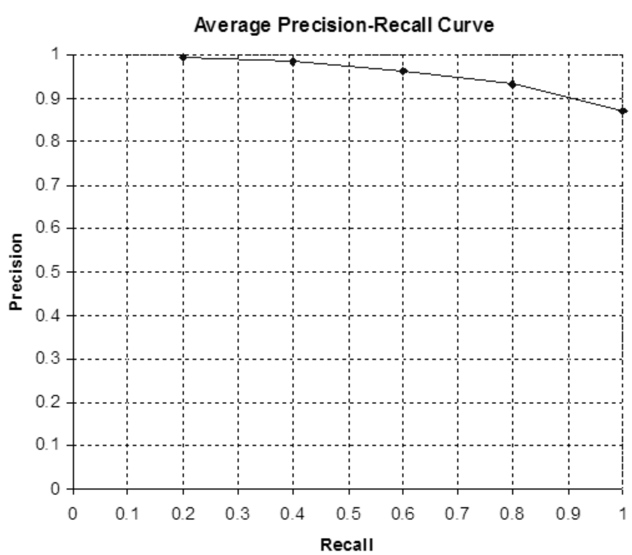


Figure 4.8: Average precision-recall graph (image taken from [8]).

In [7], Baseski identified four typical reasons why his method might return a mismatch or a dissimilarity value which is beyond our visual judgments. Two of these are related to how the edit cost functions are defined whereas the other two are in fact related to one of the shortcoming of disconnected skeleton representation that we mentioned in Section 2.2.1, *i.e.* information about boundary details are absent in the skeleton descriptions. To resolve this issue we will propose a coarse-to-fine strategy in Section 5.1.5, which is based on the category-influenced matching method presented in Section 5.1.

The time complexity of the shape matching method proposed by Baseski and Tari can be analyzed as follows. When each edit operation has unit cost, the time complexity of matching two ordered-depth-1 trees is $\mathcal{O}(mn)$, where m and n respectively denote the number of leaf nodes in the trees [92]. However, a critical issue in tree edit distance-based skeletal shape matching is how the cost of each edit operation is computed because these costs might dominate over the cost of tree matching as in [90, 98]. In this respect, the method of Baseski and Tari has two main advantages. First, the edit cost computations are nearly negligible and second, the number of leaf nodes is significantly small as disconnected skeleton is a very coarse skeletal representation.

4.3 Summary and Discussions

In this chapter, we discuss two previously proposed approaches to shape matching using disconnected skeleton representation. In the matching method of Aslan and Tari [3], disconnected skeletons are represented by their disconnection points as unlabeled attributed point sets, and a branch-and-bound strategy is used in order to match the disconnected skeleton structures of two shapes. In the matching method of Baseski [7], however, a structural approach is employed and skeletons are represented as (shape) trees, which reduces the problem into matching two shape trees, and accordingly, the authors proposed a tree edit distance-based algorithm to find a partial match between two given shape trees. The experiments performed on the same shape database reveal that the retrieval rates of both methods are nearly the same, yet the method of Baseski is superior to that of Aslan and Tari in terms of computational complexity.

CHAPTER 5

INCORPORATING SEMANTIC CATEGORY KNOWLEDGE INTO SHAPE MATCHING

In the previous chapter, we reviewed two different shape matching methods that were built upon disconnected skeleton representation. While neither of these methods outperforms the other one, they both demonstrated that despite its coarseness, disconnected skeleton is quite stable compared to other skeletal representations, thus making it an effective representation for visual shape recognition.

In this chapter, we investigate the effect of context in shape (dis)similarity computation. Borrowing the definition from Toussaint [112], the effect of context in a recognition task can be stated that "*some entity Z can have certain properties, when Z is viewed in isolation, which change when Z is viewed in some context. Alternately, an entity Z is seen as one thing in context A and another in context B*". In our study, context refers to the set of shapes belonging to the same category and accordingly, we extend and refine the shape matching methods described in Chapter 4 in a number of ways by incorporating semantic category knowledge into the matching process. Each modification offers a higher retrieval accuracy than the original algorithms. Moreover, each one results in a *non-metric* shape similarity (or dissimilarity) measure that is more consistent with our visual judgments in terms of its formulation.

The conventional approach in the shape matching literature is to define shape (dis)similarity by means of metrics. On the other hand, starting from the influential work of Tversky [113], there has been a long history of empirical research in psychology which suggests otherwise that human similarity judgments are in fact not metric, meaning that our judgments may vi-

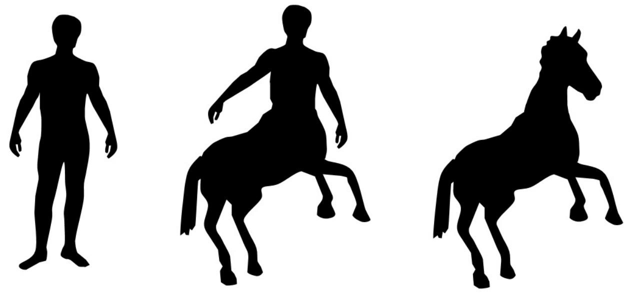


Figure 5.1: An example from Basri *et al.* [9] used to illustrate the violation of triangle inequality axiom in visual dissimilarity relationships, see text for the explanation (shapes taken from the `mythological creatures` data set used in [15]).

olate metric axioms, *i.e. minimality*, *symmetry* and *triangle inequality.* For some discussions on this issue from computational point of view, see Basri *et al.* [9] and Mumford [64].

Figure 5.1 is an illustrative example from [9] that demonstrates a case where our visual perception system does not satisfy the triangle inequality. Note that the centaur shape shares some similar parts with both of the other two shapes. Psychological judgment data indicates that many human observers report that the dissimilarity between the human shape and the horse shape is far more than the sum of the dissimilarities between the centaur shape and either of the human and horse shapes.

Above all, the reason why metric similarity measures have been prevalent in the shape matching literature is because many computational tools of pattern recognition cannot successfully deal with non-metric data. However, this situation starts to change with the introduction of new generation of tools and the paradigm shifts happening in the pattern recognition community, *e.g.* [24, 25, 26, 71]. As a consequence, a less but growing number of studies, which utilize non-metric similarity measures for shape matching, began to appear in the literature, *e.g.* [15, 39, 51]. These studies mainly concentrate on the violation of triangle inequality and solely depend on identifying part correspondences between shapes based on contour fragments or regions. On the other hand, our approach is conceptually different than the cited works in the way that in the studies presented in this chapter, we investigate and utilize the effects of context on measuring visual similarity, context being specified as the existing category structure.

The influence of context on similarity judgments is a well-studied topic in psychology [27, 35, 64, 85, 113]. For instance, in an experiment conducted by Tversky [113], human subjects were asked to chose among the countries Sweden, Norway and Hungary, the most similar one to Austria. 60% of the subjects chose Hungary. However, when he repeated the experiment with a new answer set, which included Poland instead of Norway, 49% of those chose Sweden. Now, note that this experiment was carried out in the Cold War era. Hence, it is most likely that the subjects tended to consider Hungary and Poland dissimilar to other countries since these were two Eastern Bloc countries.

Context information can also be used to account for asymmetric similarity relationships [64, 113]. This follows from the interpretation that two objects to be compared, say A and B, have in fact two separate roles such that while A is considered as a newly encountered stimulus input, B is thought as a memory benchmark belonging to a category. Accordingly, this view suggests that in measuring the similarity between A and B, human mind analyzes these objects differently, for example, that it might be the case that it immediately searches for the salient features of B whether they are found in A or not.

In Section 5.1, we follow the above interpretation and present a novel extension to the shape matching method of Baseski [7] by incorporating semantic category knowledge into matching process[1]. In this modified version of the method, which we refer to as *category-influenced matching*, each database shape is associated with a category and the cost function of each edit operation is redefined in a way to reflect the information coming from the category of the database shape. Hence, the category knowledge directly influences the dissimilarity between the query shape and the database shape. In Section 5.1.5r, we present a coarse-to-fine strategy to incorporate categorical boundary similarity into category-influenced matching method by utilizing the representation of *approximate radius functions* described in Section 2.3.1.

In Section 5.2, we use category knowledge to achieve *contextual sensitivity to articulations* in shape matching[2]. Our approach depends on the disconnected skeleton representation in that we first formulate a representation space for articulations of parts using the structure of extracted skeletons. This *articulation space* enables us to make inferences about likely articulations based on the prior knowledge obtained from existing examples of a shape category. Following to that, we incorporate the proposed approach to the method of Aslan and Tari [3] and come up with a shape matching framework that is sensitive to unlikely articulations but insensitive to likely ones.

[1]This is a joint work with Emre Baseski and an early version of this study was partly published in MSc. thesis of Emre Baseski [7]. Full version is published in *Pattern Recognition* [8].

[2]This is a joint work with Erkut Erdem and was previously presented in the Workshop on the Representation and Use of Prior Knowledge in Computer Vision [28].

Finally, we conclude the chapter with a summary and some discussions.

5.1 Category-Influenced Shape Matching

In the following, motivated by the importance of context in human similarity judgements, we modify and extend the skeletal tree matching algorithm of Baseski and Tari [7] by incorporating contextual effects into the matching process. In the literature, the notion of context has a vast number of meanings, typically refering to either a collection of neighboring entities (*e.g.* nearby objects [35, 64], local pixel neighborhood [29, 101]), or prior knowledge and expectations [58]. See Wolf *et al.* [120] for a broad discussion on the topic. In our study, context is defined as being a collection of shapes belonging to the same category.

An interesting argument in favor of context dependence in pattern recognition comes from the *Ugly Duckling Theorem* [119] which states that categorization or recognition is impossible without an underlying bias, hence in the absence of bias any two patterns are equally similar to each other. This is quite important in the sense that it also implies that there are in fact no privileged primitives. In this sense, one approach could be to start with many primitives, each of which provides a rough representation, and then to select the best ones in a given context [76, 118]. Our approach is in the opposite direction that we start with a very coarse yet very stable skeletal description and a context provides extra information about the extracted primitives.

In defining the dissimilarity between two shapes in a context, we extend the tree edit distance measure of Baseski [7] by following the interpretation mentioned in [64, 113] in which different roles are assigned to the shapes in comparison. That is to say a query shape A (input stimulus) is compared to a database shape B (memory benchmark) whose category is known. The category knowledge of B, *i.e.* all the category members (including B), forms a context that influences the dissimilarity computation by modifying the saliency of primitives and the distances between attributes, as in the philosophy of some recent works such as [65, 67].

5.1.1 Representing Category Knowledge with Category Trees

To form and utilize the relevant category knowledge of a given set of shape trees belonging to the same category, we propose a special tree structure, which we name *category tree*. Built like a union of the shape trees, a category tree is a depth-1 tree whose leaf nodes represent a specific primitive observed in the category. In particular, each leaf node of a category tree is linked to a corresponding leaf node of one or more shape trees and in addition, stores some basic information about attribute statistics. Once formed, each category tree provides a context for each primitive of a database shape. In this regard, we come up with two different procedures for constructing category trees, referred to as *static formation* and *dynamic formation*, respectively.

Note that in the context of shape matching, forming a union of tree representations has been previously addressed by Torsello and Hancock [109]. However, unlike Torsello and Hancock's construction, both of our constructions naturally preserve the tree structure in the union, and moreover, the resulting category trees are depth-1 trees as well, just like the shape trees. In fact, this is a direct implication of the depth-1 property of shape trees. Representing both individual shapes and categories using the same data structure is noteworthy that this makes the necessary constructions and computations trivial.

In the static formation, shapes to be united should be given in advance. In the beginning, the shape tree with the maximum number of nodes is designated as a base tree and then all the remaining trees are matched to the base tree (using the method of Baseski) and the category tree is formed solely based on the found correspondences. This procedure has two major drawbacks though. First, the structure of the category tree is fixed and hence addition of a new shape may require a re-formation from scratch. Second, the procedure does not guarantee the inclusion of all the available information. This drawback is visible in the illustration given in Figure 5.2.

In the dynamic formation, the category tree is formed by using an incremental procedure. In this regard, it resembles formation of *Tree-Unions* of Torsello and Hancock [109]. First, the pairwise distances between the given set of shapes are computed (again using the method of Baseski). Then, according to the descending order of total distances, the category tree is progressively expanded by using the correspondences between the category tree and the shape trees (obtained with the modified matching method in the way described in Section 5.1.3). The dynamic formation procedure is superior to the static one because it does not suffer from any of the drawbacks mentioned for the static formation. Moreover, since it operates in a dynamic way, it is computationally more effective in updating categories as new shapes are observed and categorized.

5.1.2 The Revised Formulation of Tree Edit Distance Algorithm

Let $\mathcal{T}_1$ denote the shape tree of the query shape which is being compared to a database shape whose shape tree is denoted by $\mathcal{T}_2$. Each leaf node of $\mathcal{T}_2$ is linked with a specific leaf node

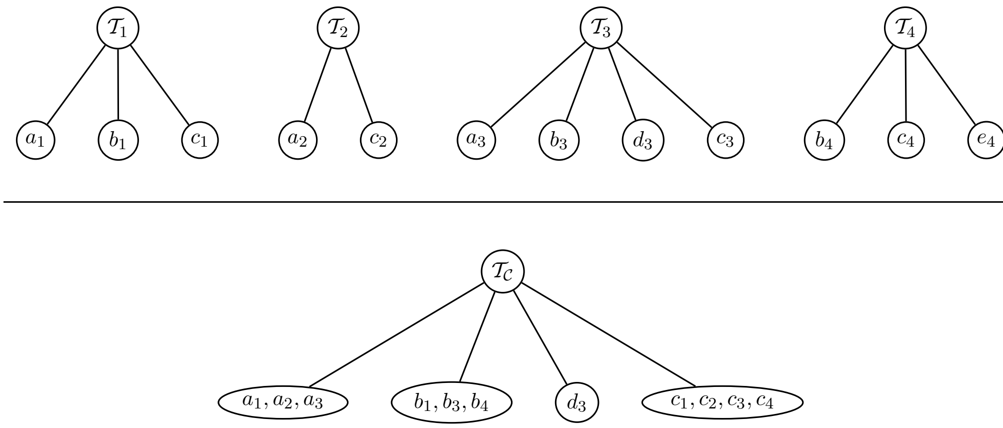


Figure 5.2: Static formation of a category tree. $\mathcal{T}_3$ is the base tree and the correspondences among nodes are specified by labeling the matched nodes with identical letters. Note that the procedure is not perfect since the node e_4 in $\mathcal{T}_4$ is eliminated in forming the category tree $\mathcal{T}_\mathcal{C}$ since it does not match to any node of the base tree.

of a category tree $\mathcal{T}_\mathcal{C}$. Say it is denoted by $\mathcal{B}_k$, this leaf node not only provides a context for the corresponding leaf node in $\mathcal{T}_2$ but for all the related m number of category members (including $\mathcal{T}_2$ and $m \leq M$, where M is the total number of shapes in that category). In the node $\mathcal{B}_k$, in addition to the associations with other category member shape trees, each leaf node of $\mathcal{T}_\mathcal{C}$ by providing some basic information about attribute statistics:

- the observed ranges for r, θ and l of the branch $(r_{min}, r_{max}, \theta_{min}, \theta_{max}, l_{min}, l_{max})$;
- the categorical saliency of the branch, defined by its frequency $freq(\mathcal{B}_k) = m/M$.

Following these denotations, the shape trees $\mathcal{T}_1$ and $\mathcal{T}_2$ and the category tree $\mathcal{T}_\mathcal{C}$ can all be expressed as a list of nodes (excluding their root nodes) as follows:

$$
\begin{aligned}
\mathcal{T}_1 &= \left\{ u_i = \left(u_i^r, u_i^\theta, u_i^l, u_i^{type} \right) \;\middle|\; u_i \in \mathcal{N}_1 \right\} \\
\mathcal{T}_2 &= \left\{ v_j = \left(v_j^r, v_j^\theta, v_j^l, v_j^{type} \right) \;\middle|\; v_j \in \mathcal{N}_2 \right\} \\
\mathcal{T}_\mathcal{C} &= \left\{ \mathcal{B}_k = \left(\mathcal{B}_k^{r_{min}}, \mathcal{B}_k^{r_{max}}, \mathcal{B}_k^{\theta_{min}}, \mathcal{B}_k^{\theta_{max}}, \mathcal{B}_k^{l_{min}}, \mathcal{B}_k^{l_{max}}, \mathcal{B}_k^{type}, freq(\mathcal{B}_k) \right) \;\middle|\; \mathcal{B}_k \in \mathcal{N}_\mathcal{C} \right\}
\end{aligned}
$$

where i, j, k denote the order of nodes, (r, θ) is the normalized location of the disconnection point in polar coordinates, *type* denotes the type of the branch (either positive or negative) and l is the normalized length of the corresponding skeleton branch. $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_\mathcal{C}$ are the

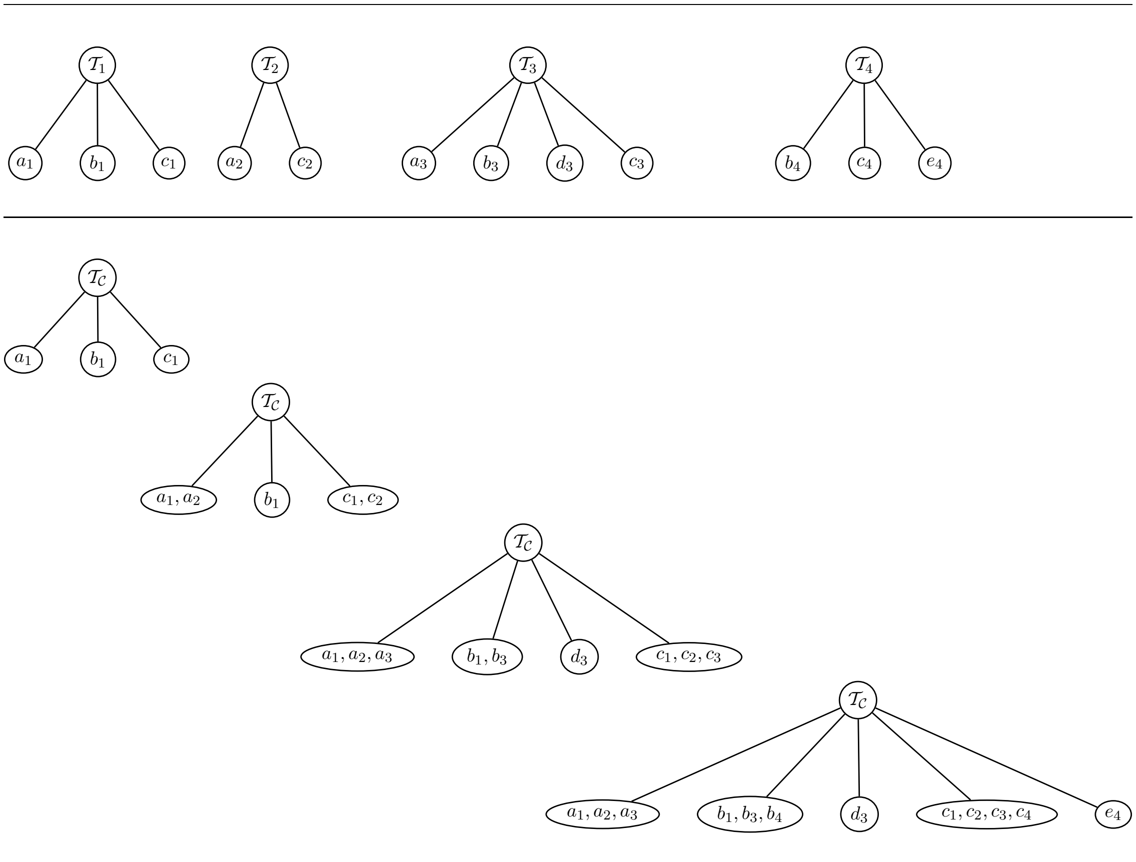


Figure 5.3: Dynamic formation of a category tree. The category tree $\mathcal{T}_\mathcal{C}$ is enlarged incrementally with the shape trees $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$ and $\mathcal{T}_4$. Matched nodes are labeled with identical letters. Note that the procedure does not suffer from any of the drawbacks of the static formation procedure.

set of leaf nodes of $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{N}_\mathcal{C}$, respectively.

To calculate the distances between attributes in the presence of category statistics, Baseski proposed the generic function $f\,(x|y,[min,max])$ (Figure 5.4) [7]. In the experiments, ϕ_1 and ϕ_2 is taken as $\phi_1 = \frac{\pi}{4}$ and $\phi_2 = \frac{4\pi}{9}$. x is defined on the horizontal axis and the function is fixed for $[min, max]$ and a given y. Notice that value of function f depends not only to the difference $x-y$, but also to the range $[min, max]$. When x falls in the range, $x - y$ difference is taken as it is. On the other hand, when x fall out of the range, $x - y$ difference is boosted. That is, numerically equal differences are perceived smaller within categories and larger between categories. The idea is not so different than a Mahalanobis distance or the distance used in [86]. It gives a distance weighted in a context.

The modified cost functions are given by:

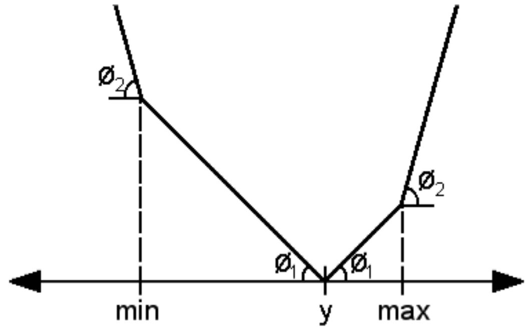


Figure 5.4: The generic cost function $f\left(x|y,[min,max]\right)$ (image taken from [7]).

$$\texttt{change}(u,v,\mathcal{B}) = \begin{cases} 1 & \text{if } u^{type} \neq v^{type} \\ \frac{f(u^r|v^r,\mathcal{B})+f(u^\theta|v^\theta,\mathcal{B})+3f(u^l|v^l,\mathcal{B})}{5} \times freq(\mathcal{B}) & \text{otherwise} \end{cases}$$

$$\texttt{remove}\left(u\right) = \left(\frac{u^l}{l_{max}(\mathcal{T}_1)}\right)\left(1-u^r\right)$$

$$\texttt{insert}\left(v\right) = \left(\frac{v^l}{l_{max}(\mathcal{T}_2)}\right)\left(1-v^r\right) \times freq\left(\mathcal{B}\right)$$

Note that there is no change in the cost function of `remove` operation however, the `insertion` cost of a node is multiplied by a factor of categorical significance of the skeleton branch since the category of the shape denoted by $\mathcal{T}_2$ is known.

5.1.3 Matching a Shape Tree with a Category Tree

We can exploit the structural equivalences of shape trees and categories and formulate a simple way of matching a shape tree with a category tree: Let $\mathcal{T}_1$ be the input shape tree to be matched with the category tree $\mathcal{T}_\mathcal{C}$. We construct a *mean shape tree* $\overline{\mathcal{T}_\mathcal{C}}$ having equal number of nodes with $\mathcal{T}_\mathcal{C}$ whose leaf nodes hold ordinary average values of the attributes collected from the shape trees of the category members. Since a mean shape tree is indistinguishable from a shape tree, we can apply the matching algorithm given in the previous section in order to determine the correspondences between a shape tree and a category tree.

We make two remarks to make. First, the mean tree is uniquely defined as an ordinary average. Hence, it differs from a mean or median structure which has equal edit distances to all the contributing shapes as in [18, 40]. Second, a comparison of a shape tree with a mean tree is guided by the category tree from which the mean tree is calculated. Even though the mean tree does not capture within group variability, the category tree does.

As noted in Section 5.1.1, we utilize this way of matching a shape tree with a category tree in the dynamic procedure proposed for the construction of category trees in which a category tree is expanded progressively, via computing the correspondences with the given shape trees. In Chapter 6, we will use this method for the purpose of categorization that the method will provide a starting point for us in devising a novel shape classification framework.

5.1.4 Experimental Results

To observe the effect of context in shape matching, we first repeated the experiments given in Section 4.1 and Section 4.2 with our category-influenced matching method. Figure 5.5 shows the average precision-recall curve for this experiment. Incorporating category knowledge into matching process aids resolving the erroneous situations that is faced with the methods that don't use contextual knowledge, thus we attain precision values are above 99.4% for all the recall levels. Moreover, the new dissimilarity measure gives a better within group versus between group separation, and it mimics the asymmetric nature of human similarity judgements. Compare Table 5.1 with Table 5.2.

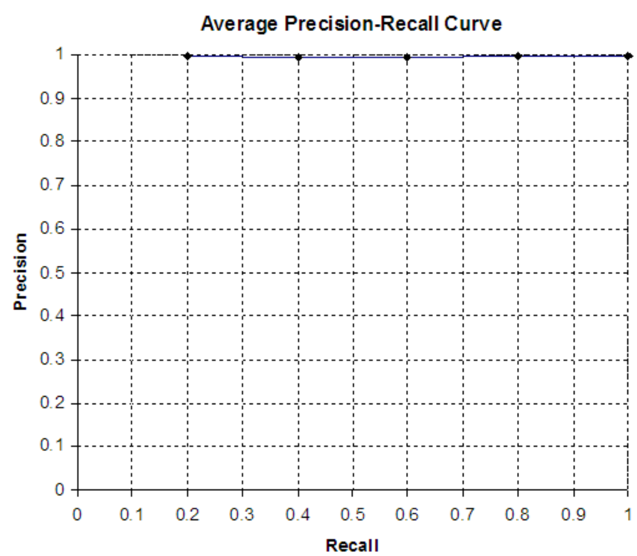


Figure 5.5: Average precision-recall graph (cf. Figure 4.3 and Figure 4.8).

Table 5.1: The top 8 retrievals for 6 elephants. The dissimilarity between an elephant and a squirrel is very close to the dissimilarities among elephants. (image taken from [7]).

	0.906	**0.994**	**1.040**	**1.157**	**1.459**	1.492	1.536	1.632
	0.846	**0.994**	**1.033**	**1.420**	1.580	1.676	1.691	1.695
	0.901	**0.906**	**1.033**	**1.122**	1.657	**1.663**	1.842	1.843
	1.122	**1.157**	**1.420**	**1.536**	**1.562**	1.772	1.782	1.798
	1.459	**1.562**	**1.663**	**1.708**	**1.744**	1.958	1.963	1.967
	0.846	**0.901**	**1.040**	**1.536**	1.646	**1.708**	1.733	1.743

Table 5.2: The top 8 retrievals for 6 elephants. Compare the results to the ones in Table 5.1 (image taken from [7]).

	0.356	**0.414**	**0.418**	**0.431**	**0.693**	2.101	2.119	2.140
	0.375	**0.446**	**0.450**	**0.514**	**0.815**	2.069	2.091	2.098
	0.378	**0.452**	**0.457**	**0.621**	**0.968**	2.426	2.475	2.477
	0.675	**0.756**	**0.828**	**0.848**	**0.857**	2.224	2.240	2.327
	0.898	**1.004**	**1.106**	**1.170**	**1.254**	2.183	2.188	2.260
	0.401	**0.467**	**0.480**	**0.615**	**0.908**	2.364	2.391	2.464

For a detailed analysis of the performance, we form a much larger shape database by extending the database of Aslan [1] with new shapes and additional shape categories, which are collected from various sources, including [50, 88]. As shown in Figure 5.6, the database contains 50 categories, each having 20 examples among which there are differences in orientation, scale, articulation of parts and small boundary details.

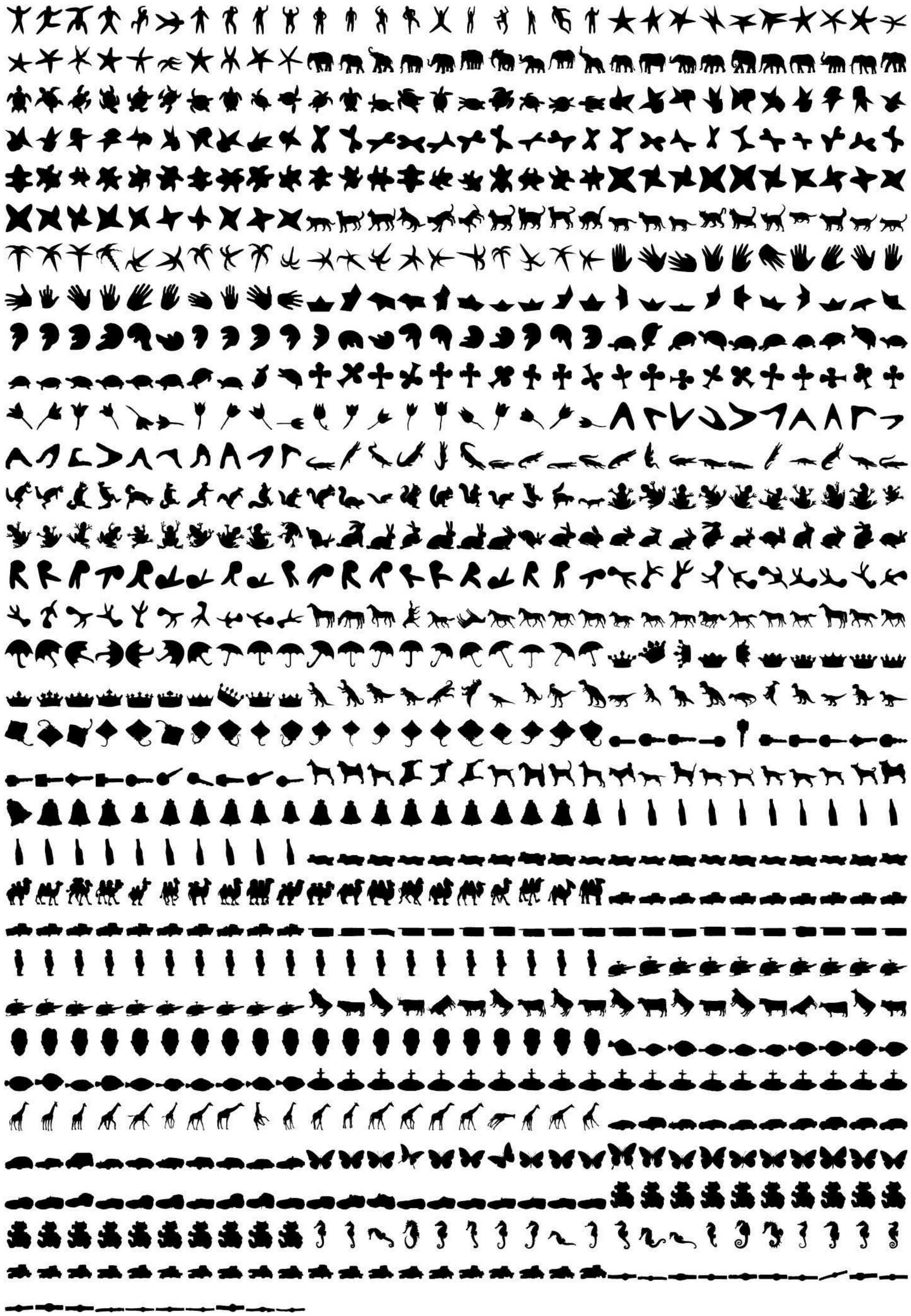


Figure 5.6: The shape database used in the experiments. It contains a total of 1000 shapes (50 categories, each having 20 examples).

We evaluate the performance of the matching method of Baseski [7] and the proposed category-influenced matching by performing 100 experiments where in each run the shape database is randomly partitioned into two: 750 shapes for training (15 examples for each category) and the remaining 250 shapes for testing. A sample partition is given in Appendix A. For each partition, each shape in the test set is used as a query shape and matched with all the remaining 750 shapes in the training set. The knowledge about a specific category is represented by a category tree formed using the shapes in the training set which belong to that category. The average precision-recall curves are presented in Figure 5.7. Notice that the importance of context is clearly visible at high recall levels, where the improvement obtained by incorporating semantic category knowledge into matching shows 50% improvement in the precision at 100% recall. For the partition given in Appendix A, the results of the matching method of Baseski [7] and the category-influenced matching method are respectively shown in Appendix B and Appendix C.

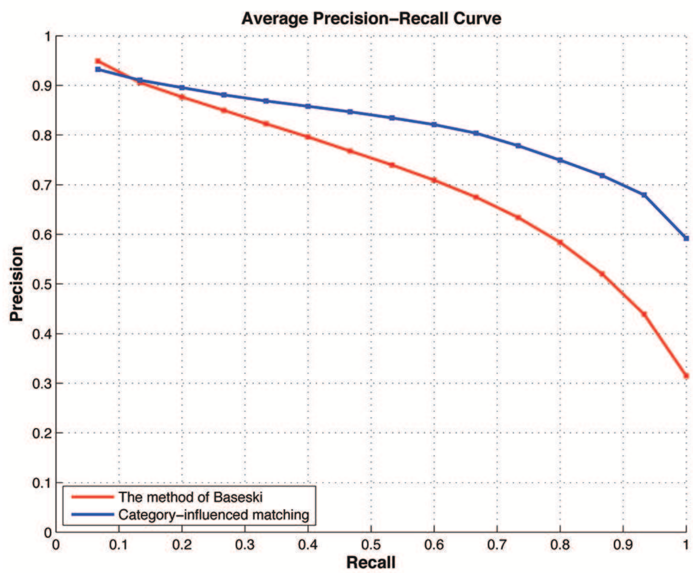


Figure 5.7: Average precision-recall curves. At each recall level, compare the precision values of the category-influenced matching method (shown in blue) to those of the method of Baseski [7] (shown in red).

5.1.5 A Coarse-To-Fine Strategy To Incorporate Categorical Boundary Similarity into Category-Influenced Matching

In Section 2.3.1, we have demonstrated a way to obtain approximate radius function of a positive skeleton branch from a corresponding TSP surface. Recall that when computed with a sufficiently large value of ρ, the resulting TSP surface becomes a smoothed version of distance surface [106, 107]. Thus, the extracted radius functions are very smooth, representing only the most significant boundary details. Once the disconnected skeleton representation is enriched with the radius functions, we can compare boundary similarity of two positive branches based on the corresponding radius functions, as traditionally utilized in skeletal matching methods such as [5, 32, 74, 90, 98]. This provides us a way to incorporate boundary similarity into category-influenced matching method. Our strategy is to adopt a coarse-to-fine approach that we first find a match between two shapes, and then recalculate the dissimilarity by taking boundary similarity into account. The details are as follows:

In constructing a shape tree, we uniformly sample 32 points along each extracted positive branch and store the corresponding vector of radii values as an additional attribute in the corresponding node of the shape tree. Note that this vector is null for the nodes which correspond to negative skeleton branches. In forming a category tree, we employed the approach in [127]. That is, we model deformations of shape section associated with a positive branch in the presence of category knowledge. Here, we first collect all the information about boundary details coming from the category members and then apply principal component analysis (PCA) to form a low-dimensional linear space for the observed deformations. Subsequently, in the related nodes of the category tree, we additionally store the mean of the approximate radius functions together with a reduced set of principle components. In the experiments, these deformation spaces are all represented with the first five principal components. In Figure 5.8-Figure 5.11, we give some illustrative examples showing the observed variations some shape sections with different characteristics, captured by the uniformly sampled radius functions of corresponding skeleton branches.

Once the descriptions of shape trees and category trees are enriched in this way, we define the following two-step procedure to incorporate boundary similarity into category-influenced matching of shapes: Let $\mathcal{T}_1$ denote the shape tree of the query shape which is being compared to a database shape whose shape tree is denoted by $\mathcal{T}_2$ and suppose $\mathcal{T}_2$ is associated with the category tree $\mathcal{T}_\mathcal{C}$. The enriched version of $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_\mathcal{C}$ can all be expressed as follows:

$$\mathcal{T}_1 = \left\{ u_i = \left(u_i^r, u_i^\theta, u_i^l, u_i^f, u_i^{type} \right) \mid u_i \in \mathcal{N}_1 \right\}$$
$$\mathcal{T}_2 = \left\{ v_j = \left(v_j^r, v_j^\theta, v_j^l, v_j^f, v_j^{type} \right) \mid v_j \in \mathcal{N}_2 \right\}$$
$$\mathcal{T}_\mathcal{C} = \left\{ \mathcal{B}_k = \left(\mathcal{B}_k^{r_{min}}, \mathcal{B}_k^{r_{max}}, \mathcal{B}_k^{\theta_{min}}, \mathcal{B}_k^{\theta_{max}}, \mathcal{B}_k^{l_{min}}, \mathcal{B}_k^{l_{max}}, \mathcal{B}_k^{f_\mu}, \mathcal{B}_k^{f_\Phi}, \mathcal{B}_k^{type}, freq(\mathcal{B}_k) \right) \mid \mathcal{B}_k \in \mathcal{N}_\mathcal{C} \right\}$$

where the additional attribute f in $\mathcal{T}_1$ and $\mathcal{T}_2$ denotes the uniformly sampled approximate radius function of the corresponding branch and f_μ and f_Φ in $\mathcal{T}_\mathcal{C}$ denote the mean of the approximate radius functions of the associated branches and the reduced set of corresponding principle components, respectively. Note that for each leaf nodes corresponding to a negative branch, these additional attributes are all null.

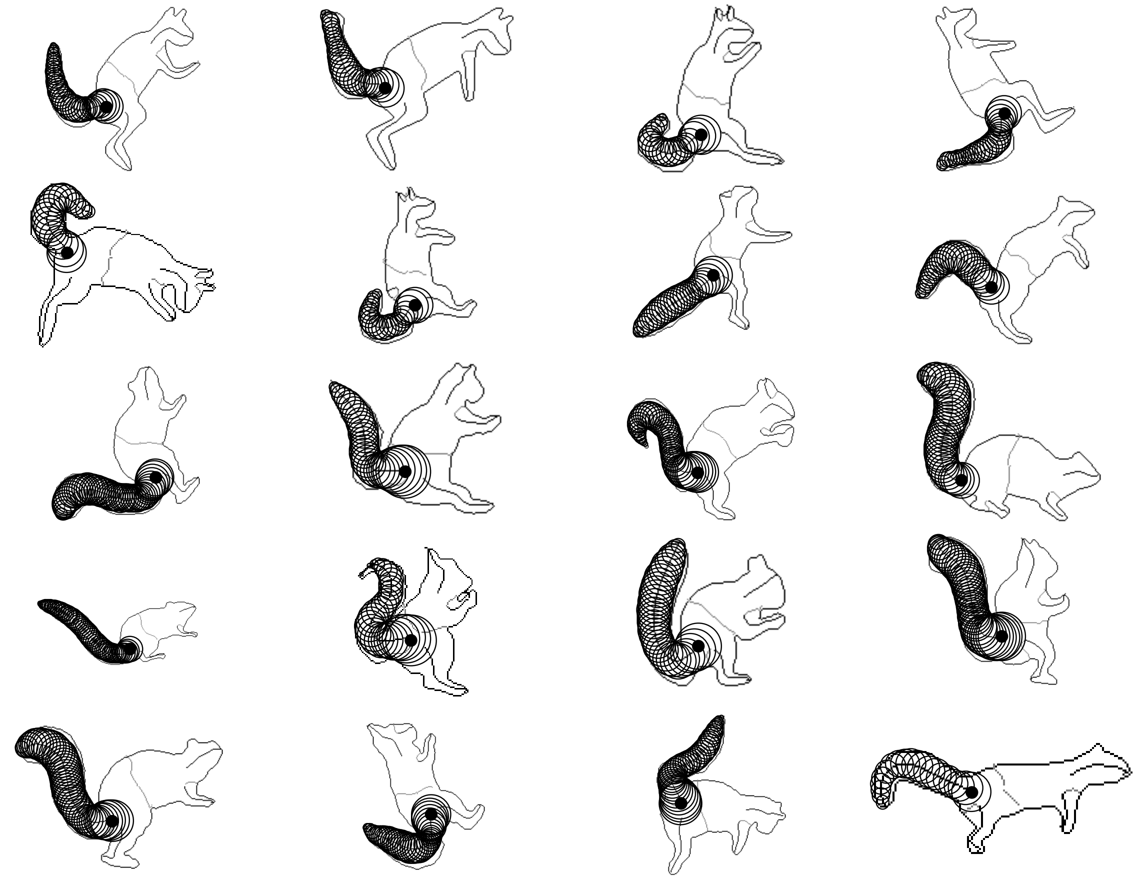
First, category-influenced matching between $\mathcal{T}_1$ and $\mathcal{T}_2$ is performed in the way described previously. In the refinement step, the overall dissimilarity is re-calculated according to Equation 5.1, this time by considering the boundary similarities between every matched pair of branches. The definition of this boundary similarity measure is given in Equation 5.2 and requires projecting the corresponding uniformly sampled radius functions onto the related low-dimensional deformation space.

$$d\left(\mathcal{T}_1, \mathcal{T}_2\right) = \sum_{u \in \Lambda} \texttt{remove}\left(u\right) + \sum_{v \in \Delta} \texttt{insert}\left(v\right) + \sum_{(u,v) \in \Omega} \Big((1 - \phi(u,v)) \times \texttt{change}\left(u, v, \mathcal{B}\right) \Big) \tag{5.1}$$

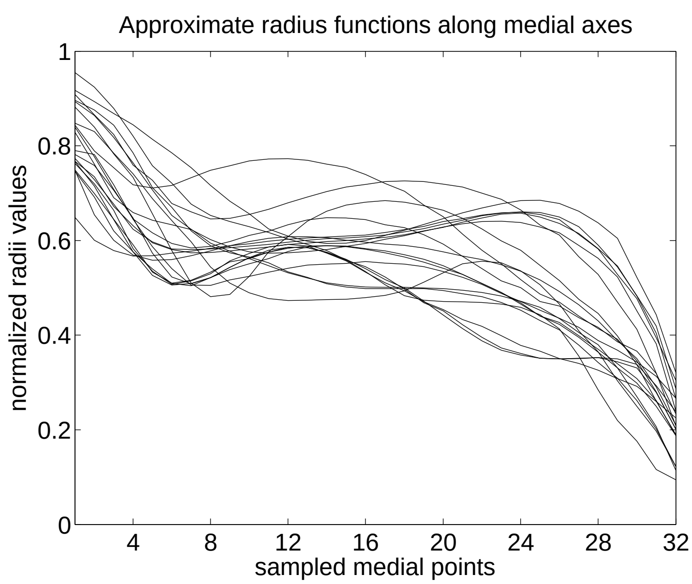
$$\phi(u,v) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\sum_{i=1}^{5} \frac{\left(\alpha_i - \beta_i\right)^2}{2\sigma^2} \right) & \text{if } u^{type} \text{ and } v^{type} \text{ are } positive \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

where α and β correspond to the projected values of approximate radius functions of u and v and σ is taken as $\sigma = 0.4$ in the experiments.

In the following, we investigate the effect of incorporating categorical boundary similarity into category-influenced matching using the training and test shape sets given in Appendix A. Figure 5.12-Figure 5.19 shows some sample matching results with and without boundary information, where for each category, the corresponding category tree influencing the distance computation and the suggested deformations spaces are constructed using 15 example shapes belonging to that category. As it can be clearly seen, the proposed approach results in perceptually more meaningful matching scores, as compared to the original category-influenced matching formulation.

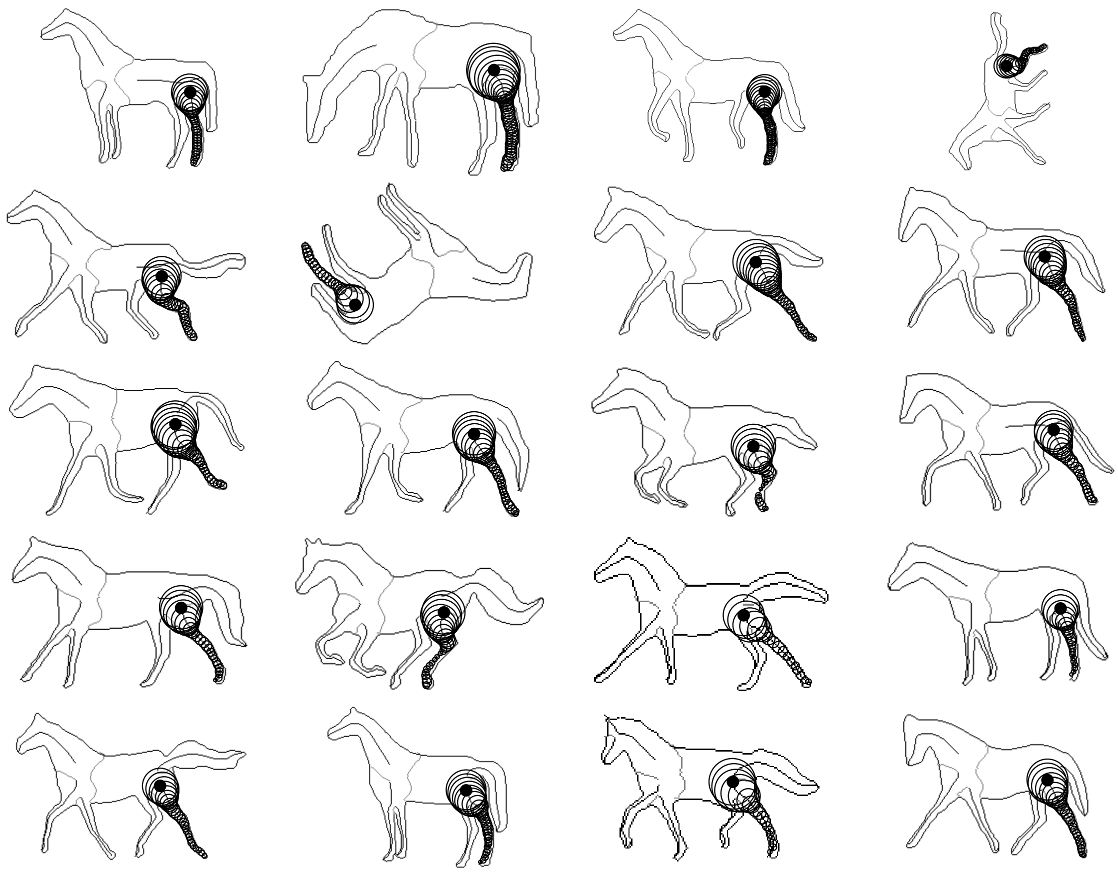


(a)

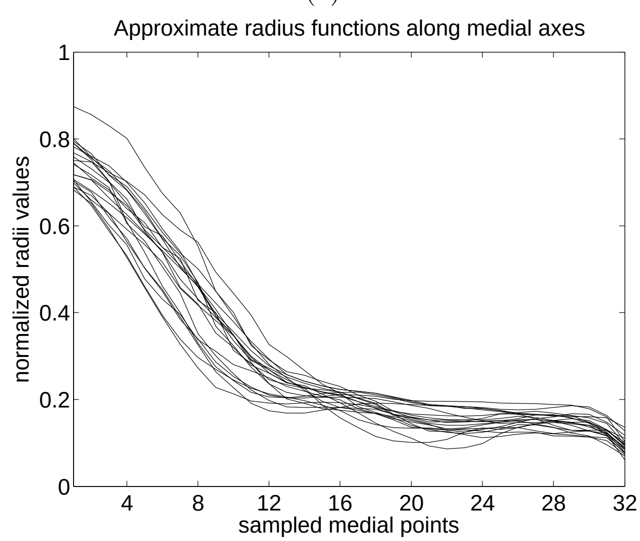


(b)

Figure 5.8: An analysis of boundary deformations using approximate radius functions. (a) Equivalent shape sections of 20 `squirrel` shapes, each associated with a positive skeleton branch. (b) The corresponding set of uniformly sampled approximate radius functions.

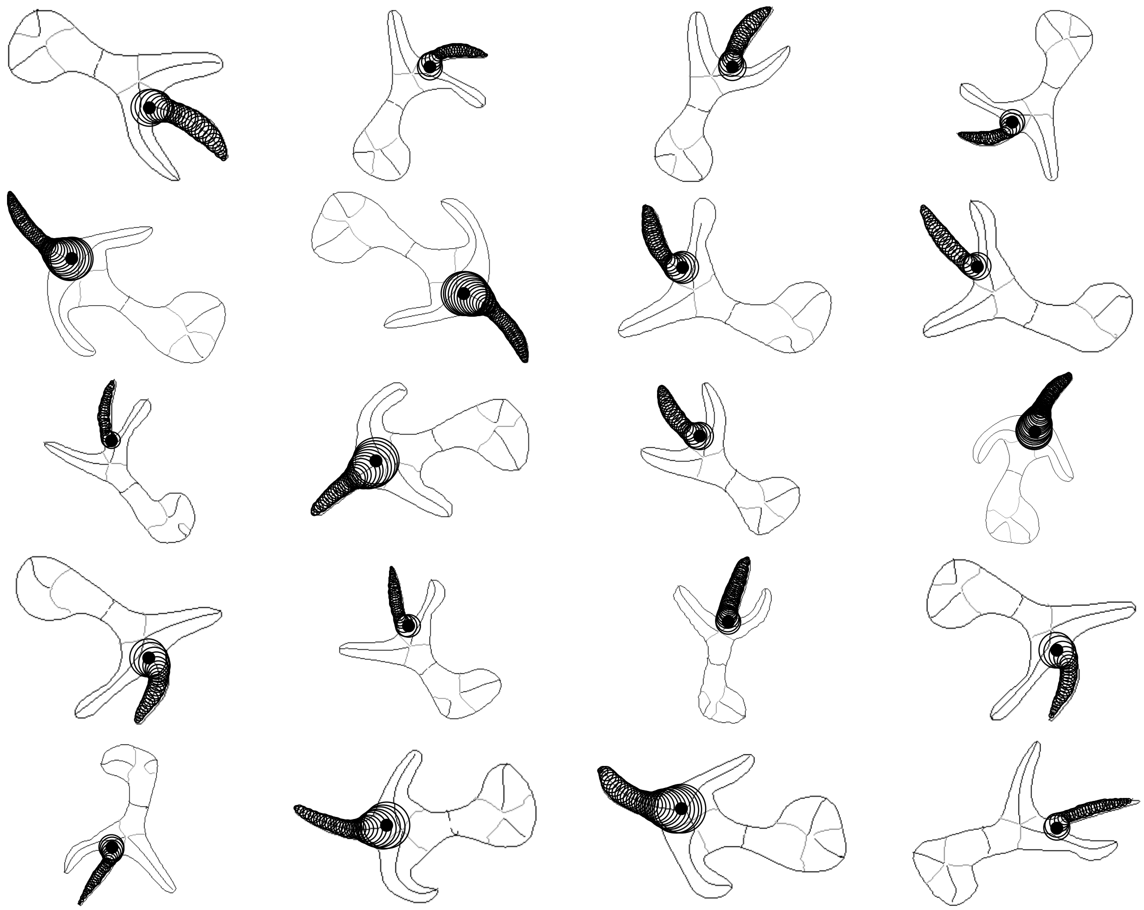


(a)

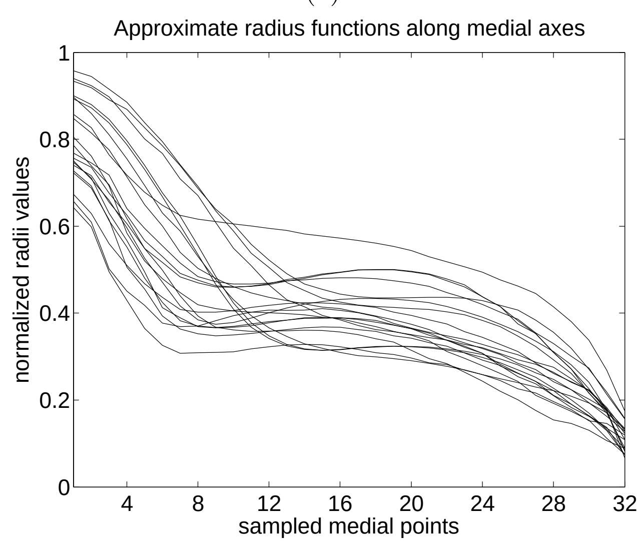


(b)

Figure 5.9: An analysis of boundary deformations using approximate radius functions. (a) Equivalent shape sections of 20 `horse` shapes, each associated with a positive skeleton branch. (b) The corresponding set of uniformly sampled approximate radius functions.

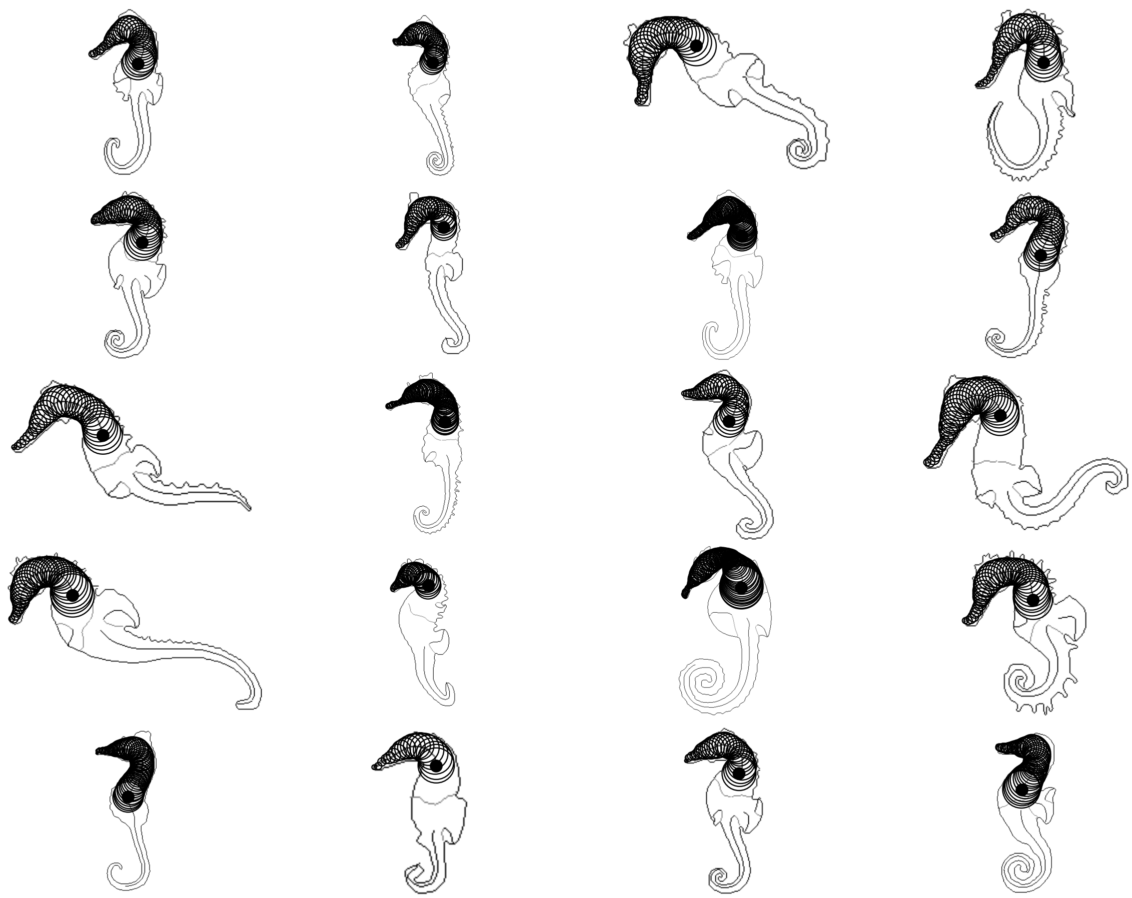


(a)

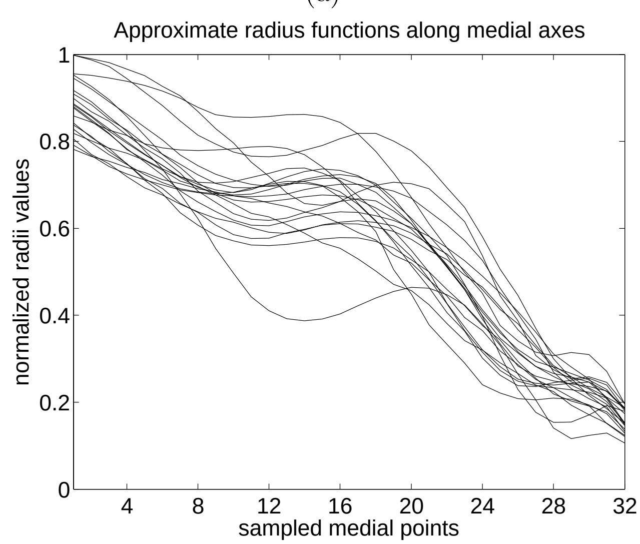


(b)

Figure 5.10: An analysis of boundary deformations using approximate radius functions. (a) Equivalent shape sections of 20 shapes from the same artificial shape category, each associated with a positive skeleton branch. (b) The corresponding set of uniformly sampled approximate radius functions.

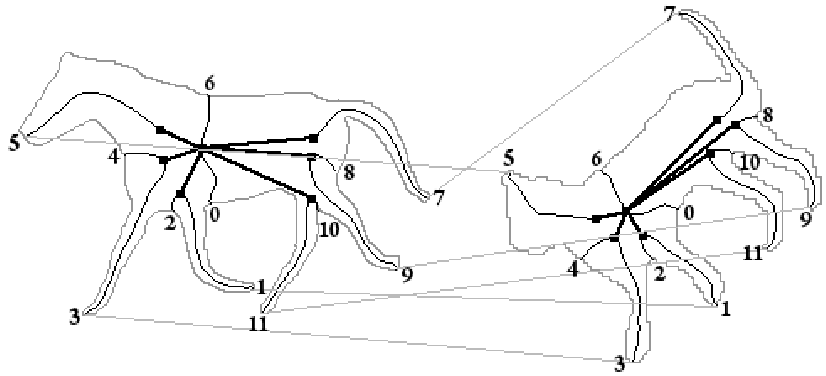


(a)

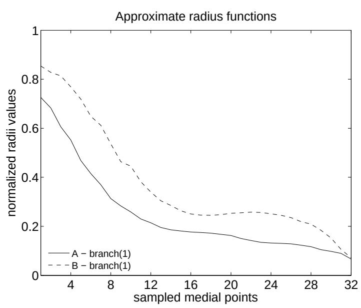


(b)

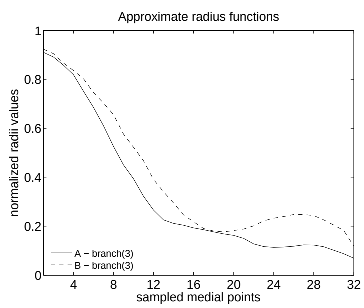
Figure 5.11: An analysis of boundary deformations using approximate radius functions. (a) Equivalent shape sections of 20 `seahorse` shapes, each associated with a positive skeleton branch. (b) The corresponding set of uniformly sampled approximate radius functions.



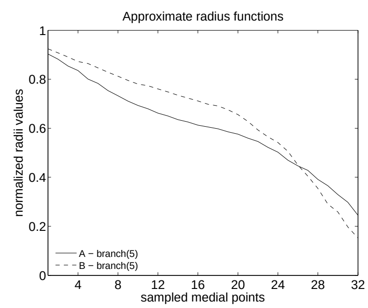
(a)



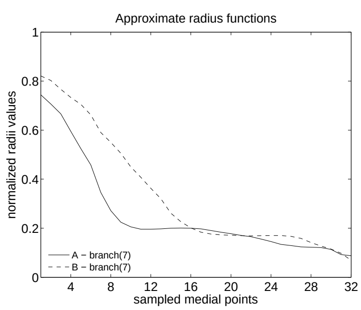
(b)



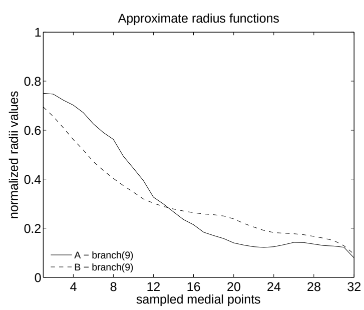
(c)



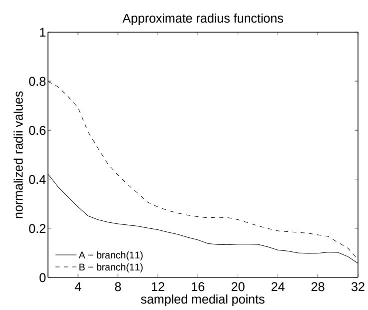
(d)



(e)

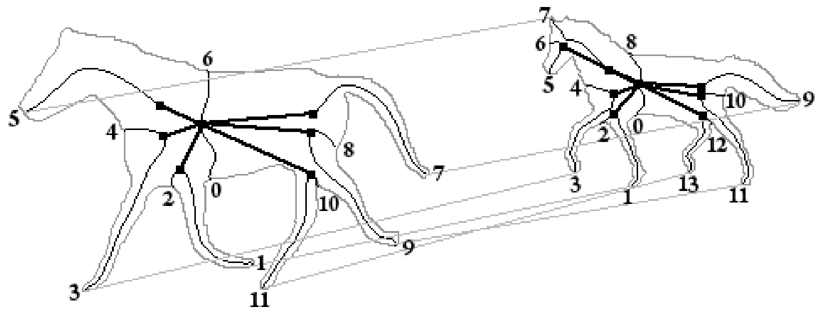


(f)

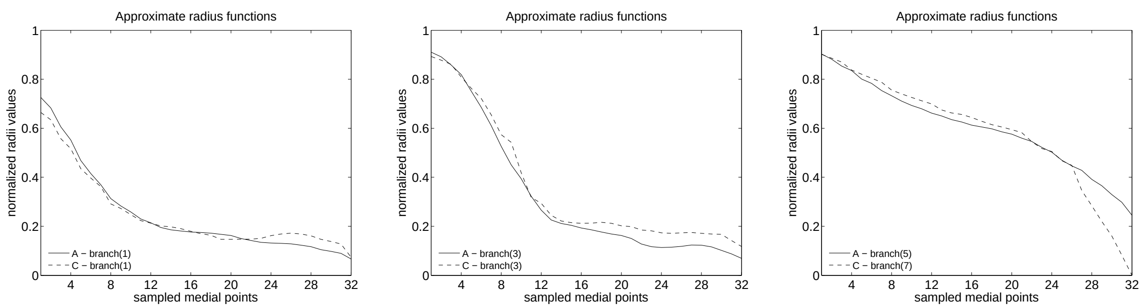


(g)

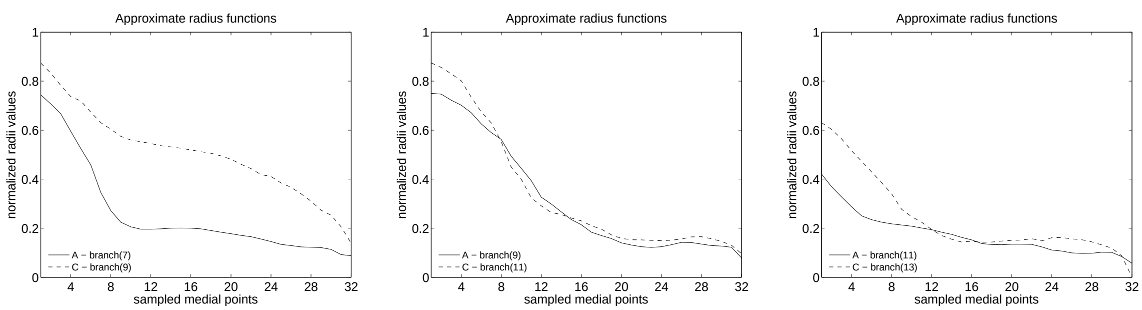
Figure 5.12: (a) Category-influenced skeletal matching result between the shapes $\mathcal{A}$ and $\mathcal{B}$. Total matching cost is reduced from 0.7240 to 0.5800 when boundary similarity is incorporated. (b)-(g) Radius profiles of matched pair of branches $\mathcal{A}_1$ and $\mathcal{B}_1$, $\mathcal{A}_3$ and $\mathcal{B}_3$, $\mathcal{A}_5$ and $\mathcal{B}_5$, $\mathcal{A}_7$ and $\mathcal{B}_7$, $\mathcal{A}_9$ and $\mathcal{B}_9$, $\mathcal{A}_{11}$ and $\mathcal{B}_{11}$, respectively.



(a)



(b) (c) (d)



(e) (f) (g)

Figure 5.13: (a) Category-influenced skeletal matching result between the shapes $\mathcal{A}$ and $\mathcal{C}$. Total matching cost is reduced from 0.7823 to 0.5368 when boundary similarity is incorporated. (b)-(g) Radius profiles of matched pair of branches $\mathcal{A}_1$ and $\mathcal{C}_1$, $\mathcal{A}_3$ and $\mathcal{C}_3$, $\mathcal{A}_5$ and $\mathcal{C}_7$, $\mathcal{A}_7$ and $\mathcal{C}_9$, $\mathcal{A}_9$ and $\mathcal{C}_{11}$, $\mathcal{A}_{11}$ and $\mathcal{C}_{13}$, respectively.

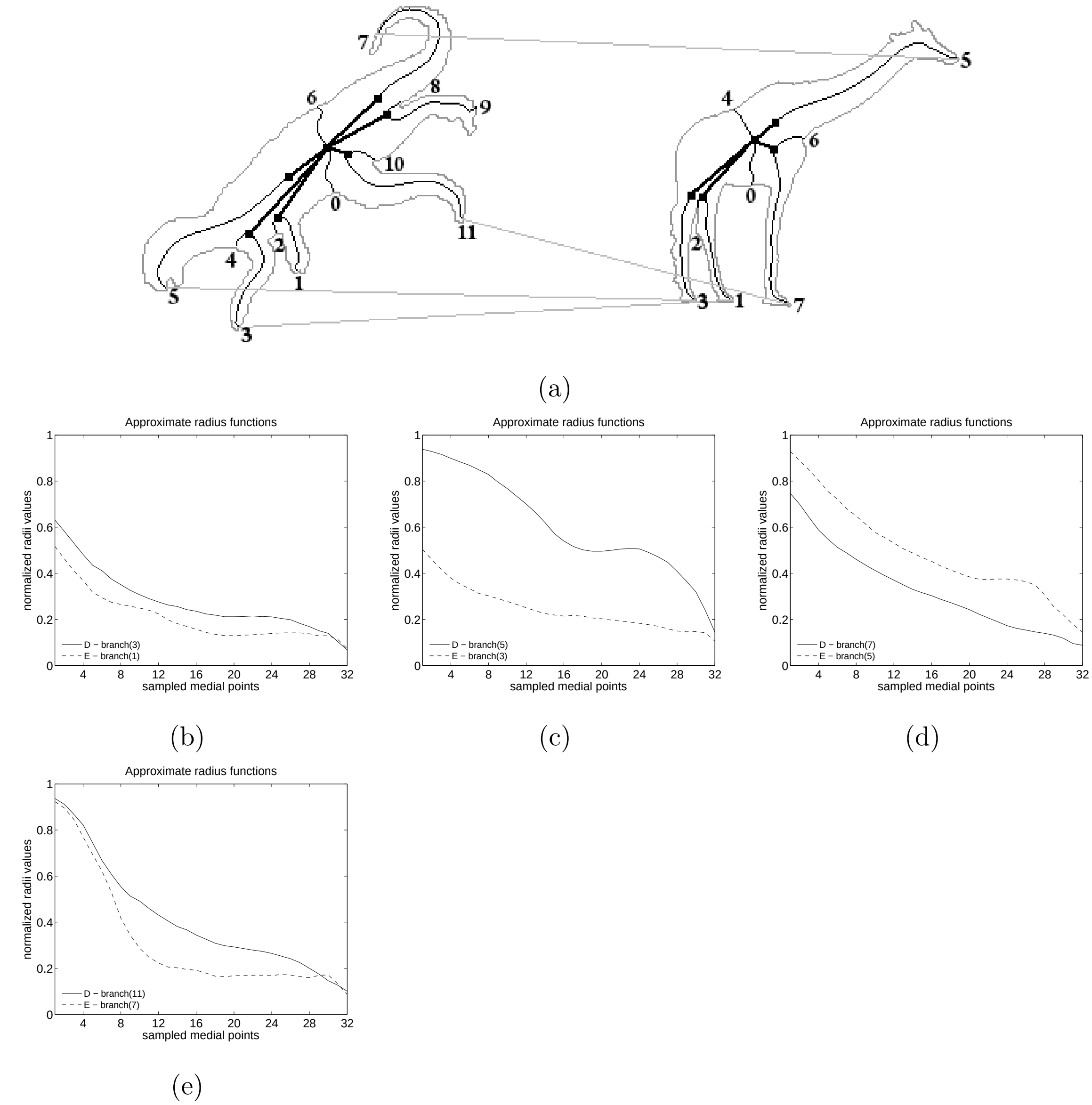


Figure 5.14: (a) Category-influenced skeletal matching result between the shapes $\mathcal{D}$ and $\mathcal{E}$. Total matching cost is reduced from 1.2904 to 1.1989 when boundary similarity is incorporated. (b)-(e) Radius profiles of matched pair of branches $\mathcal{D}_3$ and $\mathcal{E}_1$, $\mathcal{D}_5$ and $\mathcal{E}_3$, $\mathcal{D}_7$ and $\mathcal{E}_5$, $\mathcal{D}_{11}$ and $\mathcal{E}_7$, respectively.

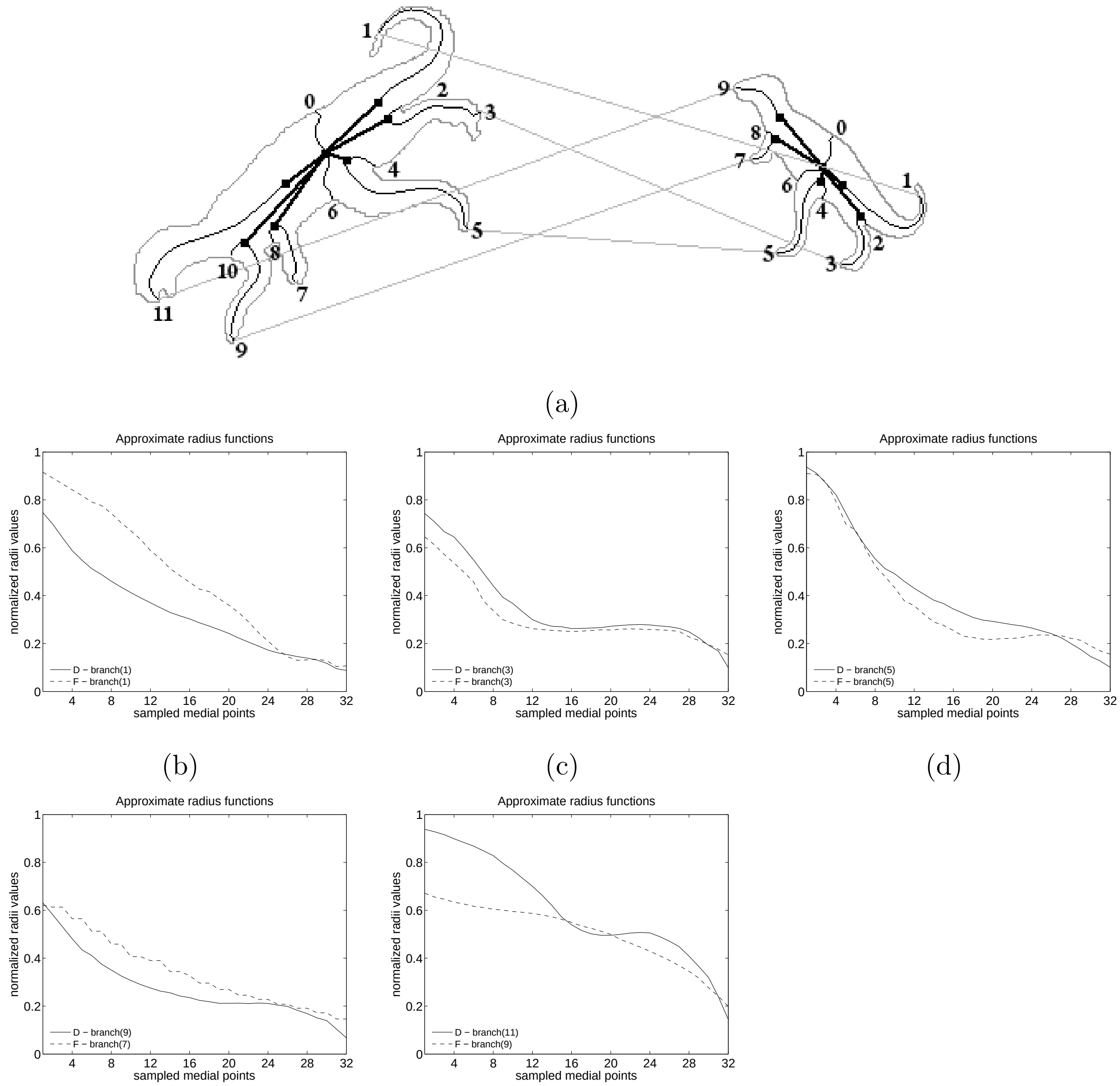
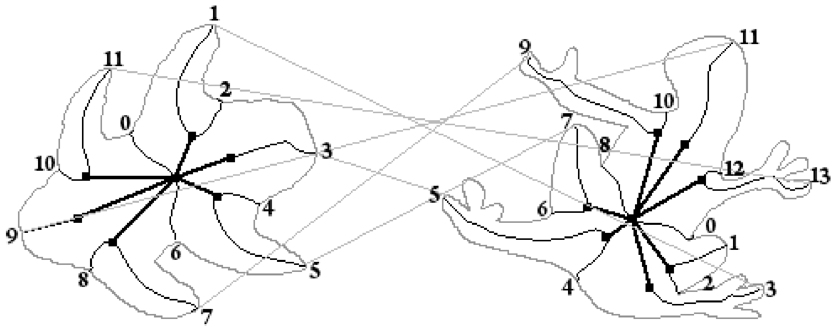
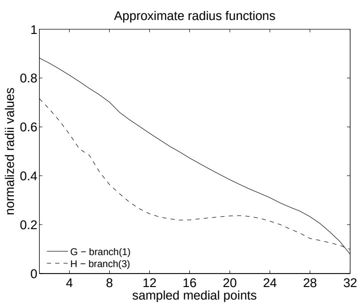


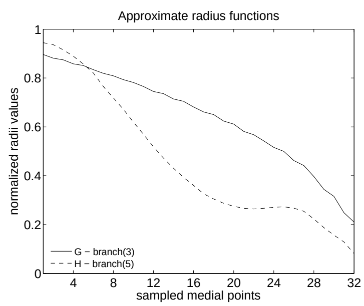
Figure 5.15: (a) Category-influenced skeletal matching result between the shapes $\mathcal{D}$ and $\mathcal{F}$. Total matching cost is reduced from 1.4936 to 0.9458 when boundary similarity is incorporated. (b)-(f) Radius profiles of matched pair of branches $\mathcal{D}_1$ and $\mathcal{F}_1$, $\mathcal{D}_3$ and $\mathcal{F}_3$, $\mathcal{D}_5$ and $\mathcal{F}_5$, $\mathcal{D}_9$ and $\mathcal{F}_7$, $\mathcal{D}_{11}$ and $\mathcal{F}_9$, respectively.



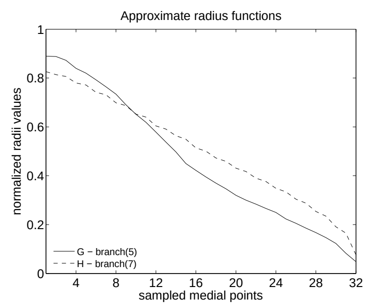
(a)



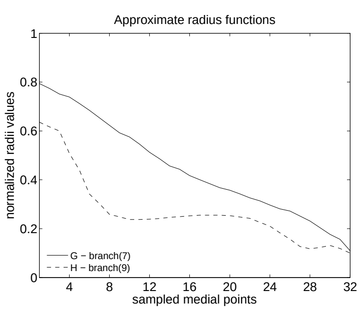
(b)



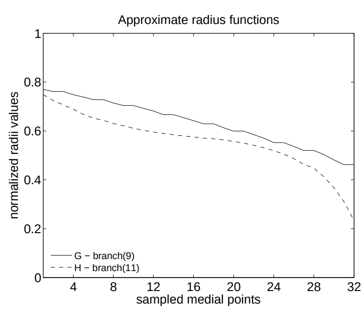
(c)



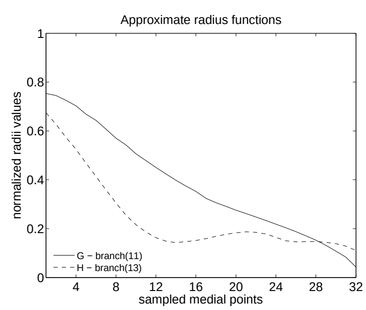
(d)



(e)

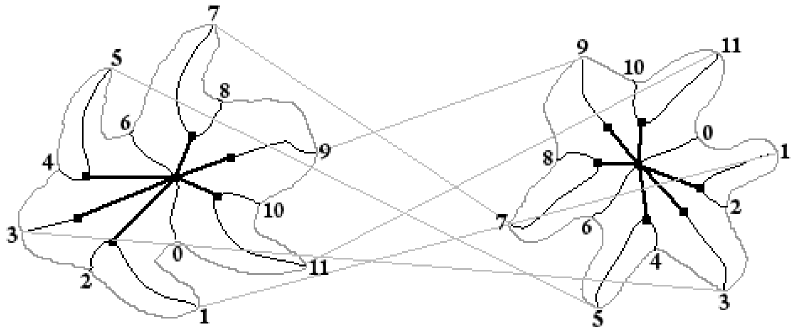


(f)

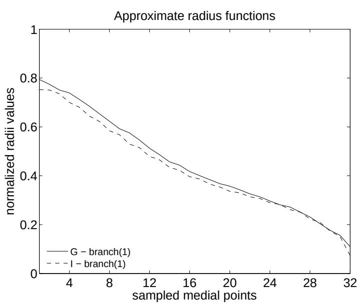


(g)

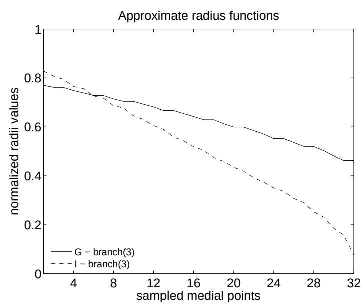
Figure 5.16: (a) Category-influenced skeletal matching result between the shapes $\mathcal{G}$ and $\mathcal{H}$. Total matching cost is reduced from 2.1879 to 1.9576 when boundary similarity is incorporated. (b)-(g) Radius profiles of matched pair of branches $\mathcal{G}_1$ and $\mathcal{H}_3$, $\mathcal{G}_3$ and $\mathcal{H}_5$, $\mathcal{G}_5$ and $\mathcal{H}_7$, $\mathcal{G}_7$ and $\mathcal{H}_9$, $\mathcal{G}_9$ and $\mathcal{H}_{11}$, $\mathcal{G}_{11}$ and $\mathcal{H}_{13}$, respectively.



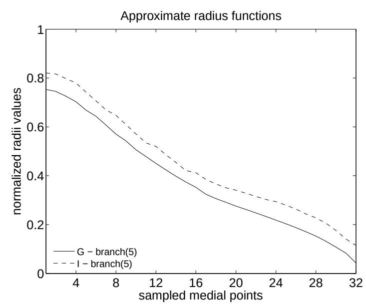
(a)



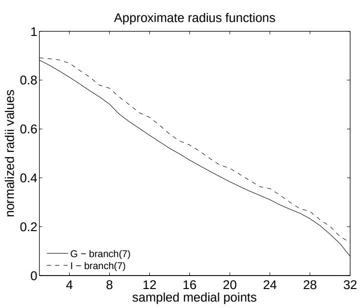
(b)



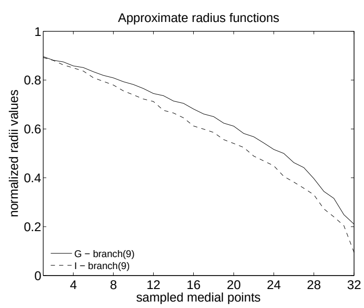
(c)



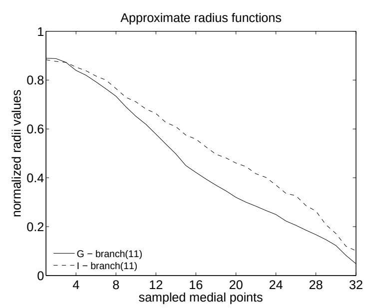
(d)



(e)

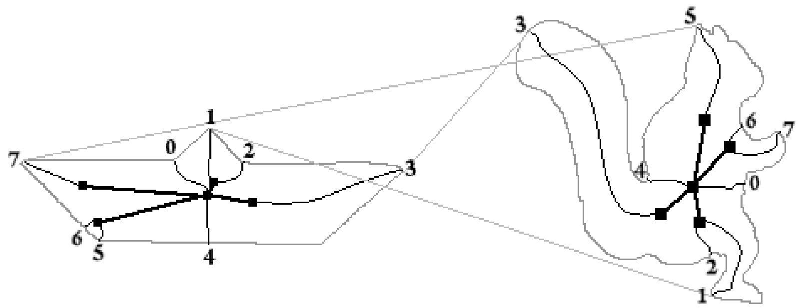


(f)

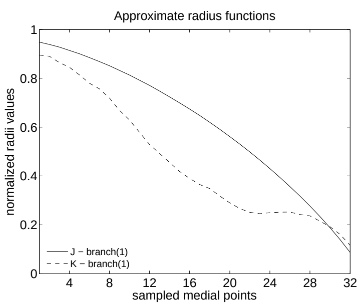


(g)

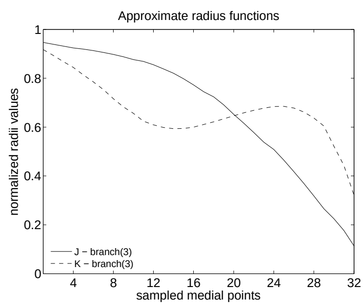
Figure 5.17: (a) Category-influenced skeletal matching result between the shapes $\mathcal{G}$ and $\mathcal{I}$. Total matching cost is reduced from 3.0387 to 1.8744 when boundary similarity is incorporated. (b)-(g) Radius profiles of matched pair of branches $\mathcal{G}_1$ and $\mathcal{I}_1$, $\mathcal{G}_3$ and $\mathcal{I}_3$, $\mathcal{G}_5$ and $\mathcal{I}_5$, $\mathcal{G}_7$ and $\mathcal{I}_7$, $\mathcal{G}_9$ and $\mathcal{I}_9$, $\mathcal{G}_{11}$ and $\mathcal{I}_{11}$, respectively.



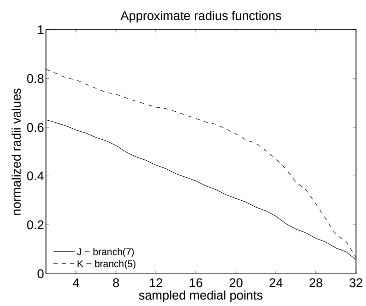
(a)



(b)

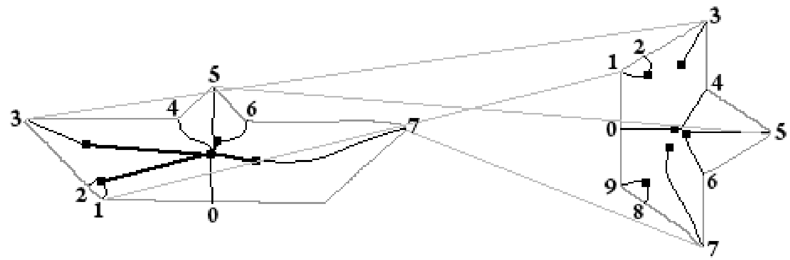


(c)

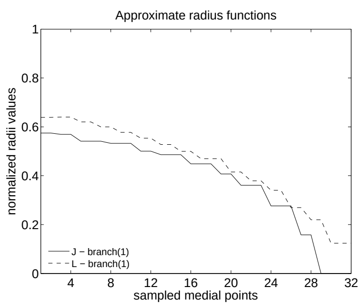


(d)

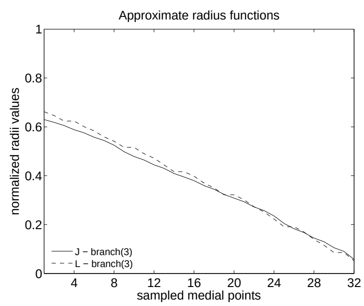
Figure 5.18: (a) Category-influenced skeletal matching result between the shapes $\mathcal{J}$ and $\mathcal{K}$. Total matching cost is reduced from 0.8105 to 0.8052 when boundary similarity is incorporated. (b)-(d) Radius profiles of matched pair of branches $\mathcal{J}_1$ and $\mathcal{K}_1$, $\mathcal{J}_3$ and $\mathcal{K}_3$, $\mathcal{J}_7$ and $\mathcal{K}_5$, respectively.



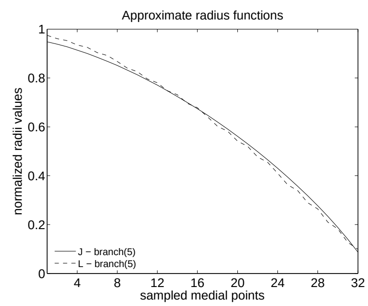
(a)



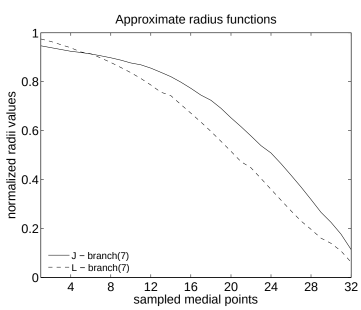
(b)



(c)



(d)



(e)

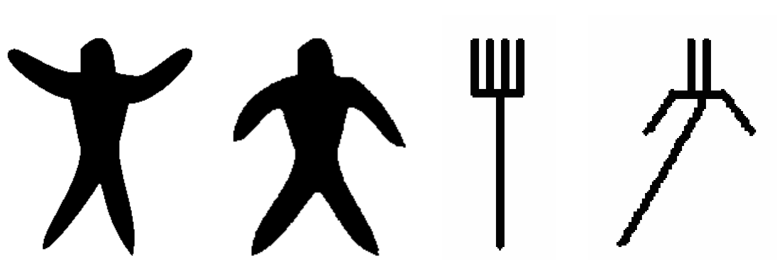
Figure 5.19: (a) Category-influenced skeletal matching result between the shapes $\mathcal{J}$ and $\mathcal{L}$. Total matching cost is reduced from 1.0875 to 0.6738 when boundary similarity is incorporated. (b)-(e) Radius profiles of matched pair of branches $\mathcal{J}_1$ and $\mathcal{L}_1$, $\mathcal{J}_3$ and $\mathcal{L}_3$, $\mathcal{J}_5$ and $\mathcal{L}_5$, $\mathcal{J}_7$ and $\mathcal{L}_7$, respectively.

5.2 Contextual Sensitivity to Articulation of Parts in Skeletal Shape Matching

The complexity of visual shape recognition requires representation and matching schemes that are invariant or insensitive to visual transformations such as deformations and articulation of parts. In this regard, skeletal representation schemes have been widely used for generic shape recognition as they lead to articulation insensitive representations while capturing the part structure of shapes [2, 5, 8, 57, 61, 74, 87, 94, 98, 110, 127].

Despite their desirable strengths, presemantic and purely syntactic level of skeletal representations fail to distinguish a likely articulation from an unlikely one. In this regard, consider the shapes given in Figure 5.20. On one hand, certain context might require articulation invariance such as asserting that the shapes shown in (a) and (b) are the same shape. On the other hand, it is less natural to make the same claim for the shapes in (c) and (d). We refer this as *contextual sensitivity to articulations*. In fact, the distinction between such cases lies in the previous encounters to the shapes in consideration and hence, it requires the interpretation which we considered in the beginning of this chapter.

The previous example shows that it is essential for a skeleton-based recognition framework to have an additional representation scheme to handle sensitivity to articulations depending on the context. In this section, we present such a complementary work to the method of Aslan and Tari [3]. Motivated both by the hybrid (axis vs. point) nature of disconnected skeleton representation and the techniques developed for landmark-based shape analysis [13, 42], we propose a part-centered coordinate frame, referred to as *semi-local coordinate frame*, that provides us a representation space for making inferences about articulations, in which similar articulations or bendings yield closer coordinates. Then, we demonstrate the use of semi-local



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

Figure 5.20: Contextual sensitivity to articulation of parts. See text for explanation (images taken from [1]).

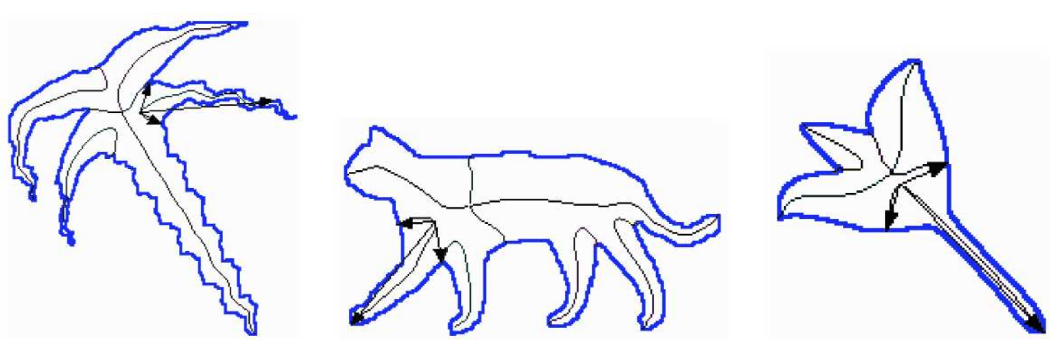
coordinate frame on a set of human shapes with different postures.

Note that the tools for landmark-based shape analysis [13, 42] are previously adopted by Burl *et al.* to design a recognition scheme by considering the relative spatial arrangement of shape sections [19]. However, our goal is completely different from the aim in [19] and the related followup works in the sense that they use these ideas to filter out global transformations in order to capture shape information. On the other hand, we filter out the shape information in addition to global transformations to capture the articulations.

5.2.1 Semi-local Coordinate Frame

Recall that in the disconnected skeleton, the extracted skeleton is in the form of a set of disconnected skeleton branches, each corresponding to a salient part of the shape. Moreover, a positive branch is neighbored by two negative branches. Typically, the start points of the negative branches as well as all the disconnection locations, are quite stable under bendings and articulations. It is the tip of the positive branch that moves freely if the branch is denoting a deformable section. Consider the disconnected branches of some shapes as displayed in Figure 5.21. Four points define three vectors, starting from the disconnection point of the protrusion branch and ending respectively at the starts of the two indentations and the protrusion. The third vector can be represented as a linear combination of the remaining two.

When these vectors are transformed to standard bases, each configuration can be represented by only a single point, which denotes the local pose of a shape section, which may or may not articulate or bend. We name this coordinate frame as *semi-local coordinate frame* and a point in this coordinate frame as $\mathcal{LA}$ *coordinate*. Note that any measurement



(a) (b) (c)

Figure 5.21: Vector combinations of some skeleton branches.

in semi-local coordinate frame is deprived of any shape information as well as Euclidean transformations, as illustrated in Figure 5.22. By separating visual transformations, we can produce descriptions that are sensitive to any combination of changes in scale, location, orientation, and articulations in addition to descriptions that are invariant to these changes.

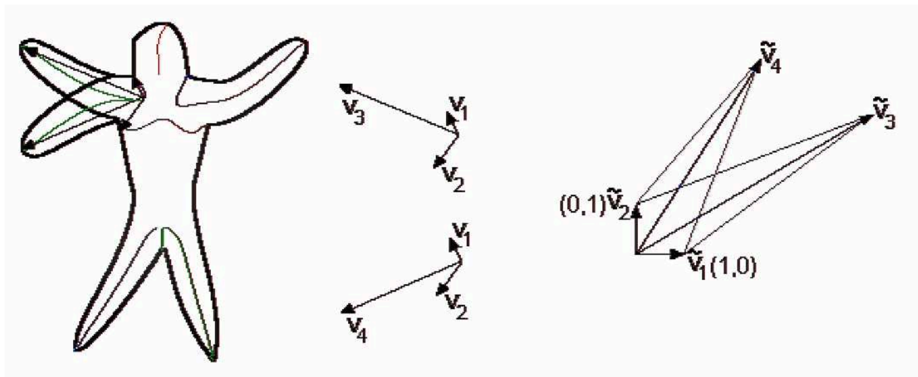


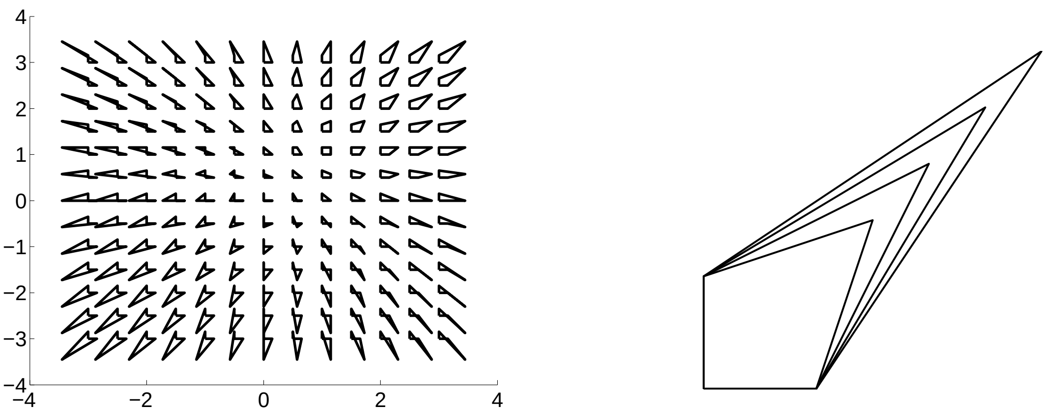
Figure 5.22: Articulation of a section can be described by a single point in the semi-local coordinate frame.

5.2.2 Articulation Space

From a geometric point of view, shape is defined as the geometric information that remains when location, scale and rotational effects are filtered out [42]. On the other hand, it is the shape information that has to be filtered out in order to make articulations explicit. In this regard, we show that semi-local coordinate frame can provide us such an *articulation space* to represent solely articulation information.

Notice that the three vectors used in representing the local pose of a shape section define a quadrangle. Therefore it is possible to associate each $\mathcal{LA}$ coordinate with a set of affine related quadrangles or equivalently a canonical quadrangle represented in $\mathcal{LA}$ coordinates. The collection of such quadrangles may be considered as an articulation space (Figure 5.23). Note that this space is qualitatively similar to the triangle shape space of Kendall [42].

In articulation space, similar articulations or bendings yield closer coordinates. Consider the two human silhouettes shown in Figure 5.24. Since their left arms have similar postures, the corresponding articulations are represented by two nearby points in $\mathcal{LA}$ coordinates. On the contrary, $\mathcal{LA}$ coordinates of right arms are far distant from each other. Notice that a

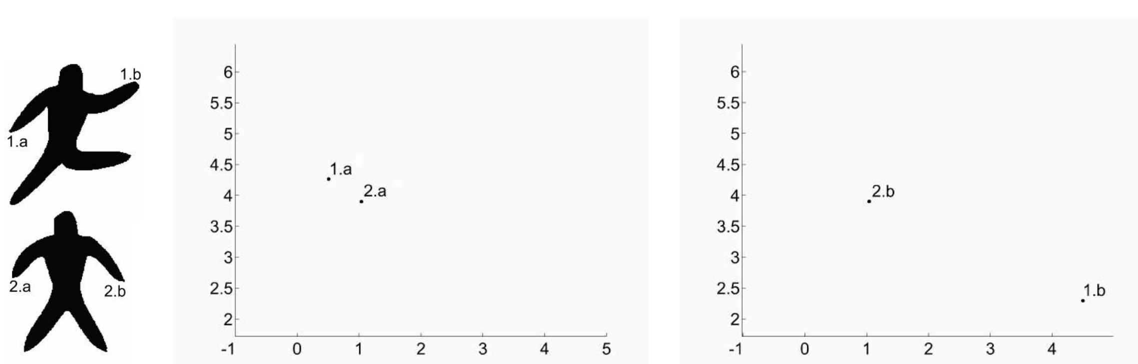


(a) (b)

Figure 5.23: Articulation space. (a) Each point in the articulation space can be associated with a quadrangle (b) four quadrangles that fall on $x = y$ line in the articulation space.

horizontal arm will be on $x = y$ line, whose polar representation is $(l, \pi/4)$ where l is the dimensionless arm length.

Assuming that the arm is a single rigid body, possible coordinates should fall into a circle whose radius is l since the size information is already filtered out. One may think that quadrangles that lie on a constant angle line in the articulation space (such as any two quadrangles shown in Figure 5.23(b)) can not both belong to the same shape section and may come to a conclusion that the whole space is not utilized and the articulations lie on a 1D manifold. This is not the case. Figure 5.25(d) shows three different postures of human



(a) (b) (c)

Figure 5.24: $\mathcal{LA}$ coordinates. (a) two human silhouettes with different postures (b) $\mathcal{LA}$ coordinates of the left arms (c) $\mathcal{LA}$ coordinates of the right arms.

arm in a single image consisting of two rigid body movements of arm (Figure 5.25(a)-(b)), and a case where a bending occurs (Figure 5.25(c)). The corresponding $\mathcal{LA}$ coordinates of left arms (which are determined from the disconnected skeleton representations computed from extracted silhouettes) are given in Figure 5.25(e). Notice that due to initially ignored joints such as elbows, $\mathcal{LA}$ coordinates of a shape section may not always lie on a circular arc.

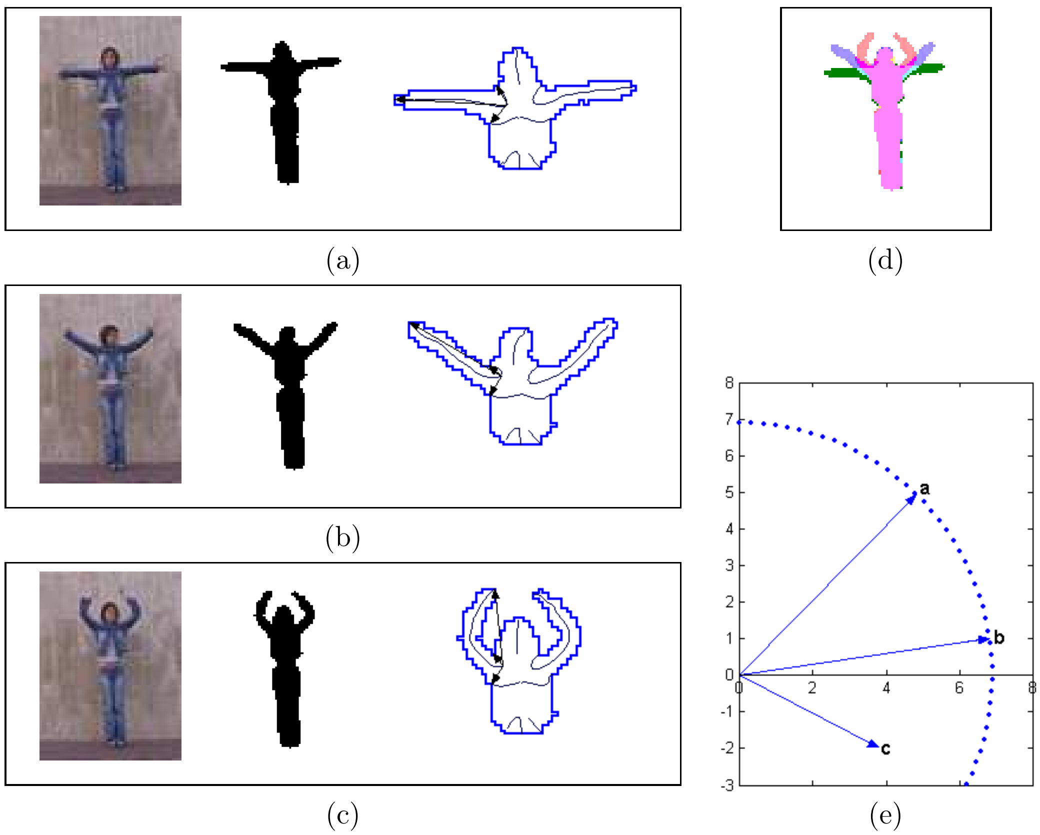


Figure 5.25: Articulations and bendings in the articulation space. (a)-(c) three different postures of a human figure (taken from `ira_wave2` video sequence from action-silhouette database of [10]), the corresponding binary silhouettes and disconnected skeletons extracted from upper body portions (d) these three postures combined (e) $\mathcal{LA}$ coordinate representations of left arms in the articulation space.

Representing articulations in semi-local coordinate frame deteriorates when the points defining the coordinate system are nearly colinear, *e.g.* the head section in Figure 5.26(a). Even a very small change in the location of the tip of the positive branch might lead to a significant change in its $\mathcal{LA}$ coordinate. Another degenerate situation is encountered when the second indentation is not close enough, *e.g.* the leg sections in Figure 5.26(a). In this case, the length of one of the vectors defining the coordinate frame becomes too large. Hence the corresponding semi-local coordinate frame fails to capture the variations of the tip of protrusion. But note that this latter degenerate case is a side effect of the skeleton extraction and may be alleviated by modifying it.

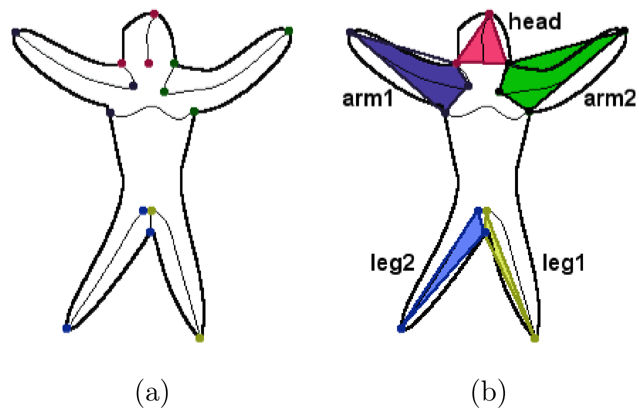


Figure 5.26: Deformable sections of a human shape via its disconnected skeleton. (a) Starting and ending points of skeleton branches (b) quadrangle or triangle representations of deformable sections.

In these degenerate situations, it is possible to define the local frame using only two points and the coordinate representation becomes equivalent to that of Bookstein used for analyzing landmark data [13]. At such object sections the set of quadrangles are replaced by a set of triangles.

5.2.3 Inferences in Articulation Space

A collection of possible postures or deformations defines either a subset of the articulation space (static view) or a trajectory in the articulation space (dynamic view). In literature, articulation priors are considered particularly in applied problems involving motion and tracking [121] and pose configurations are mostly represented as data determined manifolds embedded in high dimensional measurement spaces [14, 121, 123].

In this section, we adopt the static view and discuss very basic inferences that can be made in the articulation space. We restrict our discussion to a set of twenty human shapes with different postures given in Figure 5.27. The main reason for selecting this data set is

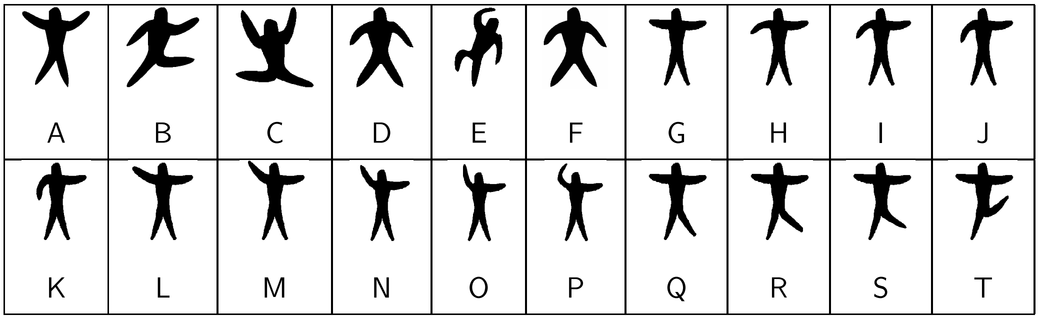


Figure 5.27: Set of 20 human silhouettes used in the experiments.

that the sections as captured by protrusions and their movements are intuitive and one can judge relative closeness of two different postures, such as arms being up or down. Secondly, human figure provides a rich data set since each figure contains five flexible sections to cover all possible situations that may arise in terms of degeneracies.

Once the deformable shape sections are extracted from disconnected skeletons of the training set and then mapped to $\mathcal{LA}$ coordinates, the distributions of the points can serve as prior knowledge about possible degrees of articulation in each section. In this study, we model these distributions as Gaussians although one can also employ a non-parametric approach.

The collected statistics about part articulations for the shape set $\mathcal{S}_1 = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}\}$ is illustrated in Figure 5.28. The largest ellipse corresponds to the distribution of arm1 coordinates where the postural variability is the highest whereas the very small ellipse shown in the square window corresponds to the distribution of the head coordinates practically having no variability at all. The individual plots of each part are provided in Figure 5.29. One can observe that similar articulations of a part are expressed with nearby points in the corresponding articulation space. For example, the articulations of arm1 for shapes B and D and the articulations of leg2 of shapes A and B are close to each other.

When we consider the set $\mathcal{S}_2 = \{\mathsf{A}, .., \mathsf{P}\}$ and concentrate only on the degrees of articulation in arm1, the distribution of this articulated section becomes as shown in Figure 5.30. Table 5.3 shows the pairwise similarities between articulations of arm1, each computed with the similarity measure given in Equation 5.3. Observe that the similar configurations have relatively high similarity scores.

$$sim(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + d^2(\mathbf{x}, \mathbf{y})/\epsilon^2} \tag{5.3}$$

where $\mathbf{x}$, $\mathbf{y}$ are $\mathcal{LA}$ coordinates of two articulations, $d(\mathbf{x}, \mathbf{y})$ is the Mahalanobis distance between $\mathbf{x}$ and $\mathbf{y}$ measured using the estimated covariance matrix and ϵ is a scalar which determines the decay rate of the similarity and is taken as 4 in the following experiments.

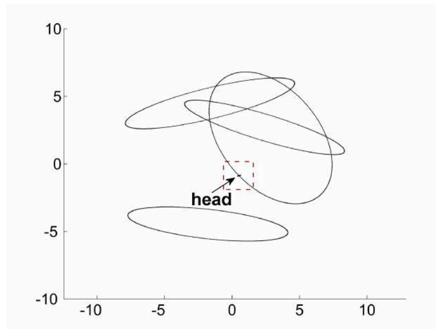


Figure 5.28: Collected statistics of each part for shape set $\mathcal{S}_1 = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}\}$ in the articulation space. The ellipses are drawn at 2σ. The largest one corresponds to arm1 and the small dot corresponds to head.

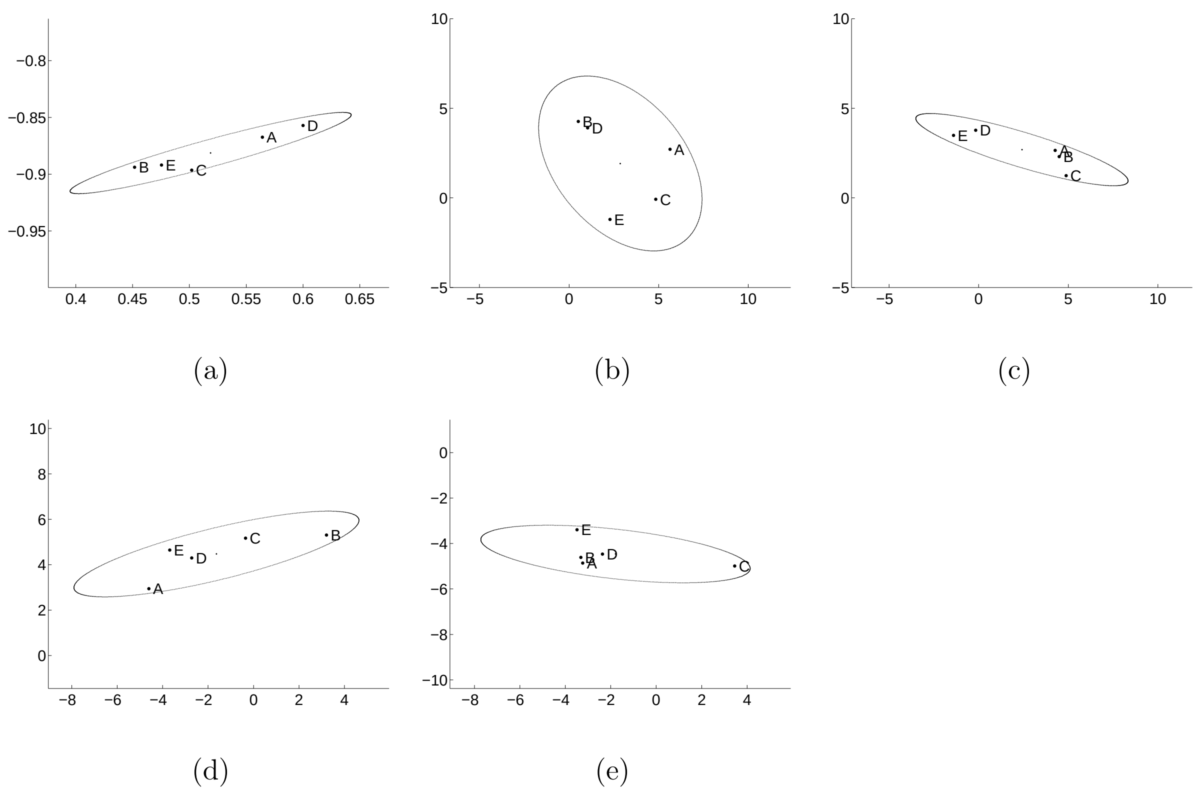


Figure 5.29: For the shape set $\mathcal{S}_1 = \{\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}, \mathsf{E}\}$, the distributions of (a) head, (b) arm1, (c)arm2, (d) leg1 and (e) leg2 in articulation space. Note that the scales are not equal.

Table 5.3: The pairwise similarities between articulations of $\mathsf{arm1}$ in $\mathcal{S}_2 = \{\mathsf{A}, .., \mathsf{P}\}$.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	1.000	0.158	0.249	0.186	0.084	0.236	0.755	0.434	0.337	0.197	0.125	0.928	0.824	0.622	0.468	0.133
B	×	1.000	0.161	0.945	0.128	0.797	0.247	0.180	0.455	0.900	0.873	0.160	0.162	0.144	0.103	0.113
C	×	×	1.000	0.192	0.262	0.212	0.304	0.124	0.175	0.171	0.147	0.203	0.375	0.468	0.358	0.597
D	×	×	×	1.000	0.139	0.919	0.300	0.199	0.519	0.939	0.766	0.186	0.194	0.172	0.120	0.129
E	×	×	×	×	1.000	0.131	0.106	0.060	0.091	0.117	0.140	0.076	0.106	0.119	0.099	0.415
F	×	×	×	×	×	1.000	0.394	0.250	0.673	0.927	0.581	0.237	0.241	0.208	0.141	0.133
G	×	×	×	×	×	×	1.000	0.432	0.529	0.316	0.189	0.707	0.733	0.558	0.352	0.156
H	×	×	×	×	×	×	×	1.000	0.479	0.236	0.137	0.564	0.294	0.220	0.170	0.076
I	×	×	×	×	×	×	×	×	1.000	0.633	0.315	0.372	0.294	0.232	0.159	0.106
J	×	×	×	×	×	×	×	×	×	1.000	0.662	0.203	0.196	0.169	0.118	0.113
K	×	×	×	×	×	×	×	×	×	×	1.000	0.125	0.132	0.121	0.088	0.111
L	×	×	×	×	×	×	×	×	×	×	×	1.000	0.650	0.472	0.361	0.113
M	×	×	×	×	×	×	×	×	×	×	×	×	1.000	0.899	0.638	0.185
N	×	×	×	×	×	×	×	×	×	×	×	×	×	1.000	0.790	0.229
O	×	×	×	×	×	×	×	×	×	×	×	×	×	×	1.000	0.203
P	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	1.000

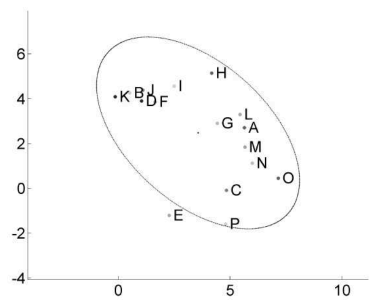
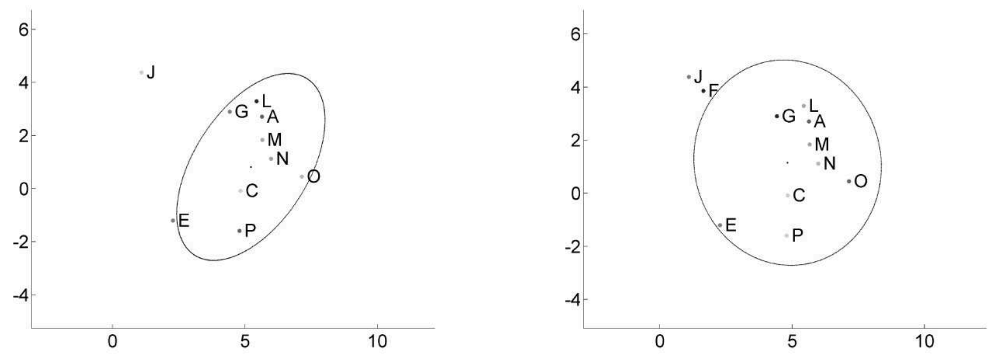


Figure 5.30: The distribution of articulations of $\mathsf{arm1}$ in the shape set $\mathcal{S}_2 = \{\mathsf{A}, .., \mathsf{P}\}$. Observe that in semi-local coordinate frame, $\mathcal{LA}$ coordinate of $\mathsf{arm1}$ belonging to shape G (straight arm posture) is close to $x = y$ line.

Next, we consider the set $\mathcal{S}_3 = \{\mathsf{A}, \mathsf{C}, \mathsf{E}, \mathsf{L}, \mathsf{M}, \mathsf{N}, \mathsf{O}, \mathsf{P}\}$ which contains only the shapes having their $\mathsf{arm1}$s up (Figure 5.31(a)). In this particular case, the past experience is incomplete, therefore when a human shape whose $\mathsf{arm1}$ has a different posture is encountered, it must be considered as impossible. The Mahalanobis distances from $\mathsf{arm1}$ of shapes J (arm down) and G (horizontal arm) to the distribution reflect this fact with the values 5.008 and 1.791 respectively. We can expand our knowledge about $\mathsf{arm1}$ by inserting the instance F where $\mathsf{arm1}$ is down. New distribution covers the cases where $\mathsf{arm1}$ is down (Figure 5.31(b)).

As expected, the distances of configurations for shapes J and G are reduced to 2.638 and 0.923, respectively.



(a) (b)

Figure 5.31: The distributions of articulations of $\mathsf{arm1}$ in the shape sets (a) $\mathcal{S}_3 = \{\mathsf{A},\mathsf{C},\mathsf{E},\mathsf{L},\mathsf{M},\mathsf{N},\mathsf{O},\mathsf{P}\}$ (b) $\overline{\mathcal{S}_3} = \{\mathsf{A},\mathsf{C},\mathsf{E},\mathsf{F},\mathsf{L},\mathsf{M},\mathsf{N},\mathsf{O},\mathsf{P}\}$. Notice the change in the distribution when shape F ($\mathsf{arm1}$ down) is added.

Similar inferences are also valid for the degenerate cases. When the articulation distribution of $\mathsf{leg1}$ for the shape set $\mathcal{S}_4 = \{\mathsf{A},\mathsf{C},\mathsf{D},\mathsf{E},\mathsf{F},\mathsf{G},\mathsf{Q},\mathsf{R},\mathsf{S}\}$ is considered, the articulation of shape T can be regarded as impossible (see Figure 5.32) since the Mahalanobis distance from it to the distribution is very high (6.290) compared to the others.

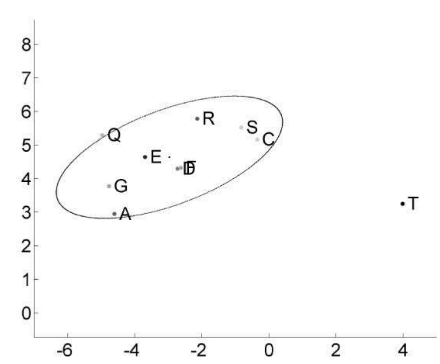


Figure 5.32: The distribution of articulation of $\mathsf{leg1}$ in the set $\mathcal{S}_4 = \{\mathsf{A},\mathsf{C},\mathsf{D},\mathsf{E},\mathsf{F},\mathsf{G},\mathsf{Q},\mathsf{R},\mathsf{S}\}$. The articulation of shape T is far distant from the distribution.

5.2.4 Incorporating Contextual Sensitivity to Articulations into Skeletal Shape Matching

We now utilize the developed ideas to incorporate contextual sensitivity to articulations into the skeletal shape matching method of Aslan and Tari [3]. But note that our approach is in fact independent of the matching method. However, it depends on the disconnected skeleton representation in order to construct the semi-local coordinate frame. Recall that in the matching method of Aslan and Tari, the total similarity of two shapes is determined by the weighted sum of matched branch pairs, in which the weights are the normalized lengths of the branches (Equation 4.1). As a refinement step, we propose to reevaluate the overall matching score by integrating the measurements in the articulation space as additional weights. But as in Section 5.1.3, it is sufficient to revise only the weights of the positive skeleton branches, because these branches actually represent the deformable sections of shapes.

Consider the matching between shapes A and T (Figure 5.33). In the syntactic level, these two shapes are found to be similar with a score of 0.826. However, when we take into consideration the prior knowledge about likely articulations obtained from the set $\mathcal{S}_5 = \{\mathsf{A}, .., \mathsf{S}\}$, this score is reduced to 0.458. This updated matching score reflects the difference in the posture of $\mathsf{leg1}$s of the given shapes.

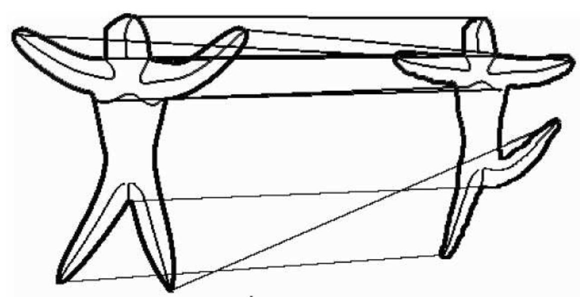


Figure 5.33: Matching result of two human shapes. The original matching score is 0.826 but it reduces to 0.458 in the context of articulations in $\mathcal{S}_5 = \{\mathsf{A}, .., \mathsf{S}\}$.

Figure 5.34 illustrates the effect of contextual sensitivity to articulations on some example queries when the prior knowledge is expressed with the shape set $\mathcal{S}_5$. For each query, we list the five best matches with and without contextual sensitivity. See how the five best matches to shape G are re-ordered. Also notice the drastic change in the best match list of shape A.

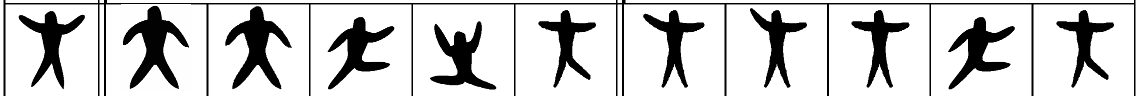
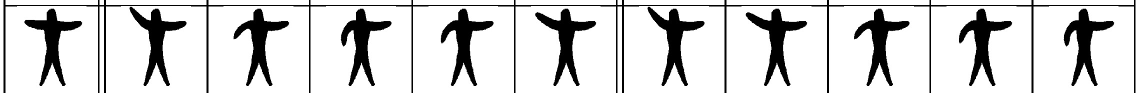
Query	5 best matches without feedback					5 best matches without feedback				
										
A	F	D	B	C	R	L	M	G	B	R
	0.975	0.962	0.940	0.917	0.890	0.582	0.559	0.550	0.532	0.515
										
G	M	I	K	J	L	M	L	I	J	K
	0.999	0.996	0.995	0.994	0.991	0.945	0.932	0.901	0.869	0.849

Figure 5.34: Some query results with and without contextual sensitivity.

5.3 Summary and Discussions

In this chapter, the effects of context, in particular the function of semantic category knowledge, in shape dissimilarity computation is investigated. In doing so, we adopt the interpretation of Tversky about the asymmetric nature of human (dis)similarity judgements [113], in which different roles are assigned to the shapes in consideration and extend or refine the shape matching methods of Aslan and Tari [3] and Baseski [7] accordingly. In the proposed versions of these algorithms, a query shape is being compared with a database shape that belongs to a familiar category. Hence, the knowledge about the category of the database shape guides the dissimilarity computation. Note that in shape matching literature, the classic view is to define shape (dis)similarity by means of metrics, whereas in our formulation, the resulting (dis)similarity measures are asymmetric due to influence of category knowledge.

Our motivation in extending the tree edit distance-based algorithm of Baseski [7] is to utilize the extra information inferred from all the members of the category of which the database shape in comparison belongs. Availability of the knowledge about the category of the database shape allows us both to modify the importance of extracted skeleton branches and the distances between attributes in the matching process. Early experimental results demonstrate that incorporating category knowledge into matching drastically improve the performance of the originating matching method.

The key to our category-influenced matching algorithm is the category tree data structure which we construct as a union of shape trees belonging to the same category. In the context of skeletal shape matching, forming a union of tree representations was previously addressed by Torsello and Hancock [109]. However, in contrast to their way of utilizing tree unions that they embed shock trees into a vector space, we utilize category trees in order to provide a context for each primitive of a database shape. Moreover, unlike the case in forming the union shock trees, our constructions naturally result in tree structures as a consequence of depth-1 property of shape trees. In fact, category trees are also depth-1 trees. Recently, Torsello and Hancock utilized tree unions in an unsupervised setting to learn shape categories [111]. In Chapter 6, we will make use of category trees in a supervised shape categorization framework.

As noted before in Chapter 2, disconnected skeleton representation does not carry any information about the boundary details and in this regard, we propose to obtain approximate radius functions from TSP surfaces to enrich the disconnected skeleton descriptions. In Section 5.1.5, we make use of these enriched descriptions to incorporate boundary similarity into our category-influenced matching method. We first employed the approach in [127] and model a low-dimensional linear deformation space for each positive branch which appears in a shape category and then we develop a refinement procedure to revise the overall dissimilarity score by considering the boundary similarity of matched positive branches in the corresponding deformation spaces.

The widespread use of skeletal representations in visual shape recognition lies mostly in the fact that they are insensitive to articulation of parts. However, as demonstrated in [1], insensitivity to articulations becomes undesired in some circumstances which actually involves prior knowledge about the degree of possible articulations to come up with the correct matching result. This contextual sensitivity to articulations raises another need for the incorporation of category knowledge into shape matching process. In this regard, we propose a novel part-centered coordinate frame constructed via the disconnected skeleton representation which provides a representation space for making inferences about articulations, in which similar articulations or bendings yield closer coordinates. Using illustrative examples, we demonstrated that it is possible to build articulation priors and incorporate them to the matching method of Aslan and Tari [3] to arrive at an enriched skeletal matching scheme.

CHAPTER 6

SIMILARITY-BASED CLASSIFICATION OF SHAPES USING DISCONNECTED SKELETONS

Classification (or *categorization*) is among the most primary cognitive processes, described as the ability to group a very large (or possibly infinite) number of similar objects into a relatively small number of classes (or categories) and to identify a novel object as a member of a particular class. From the perspective of information retrieval, having a classification mechanism is vital because organizing knowledge in a structured way offers efficient and economical use of limited resources when reasoning about a novel object. Moreover, the knowledge about the category of an object enables making inferences about unobserved characteristics of that object [46, 77, 82]. Lastly, it is important to note that the notion of similarity plays a central role in classification, especially in explaining *generalization* from the knowledge about previously encountered objects of the same category.

Visual object classification is one of the fundamental tasks of both human and computer vision systems. This ability, as in recognizing a novel object, requires the integration of information about the properties of the object such as shape, size, color, texture. In this regard, it is widely believed that shape information provides an informative representation that is invariant to changes in the viewpoint that objects can be identified and classified solely based on their shapes (Figure 6.1) [62]. As first demonstrated by Landau and her colleagues [49], psychological experiments suggest that infants have a tendency to name objects based on the resemblance in their shapes rather than other perceptual properties like size, color or texture, and this phenomenon is called *shape bias*.

In this chapter, we start with some theoretical preliminaries on classification and sim-

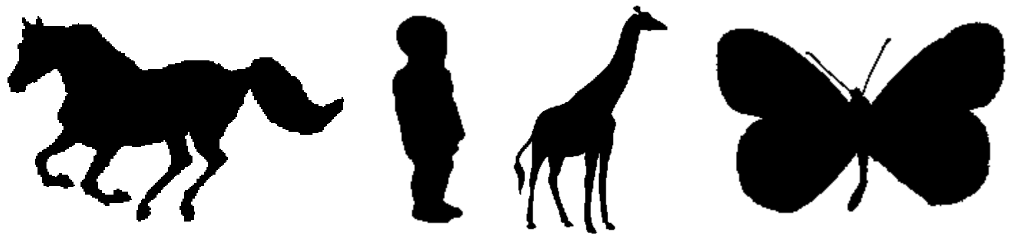


Figure 6.1: Objects can be easily classified solely based on their shapes.

ilarity. Next, we review some skeleton based shape classification methods proposed in the literature. In the following section, we present our shape classification algorithm, which is based on disconnected skeleton representation of Aslan and Tari [3].

6.1 Classification and Similarity

6.1.1 Supervised vs. Unsupervised Classification

In literature, classification studies can be divided into two broad groups, *supervised* and *unsupervised* classification, based on two different strategies used in learning. In supervised category learning, a subject or a machine learns to discriminate between different categories while members of the categories are repeatedly provided with a category label and feedback is given accordingly about their classification accuracy [63, 66, 45, 23]. In computational terms, supervised learning approach can be interpreted as a *function approximation* process with a good generalization capability [23, 99, 116]. On the other hand, in unsupervised category learning, no explicit feedback or even no information about existing categories is provided, and the objective of the subject or the machine is to find out the hidden category structure by himself or itself.

6.1.2 Theoretical Approaches to Classification

Classification studies in the literature can be grouped as the *classical*, *prototype* and *exemplar* approaches based on how category knowledge is represented [100]:

In *the classical approach* or so-called the Aristotelian view, it is believed that every category is constructed by a set of *essential features*, each of which is necessary and sufficient in defining that particular category (*e.g.*, the defining properties of bird category can be listed as `can fly`, `has feathers`, `has wings` and `can sing`). According to this line of explanation, the boundaries of categories are all well-defined that a novel object is a member of a category if and only if it satisfies all the characteristics of that category. Hence, there is no notion of a membership rating. In contrast to other two approaches, each member of a category is an equally representative example of that category.

In *the prototype approach*, the concept of prototype plays the central role in categorization where each category is a fuzzy set that is constructed as groupings of objects similar to the prototype of that category [82]. A prototype can be considered as either a summary representation formed by abstracting over previously encountered examples of a category (*e.g.* suppose instances of a category are represented with n-dimensional feature vectors then a prototype of the category can be easily formed by averaging over all these feature vectors) or just a typical example of the category (*e.g.* think about a robin or a sparrow representing the bird category). When a novel object is encountered, classification is performed by comparing it with the prototypes of categories. The instance is then assigned to the most similar category if the corresponding similarity is found to be greater than a threshold value. This kind of decision making process can utilize Luce's well-known *Choice Rule* [59, 60]: Let $\mathcal{S}(x, A)$ be the similarity between the newly encountered instance x and the category A, the probability of membership of x to category A is calculated as:

$$P(x \in A) = \frac{\mathcal{S}(x, A)}{\mathcal{S}(x, A) + \sum_{B \neq A} \mathcal{S}(x, B)} \tag{6.1}$$

In *the exemplar approach*, each category is believed to be constructed by not as a single prototype, but as a collection of exemplars, referring to the memory traces of some previously encountered examples of a category, and according to this view, classification depends on the similarities to the stored exemplars. For example, consider the following simple classification procedure. In classifying a newly encountered object, for each category, we sum up all the similarities between the novel object and the exemplars of the category (Equation 6.2). Following to that, the object is assigned to the category having the greatest cumulative similarity value. Here, Luce's choice rule can also be used, as in [63, 66].

$$\mathcal{S}(x, A) = \sum_{y \in A} \mathcal{S}(x, y) \tag{6.2}$$

The main difference between prototype and exemplar approaches to classification lies in how they define classification: whether classification relies on an abstraction over previously encountered objects, or it is a function of these stored examples [83]. In fact, these two approaches can be considered as the two extremes in a continuum. When only the most typical exemplar of a category defines the category, the exemplar-based model becomes equivalent to a prototype-based one. Similarly, the exemplars defining a category might not refer to actual copies of encountered examples but they might involve abstraction. In short, one can end up with a different classification model by combining these two alternative approaches, as demonstrated in [66, 45, 117].

6.1.3 Models of Similarity

As clearly discussed above, the notion of similarity is at the heart of classification models regardless of the approach employed. We now review two major models of similarity, which are *geometric*, *feature-based* models.

Geometric models of similarity treat objects as points in a multidimensional perceptual space where the similarity between two objects is inversely related to the distance between their representations in the perceptual space:

$$\mathcal{S}(x, y) \propto \mathcal{D}(x, y)^{-1} \text{ where } \mathcal{D}(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{6.3}$$

where n denotes the number of features and p is a positive real number.

Note that the underlying assumption behind geometric models is that similarity judgments satisfy all three metric axioms, which are *minimality*, *symmetry* and *triangle inequality*. We mentioned in the beginning of Chapter 5 that this is in fact a false proposition though.

Feature-based models of similarity are set theoretic models where each object is represented as a set of features and the similarity between two objects is a function of their common and distinguishing features. An early and well-known feature-based model is the *Contrast Model* of Tversky [113] in which the similarity is calculated with the following linear combination formula:

$$\mathcal{S}(x, y) = af(x \cap y) - bf(x - y) - cf(y - x) \tag{6.4}$$

where $(x \cap y)$ is the set of features shared by x and y, $(x - y)$ and $(x - y)$ are the disjoint sets of distinctive features of x and y, respectively, a, b and c are positive real numbers, and the function f is mostly defined as an additive function.

In [113], Tversky suggested that the rationale behind the non-metric nature of human similarity judgments was the context of comparison and as we investigated in Chapter 5, in the setting of his similarity model, he proposed that context might influence the saliency of features.

Comparing objects having a hierarchical structure (*e.g.* strings, shapes, scenes) is challenging for both geometric or feature-based models of similarity. In this regard, there are some alternative models proposed in the literature, each of which depends on finding correspondences between parts of objects in comparison. In *alignment-based models of similarity* [34], common features belonging to matched parts affect the similarity computation more, as compared to feature-based models. However, note that this leads to a *chicken and egg dilemma* since matching also depends on features common to parts of objects. In *transformation-based models of similarity* [37], the comparison of two objects involves transforming one into another and the similarity value is inversely proportional to the total cost of transformation operations.

6.1.4 Generalization

Theoretical approaches to classification, especially exemplar approaches, rely on similarity to account for *generalization* from past experience to classify novel instances, and in this regard, it is important to note that the generalization capability of a classification method is critical to its performance.

In [93], Shepard formulated what he referred to as *universal law of generalization*, according to which the probability of generalization falls of exponentially with the perceptual distance between a previously encountered example and the novel one, or in other words similarity is an exponential decay function of distance in perceptual space. In [108], Tenenbaum and Griffiths presented a Bayesian-based extension of Shepard's work to concept learning that could explain generalization from multiple examples. Morever, the authors showed that their proposed generalization function is closely related to Tversky's set theoretic approach to similarity [113].

6.2 Related Works On Shape Classification Using Skeletons

As noted before, shape information is an important clue for visual perception as objects can be recognized and classified solely based on their shapes. However, in computational terms, visual shape recognition and classification is rather challenging because objects show great variability in their shapes due to visual transformations such as articulation and deformation of parts, occlusion and changes in viewpoint. Shape skeletons, in spite of their instability issues, proved themselves more effective than boundary based shape representations since they can capture intuitive part structure of shapes.

In Chapters 3-5, we investigated the use of skeletons for shape recognition, where we first reviewed some of the existing skeleton based shape matching methods [1, 7, 32, 61, 74, 75, 90, 98, 110, 127] and then developed a number of ways to incorporate semantic category knowledge into the matching process in order to improve the performance of the methods of Aslan and Tari [3] and Baseski [7]. In this section, we investigate the use of skeletons for shape classification. Despite their common use in shape recognition, the potential of shape skeletons for shape classification has not been investigated much. This is partly because the structure of skeletons, *i.e.* the interrelationships between skeleton branches, are conventionally represented by graphs or trees, and in this regard, the variety of classification tools proposed in structural pattern recognition are not as diverse as those available in statistical pattern recognition.

The most common approach in structural pattern recognition is to use *k-nearest neighbor (kNN)* method [23]. However, despite its conceptual simplicity and asymptotic behavior (when $k = 1$ and the size of training data approaches to infinity, the error rate of kNN classifier is bounded by twice the Bayes error rate), classifying a query shape based on a naive kNN classifier involves first computing the distances between the skeleton of the query shape and all the skeletons of database shapes and hence it is computationally inefficient. To reduce this computational burden, a variety of *indexing* studies have been proposed, each of which attempt to organize a metric space for fast searching (For a survey, see [20]). The typical approach employed in these studies is to eliminate certain distance computations using triangle inequality wherein the database shapes are clustered into groups based on their distances to some prototypical objects. However, these methods face with *the curse of dimensionality*, which means their performance deteriorate as the dimensionality of the metric space increases. In this regard, an alternative approach is to encode the topological structure of graphs into low dimensional vector spaces [22, 94].

In the following sections, we will review some skeleton based shape classification methods proposed in the literature. There are some conceptual issues worth mentioning about these methods. First, all of these are supervised classification methods, *i.e.* they both involve a training phase. Although there are some interesting unsupervised methods applied on skeleton based shape classification, *e.g.* [111], they are omitted here simply because our classification method, which will be presented in the next section, is also a supervised method. Next, all of the reviewed methods (may be except the method of Sebastian *et al.* [88]) are specifically proposed for classification of shapes based on skeletons, meaning that they are not general methods, and depend on the underlying skeletal representation of shapes.

6.2.1 FORMS [127]

In Section 3.1, we reviewed skeleton based shape matching method of Zhu and Yuille [127]. Recall that the approach employed in this study differs from the others in that recognition of a shape is carried via bidirectional (bottom-up and top-down) procedure to cope with the instabilities of shape skeleton. That is, the skeleton of the query shape is initially extracted in a bottom-up manner, segmented into parts and expressed as a graph. But the extracted skeleton is subject to change based on the information coming from the matching process.

In the matching process, the skeleton graphs of the query and the database shapes are not directly compared but instead they are first matched against a skeleton model associated with the database shape in comparison and their extracted skeletons are revised according to the matching residue. This skeleton model is a prototypical skeleton graph which represents a common skeleton structure for a category of shapes and in addition provides for each shape part a low dimensional representation space, which is formed by applying Principle Component Analysis (PCA) on the observed deformations. Once the query and the database shapes are matched with the associated skeleton model, the distance between each matched pair of parts are computed as in the corresponding PCA subspace. However, note that the distance computation also depends on some other measurable properties such as the area, the radius of the maximum circles of joint points.

As clearly seen from the above description, the approach of Zhu and Yuille for skeleton based shape matching in fact involves a phase that can be utilized as a prototype based classification method, as demonstrated in [127]. Remember that each shape part is represented as a set of skeletal attributes, which are the length (for worms), the angle specifying the angular region in which the deformations occur (for circles), its area and the radius of the maximum circles of joint points. The only difference in utilizing FORMS for shape classification is that the query shape is matched with a prototypical shape formed based on the skeleton model of the shape category where each part is represented by the average values of the corresponding skeletal attributes in the category.

As we emphasized in Section 3.1, the problem with this approach is that its success is mainly depends on how well the skeleton of the query shape can be extracted in terms of the specified generic shape primitives since these primitives are designed especially to represent articulated or deformable objects.

6.2.2 The Method of Sebastian *et al.* [88]

In [88], Sebastian *et al.* presented an exemplar based approach for shape classification, which is built upon shock graph matching of shapes [87] . It is essential to note that, as the title of the paper clearly indicates, the main focus of this study is not classification but to investigate several indexing strategies for fast retrieval of shapes, and in this regard, the proposed classification method is interpreted as an indexing method as well. That is, the function of the classification method is to eliminate unrelated shape comparisons in a retrieval task. The proposed method is explored only in terms of its indexing performance, but the details of exemplar selection process are missing in the text.

As noted above, the method of Sebastian *et al.* is developed as an exemplar based method, wherein each shape category is represented with a small set of representative members of that category. More formally, let Q be a query shape, and N be the number of categories in the database, then each category C_i, $i = 1 \ldots, N$, is expressed by the set $\mathbf{E_i} = \{E_i^k | k = 1, \ldots, N_i\}$ where E_i^k is one among N_i exemplars of the category C_i.

Given a query shape Q, the proposed classification method works as follows. First, the similarities between the query shape Q and the exemplars are computed using Equation 6.5, where $d(Q, E_i^k)$ corresponds to the distance between Q and E_i^k obtained with the edit distance based method in [87]. Followingly, a fuzzy membership value is assigned to Q for each category C_i as the sum of similarities of the query to all exemplars of C_i and the closest categories are identified accordingly. In regard to indexing, the cost of computing the edit distances between the query and the exemplars is much lower than matching the query shape against all of the shapes in database.

$$\mathcal{S}(Q, E_i^k) = exp\left(-\frac{d(Q, E_i^k)}{\min\limits_{\substack{i=1,\ldots,N \\ k=1,\ldots,N_i}} d(Q, E_i^k)} \right) \tag{6.5}$$

$$\nu(Q, C_i) = \sum_{k=1,\ldots,N_i} \mathcal{S}(Q, E_i^k) \tag{6.6}$$

In the method of Sebastian *et al.*, unlike the case in FORMS [127], each shape category is represented with a small set of typical examples of the shape category, not just a single prototype. This brings an advantage over the approach of Zhu and Yuille in that a single prototype model might be incapable of representing a shape category if the category is very diverse in itself. Moreover, the method seems scalable to larger sets. But it is important to note that the authors did not discuss the selection process of the optimal exemplars, which is in fact the most challenging issue for exemplar based approaches.

6.2.3 The Method of Yang *et al.* [122]

In [122], Yang *et al.* presented a skeleton based Bayesian framework for shape classification. The proposed classification method strongly relies on two previous studies. The first one is the skeleton pruning work of Bai *et al.* [6], which is based on contour partitioning via Discrete Curve Evolution [6] and the second one is the shape classification method of Sun an Super [104], which learns a Bayesian classifier to classify each shape category from the distributions of the boundary segments extracted from the database shapes that belong to that category.

Yang *et al.* utilize the pruning algorithm proposed in [6] to obtain stable descriptions of extracted skeletons. That is, as in the matching method of Bai *et al.* [5], a skeleton is not represented by its topological structure, but by the set of the shortest paths between every pair of end points of its branches, each of which is described with the corresponding radius functions, *i.e.* the sequence of radii of the maximal circles at the successive skeleton points on the path (Figure 3.18).

To classify shapes based on the proposed skeleton representation, the authors employed the approach in [122], where in the place of contour segments, they use the shortest skeleton paths in learning a Bayesian classifier for each shape category. Assuming that all paths within a shape category are equally probable, the probability of a shape belonging to a category is calculated as the sum of posterior probabilities of all the paths of the query shapes with the distributions of all paths in the category.

As the authors themselves reported, the main drawback of this method lies in the assumption that radius functions of the shortest skeleton paths of the query shape should be close to the ones in the category it belongs since the dissimilarity between two skeleton paths depends solely on the differences in the corresponding radius functions. Moreover, the method does not compute the correspondences between the skeleton paths of the query shape and the shape category. The lack of this matching information is a limitation that one cannot update the prior knowledge once the input shape is classified.

6.3 Similarity-Based Classification of Shapes using Disconnected Skeletons

In the previous section, we have reviewed some shape classification methods which are all based on skeletal representations of shapes. Among these studies, the method of Sebastian *et al.* [88] is conceptually different than the other two classification methods, (*i.e.* the methods of Zhu and Yuille [127] and Yang *et al.* [122]) as it is based on an exemplar approach. This provides robustness against outlier shapes within a category. On the other hand, in contrast to the case in other approaches, Sebastian *et al.* specifically didn't mention any details about the training phase, though the selection process of exemplar shapes to represent shape categories is a challenging problem in itself. In this section, keeping these issues in mind, we propose an novel shape classification algorithm which is based on disconnected skeleton representation of Aslan and Tari [3].

Recall that in Section 5.1.3, where we presented our *category-influenced shape matching* method, we formulated a straightforward procedure to match a shape tree with a category tree by exploiting the structural equivalence of shape and category trees. Although in the past we have utilized that procedure to form category trees in a dynamic way, it can also be considered as a naive prototype-based shape classification method, as we demonstrated in [8]. That is, one can classify a given shape by matching its shape tree against all of the category trees, each of which represents a specific shape category, and then assigning the category of the closest one. In the next section, we review this procedure in detail by investigating its use as a classification scheme.

In Section 6.3.2, we present a novel classification method, which is founded on the approach mentioned above. More specifically, we make a key change in the matching procedure and introduce additional mechanisms to come up with a more complete and robust classification method. In short, these contributions can be listed as follows. First, we replace the cost function of `change` operation with a new one, which is based on a generalization function proposed by Tenenbaum and Griffiths [108]. Second, we devise a recursive clustering strategy to form multiple category trees for each shape category so that our classification method doesn't suffer from the outlier shapes in a category. Finally, we employ a similarity-based representation paradigm [17, 24, 25, 26, 71] in which the computed distances to all category trees are embedded into a similarity space wherein the final decision is made.

6.3.1 Shape Classification By Matching Shape Trees with Category Trees

In Section 5.1.3, the structural equivalence of shape and category trees (*i.e.* each is a depth-1 tree) helped us to formulate a simple procedure to compare instances of these two structures. The proposed matching process is based on the category-influenced matching method we presented in Section 5.1 and can be summarized as follows:

Let $\mathcal{T}_1$ be the shape tree in comparison with the category tree $\mathcal{T}_\mathcal{C}$. In order to compare $\mathcal{T}_1$ with $\mathcal{T}_\mathcal{C}$, we construct a *mean shape tree* $\overline{\mathcal{T}_\mathcal{C}}$ from $\mathcal{T}_\mathcal{C}$ on the fly. It serves as a hypothetical shape tree representing the structure of $\mathcal{T}_\mathcal{C}$, *i.e.* it has equal number of nodes with $\mathcal{T}_\mathcal{C}$, each of which holds nothing but the ordinary average values of the attributes collected from the shape trees associated with $\mathcal{T}_\mathcal{C}$. Hence, since a mean shape tree constructed in this way is indistinguishable from a regular shape tree, one can directly apply the category-influenced matching algorithm to determine the correspondences between a shape tree and a category. Note that at the time of matching, $\overline{\mathcal{T}_\mathcal{C}}$ is associated with $\mathcal{T}_\mathcal{C}$, just like the shape trees of database shapes used in the formation of $\mathcal{T}_\mathcal{C}$, but this association has nothing to do with the content of $\mathcal{T}_\mathcal{C}$ and released after the matching.

Previously, we have utilized this matching procedure in dynamically constructing a category tree where the shape trees in the given set are progressively matched with the category tree in consideration following the steps described above. Apart from this use, the same procedure can also be employed as a straightforward shape classification method as follows. Observe that the mean shape tree that is formed from a specific category tree in fact functions as a prototypical representation of the corresponding shape category. This suggests that a shape can be easily classified by using *1-nearest neighbor* (1NN) approach. That is, the shape tree of the query shape is matched against all the category trees formed for each shape category in the database, and finally, it is classified as a member of the most similar category.

The main problem with the above classification scheme is that only a single category tree is formed for each shape category so there is an underlying assumption that the database shapes of the same category should all contain some common substructures in their shape tree descriptions. Hence, if there exist some outlier shapes or some subcategories within a category, the resulting category tree might contain misleading correspondences. To overcome this drawback the obvious solution is to form multiple category tree to represent a shape category. In Section 6.3.2, we present such a modification, in which we incorporate a recursive clustering step into the formation process of category trees.

6.3.2 The Proposed Classification Method

In the previous section, we have discussed a simple shape classification approach, which is based on the matching procedure for comparing a shape tree with a category tree. That is, to determine the category of a input shape, its shape tree is matched with each category tree, which is formed to represent a single shape category. Hence, the proposed method can be considered as a prototype-based classification algorithm. In this section, we built on this method, but employ an exemplar based approach instead to enhance its classification capability. As noted earlier, our newly proposed classification scheme involves three key changes.

Firstly, we replace the cost function of `change` operation with a new one, which is more appropriate for use in an exemplar based approach. Recall that in the original version, the skeletal attributes of the query shape is compared with the average attribute values of the category members, where, as described in Section 5.1.2, this comparison is influenced by the categorical context. Based on a generalization function proposed by Tenenbaum and Griffiths [108], our alternative cost function does not compute a difference score between the attributes of the query shape and the average attribute values but, instead, computes a membership value by considering the whole set of attributes.

Secondly, we incorporate a clustering mechanism into the procedure for forming category trees so that multiple category trees are formed for each shape category in a recursive manner. In this way, as we discuss previously, our new classification scheme doesn't suffer from the outlier shapes or a subcategory structure, which may exist in a category. In short, the proposed clustering strategy make use of both the pairwise distances between the category members and their distances to the corresponding category tree.

Lastly, we combine our previous classification strategy with a similarity-based approach to come up with a more effective classification scheme. The whole idea of our category influenced matching is to incorporate within category knowledge into distance computation. By further employing a similarity-based approach, we rise a level up in the *context* and model the between-category differences as well. That is, the computed distances to all category trees are embedded into a similarity space and a Support Vector Machine (SVM) classifier [116, 84] for each shape category is trained in this representation space.

An overview of our newly proposed classification scheme is given in Figure 6.2. Having a three-layered structure, the input to the system is the shape tree obtained from the disconnected skeleton of the query shape. The units in the hidden layer, which we refer to as *pseudo-exemplar units*, are associated with the category trees, representing the shape categories in the database. Each of these returns a similarity value between 0 and 1, which is computed as the negative exponential of the edit-distance between the input shape tree and the corresponding category tree. The units in the hidden layer are fully connected to the nodes in the output layer, each of which corresponds to a specific shape category and outputs a membership score for the input shape. The weights of the connections between the hidden layer and the output layer are learned using SVMs based on a training set of previously categorized shapes (*i.e.* the database shapes used in the formation of shape trees).

It is important to note that the proposed framework is an exemplar model since we form multiple category trees for each shape category. However, when the number of pseudo-exemplar units in the hidden layer is equal to the number of shape categories in the database

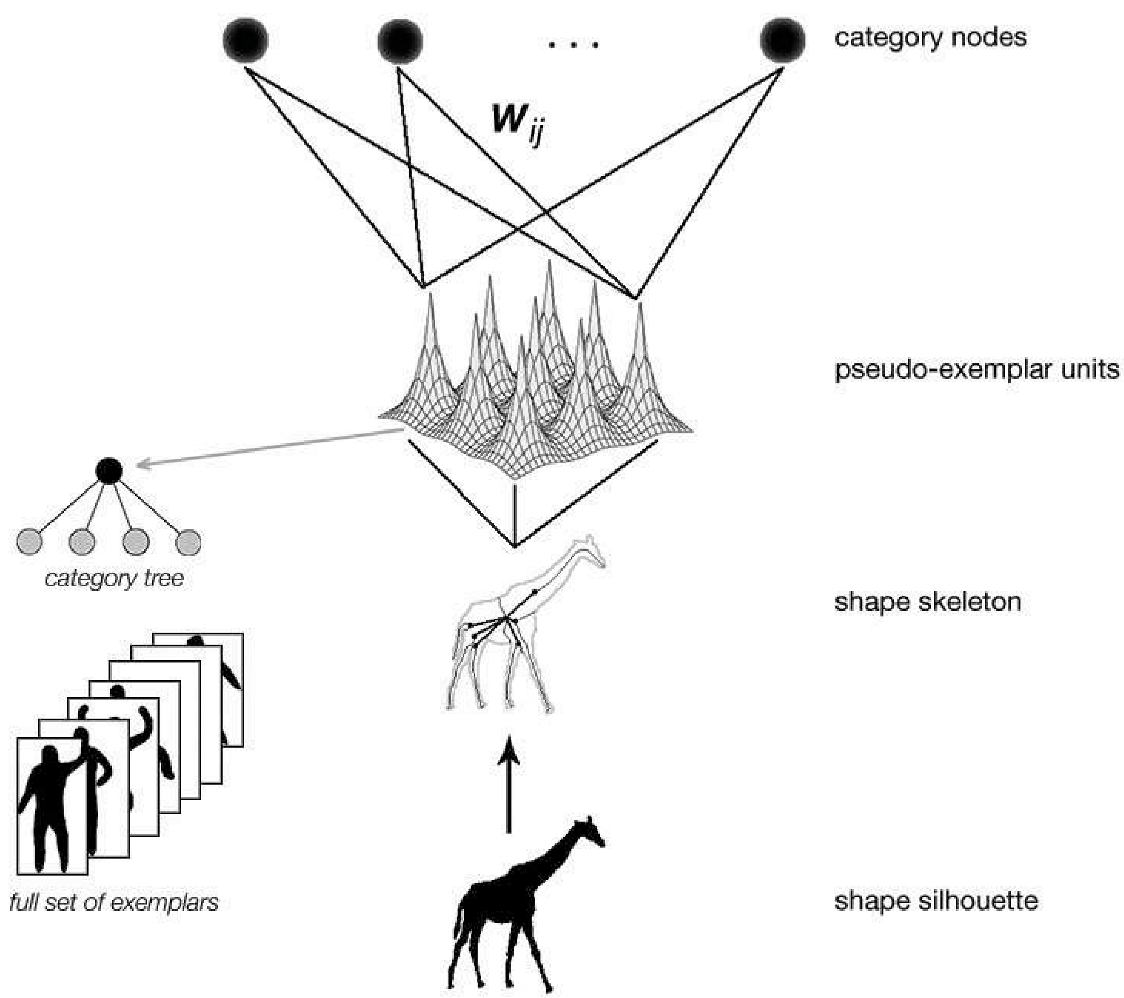


Figure 6.2: Overview of the proposed classification framework.

(*i.e.* a single category tree is formed for each category, as in our previous approach), the framework can be considered a prototype-based method. Moreover, the organization of our classification framework resembles a Radial Basis Function (RBF) network with Gaussian kernels. As demonstrated by Poggio *et al.*, these networks are biologically plausible and effective cognitive models of recognition and generalization [79, 78]. The definition of our framework differs from the definition of RBFs in two terms. First, the input to the system is not a vectorial representation but a structural representation. Second, the pseudo-exemplar units in our framework plays the role of Gaussian functions, returning a similarity value based on the tree-edit distances.

A New Cost Function for `change` Operation

In comparing the input shape with the category trees by using the category-influenced matching method, we utilize a new cost function for `change` operation, which depends on the generalization function proposed by Tenenbaum and Griffiths [108], instead of the generic cost function given in Figure 5.4. Based on a Bayesian formulation, this generalization function is an extension of Shepard's Universal Law of Generalization [93] to the cases of multiple examples and moreover, the function is also closely related to Tversky's set theoretic approach to similarity [113].

Let $\mathcal{T}_{query}$ be the input shape tree to be compared with the category tree $\mathcal{T}_{\mathcal{C}}$ and let $\overline{\mathcal{T}_{\mathcal{C}}}$ be the corresponding mean shape tree. Suppose $\mathcal{B}$ and u denote nodes in $\mathcal{T}_{\mathcal{C}}$ and $\mathcal{T}_{query}$, respectively, and $\mathcal{X} = \left\{x^{(1)}, x^{(2)}, \ldots, x^{(freq(\mathcal{B}))}\right\}$ be the set of corresponding nodes associated with the node $\mathcal{B}$. The following generalization function $g_{\mathcal{B}}(u)$ is derived by approximating the conditional probability $p(u \in \mathcal{B}|\mathcal{X})$ [108] where u corresponds to a leaf node in $\mathcal{T}_{query}$:

$$g_{\mathcal{B}}(u) = \frac{exp\left(-\left(\tilde{d}_r/\sigma_r + \tilde{d}_\theta/\sigma_\theta + \tilde{d}_l/\sigma_l\right)\right)}{\left[\left(1 + \frac{\tilde{d}_r}{(r_{max}-r_{min})}\right)\left(1 + \frac{\tilde{d}_\theta}{(\theta_{max}-\theta_{min})}\right)\left(1 + \frac{\tilde{d}_l}{(l_{max}-l_{min})}\right)\right]^{freq(\mathcal{B})-1}} \tag{6.7}$$

where the value of $\tilde{d}_i$ equals to 0 if u falls inside the observed range of corresponding attribute space spanned by $\mathcal{X}$. If this is not the case, its value is determined as the distance from u to the nearest example in $\mathcal{X}$ along the corresponding attribute space. σ_r, σ_θ and σ_l are the scaling parameters which are taken as $\sigma_r = 1$, $\sigma_\theta = 2\pi$ and $\sigma_l = 2$ in the experiments.

When the skeletal attributes specified in u moves away from the observed ranges for r, θ and l, similarity decreases based on an exponential decay function. Adaptive behavior of the generalization function with an increasing number of examples is demonstrated in Figure 6.3. For illustrative purposes we neglect the length attribute l and describe the effect in 2D, based

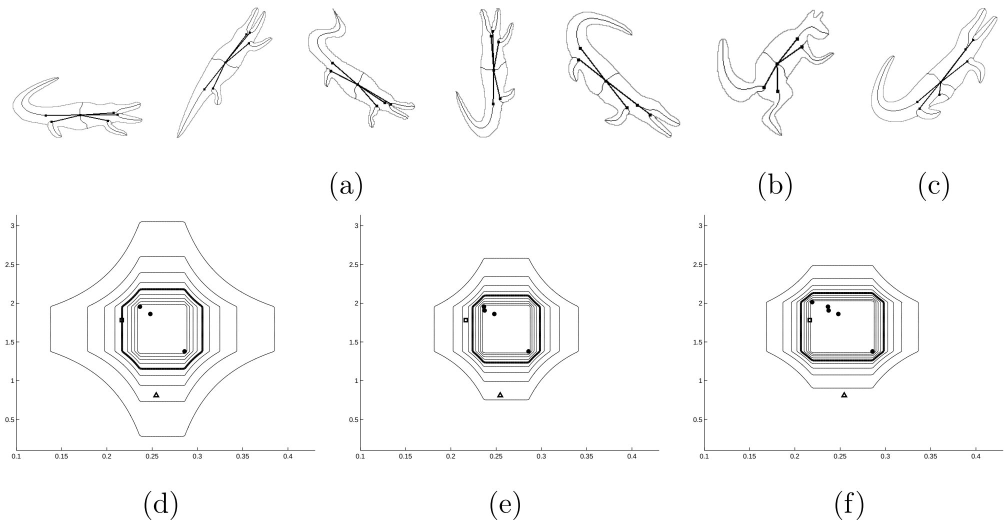


Figure 6.3: Adaptive behavior of generalization function with increasing number of examples. (a) Five examples from crocodile category (b)-(c) A squirrel and a crocodile shape used as query shapes (d)-(f) The behavior of the generalization function associated with the positive local symmetry branch corresponding the back leg in the crocodile category, when hypothesized from three, four and five examples respectively. The encountered examples are denoted with circular spots whereas triangle and square denote respective skeleton branches of the squirrel and the crocodile shapes. Contours show the value of generalization function in increments of 0.1 where thick ones correspond to $p(u \in \mathcal{B} \mid \mathcal{X}) = 0.5$. For illustrative purposes length attribute l is neglected and only the location attributes r and θ are considered.

on only r and θ attributes corresponding the location of disconnection points. As the number of encountered examples increases, the observed range enlarges to cover all examples and the degree of generalization is adjusted accordingly, describing the characteristics of the corresponding skeleton branch in the category more precisely.

Following this generalization function, we define the new cost function for `change` operation used in comparing a shape tree with a category tree as follows:

$$\texttt{change}^*(u, \mathcal{B}) = 1 - g_{\mathcal{B}}(u) \tag{6.8}$$

Multiple Category Trees For Each Shape Category

As we mentioned previously, our former classification approach is prone to outlier shapes or subcategories in a category, and the reason for that is we form a single category tree for each shape category in the database. When a category contains some outlier shapes or has two or more subcategories, the resulting category tree might have a wrong structure and contain misleading associations between the category members. Evidently, this might degrade the performance of the category-influenced matching when used in a classification task. In order to overcome this drawback in our new classification scheme, we incorporate a recursive clustering phase into the formation process of category trees so that multiple category trees are formed for each shape category. This also makes our new framework an exemplar-based classification method.

Recall the structure of our new classification scheme given in Figure 6.2 that once the category trees are formed, they provide us a representation set for similarity-based classifiers. In this sense, formation process of category trees share conceptual similarity with the selection strategies utilized in similarity-based classification studies, *e.g.* [21, 70, 72, 102, 124], which are used for choosing a reduced set of representative examples from a set of objects. However, note the difference that forming a category tree is more like generating a pseudo-exemplar than selecting actual category members to represent a category of shapes.

The revised version of the procedure for forming category trees involves an additional recursive clustering phase in order to construct multiple category tree for a shape category. The steps of the procedure is as follows. Given a set of shape trees $\mathfrak{T}$, a temporary category tree $\mathcal{T}_{\mathcal{C}}$ is formed either by using the static or dynamic formation procedure. Next, similar to approach in [17], the most representative member of the set, which is denoted by $\mathcal{T}_{median}$, is identified using Equation 6.9 as the shape tree that is most closest to $\mathcal{T}_{\mathcal{C}}$. Consequently, $\mathfrak{T}$ is partitioned into two groups according to the measure $S(\mathcal{T})$ given in Equation 6.10, which simply returns a similarity value between 0 and 1. The shape trees having $S(\mathcal{T}) > 0.5$ are removed from original set $\mathfrak{T}$ and used to form a new category tree representing a subcategory structure and this procedure is repeated recursively until $\mathfrak{T}$ contains no shape trees. These steps are summarized in Algorithm 1.

$$\mathcal{T}_{median} = \arg\max_{\mathcal{T}\in\mathfrak{T}} \sum_{i=1}^{n} exp\left(-dist(\mathcal{T}, \mathcal{T}_{\mathcal{C}})\right) \tag{6.9}$$

$$S(\mathcal{T}) = exp\left(-sim(\mathcal{T}, \mathcal{T}_{median}) \times \sum_{\substack{\mathcal{T}_i\in\mathfrak{T} \\ \mathcal{T}_i\neq\mathcal{T}}} \left(sim(\mathcal{T}, \mathcal{T}_i) - sim(\mathcal{T}_{median}, \mathcal{T}_i)\right)^2\right) \tag{6.10}$$

where $sim(\mathcal{T}_1, \mathcal{T}_2) = exp\left(-dist(\mathcal{T}_1, \mathcal{T}_2)\right)$.

Figure 6.4 shows the clustering results of the two different shape sets, which are obtained while forming the category trees by using the revised formation procedure. For each of these

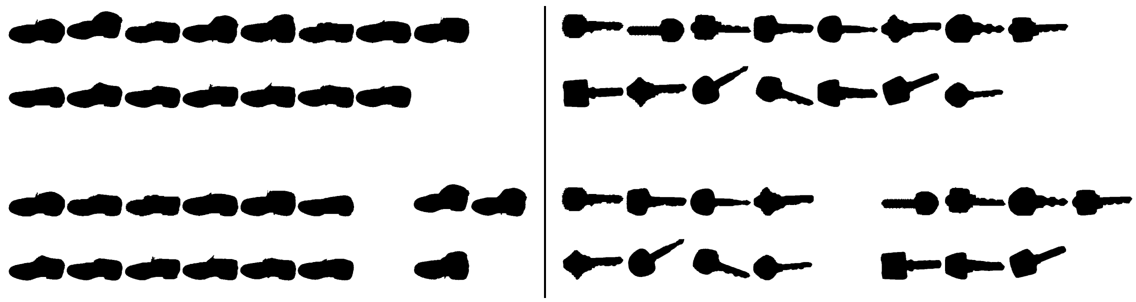


Figure 6.4: Some clustering results obtained while forming multiple category trees for the given set of shapes. While the top row shows the given shape sets, the bottom row presents the clustered shapes used in the formation of multiple category trees for the corresponding category.

shape sets, the procedure obtains two partitions in the end, and, accordingly, two separate category trees are formed for each one of the clusters.

Similarity-Based Classification of Shapes Using Support Vector Machines

In classical approaches to pattern recognition, objects are recognized or in terms of their inherent characteristic features, and hence the concept of *feature* is at the heart of the proposed techniques. To be more specific, in statistical pattern recognition, objects are expressed as a vector in a feature space, where each dimension represents a measurement of feature. In structural pattern recognition, however, objects are expressed by a syntactic scheme (*e.g.* a string, a graph or a tree), which represents the interrelationships between the structural features (primitives) of objects. A major challenge common to all recognition or classification systems, either based on the statistical or structural approaches, is the *feature selection problem*, *i.e.* determining an optimal set of essential features. This issue can be resolved only if one has domain knowledge about the problem at hand. However, in most of the real-world problems, gathering this information might be hard or even impossible.

Recently, similarity-based approaches begins to gain popularity in pattern recognition community. In this paradigm shift, as opposed to the traditional approaches, objects are represented by distances or similarities to some reference objects, not by features that are hard to choose [25, 24, 71, 26, 17]. Hence, similarity-based approaches require no domain-specific knowledge other than an distance or a similarity measure. Moreover, they have vital significance in structural pattern recognition, as they provide a natural embedding to representation spaces in which the learning tools already present in the statistical pattern recognition can be used to cluster or classify structurally represented object (*e.g.*, see [17, 109]). In this regard, they bring together the advantages of both the structural and the statistical pattern recognition, *i.e.* structural approaches provide richer representation of objects whereas the learning tools proposed in statistical pattern recognition literature are much diverse than those exist in structural pattern recognition.

In our classification framework, the reference set used for defining a similarity space is comprised of the set of the category trees which are constructed using the procedure in 6.3.2. Denoting this reference set by $\mathcal{R} = \left\{\mathcal{T}_{\mathcal{C}_1},\ \mathcal{T}_{\mathcal{C}_2},\ \ldots\ \mathcal{T}_{\mathcal{C}_M}\right\}$, where m is the total number of category trees formed for N number of categories ($M \geq N$), a given query shape can be embedded as a point in the similarity space by taking negative exponential of the vector of distances between the shape tree representation of a shape and the existing category trees:

$$\mathbf{S}(\mathcal{T}_{query}, \mathcal{R}) = exp\Big(-\big(d(\mathcal{T}_{query}, \mathcal{T}_{\mathcal{C}_1}),\ d(\mathcal{T}_{query}, \mathcal{T}_{\mathcal{C}_2}),\ \ldots,\ d(\mathcal{T}_{query}, \mathcal{T}_{\mathcal{C}_m})\big)\Big) \tag{6.11}$$

Algorithm 1 Recursively Forming Multiple Category Trees for Each Shape Category

Require: A set of shape trees $\mathfrak{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}, |\mathfrak{T}| > 0$

repeat
- $n \Leftarrow |\mathfrak{T}|$
- Construct a temporary category tree $\mathcal{T}_{\mathcal{C}}$ for the set of shape trees $\mathfrak{T}$
- $\mathcal{T}_{median} \Leftarrow \arg\max_{\mathcal{T}\in\mathfrak{T}} \sum_{i=1}^{n} sim(\mathcal{T}_i, \mathcal{T}_{\mathcal{C}})$ $\{$where $sim(\mathcal{T}_1, \mathcal{T}_2) = exp(-d(\mathcal{T}_1, \mathcal{T}_2)\}$
- $\{$Partitition $\mathfrak{T}$ into two subsets based on the distances to $\mathcal{T}_{median}\}$
- $\mathfrak{T}* \Leftarrow \emptyset$
- **for** $i = 0$ to n **do**
  - $\{$Iterate over all the shape trees in $\mathfrak{T}\}$
  - $S(\mathcal{T}_i) \Leftarrow exp\left(-sim(\mathcal{T}_i, \mathcal{T}_{median}) \times \sum_{\substack{\mathcal{T}_j\in\mathfrak{T}\\ \mathcal{T}_j\neq\mathcal{T}_i}} \Big(sim(\mathcal{T}_i, \mathcal{T}_j) - sim(\mathcal{T}_{median}, \mathcal{T}_j)\Big)^2\right)$
  - **if** $S(\mathcal{T}_i) > 0.5$ **then**
    - $\mathfrak{T}^* \Leftarrow \mathfrak{T}^* \bigcup \{\mathcal{T}_i\}$
  - **else**
    - $\mathfrak{T} \Leftarrow \mathfrak{T} - \{\mathcal{T}_i\}$
  - **end if**
- **end for**
- Construct the category tree for the set of shape trees $\mathfrak{T}^*$, $\mathfrak{T}^* \subseteq \mathfrak{T}$

until $\mathfrak{T} = \emptyset$

In this similarity space, the simplest classification strategy, which does not involve any learning at all, is to use 1NN rule. Note that this in fact equivalent to our former classification approach explained in Section 6.3.1, where the query shape is simply assigned to the category described by the category tree that is found most similar. As we mentioned previously, the main novelty of a similarity-based approach, however, lies in the ability to use conceptually more complex classification techniques. To demonstrate the idea, consider the two dimensional similarity space defined by the two category trees that are respectively formed to represent the category knowledge about the `camel` and `tulip` shapes shown in Figure 6.5(a). The similarity representations of some query shapes are given in Figure 6.5(b), where the dimensions corresponds to the negative exponentials of distances to the category trees of `camel` and `tulip` categories, respectively. Note that the performance of 1NN classifier is 70%, as three of the camel shapes are misclassified. Now, suppose that we have trained a linear classifier in the similarity space to discriminate between the shape categories. This hypothetical classifier is shown with a thick black line in the plot given in Figure 6.5(c), wherein the similarity representations of training shapes of `camel` and `tulip` categories are respectively displayed with blue and red points, and the similarity representations of query shapes are displayed by themselves. The classification rate of this classifier is 90%, where only a single shape is misclassified. At this point, it should be clear that employing a similarity-based approach can boost the performance of an underlying structural classification approach.

Recall the structure of our new classification scheme shown in Figure 6.2 that the function of the hidden layer is to compute a similarity representation of the input shape based on the distances between its shape tree and the category trees formed for each category in the database. In this similarity space, for each shape category, we train a separate SVM classifier with Gaussian kernel [116, 84] based on one-vs-all approach, where in the training phase, the similarity representation of members of that shape category is labeled as positive examples with (`1`) whereas the members of all other categories are labeled as negative examples with (`-1`).

In classifying a novel shape, the vector of computed similarities is fed to all of learned SVM classifiers, each outputting a scalar value. Then, a membership score for each category is obtained by normalizing these outputs according to Luce's choice rule [59, 60], as follows:

$$p(\mathcal{T}_{query} \in \mathcal{C}_i) = \frac{exp(\mathcal{Y}_i)}{\sum_{j=1}^{N} exp(\mathcal{Y}_j)} \tag{6.12}$$

where $\mathcal{T}_{query}$ and $\mathcal{Y}_i$ denote the shape tree of the input shape and the output of the SVM

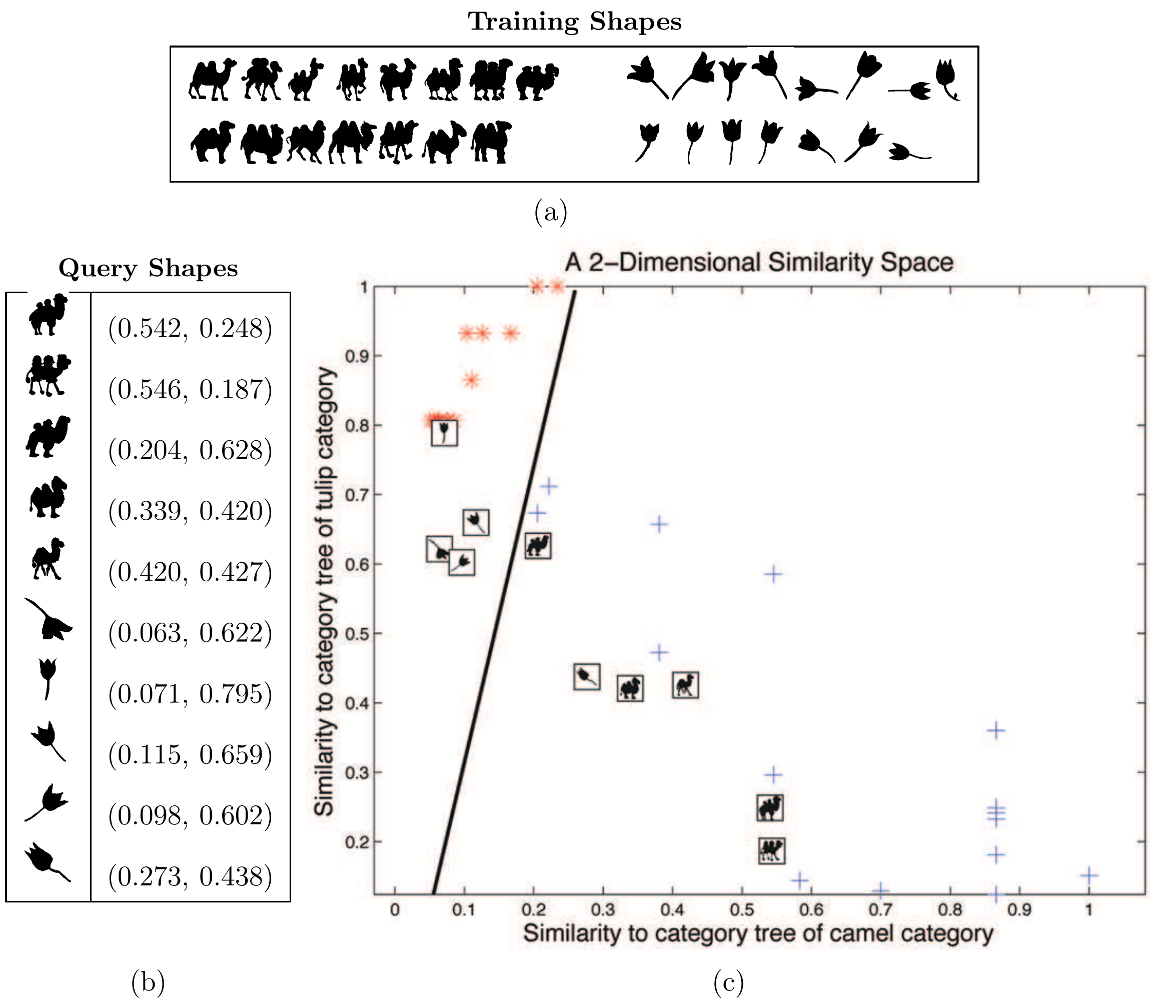


Figure 6.5: Example of a 2D similarity space and a linear classifier to discriminate between two shape categories. (a) Training set of shapes used in forming category trees to represent `camel` and `tulip` shape categories. (b) Query shapes and their similarity representation in the space defined by the distances to the formed category trees where the first dimension corresponds to the similarity to `tulip` category and the second dimension corresponds to the similarity to `camel` category. (c) The representation of training and query shapes plotted in the similarity space, where the training shapes of `camel` and `tulip` categories are shown with blue and red points, respectively and the query shapes are shown by themselves. Note that the classification performance is 90%, as compared to the 70% performance rate of 1NN classifier. (adapted from a figure provided by Pekalska *et al.* [72]).

classifier trained specifically for the i^{th} shape category, $\mathcal{C}_i$, respectively.

6.4 Experimental Results

In this section, we investigate the performance of the proposed shape classification framework using the database shown in Figure 5.6. In particular, we compare our *similarity-based* approach with the one that utilizes *1-nearest neighbor* (1NN) rule. Moreover, we analyze the classification accuracy when single or multiple category trees are formed for each shape category (*prototype* vs. *exemplar*) and the two alternative cost functions (`change` vs. `change`*) are used in computing the distances to the category trees. Our experimental setting is same with the one given in Section 5.1.4, where we randomly generate 100 partitions, each of which contains 750 shapes for training (15 examples from each shape category) and the remaining 250 shapes for testing. In each run, we first form the category trees, and afterwards use them as a reference set to define a similarity space wherein we train SVM classifiers for each category[1]. Following the training phase, we classify each shape in the test set by either using the proposed classification framework or 1NN rule.

Table 6.1 shows the average classification rates of each strategy and the corresponding standard deviations. The proposed similarity-based approach results in an average performance rate of 91.18% (when multiple category trees are formed for each shape category and the proposed `change`* cost function is used), boosting the 83.09% classification accuracy of 1NN classifier. In Figure 6.6, we also provide the average classification performances for each category. As an example case, for the sample partition given in Appendix A, the corresponding classification results are presented in Appendix D. As it can be clearly seen in Table 6.1, the similarity-based approach introduces considerable improvements in terms of accuracy when compared to the nearest-neighbor strategy. Moreover, as expected, forming multiple category trees for each category increases the performance of both 1NN and similarity based approaches, and the newly proposed `change`* cost function is more effective for the proposed classification framework.

As we mentioned previously, from an information retrieval perspective, one of the functions of classification is to eliminate unrelated comparisons in a retrieval task. In this regard, as a supplementary experiment, we evaluate the indexing performance of the proposed classification framework on our category-influenced matching method. In a retrieval task, we first perform classification to identify the top five most similar categories for the given shape.

[1] In the training phase, we use SVM^{light} [41] package and perform 5-fold cross validation, and automatically select the best values for the parameters γ and C, which respectively correspond to the radius of RBF kernel and the weighting factor for misclassification penalty.

Table 6.1: Average Classification Performances

		1NN Approach		*Similarity-Based Approach*	
		Classification Rate	*Std. Dev.*	*Classification Rate*	*Std. Dev.*
Prototype	`change`	79.07	2.43	87.19	3.88
	`change*`	77.13	2.59	87.80	1.95
Exemplar	`change`	84.18	2.22	88.65	2.67
	`change*`	83.09	2.99	91.18	1.66

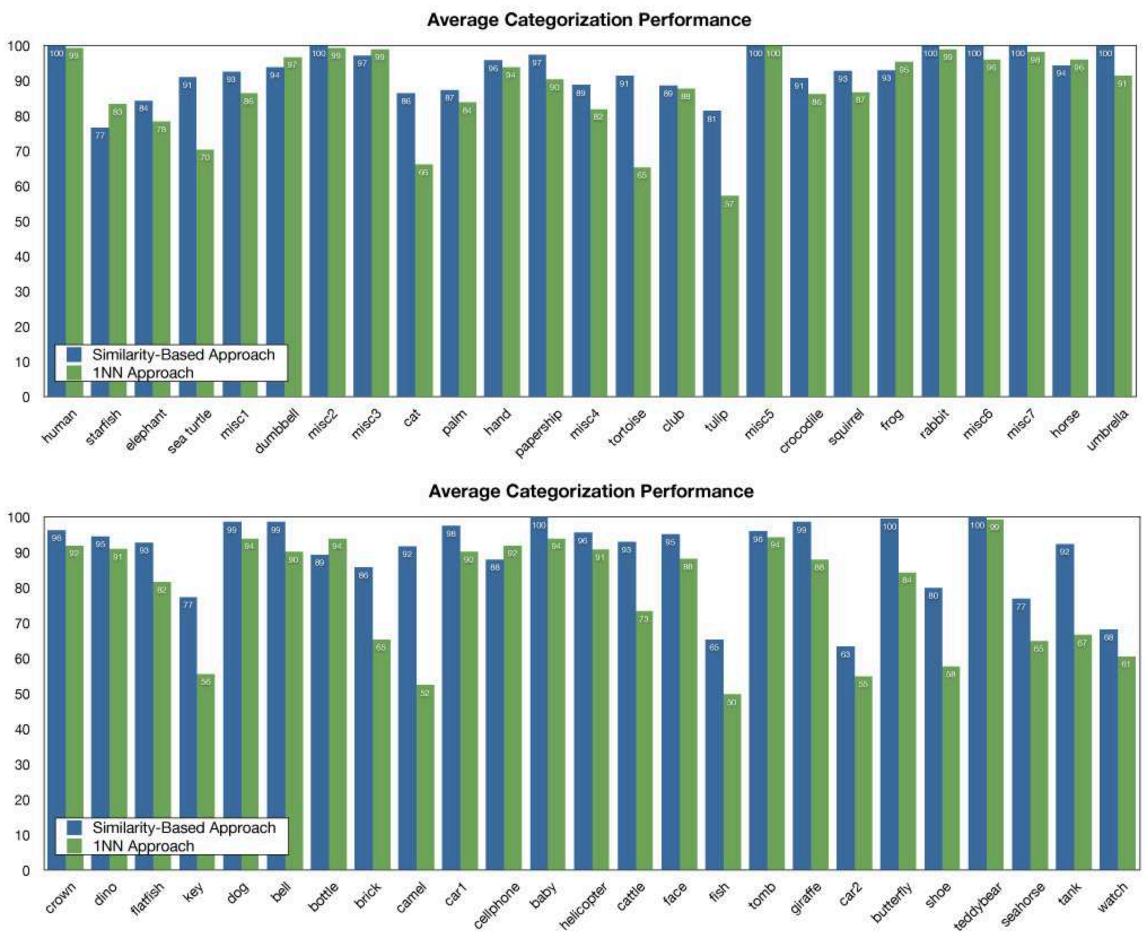


Figure 6.6: Average classification performances for each category.

Then, we eliminate some of these retrieved categories using a simple thresholding mechanism after normalizing the corresponding membership scores. Following to that, the query shape is compared to only the shapes belonging to the categories in the final candidate list. The resulting distance values along with the associated normalized membership information are then used to compute a new matching score as:

$$d^*(\mathcal{T}_1, \mathcal{T}_2) = 1 - exp(-d(\mathcal{T}_1, \mathcal{T}_2)) \times p^*(\mathcal{T}_1 \in \mathcal{C}) \tag{6.13}$$

where $d(\mathcal{T}_1, \mathcal{T}_2)$ denotes the category-influenced matching score and $p^*(\mathcal{T}_1 \in \mathcal{C})$ is the membership score normalized with respect to the retrieved shape categories.

We evaluate the effect of the proposed indexing strategy by repeating the experiments in Section 5.1.4 with a prior classification step. In Appendix E, we present the results of the category-influenced matching method with indexing for the sample partition given in Appendix A. Figure 6.7 shows the average precision-recall curves for the category-influenced matching that includes a prior classification step. The experimental results reveal that performing a prior classification step contributes in achieving better precision values at each recall level with a less computation effort.

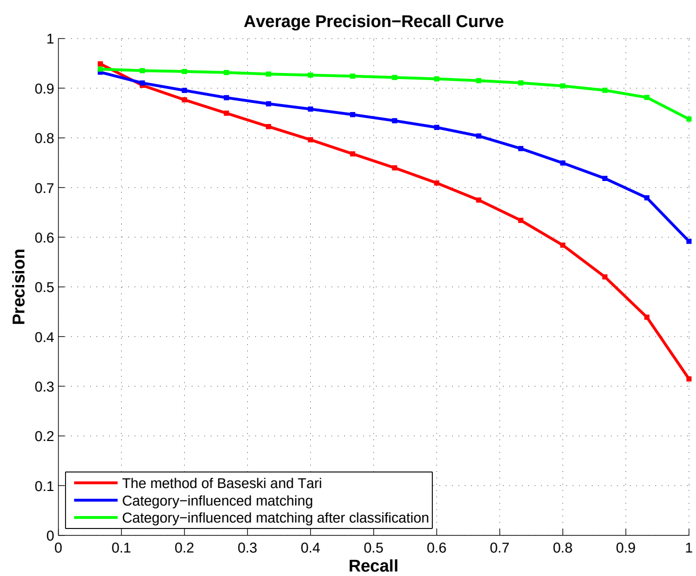


Figure 6.7: Average precision-recall curves. At each recall level, compare the precision values of the category-influenced matching method after classification (shown in green) to those of the category-influenced matching method (shown in blue) and the method of Baseski [7] (shown in red).

6.5 Summary and Discussions

In this chapter, we present a similarity-based supervised shape classification framework built on disconnected skeleton representation. Our starting point is the matching procedure previously developed for comparing a shape tree with a category tree. In the first step of the procedure, exploiting the structural equivalence of shape and category trees, a mean shape tree is formed from the category tree on the fly. As the mean shape tree is indistinguishable from a regular shape tree, it becomes possible to compare a shape tree with a category tree by matching it with the corresponding mean tree using the category-influenced matching. While in Section 5.1 the method is used in dynamic formation of category trees, in this chapter, we demonstrate its use as a nearest-neighbor based classification approach.

We extend this approach and come up with a more effective exemplar-based classification scheme by making some changes and incorporating an additional learning technique. In this regard, we first revise the cost function of `change` operation with a new one that is more suitable for an exemplar-based classification approach. Then, we propose a procedure to form multiple category trees for each shape category, and finally, we employ a similarity-based approach where a shape category is represented by not just a category tree, but by means of its similarities to other existing categories as well. The proposed framework has a network structure where the distances between the given shape and the existing shape categories are computed. These distances are then embedded into a similarity space in which we train a separate SVM classifier for each shape category, and subsequently, the final decision about the category of the input shape is made according to the outputs of the SVM classifiers.

As our experimental results demonstrate, the similarity-based approach brings considerable improvements in terms of performance over a nearest-neighbor strategy. Moreover, we evaluate the indexing performance of the proposed classification framework in a retrieval task, where classification precedes the pairwise comparisons between the query shape and the database shapes, eliminating unrelated distance computations.

CHAPTER 7

CONCLUSION

In this thesis, we have investigated the use of disconnected skeleton representation of Aslan and Tari [3] for shape recognition and classification. The rationale behind the choice of this particular representation is that, as compared to other skeletal representations, disconnected skeletons provide a very coarse but very stable representation of shapes, making some of the computations presented in the thesis possible. Although our experimental results have proven that disconnected skeleton, despite of its coarse structure, is a powerful representation for recognizing and classifying shapes, the representation might be criticized on two grounds: it does not preserve information about the boundary details, and the level of hierarchy is always one. In regard to these concerns, we have presented two ways of enriching the disconnected skeleton representation. First, we have proposed a procedure to roughly fetch the radius functions of positive skeleton branches (representing the approximate distance to shape boundary along the branch) from a corresponding TSP surface [107]. Second, we have devised a multi-level hierarchical approach to increase the level of detail in skeleton descriptions by first segmenting a given shape into its parts based on its skeleton and then performing the skeleton analysis on the extracted parts.

In the context of shape recognition, disconnected skeleton representation was previously utilized in [3, 7]. Particularly, in the method of Aslan and Tari [3], the authors represented disconnected skeletons by their disconnection points as unlabeled attributed point sets, and proposed a branch-and-bound algorithm to obtain correspondences between the skeleton branches of two given shapes. In the method of Baseski [7], however, a structural approach is employed and skeletons are represented as (shape) trees, reducing the problem into matching two shape trees, and the author proposed a tree edit distance-based algorithm to find a partial match between two given shape trees. In this thesis, using these methods as base shape matching algorithms, we have investigated the effect of context on shape similarity computation and proposed a number of ways to incorporate semantic category knowledge into matching process. Our approaches, unlike the view in syntactic matching of two given shapes, differentiate the semantic roles of the shapes in comparison that a query shape is being matched with a database shape, which belongs to a familiar category. The knowledge about the category of the database shape influences the similarity computation, making the resulting similarity measures asymmetric. It is important to note that the conventional way in shape matching literature is to define shape similarity by means of metrics, although it is widely believed that human similarity judgments are non-metric in nature.

In that direction, we have first extended the method of Baseski where our motivation was to improve the quality of matching in comparing a query shape with a database shape by incorporating the information inferred from all the shapes belonging to the same category as the database shape in comparison belongs[1]. The proposed revision relies on a novel data structure, which is referred to as category tree that is formed as a union of shape trees of database shapes belonging to the same category. It should be mentioned that the depth-1 property of shape trees really simplifies the construction of category trees as the resulting category trees are always (depth-1) trees. A constructed category tree holds the associations between the primitives (*i.e.* the extracted skeleton branches) of the category members and moreover, provides information regarding attribute statistics, which allows modifying both the importance of primitives and the distances between attributes in the matching process. Thus, we name this modified version of the algorithm category-influenced matching method.

As a further extension of our category-influenced matching method, we have also incorporate the boundary similarities by employing a coarse-to-fine approach and utilizing enriched disconnected skeleton descriptions. Being similar to the approach of Zhu and Yuille [127], we proposed to model a low-dimensional linear deformation space for each positive branch which appears in a shape category, and following this, we developed a refinement procedure that recalculated the overall dissimilarity score according to the boundary similarity of matched positive branches in the corresponding deformation spaces[2].

Next, we demonstrate another important use of category knowledge in recognition, the contextual sensitivity to articulation of parts. Note that skeleton-based representations are one of the most commonly used representations for shape matching as they provide insensitivity to articulations. However, as mentioned in [1], insensitivity to articulations might

[1]This is a joint work with Emre Baseski and an early version of this study was partly published in MSc. thesis of Emre Baseski [7]. Full version is published in *Pattern Recognition* [8].

[2]Reported in Section 2.3.1 & Section 5.1.5 and to be submitted as an article.

be undesired in some situations that requires a combination of semantics with syntax, *i.e.* prior knowledge about the degree of possible articulations is required to come up with the correct matching result (Recall Figure 5.20). In this respect, based on disconnected skeleton representation, we presented a novel part-centered coordinate frame which provides a representation space for reasoning about observed articulations. In the proposed space, similar articulations or bendings are represented with nearby points. This opens the possibility of building articulation priors and making inferences about them in a fairly easily way. In this thesis, we incorporate this idea into the matching method of Aslan and Tari [3] where articulation priors are modeled as Gaussians[3]. A possible future direction could be using a non-parametric density estimation approach in order to construct more accurate priors. Certainly, one should also need a novel shape database specifically designed for reflecting the importance of contextual sensitivity to articulations.

Finally, we present a similarity-based supervised shape classification method that is built on a matching procedure proposed for dynamic formation of category trees in which the given shape trees are incrementally matched with the category tree in construction[4]. This procedure exploits the structural equivalence of shape and category trees (*i.e.* they are both depth-1 trees) and compares a shape tree with a category tree by first forming a pseudo-shape tree formed from the category tree, and then comparing that with the shape tree by category-influenced matching method. Previously in [8], we proposed to use this matching procedure as a simple classification algorithm based on nearest-neighbor rule. In this thesis, we extend the approach by employing a similarity-based learning strategy to learn relationships between shape categories. To be specific, a shape category is represented by not just a category tree, but by means of its similarities to other existing categories as well. The proposed framework has a network structure where the distances between the given shape and the existing shape categories are computed first, as described above. Then, these computed distances are embedded into a similarity space in which we previously train an SVM classifier for each shape category exist in the database, and subsequently, the final decision about the category of the input shape is made by combining the outputs of the SVM classifiers. As our experimental results demonstrate, the similarity-based approach brings considerable improvements in terms of performance over our previous approach, *i.e.* classifying shapes based on a nearest-neighbor strategy.

[3]This is a joint work with Erkut Erdem and was previously presented in the Workshop on the Representation and Use of Prior Knowledge in Computer Vision [28].

[4]Reported in Section 6.3 and to be submitted as an article.

7.1 Future Directions

In the scope of this thesis, we aimed to develop a shape classification framework using disconnected skeletons, which is based on a similarity-based approach where shape categories are learned in a supervised manner. In this regard, it is important to note that there are also some unsupervised shape classification or clustering studies based on skeletal representations, *e.g.* [111]. As a future work, it will be quite interesting to explore the use of disconnected skeleton representation along with the data structures presented within this thesis in unsupervised learning of shape categories in a given collection of shapes.

Another interesting topic worth exploring is visualization of shape similarity or dissimilarity data. Traditionally, *multidimensional scaling* [47] or its variants are used in visualization of any type of similarity data where the idea is to compute a low dimensional (possibly 2D or 3D) map in which objects that are similar to each other lie close to each other whereas dissimilar objects are placed far away from each other. The problem with these methods is that the similarity data should be metricized in some way before applying these techniques. Although there exist some alternative formulations (*e.g.* [81]), exploring how to visualize non-metric similarity data is still an open problem.

REFERENCES

[1] C. Aslan. Disconnected skeleton for shape recognition. Master's thesis, Dept. of Computer Engineering, Middle East Technical University, May 2005.

[2] C. Aslan, A. Erdem, E. Erdem, and S. Tari. Disconnected skeleton: Shape at its absolute scale. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(12):2188–2203, 2008.

[3] C. Aslan and S. Tari. An axis-based representation for recognition. In *ICCV 2005*, volume 2, pages 1339– 1346, 2005.

[4] J. August, K. Siddiqi, and S. W. Zucker. Ligature instabilities in the perceptual organization of shape. *Comput. Vis. Image Underst.*, 76(3):231–243, 1999.

[5] X. Bai and L. J. Latecki. Path similarity skeleton graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1282–1292, 2008.

[6] X. Bai, L. J. Latecki, and W.-Y. Liu. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):449–462, 2007.

[7] E. Baseski. Context-sensitive matching of two shapes. Master's thesis, Dept. of Computer Engineering, Middle East Technical University, July 2006.

[8] E. Baseski, A. Erdem, and S. Tari. Dissimilarity between two skeletal trees in a context. *Pattern Recognition.* published online: 30 May 2008, doi:10.1016/j.patcog.2008.05.022.

[9] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38(15):2365–2385, 1998.

[10] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, volume 2, pages 1395– 1402, 2005.

[11] H. Blum. Biological shape and visual science. In *Journal of Theoretical Biology*, volume 38, pages 205–287, 1973.

[12] H. Blum and R. N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.

[13] F. L. Bookstein. *Morphometric Tools for Landmark Data–Geometry and Biology*. Cambridge Univ. Press, 1991.

[14] M. Brand. Shadow puppetry. In *ICCV*, pages 1237–1244, 1999.

[15] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel. Analysis of two-dimensional non-rigid shapes. *Int. J. Comput. Vision*, 78(1):67–88, 2008.

[16] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recogn. Lett.*, 18(9):689–694, 1997.

[17] H. Bunke, S. Günter, and X. Jiang. Towards bridging the gap between statistical and structural pattern recognition: Two new concepts in graph matching. In *ICAPR*, pages 1–11, London UK, 2001. Springer-Verlag.

[18] H. Bunke, X. Jiang, K. Abegglen, and A. Kandel. On the weighted mean of a pair of strings. *Pattern Analysis and Applications*, 5(1):23–30, 2002.

[19] M. C. Burl, T. K. Leung, and P. Perona. Recognition of planar object classes. In *CVPR*, pages 223–230, 1996.

[20] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.

[21] C. M. Cyr and B. B. Kimia. 3d object recognition using shape similiarity-based aspect graph. In *ICCV*, volume 1, pages 254–261, 2001.

[22] M. F. Demirci, R. H. van Leuken, and R. C. Veltkamp. Indexing through laplacian spectra. *Comput. Vis. Image Underst.*, 110(3):312–325, 2008.

[23] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[24] R.P.W. Duin, D. de Ridder, and D.M.J. Tax. Featureless pattern classification. *Kybernetika*, 34(4):399–404, 1998.

[25] S. Edelman. Representation, similarity, and the chorus of prototypes. *Minds and Machines*, 5(1):45–68, 1995.

[26] S. Edelman. *Representation and Recognition in Vision*. MIT Press, 1999.

[27] S. Edelman, F. Cutzu, and S. Duvdevani-Bar. Similarity to reference shapes as a basis for shape representation. In G. Cottrell, editor, *COGSCI'96*, pages 260–265, 1996.

[28] A. Erdem, E. Erdem, and S. Tari. Articulation prior in an axial representation. In *International Workshop on the Representation and Use of Prior Knowledge in Vision (WRUPKV) held in association with ECCV'06*, pages 1–14, 2006.

[29] E. Erdem and S. Tari. Mumford-shah regularizer with contextual feedback. *Journal of Mathematical Imaging and Vision*. published online: 24 July 2008, doi:10.1007/s10851-008-0109-y.

[30] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, pages 242–256, May 2004.

[31] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(12):1312–1328, 1999.

[32] D. Geiger, T. L. Liu, and R. V. Kohn. Representation and self-similarity of shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(1):86–99, 2003.

[33] P. J. Giblin and B. B. Kimia. On the local form and transitions of symmetry sets, and medial axes, and shocks in 2D. In *ICCV*, pages 385–391, 1999.

[34] R. L. Goldstone. Similarity, interactive activation, and mapping. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(1):3–28, 1994.

[35] R.L. Goldstone, D.L. Medin, and J. Halberstadt. Similarity in context. *Memory and Cognition*, 25:237–255, 1997.

[36] P. Golland, W. Eric, and L. Grimson. Fixed topology skeletons. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 10–17, 2000.

[37] U. Hahn, N. Chater, and L. B. Richardson. Similarity as transformation. *Cognition*, 87(1):1–32, 2003.

[38] A. Hurlbert and T. Poggio. Making machines (and artificial intelligence) see. *Daedalus*, 117(1):213–240, 1988.

[39] D. W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with nonmetric distances: Image retrieval and class representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(6):583–600, 2000.

[40] X. Jiang, A. Muenger, and H. Bunke. Computing the generalized mean of a set of graphs. In *Workshop on Graph Based Representations*, pages 115–124, 1999.

[41] T. Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

[42] D. G. Kendall, D. Barden, T. K. Carne, and H. Le. *Shape and Shape Theory.* John Wiley and Sons, 1999.

[43] I. Kovacs and B. Julesz. Perceptual sensitivity maps within globally defined visual shapes. *Nature*, 370:644 – 646, 1994.

[44] I. Kovacs and B. Julesz. Medial-point description of shape: A representation for action coding and its psychophysical correlates. *Vision Research*, 38:2323–2333, 1998.

[45] J. K. Kruschke. Alcove: An exemplar-based connectionist model of category learning. *Psychological Review*, 99(1):22–44, 1992.

[46] J. K. Kruschke. Category learning. In K. Lamberts and R. L. Goldstone, editors, *The Handbook of Cognition*, chapter 7, pages 183–201. London: Sage, 2005.

[47] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[48] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

[49] B. Landau, L. B. Smith, and S. S. Jones. The importance of shape in early lexical learning. *Cognitive Development*, 3:299–321, 1988.

[50] L. J. Latecki and R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1185–1190, 2000.

[51] L. J. Latecki, R. Lakamper, and D. Wolter. Optimal partial shape similarity. *Image and Vision Computing*, 23(2):227–236, 2005.

[52] L. J. Latecki, V. Megalooikonomou, Q. Wang, R. Lakaemper, C. A. Ratanamahatana, and E. Keogh. Partial elastic matching of time series. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 701–704, Washington, DC, USA, 2005. IEEE Computer Society.

[53] A. Latto, D. Mumford, and J. Shah. The representation of shape. In *IEEE Proc. of the Workshop on Computer Vision*, pages 183–198, 1984.

[54] T. S. Lee, D. Mumford, R. Romero, and V. A.F. Lamme. The role of the primary visual cortex in higher level vision. *Vision Research*, 38(15-16):2429–2454, 1998.

[55] K. Leonard. Efficient shape modeling: ϵ-entropy, adaptive coding, and boundary curves -vs- Blum's medial axis. *Int. J. Comput. Vision*, 74(2):183–199, 2007.

[56] T. Liu and D. Geiger. Approximate tree matching and shape similarity. In *ICCV*, volume 1, pages 456–462, 1999.

[57] T. L. Liu, D. Geiger, and R. V. Kohn. Representation and self-similarity of shapes. In *ICCV*, pages 1129–1135, 1998.

[58] P. Lombardi, V. Cantoni, and B. Zavidovique. Context in robotic vision: Control for real-time adaptation. In *ICINCO*, volume 3, pages 135–142, 2004.

[59] R. Duncan Luce. *Individual Choice Behavior*. Wiley, New York, 1959.

[60] R. Duncan Luce. The choice axiom after twenty years. *Journal of Mathematical Psychology*, 15:215–233, 1977.

[61] D. Macrini, K. Siddiqi, and S. Dickinson. From skeletons to bone graphs: Medial abstraction for object recognition. In *CVPR*, 2008.

[62] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman and Company, New York, 1982.

[63] D. Medin and M. M. Schaffer. Context theory of classification learning. *Psychological Review*, 85:207–238, 1978.

[64] D. Mumford. Mathematical theories of shape: Do they model perception? In B. C. Vemuri, editor, *roc. SPIE Vol. 1570, p. 2-10, Geometric Methods in Computer Vision*, pages 2–10, 1991.

[65] M. Neuhaus and H. Bunke. Automatic learning of cost functions for graph edit distance. *Inf. Sci.*, 177(1):239–247, 2007.

[66] R. M. Nosofsky. Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, 115(1):39–57, 1986.

[67] I. Omer and M. Werman. Image specific feature similarities. In *ECCV*, pages 321–333, 2006.

[68] A. Opelt, A. Pinz, and A. Zisserman. Fusing shape and appearance information for object category detection. In *BMVC*, 2006.

[69] O. C. Ozcanli, A. Tamrakar, B. B. Kimia, and J. L. Mundy. Augmenting shape with appearance in vehicle category recognition. In *CVPR*, pages 935–942, 2006.

[70] R. Paredes and E. Vidal. Learning prototypes and distances: A prototype reduction technique based on nearest neighbor error minimization. *Pattern Recognition*, 39(2):180–188, 2006.

[71] E. Pekalska and R. P. W. Duin. *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. World Scientific, 2005.

[72] E. Pekalska, R. P. W. Duin, and P. Paclík. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39(2):189–208, 2006.

[73] M. Pelillo. Matching free trees, maximal cliques, and monotone game dynamics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(11):1535–1541, 2002.

[74] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1105–1120, 1999.

[75] M. Pelillo, K. Siddiqi, and S. W. Zucker. Many-to-many matching of attributed trees using association graphs and game dynamics. In *Int. Workshop Visual Form*, pages 583–593, 2001.

[76] F. Pernus, A. Leonardis, and S. Kovacic. Two-dimensional object recognition using multiresolution non-information-preserving shape features. *Pattern Recognition Letters*, 15(11):1071–1079, 1994.

[77] S. Pinker. *How the Mind Works*. W. W. Norton Company, 1997.

[78] T. Poggio and E. Bizzi. Generalization in vision and motor control. *Nature*, 431(7010):768–774, 2004.

[79] T. Poggio and S. Edelman. A network that learns to recognize 3d objects. *Nature*, 343(6255):263–266, 1990.

[80] S. W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM J. Comput.*, 6:730–732, 1977.

[81] W. Richards and J. J. Koenderink. Trajectory mapping ("tm""): A new non-metric scaling technique. Technical Report AIM-1468, Massachusetts Institute of Technology, Cambridge, MA, USA, 1993.

[82] E. Rosch. *Principles of Categorization*, pages 189–206. Hillsdale NJ: Erlbaum 1978. (Reprinted in Concepts: Core Readings. Edited by E. Margulis and S. Laurence. Cambridge MA: MIT Press, 1999.

[83] B. H. Ross and V. S. Makin. Prototype versus exemplar models in cognition. In Robert J. Sternberg, editor, *The Nature of Cognition*, chapter 7, pages 205–241. The MIT Press, 1999.

[84] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001.

[85] P. G. Schyns, R. L. Goldstone, and J. P. Thibaut. The development of features in object concepts. *Behavioral and Brain Sciences*, 21:1–54, 1998.

[86] S. Sclaroff. Deformable prototypes for encoding shape categories in image databases. *Pattern Recognition*, 30(4):627–642, 1997.

[87] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing shock graphs. In *ICCV*, volume 1, pages 755–762, 2001.

[88] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Shock-based indexing into large shape databases. In *ECCV*, pages 731–746, London UK, 2002. Springer-Verlag.

[89] T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(1):116–125, 2003.

[90] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):550–571, 2004.

[91] J. Shah. Gray skeletons and segmentation of shapes. *CVIU*, 99(1):96–109, 2005.

[92] D. Shasha and K. Zhang. Approximate tree pattern matching. In *Pattern Matching Algorithms*, pages 341–371. Oxford University Press, 1997.

[93] R. N. Shepard. Toward a universal law of generalization for psychological science. *Science*, 237:1317–1323, 1987.

[94] A. Shokoufandeh, D. Macrini, S. Dickinson, K. Siddiqi, and S. W. Zucker. Indexing hierarchical structures using graph spectra. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1125–1140, 2005.

[95] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, pages 1–15, 2006.

[96] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker. Hamilton-Jacobi skeletons. *Int. J. Comput. Vision*, 48(3):215–231, 2002.

[97] K. Siddiqi and B. B. Kimia. A shock grammar for recognition. In *CVPR*, pages 507–513, 1996.

[98] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *Int. J. Comput. Vision*, 35(1):13–32, 1999.

[99] H. A. Simon. *The sciences of the artificial*. MIT Press, 3 edition, 1996.

[100] E. E. Smith and D. Medin. *Categories and concepts*. Harvard University Press, Cambridge, 1981.

[101] M. Sonka, V. Hlavac, and Roger Boyle. *Image Processing Analysis and Machine Vision*. Chapman and Hall, 1993.

[102] B. Spillmann, M. Neuhaus, H. Bunke, E. Pekalska, and R. P. W. Duin. Transforming strings to vector spaces using prototype selection. pages 287–296, 2006.

[103] A. Srivastava, S. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering learning and testing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):590–602, 2005.

[104] K. B. Sun and B. J. Super. Classification of contour shapes using class segment sets. In *CVPR*, volume 2, pages 727–733, Washington DC USA, 2005. IEEE Computer Society.

[105] S. Tari, C. Aslan, E. Baseski, and A. Erdem. Shape scale: Representing shapes at their absolute scales. SIAM Annual Meeting, 2006.

[106] S. Tari, J. Shah, and H. Pien. A computationally efficient shape analysis via level sets. In *MMBIA '96: Proceedings of the 1996 Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA '96)*, pages 234–243, 1996.

[107] S. Tari, J. Shah, and H. Pien. Extraction of shape skeletons from grayscale images. *CVIU*, 66(2):133–146, 1997.

[108] J. B. Tenenbaum and T. L. Griffiths. Generalization similarity and bayesian inference. *Behavioral and Brain Sciences*, 24:629–640, 2001.

[109] A. Torsello and E. R. Hancock. Matching and embedding through edit-union of trees. In *ECCV*, volume 3, pages 822–836, 2002.

[110] A. Torsello and E. R. Hancock. Computing approximate tree edit distance using relaxation labeling. *Pattern Recogn. Lett.*, 24(8):1089–1097, 2003.

[111] A. Torsello and E. R. Hancock. Learning shape-classes using a mixture of tree-unions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(6):954–967, 2006.

[112] G. T. Toussaint. The use of context in pattern recognition. *Pattern Recognition*, 10(3):189–204, 1978.

[113] A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.

[114] S. Ullman. Aligning pictorial description: An approach to object recognition. *Cognition*, 32:193–254, 1989.

[115] M. van Eede, D. Macrini, A. Telea, C. Sminchisescu, and S. Dickinson. Canonical skeletons for shape matching. In *ICPR*, volume 2, pages 64–69, 2006.

[116] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[117] T. Verbeemen, W. Vanpaemel, S. Pattyn, G. Storms, and T. Verguts. Beyond exemplars and prototypes as memory representations of natural concepts: A clustering approach. *Journal of Memory and Language*, 56:537–554, 2007.

[118] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518, 2001.

[119] S. Watanabe. *Pattern Recognition: Human and Mechanical.* John Wiley and Sons, 1969.

[120] L. Wolf and S. Bileschi. A critical view of context. *Int. J. Comput. Vision*, 69(2):251–261, 2006.

[121] Y. Wu, J. Lin, and T. S. Huang. Analyzing and capturing articulated hand motion in image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1910–1922, December 2005.

[122] X. Yang, X. Bai, D. Yu, and L. J. Latecki. Shape classification based on skeleton path similarity. In Alan L. Yuille, Song Chun Zhu, Daniel Cremers, and Yongtian Wang, editors, *EMMCVPR*, volume 4679 of *Lecture Notes in Computer Science*, pages 375–386. Springer, 2007.

[123] A. Yilmaz and M. Shah. A differential geometric approach to representing the human actions. *Comput. Vis. Image Underst.*, 109(3):335–351, 2008.

[124] H. Zhang and J. Malik. Learning a discriminative classifier using shape context distances. In *CVPR*, volume 01, pages 242–247. IEEE Computer Society, 2003.

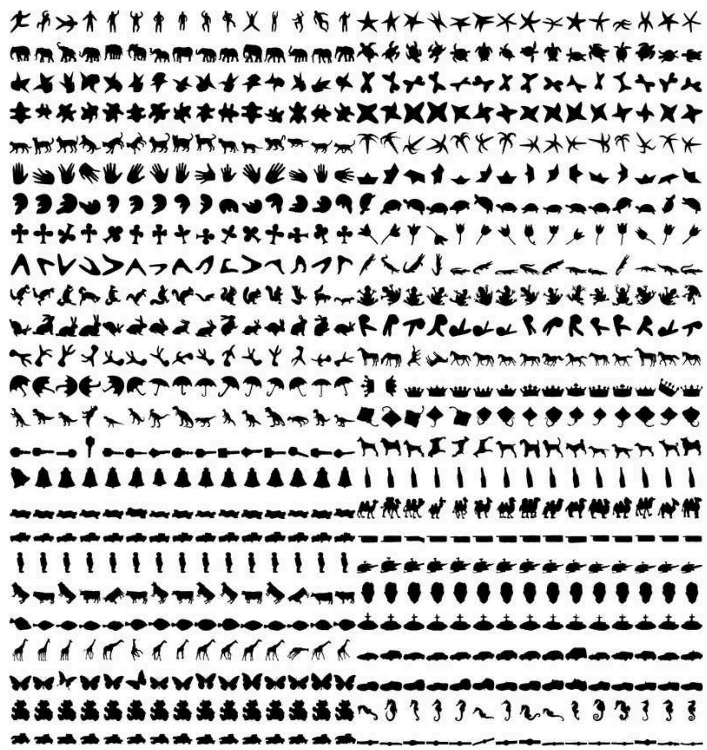
[125] S. C. Zhu. Stochastic jump-diffusion process for computing medial axes in markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1158–1169, 1999.

[126] S. C. Zhu and A. L. Yuille. Forms: A flexible object recognition and modelling system. In *ICCV*, pages 465–472, 1995.

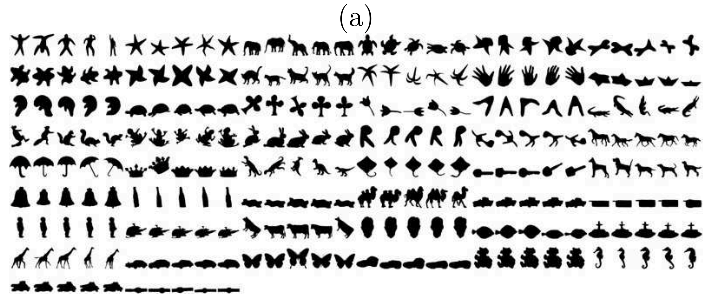
[127] S. C. Zhu and A. L. Yuille. Forms: A flexible object recognition and modeling system. *Int. J. Comput. Vision*, 20(3):187–212, 1996.

APPENDIX A

A PARTITION OF THE SHAPE DB



(a)



(b)

Figure A.1: A sample partition of the shape database. The set of (a) training, (b) test shapes.

APPENDIX B

RETRIEVAL RESULTS OF THE MATCHING METHOD OF BASESKI

Table B.1: Matching results of the method of Baseski [7].

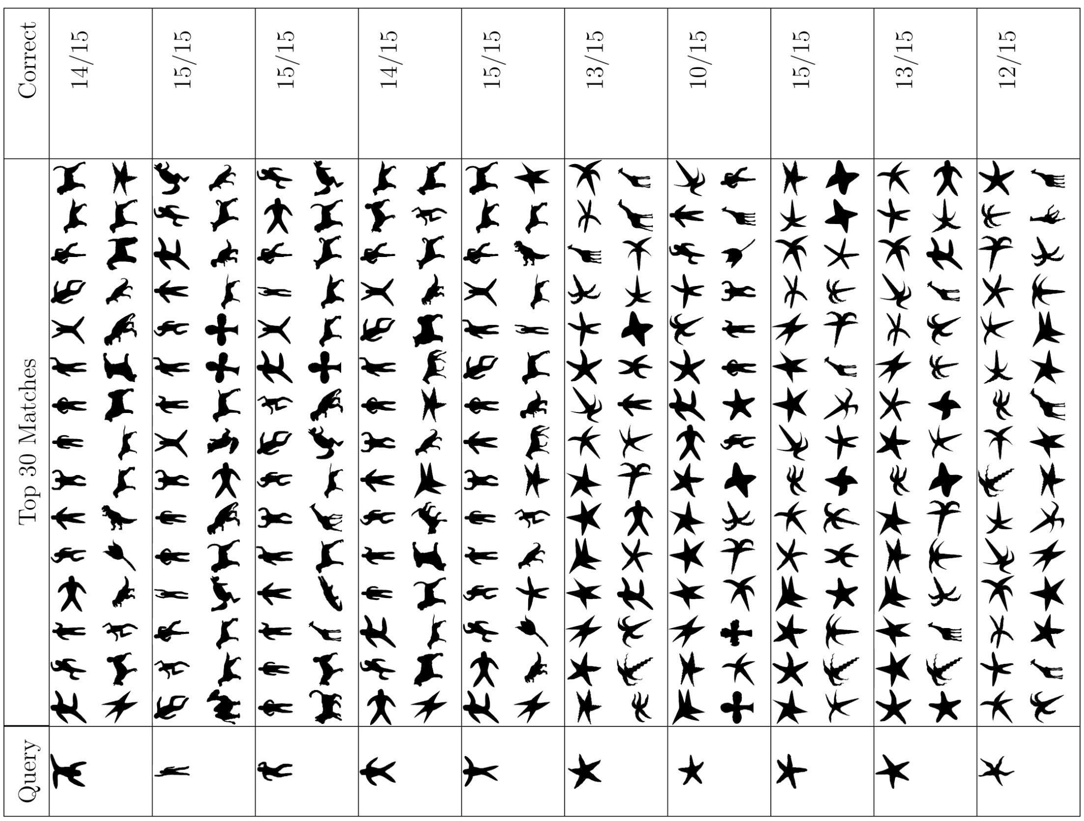
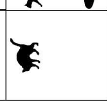
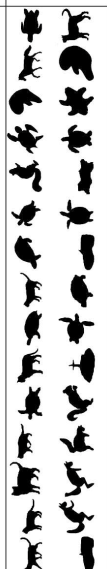
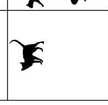
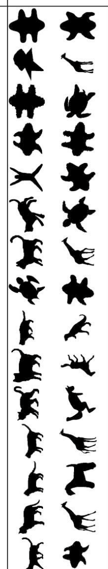
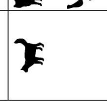
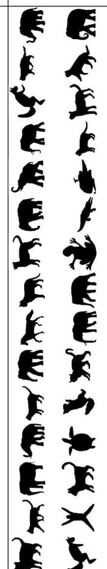
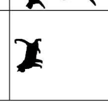
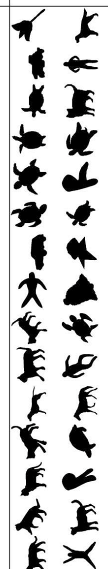
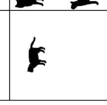
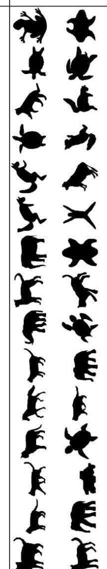
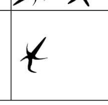
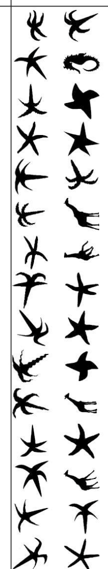
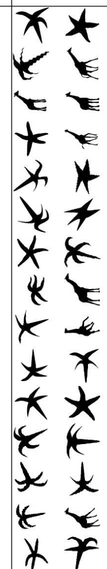
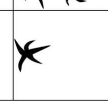
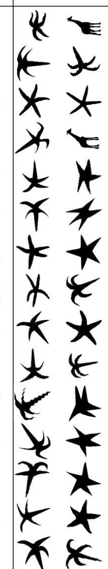
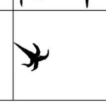
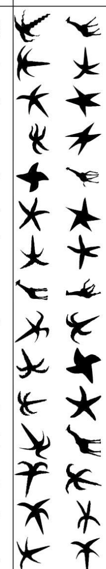
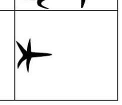
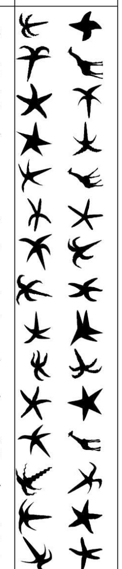


Table B.1: *continued*

Query	Top 30 Matches	Correct
		12/15
		13/15
		11/15
		14/15
		12/15
		14/15
		10/15
		6/15
		14/15
		14/15

Query	Top 30 Matches	Correct
		15/15
		15/15
		8/15
		14/15
		14/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		7/15
		9/15
		11/15
		8/15
		10/15
		15/15
		15/15
		15/15
		15/15
		15/15

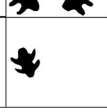
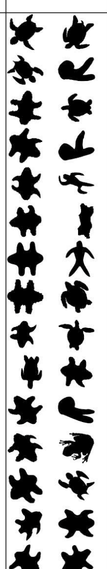
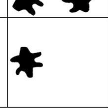
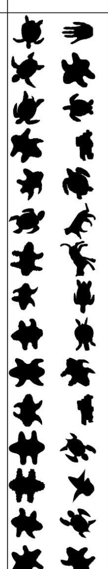
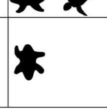
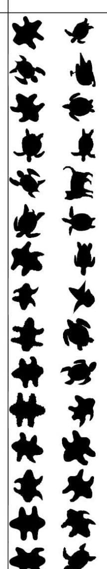
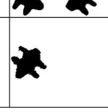
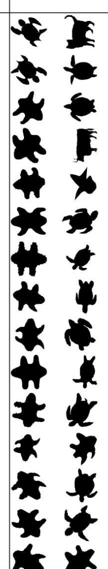
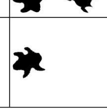
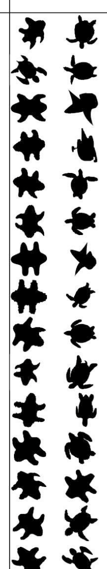
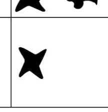
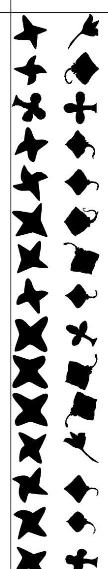
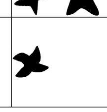
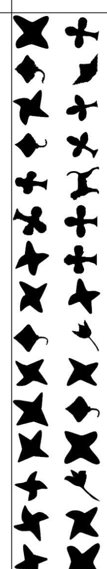
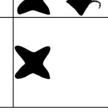
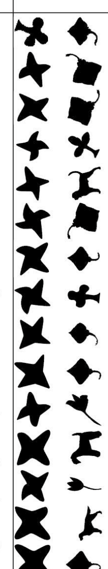
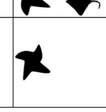
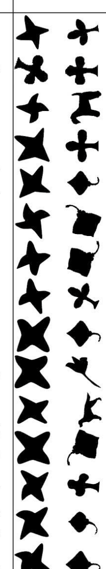
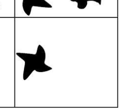
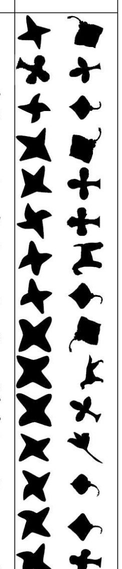
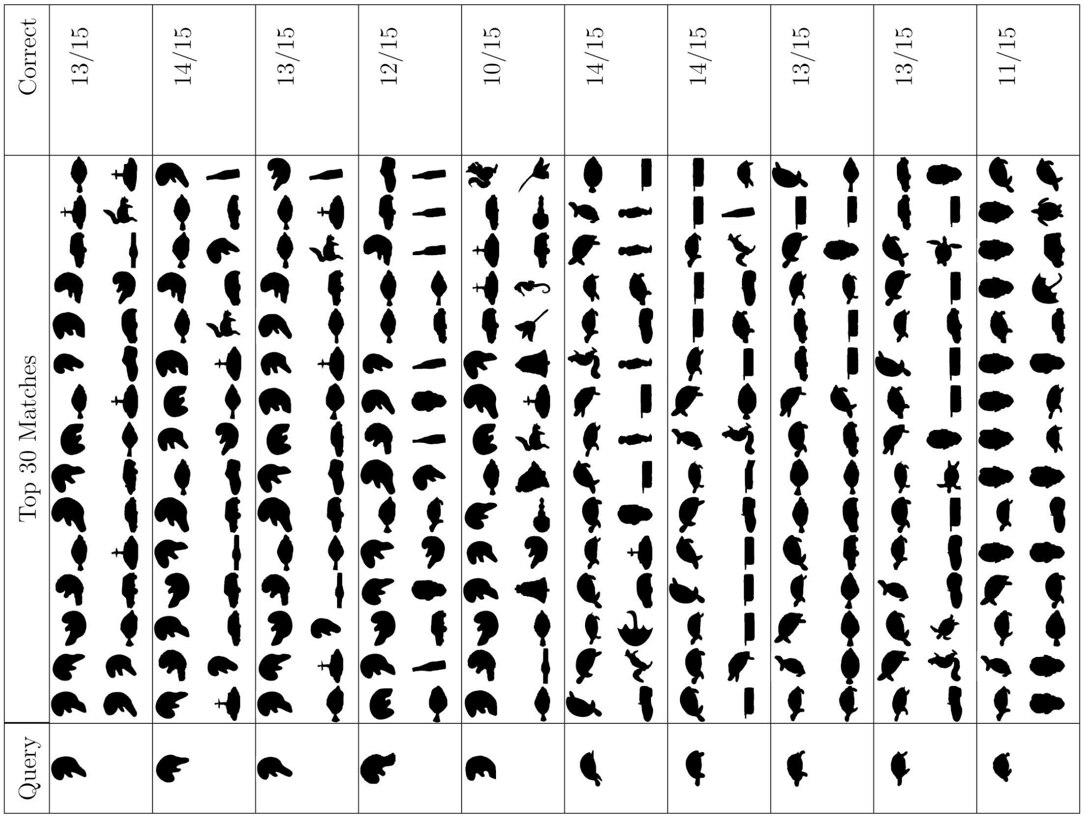
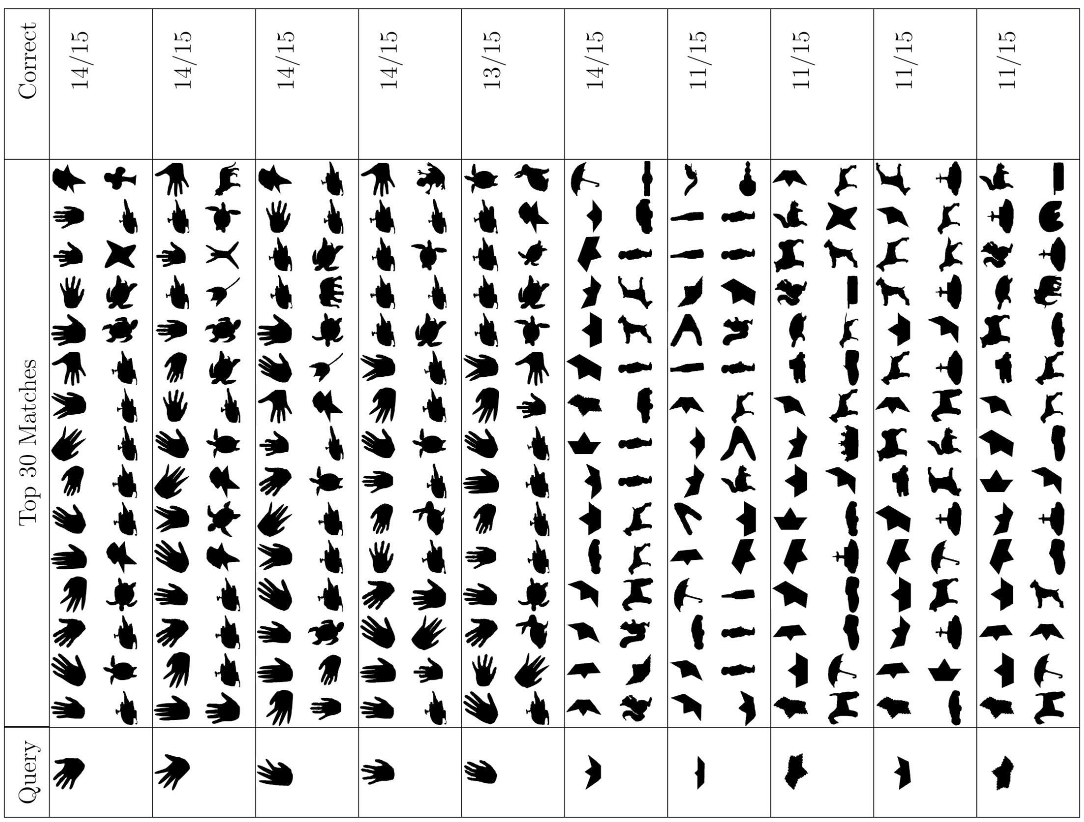
Query	Top 30 Matches	Correct
		15/15
		14/15
		15/15
		15/15
		15/15
		14/15
		15/15
		14/15
		14/15
		14/15

Table B.1: *continued*



Correct
13/15
14/15
13/15
12/15
10/15
14/15
14/15
13/15
13/15
11/15



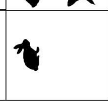
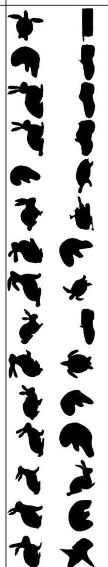
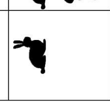
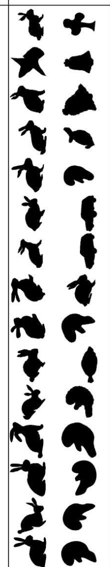
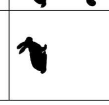
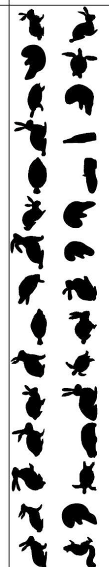
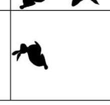
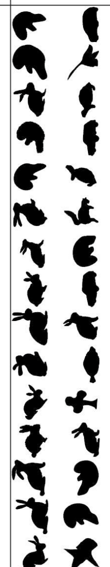
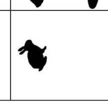
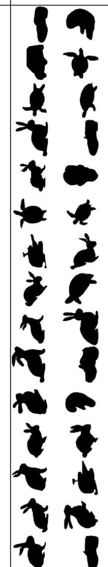
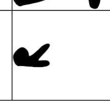
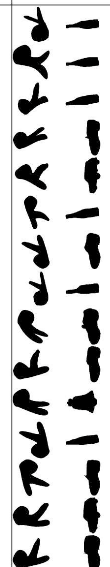
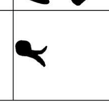
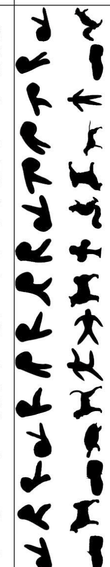
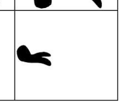
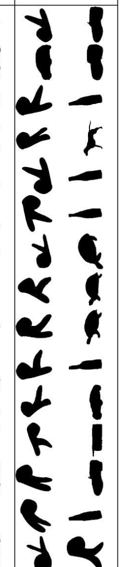
Correct
14/15
14/15
14/15
14/15
13/15
14/15
11/15
11/15
11/15
11/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		10/15
		11/15
		13/15
		13/15
		11/15
		4/15
		4/15
		5/15
		3/15
		7/15

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		11/15
		12/15
		9/15
		11/15
		12/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		13/15
		15/15
		14/15
		13/15
		14/15
		15/15
		13/15
		15/15
		15/15
		15/15

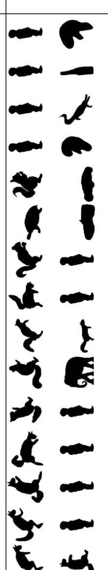
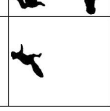
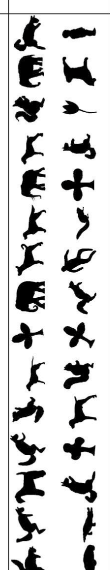
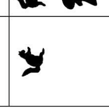
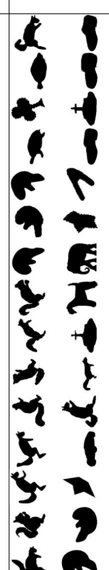
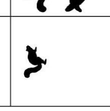
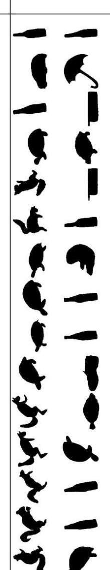
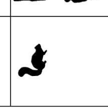
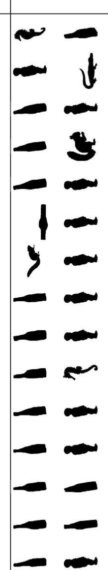
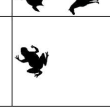
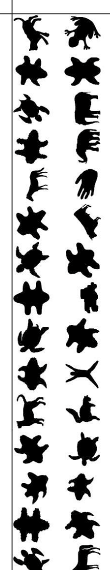
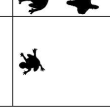
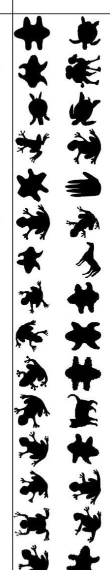
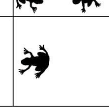
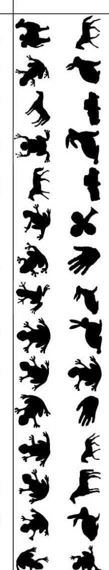
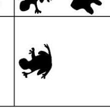
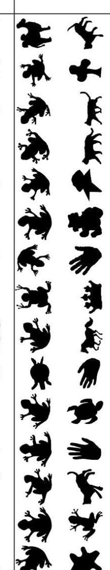
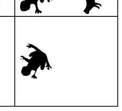
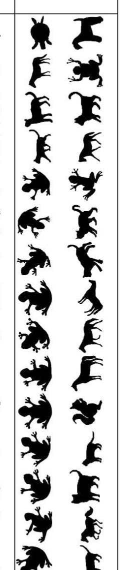
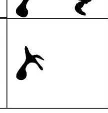
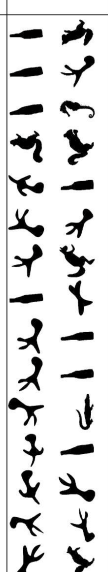
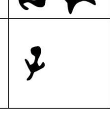
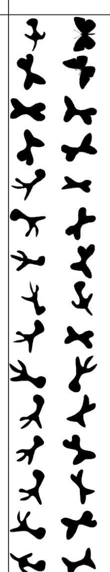
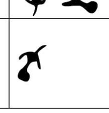
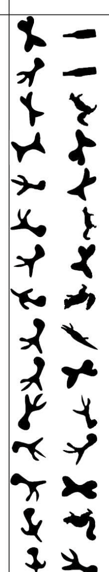
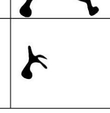
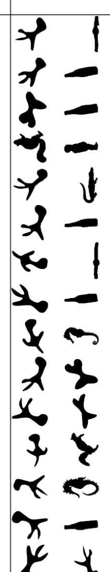
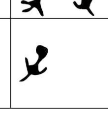
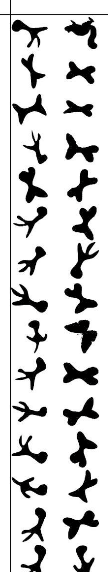
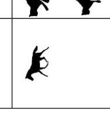
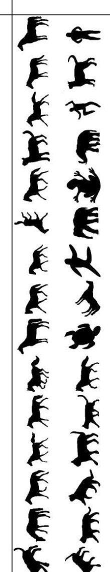
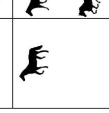
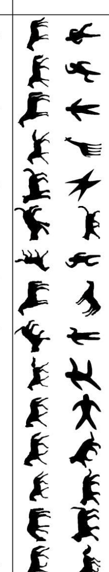
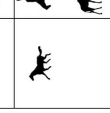
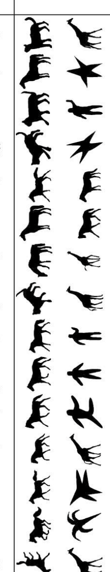
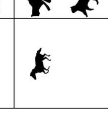
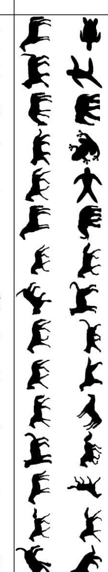
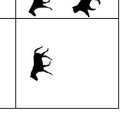
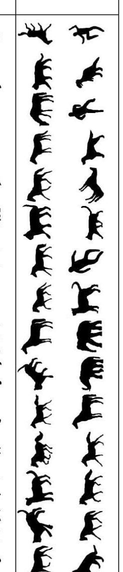
Query	Top 30 Matches	Correct
		12/15
		11/15
		11/15
		7/15
		2/15
		1/15
		14/15
		14/15
		14/15
		13/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		14/15
		14/15
		15/15
		14/15
		14/15
		15/15
		14/15
		13/15
		15/15
		15/15

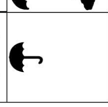
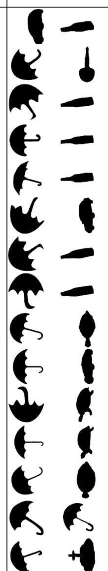
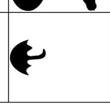
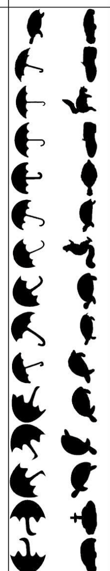
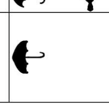
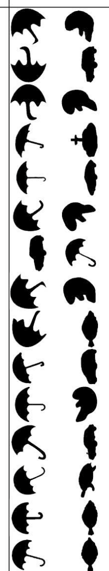
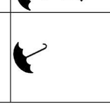
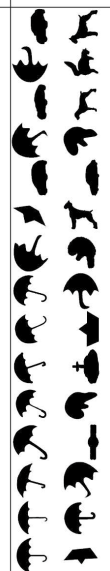
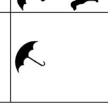
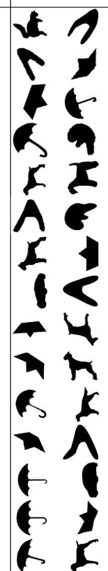
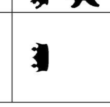
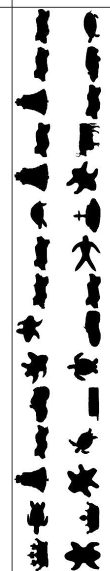
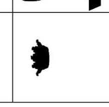
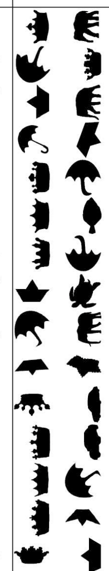
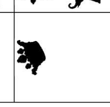
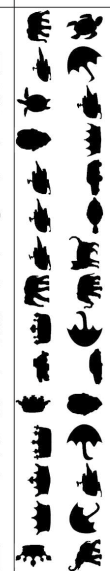
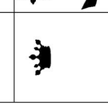
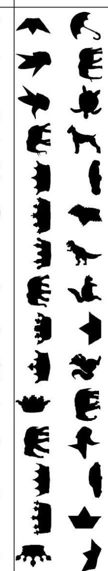
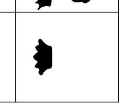
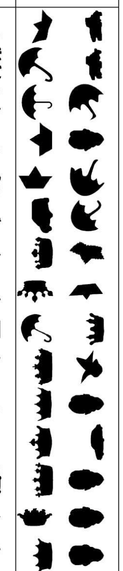
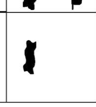
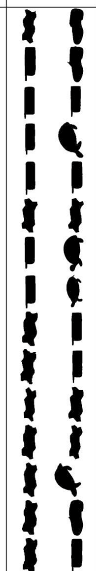
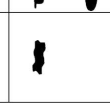
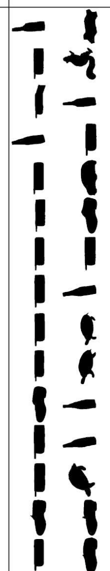
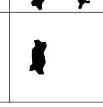
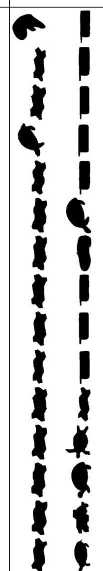
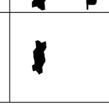
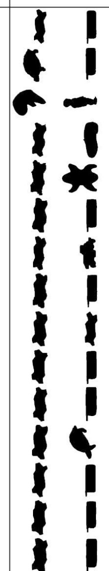
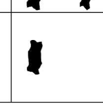
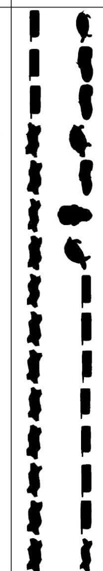
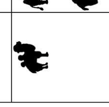
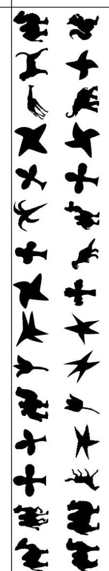
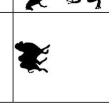
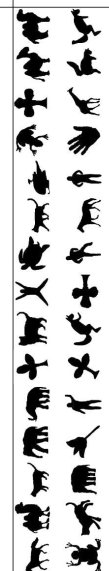
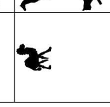
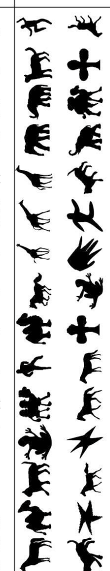
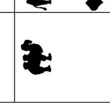
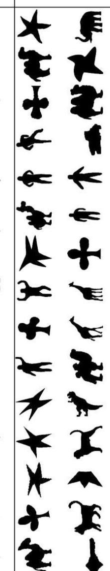
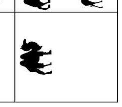
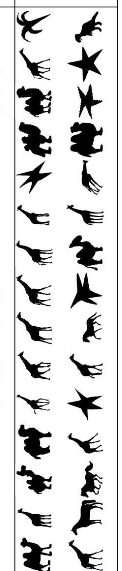
Query	Top 30 Matches	Correct
		15/15
		14/15
		15/15
		14/15
		6/15
		2/15
		10/15
		7/15
		9/15
		9/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		11/15
		12/15
		13/15
		14/15
		14/15
		10/15
		10/15
		13/15
		10/15
		14/15

Query	Top 30 Matches	Correct
		7/15
		11/15
		2/15
		12/15
		9/15
		15/15
		13/15
		15/15
		14/15
		10/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		12/15
		1/15
		14/15
		14/15
		13/15
		7/15
		3/15
		4/15
		5/15
		7/15

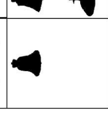
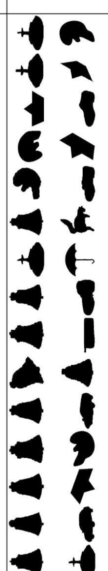
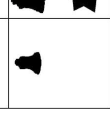
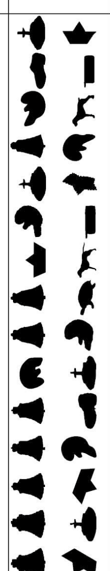
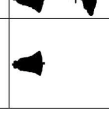
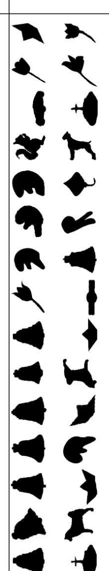
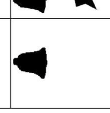
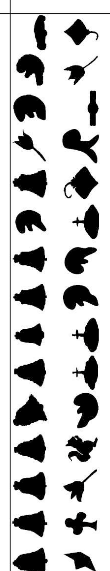
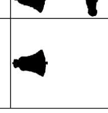
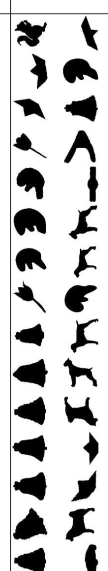
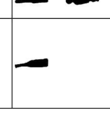
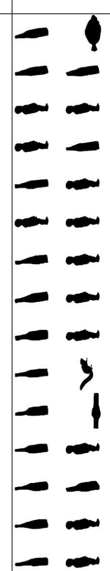
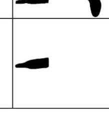
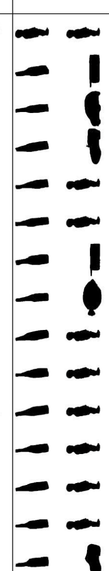
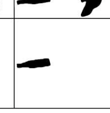
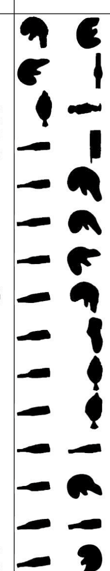
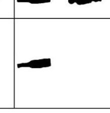
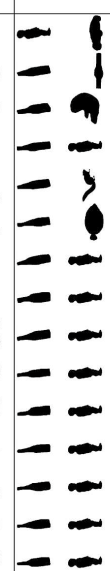
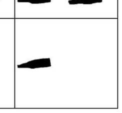
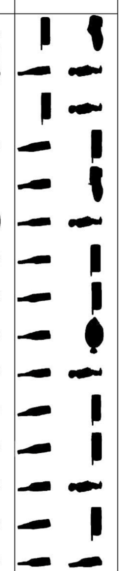
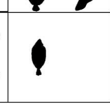
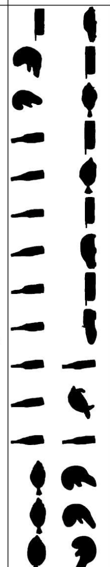
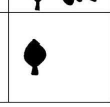
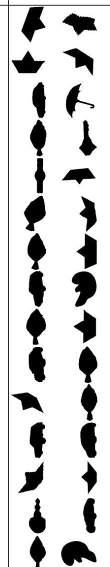
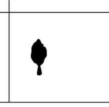
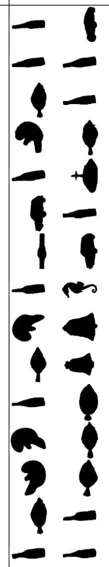
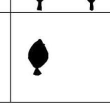
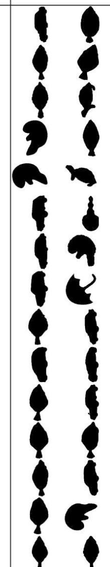
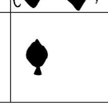
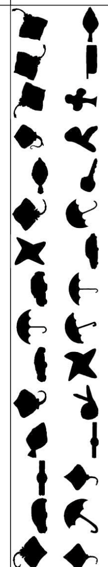
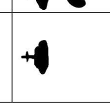
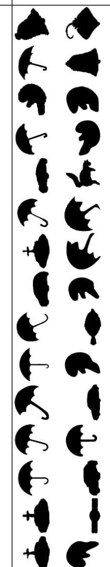
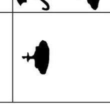
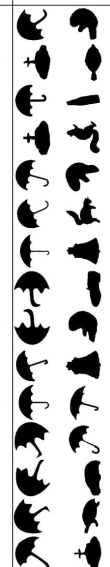
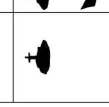
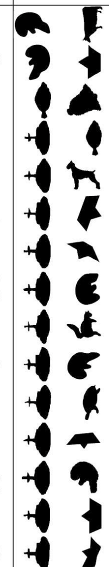
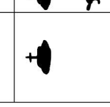
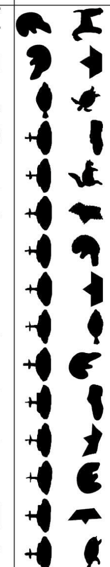
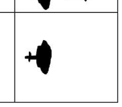
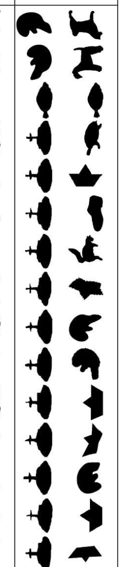
Query	Top 30 Matches	Correct
		10/15
		8/15
		8/15
		10/15
		8/15
		15/15
		14/15
		14/15
		14/15
		14/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		14/15
		15/15
		15/15
		15/15
		15/15
		12/15
		4/15
		12/15
		12/15
		15/15

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		13/15
		13/15
		10/15
		13/15
		13/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		5/15
		7/15
		7/15
		13/15
		3/15
		3/15
		3/15
		12/15
		12/15
		12/15

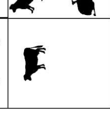
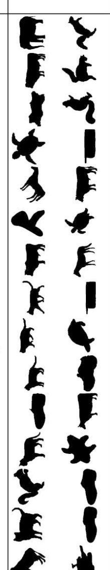
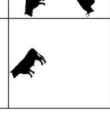
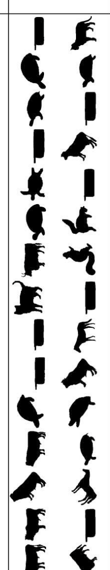
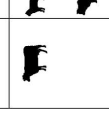
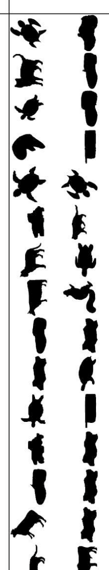
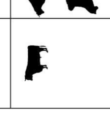
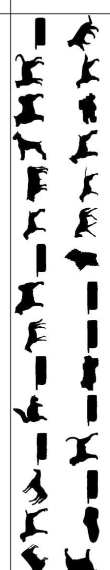
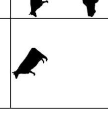
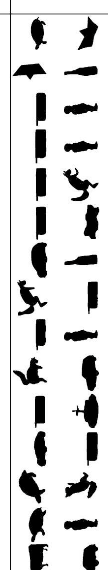
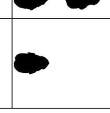
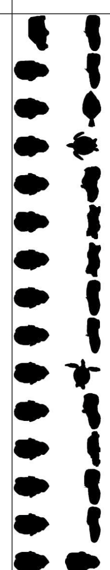
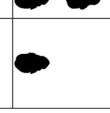
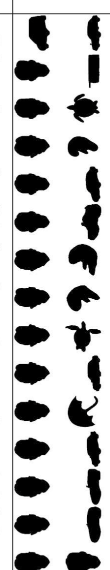
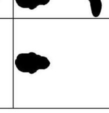
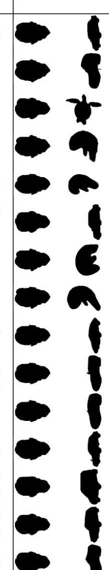
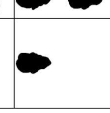
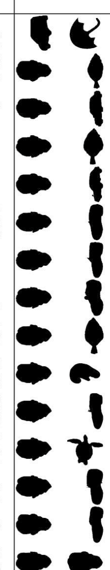
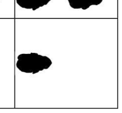
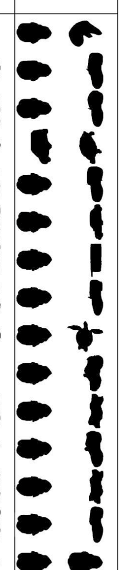
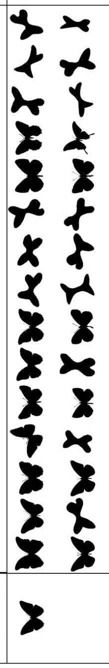
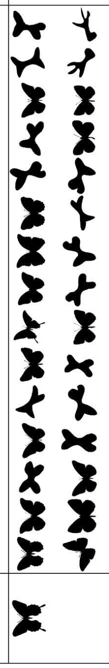
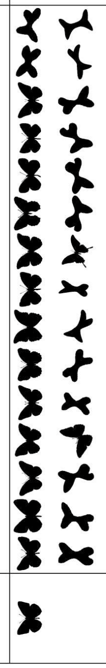
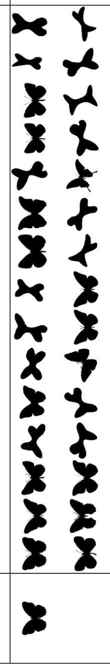
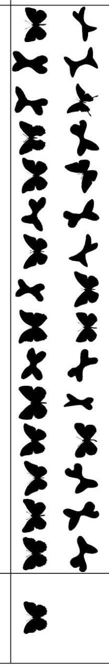
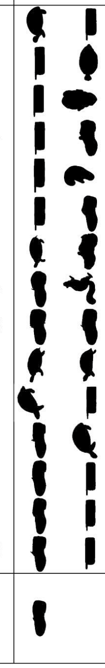
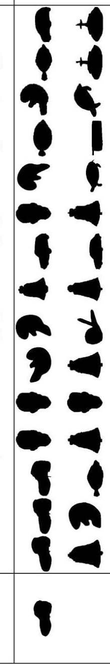
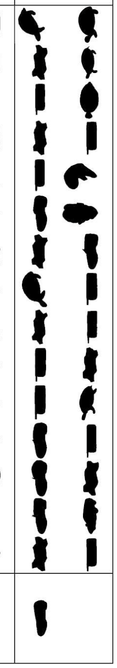
Query	Top 30 Matches	Correct
		5/15
		8/15
		3/15
		2/15
		1/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		1/15
		11/15
		10/15
		3/15
		5/15

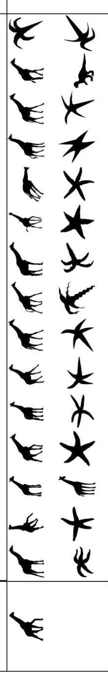
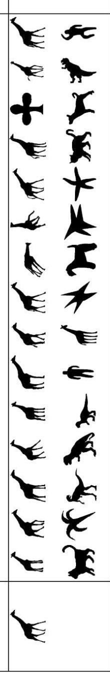
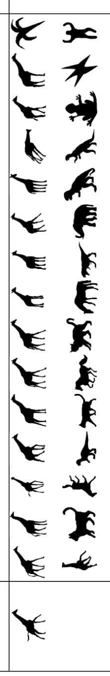
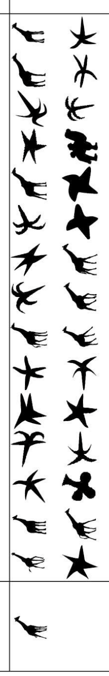
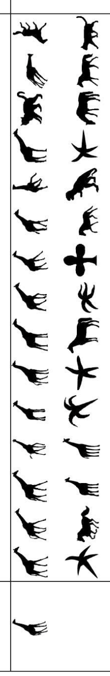
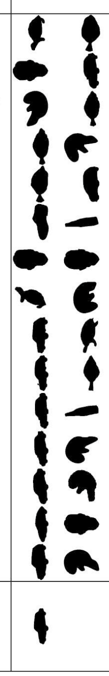
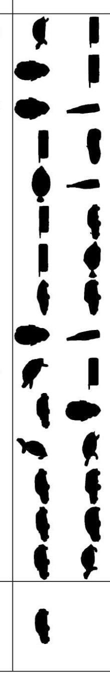
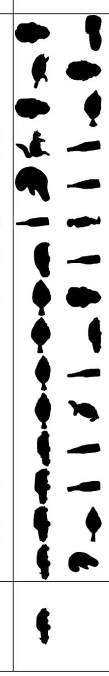
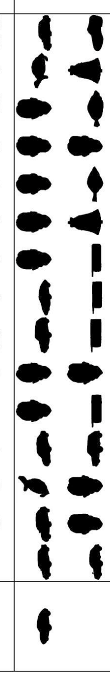
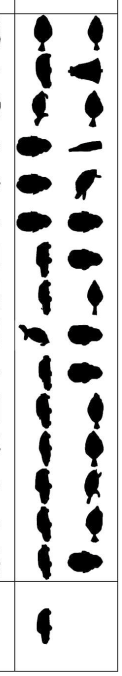
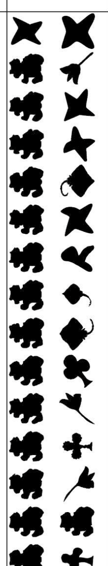
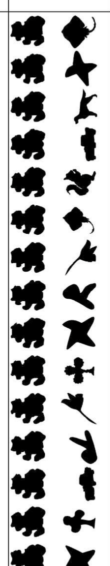
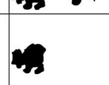
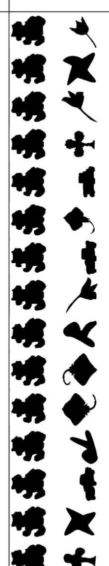
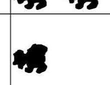
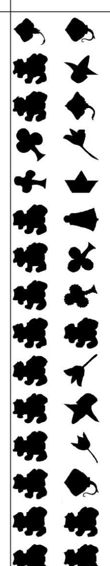
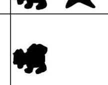
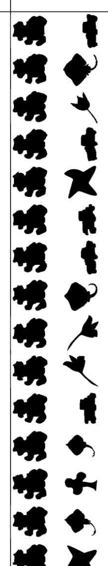
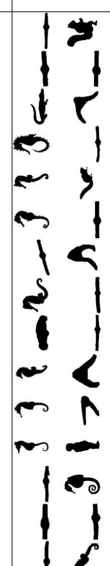
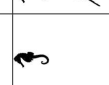
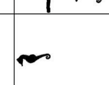
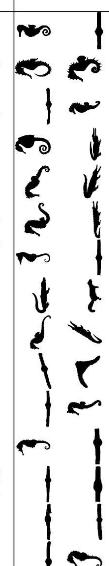
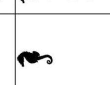
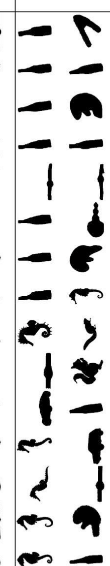
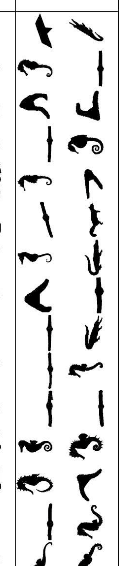
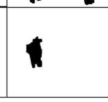
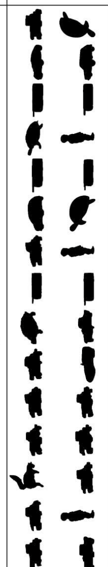
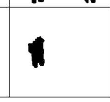
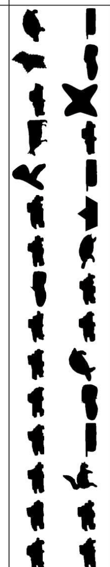
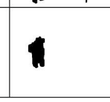
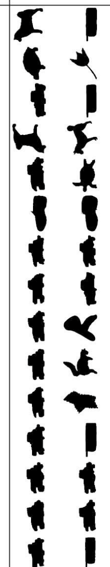
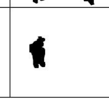
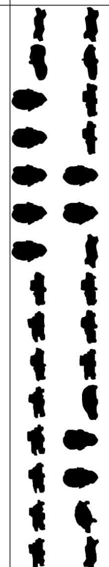
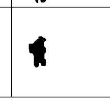
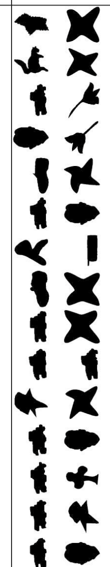
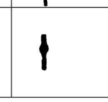
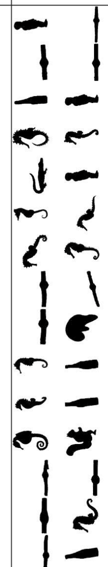
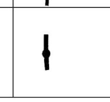
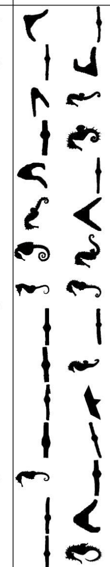
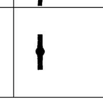
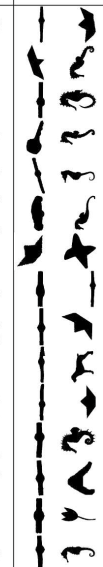
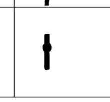
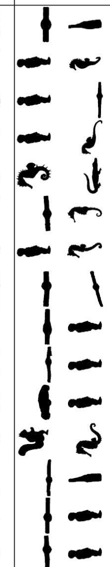
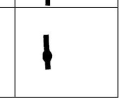
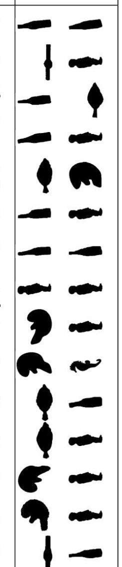
Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		10/15
		15/15
		9/15
		9/15
		6/15
		8/15
		9/15

Table B.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		9/15
		6/15
		12/15
		6/15
		11/15

Query	Top 30 Matches	Correct
		10/15
		13/15
		13/15
		7/15
		9/15
		10/15
		13/15
		12/15
		10/15
		2/15

APPENDIX C

RETRIEVAL RESULTS OF CATEGORY-INFLUENCED MATCHING

Table C.1: Results of category-influenced matching.

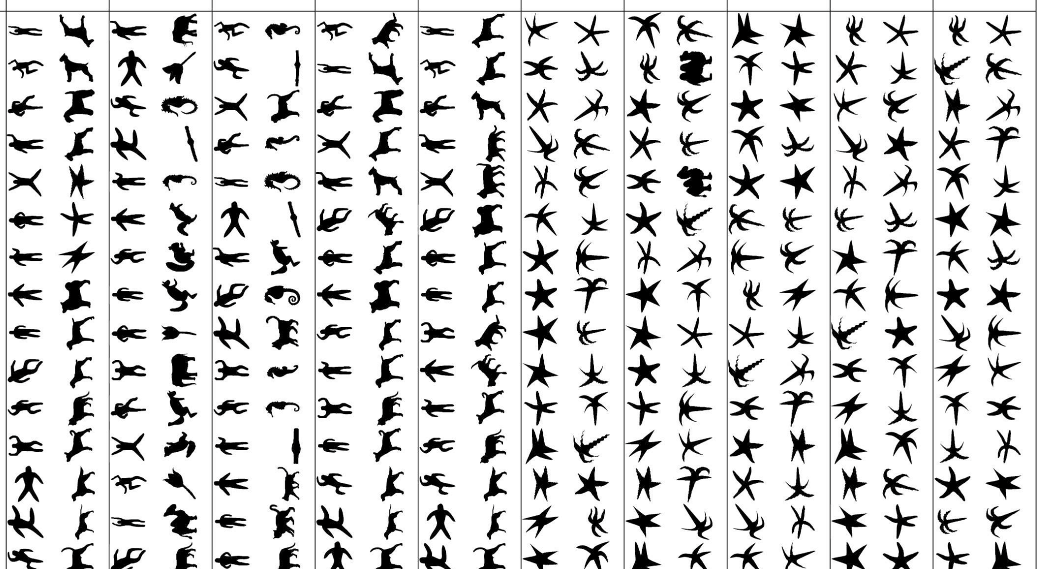
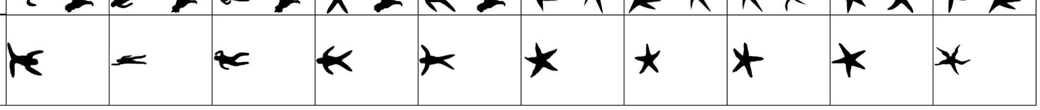
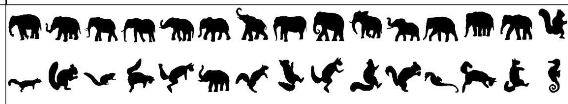
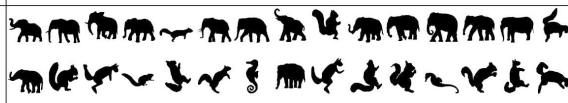
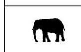
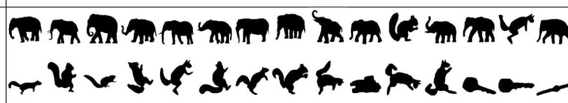
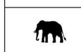
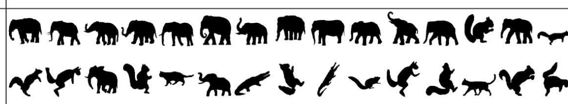
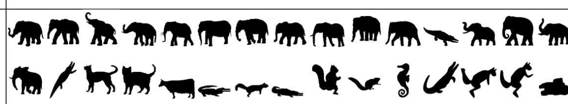
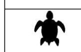
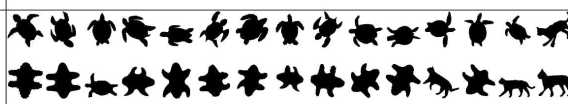
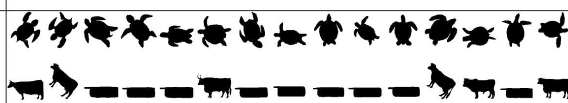
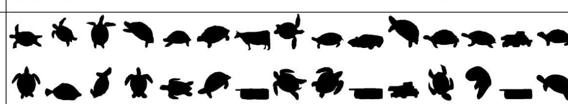
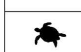
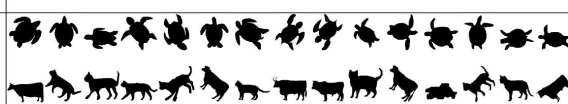
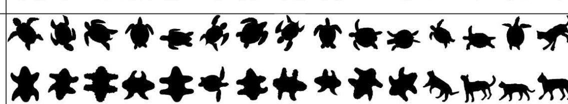
Correct	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15
Top 30 Matches										
Query										

Table C.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		14/15
		13/15
		15/15
		15/15
		15/15
		15/15
		10/15
		15/15
		15/15

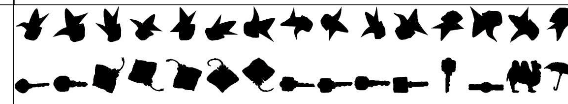
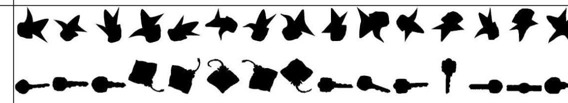
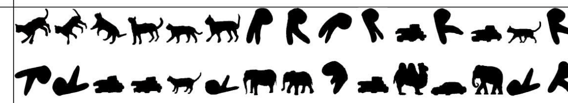
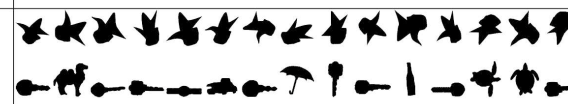
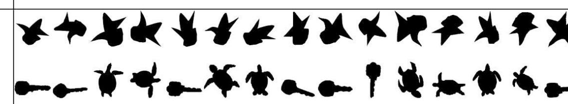
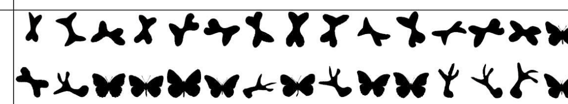
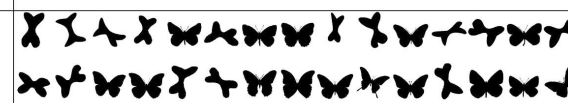
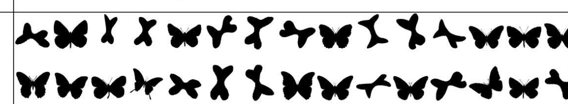
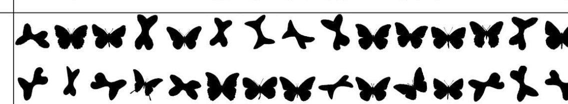
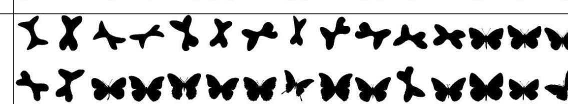
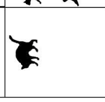
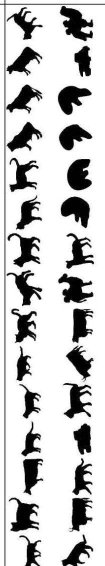
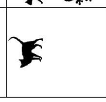
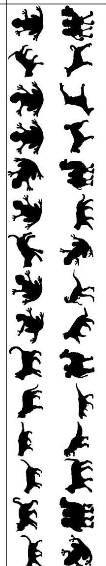
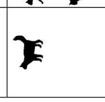
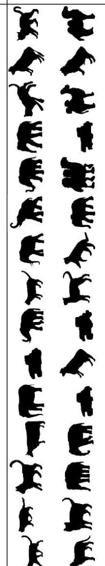
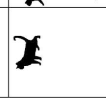
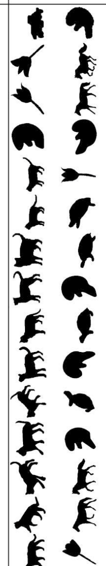
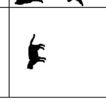
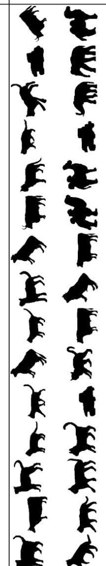
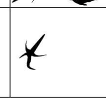
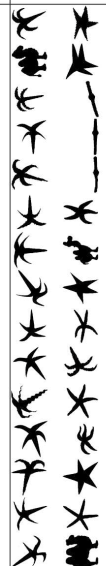
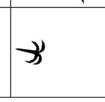
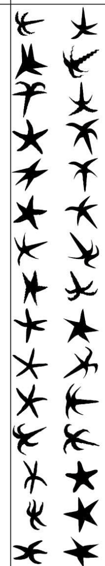
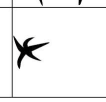
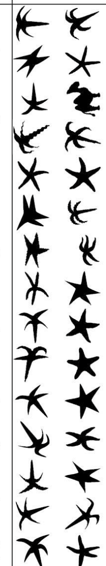
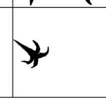
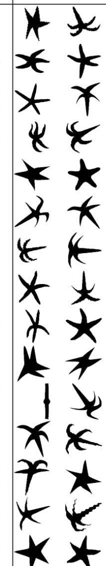
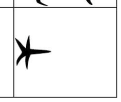
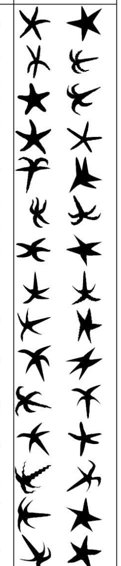
Query	Top 30 Matches	Correct
		15/15
		15/15
		0/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table C.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		11/15
		10/15
		11/15
		14/15
		15/15
		15/15
		14/15
		14/15
		15/15

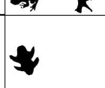
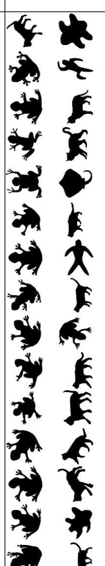
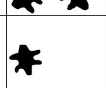
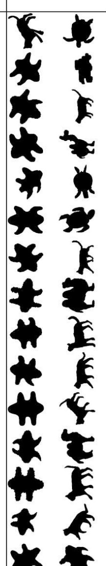
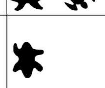
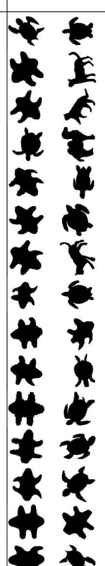
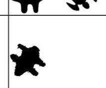
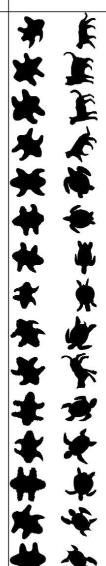
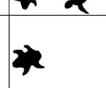
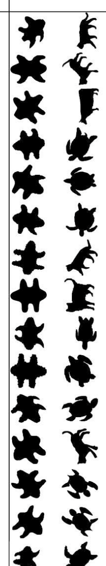
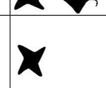
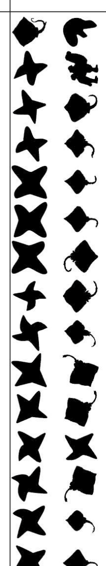
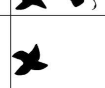
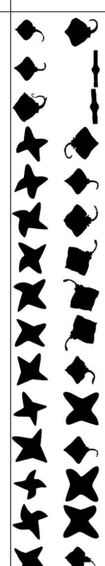
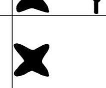
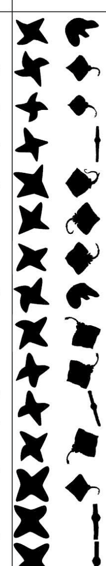
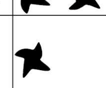
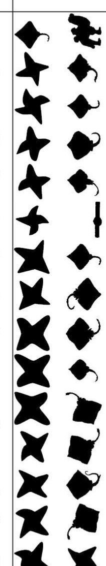
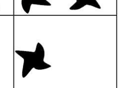
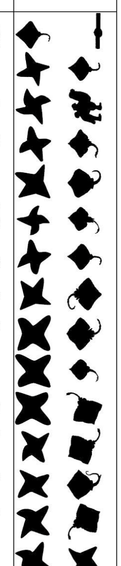
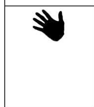
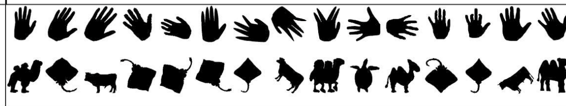
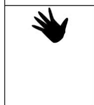
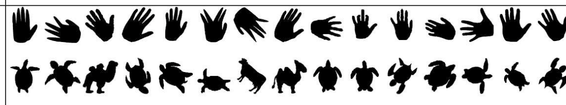
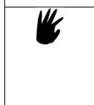
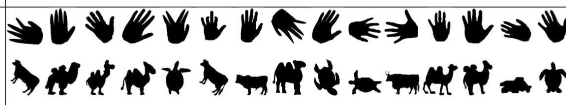
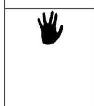
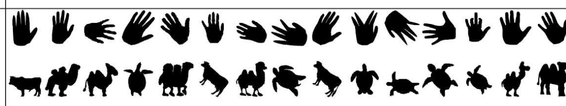
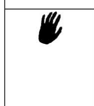
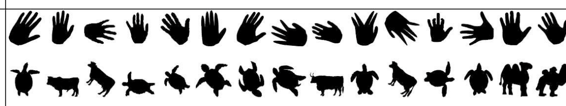
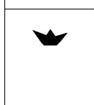
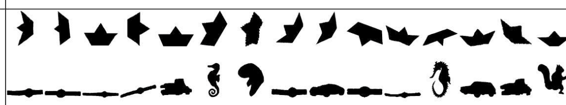
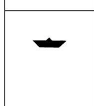
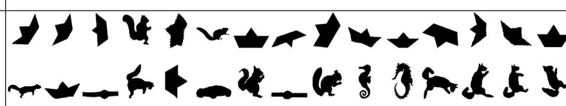
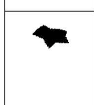
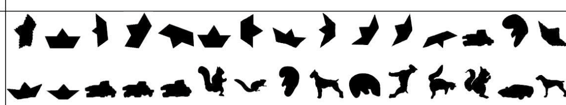
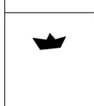
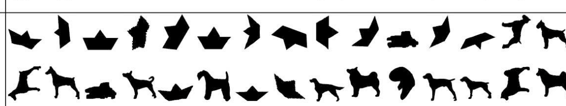
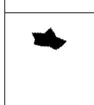
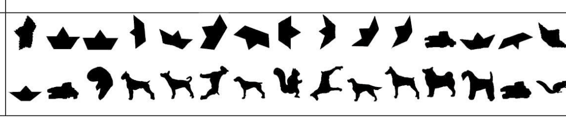
Query	Top 30 Matches	Correct
		2/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table C.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

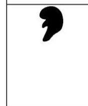
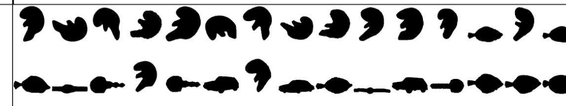
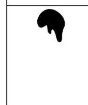
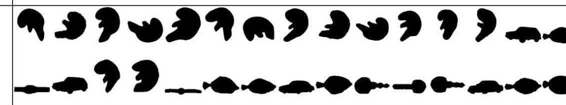
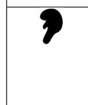
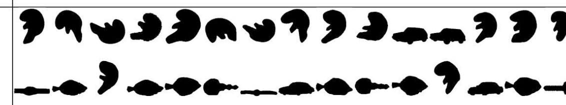
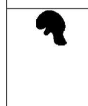
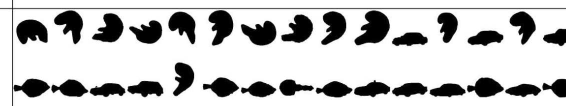
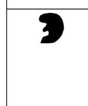
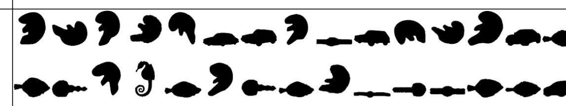
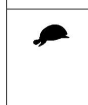
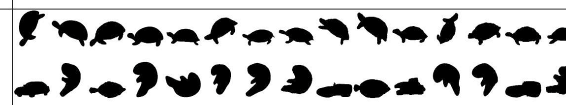
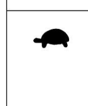
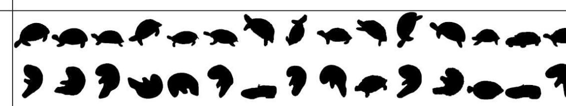
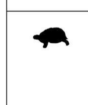
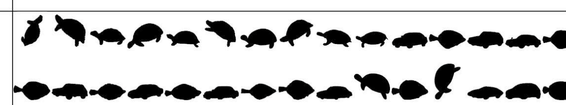
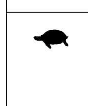
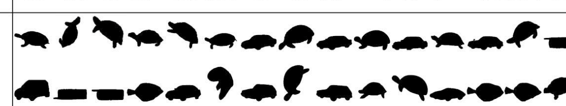
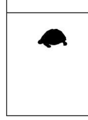
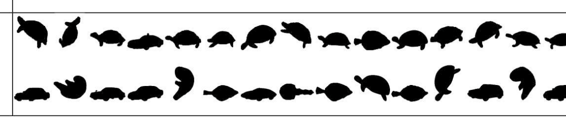
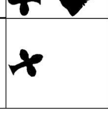
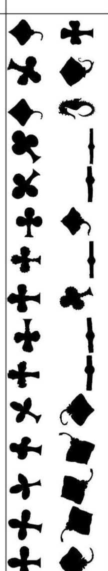
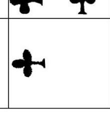
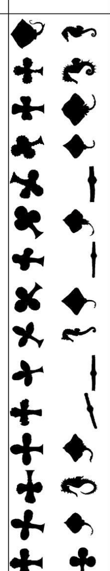
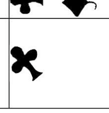
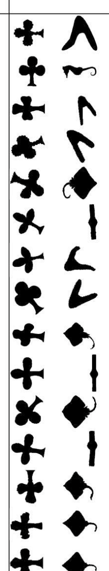
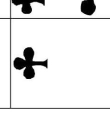
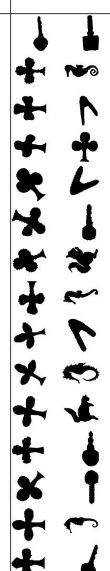
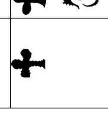
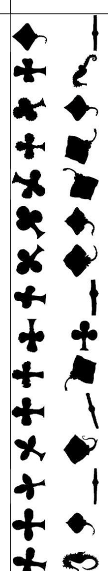
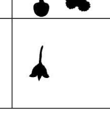
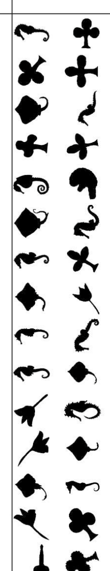
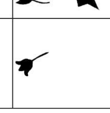
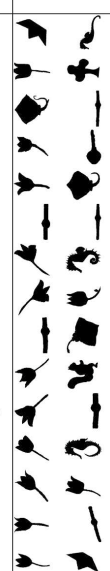
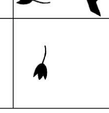
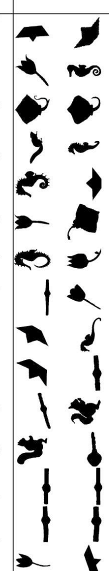
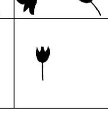
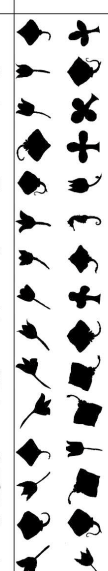
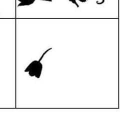
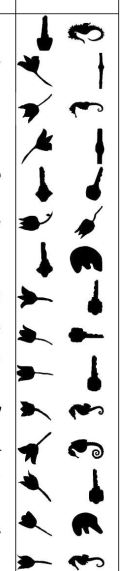
Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		13/15
		12/15
		15/15
		15/15
		12/15
		14/15
		15/15

Table C.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		4/15
		13/15
		5/15
		13/15
		13/15

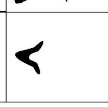
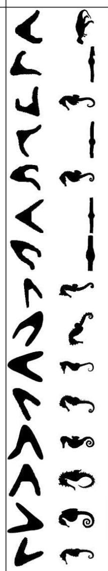
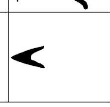
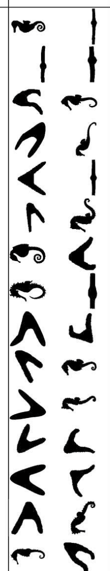
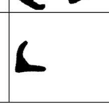
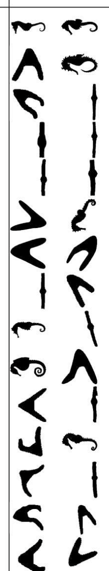
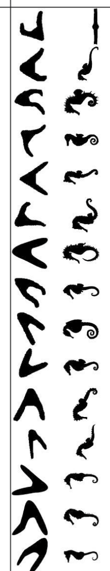
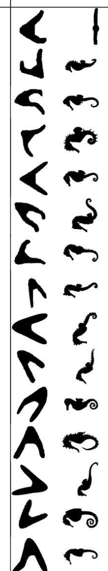
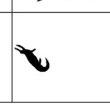
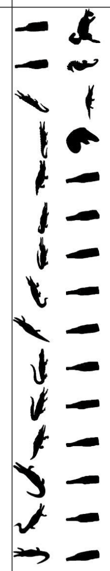
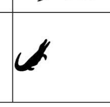
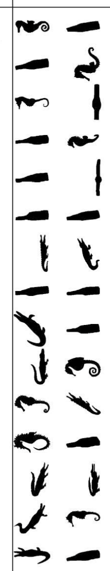
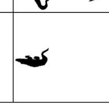
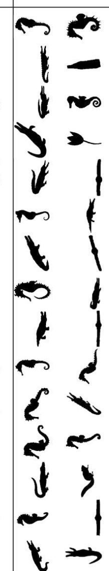
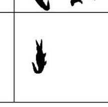
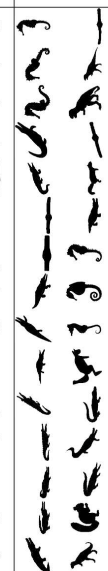
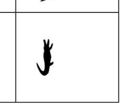
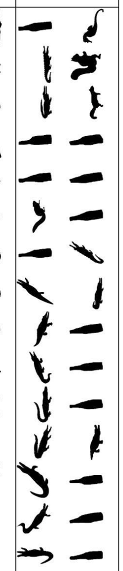
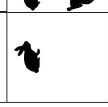
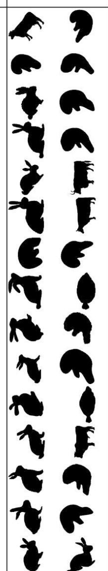
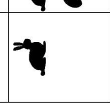
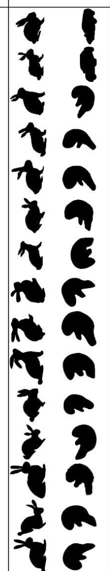
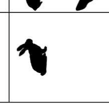
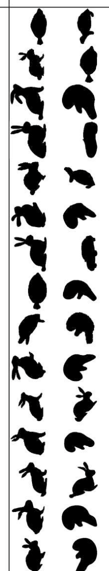
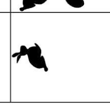
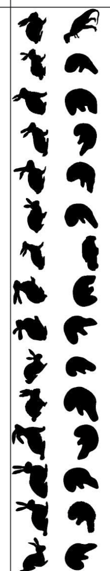
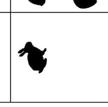
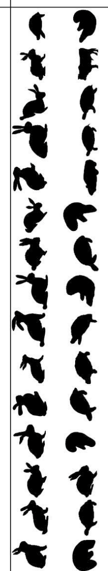
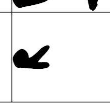
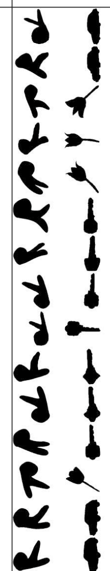
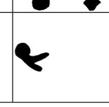
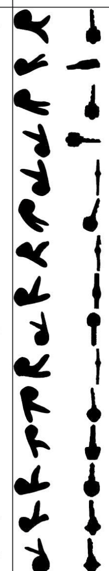
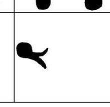
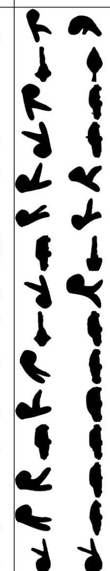
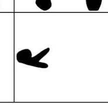
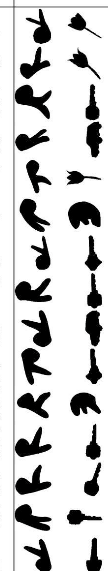
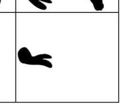
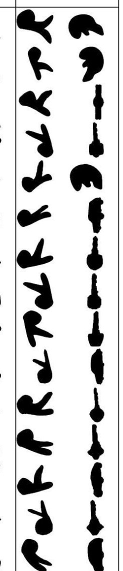
Query	Top 30 Matches	Correct
		15/15
		15/15
		14/15
		15/15
		15/15
		14/15
		9/15
		12/15
		14/15
		13/15

Table C.1: *continued*

Query	Top 30 Matches	Correct
		13/15
		15/15
		14/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

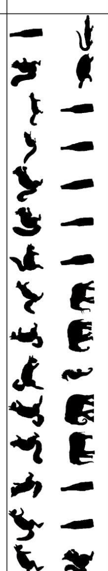
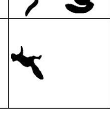
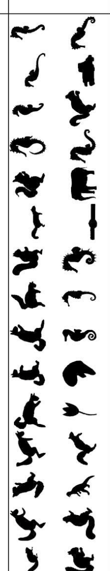
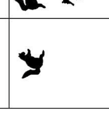
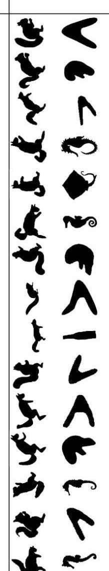
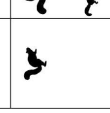
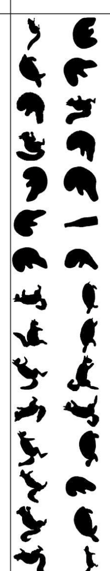
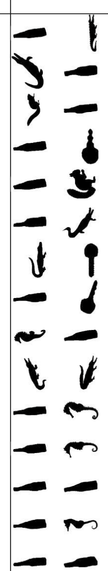
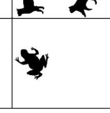
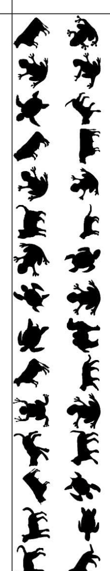
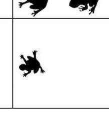
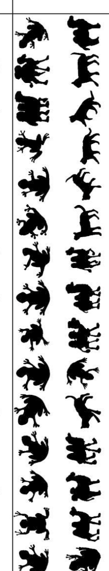
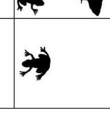
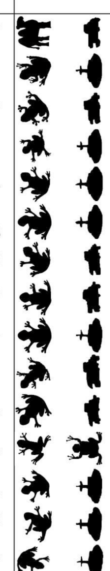
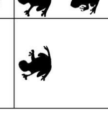
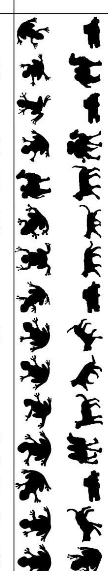
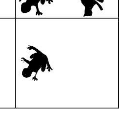
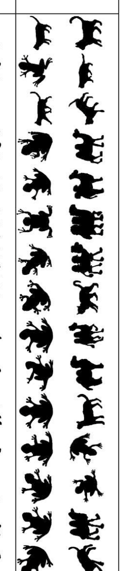
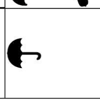
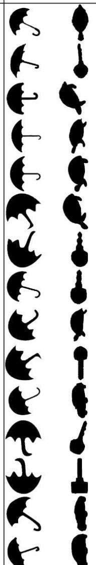
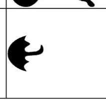
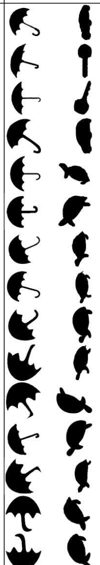
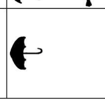
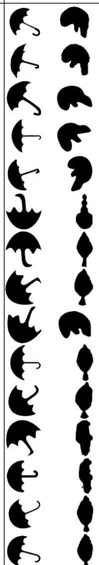
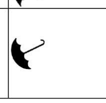
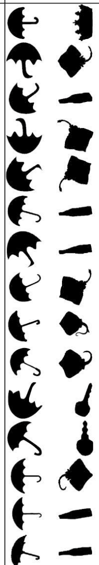
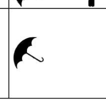
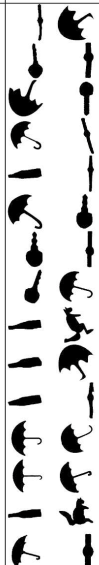
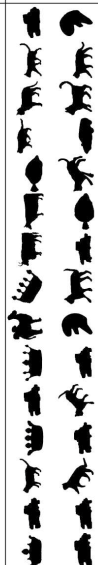
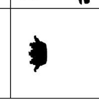
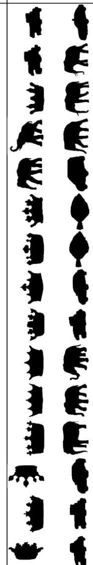
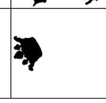
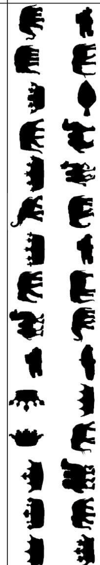
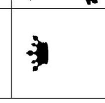
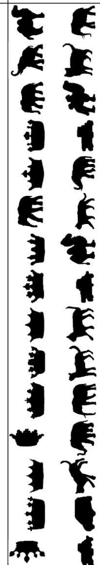
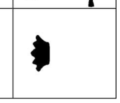
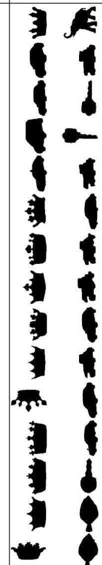
Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		14/15
		2/15
		9/15
		15/15
		15/15
		15/15
		15/15

Table C.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		11/15
		4/15
		11/15
		10/15
		11/15
		11/15

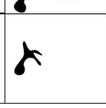
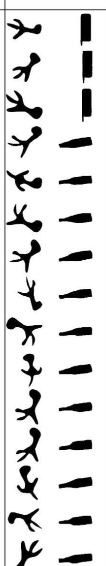
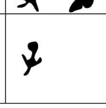
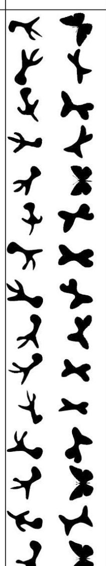
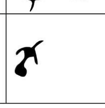
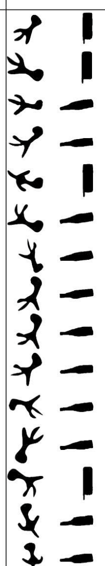
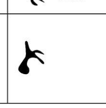
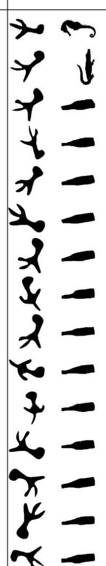
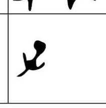
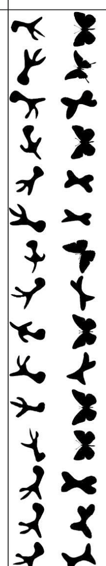
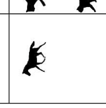
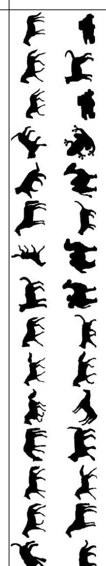
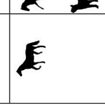
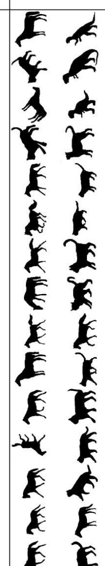
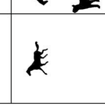
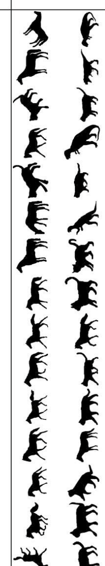
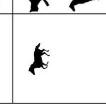
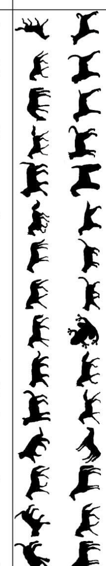
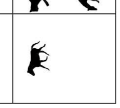
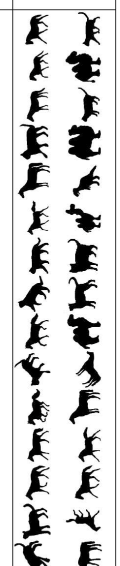
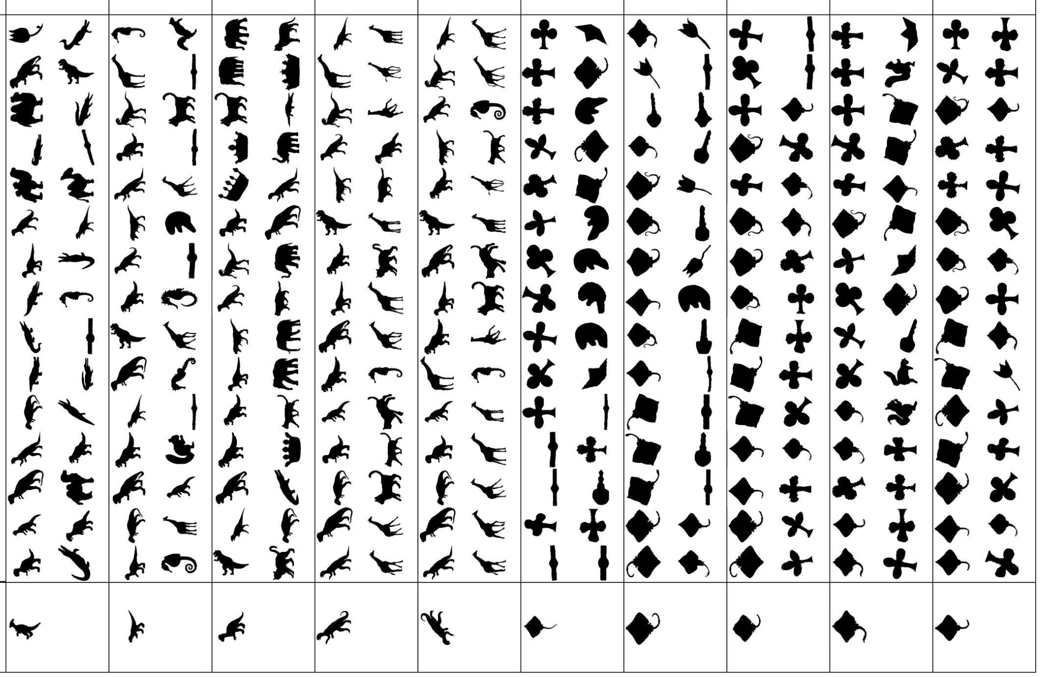
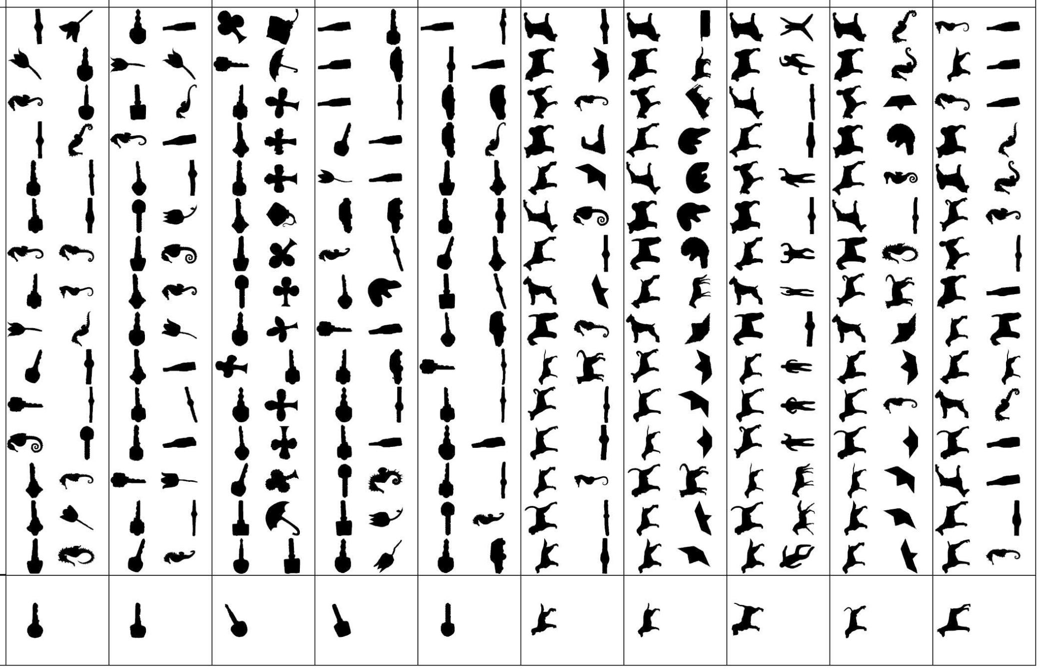
Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table C.1: *continued*

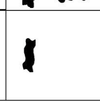
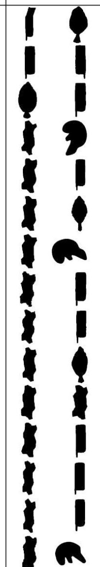
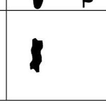
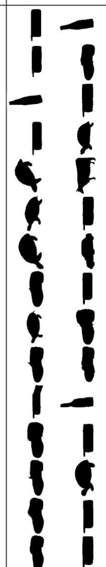
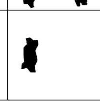
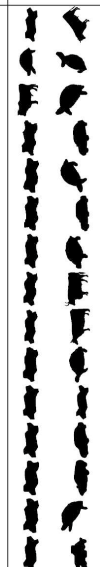
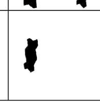
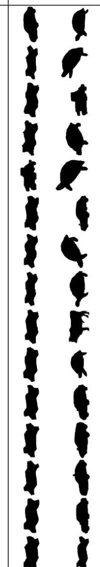
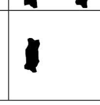
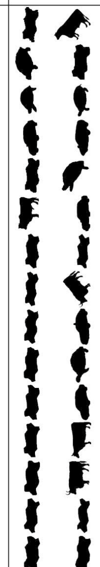
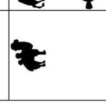
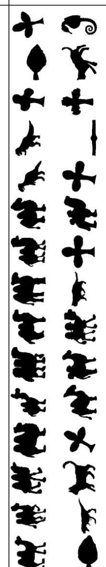
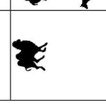
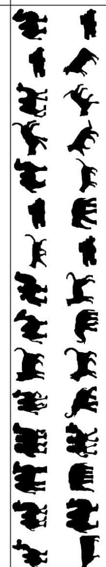
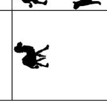
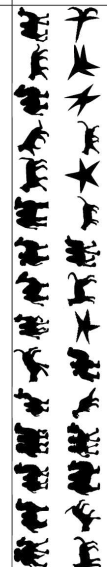
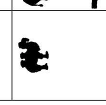
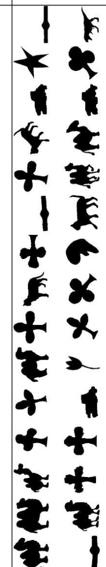
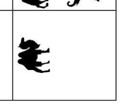
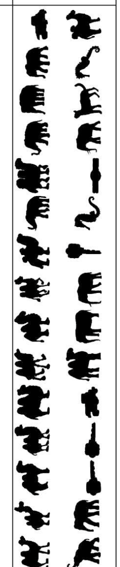


Query	Top 30 Matches	Correct
		12/15
		14/15
		13/15
		15/15
		14/15
		3/15
		15/15
		15/15
		10/15
		15/15



Query	Top 30 Matches	Correct
		11/15
		13/15
		15/15
		10/15
		13/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table C.1: *continued*

Query	Top 30 Matches	Correct
		13/15
		0/15
		14/15
		14/15
		15/15
		14/15
		12/15
		15/15
		7/15
		12/15

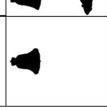
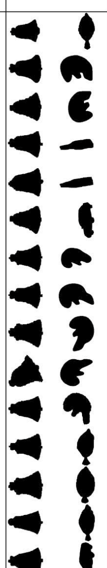
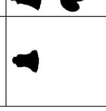
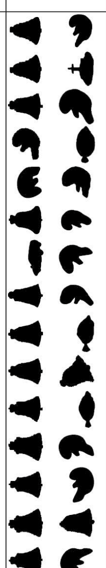
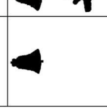
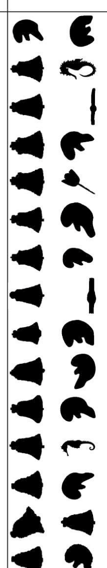
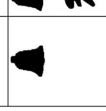
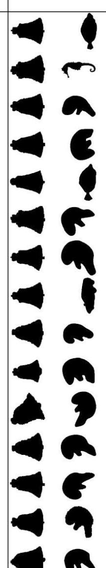
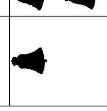
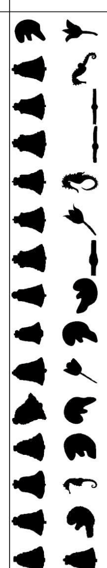
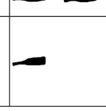
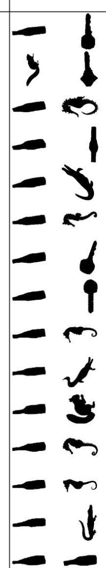
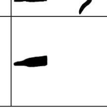
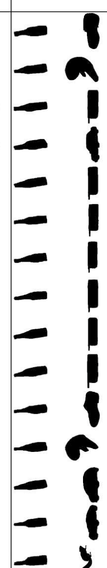
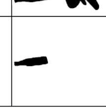
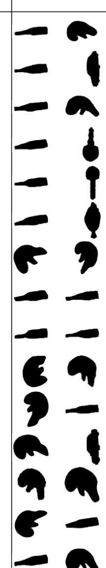
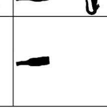
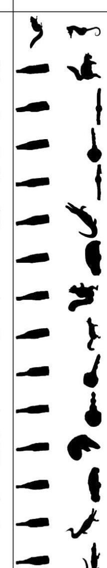
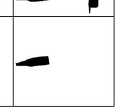
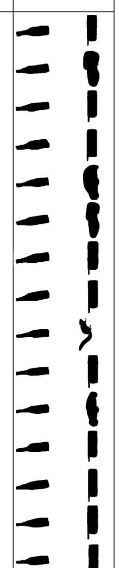
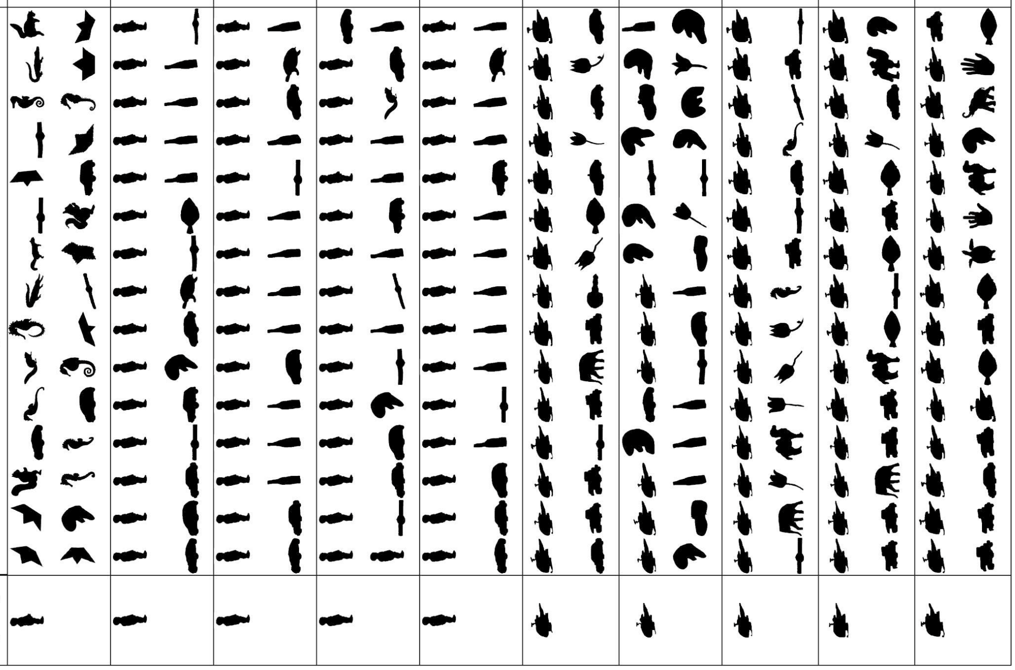
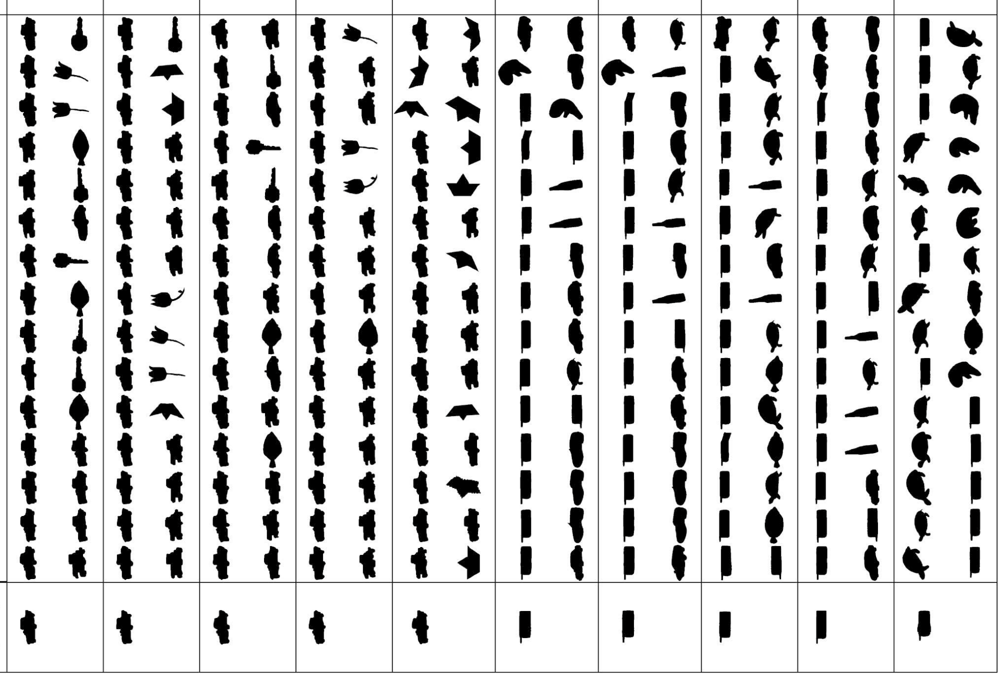
Query	Top 30 Matches	Correct
		15/15
		14/15
		15/15
		15/15
		15/15
		15/15
		15/15
		13/15
		14/15
		15/15

Table C.1: *continued*

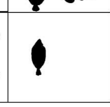
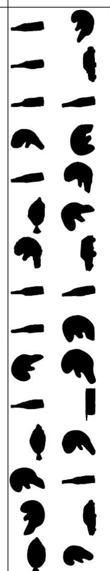
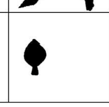
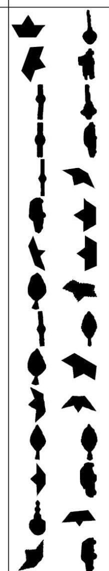
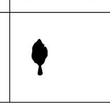
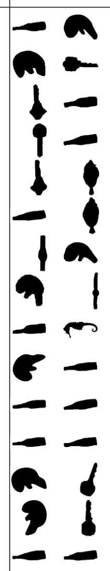
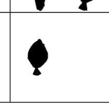
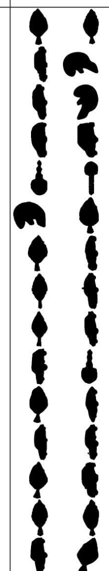
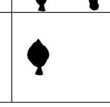
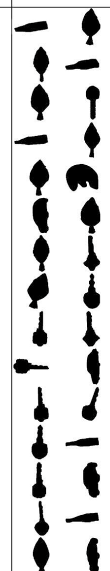
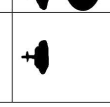
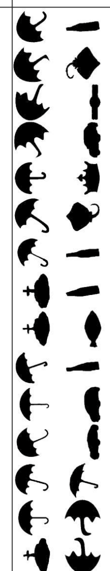
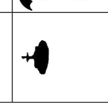
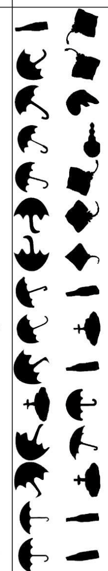
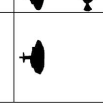
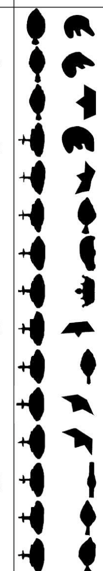
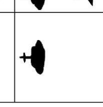
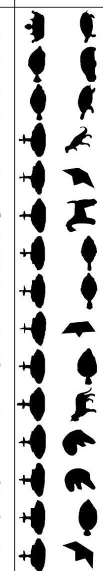
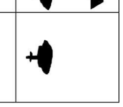
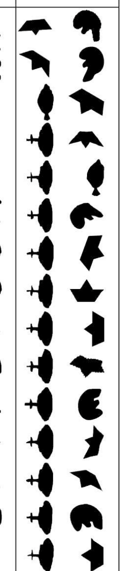


Query	Top 30 Matches	Correct
		0/15
		15/15
		15/15
		15/15
		15/15
		15/15
		6/15
		15/15
		15/15
		15/15



Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		14/15
		15/15
		14/15
		15/15
		15/15
		10/15

Table C.1: *continued*

Query	Top 30 Matches	Correct
		3/15
		5/15
		2/15
		11/15
		9/15
		3/15
		3/15
		12/15
		12/15
		12/15

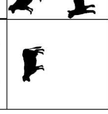
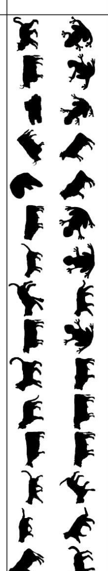
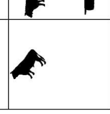
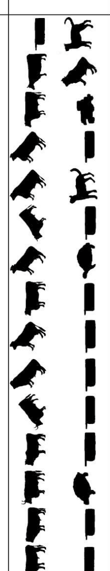
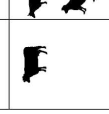
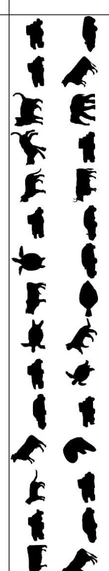
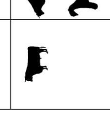
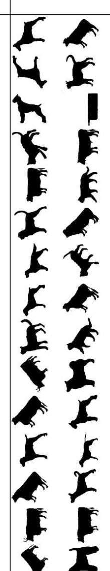
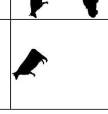
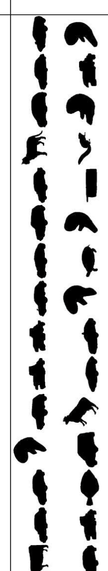
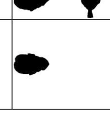
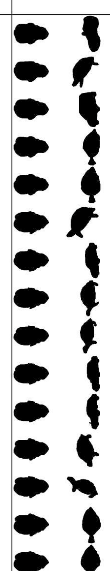
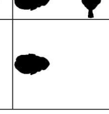
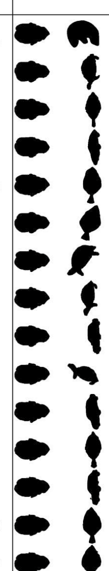
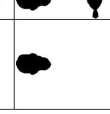
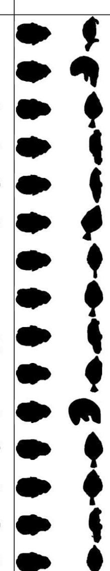
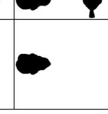
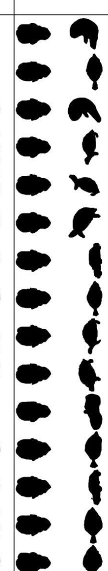
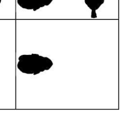
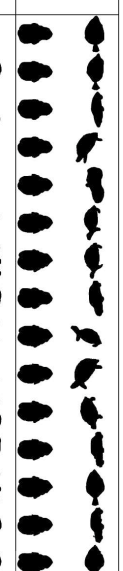
Query	Top 30 Matches	Correct
		11/15
		15/15
		6/15
		11/15
		2/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table C.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		0/15
		11/15
		9/15
		8/15
		8/15

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		12/15
		13/15
		10/15
		14/15
		13/15

Table C.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		10/15
		15/15
		11/15
		15/15
		14/15

Query	Top 30 Matches	Correct
		11/15
		13/15
		15/15
		8/15
		7/15
		14/15
		11/15
		14/15
		13/15
		3/15

APPENDIX D

CLASSIFICATION RESULTS

Table D.1: Some classification results.

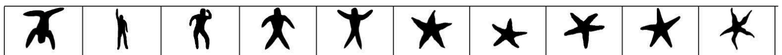
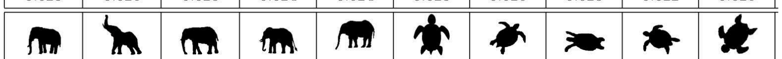
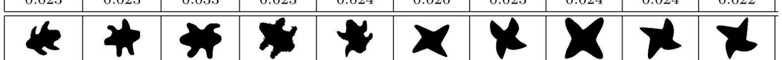
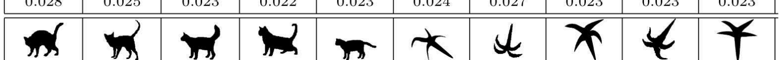
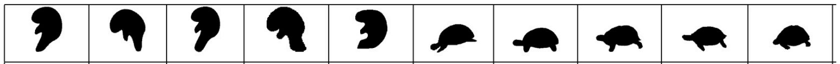
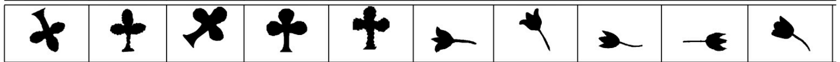
									
human	human	human	human	human	starfish	starfish	starfish	starfish	palm
0.171	0.162	0.189	0.154	0.166	0.186	0.165	0.142	0.170	0.069
starfish	dino	frog	dog	misc6	human	human	palm	frog	starfish
0.025	0.027	0.023	0.025	0.024	0.023	0.029	0.031	0.023	0.042
frog	camel	misc7	frog	frog	frog	club	frog	misc7	dumbbell
0.023	0.026	0.023	0.024	0.024	0.023	0.026	0.023	0.022	0.026
									
elephant	elephant	elephant	elephant	elephant	sea turtle	sea turtle	sea turtle	sea turtle	sea turtle
0.201	0.136	0.141	0.180	0.195	0.136	0.111	0.078	0.163	0.151
squirrel	dino	crocodile	crocodile	frog	misc2	cattle	tortoise	misc2	misc2
0.023	0.042	0.031	0.029	0.023	0.029	0.034	0.063	0.028	0.034
frog	misc4	cat	cat	misc7	misc7	tank	cat	cattle	dumbbell
0.023	0.023	0.026	0.024	0.023	0.024	0.025	0.030	0.024	0.024
									
misc1	misc1	sea turtle	misc1	misc1	butterfly	dumbbell	dumbbell	dumbbell	dumbbell
0.187	0.176	0.045	0.182	0.148	0.079	0.136	0.105	0.087	0.170
tank	frog	misc1	crown	misc5	misc7	butterfly	butterfly	butterfly	crown
0.025	0.023	0.034	0.024	0.024	0.051	0.029	0.041	0.050	0.022
frog	misc5	misc2	frog	frog	cattle	face	frog	frog	frog
0.023	0.023	0.033	0.023	0.024	0.026	0.023	0.024	0.024	0.022
									
misc2	misc2	misc2	misc2	misc2	misc3	misc3	misc3	misc3	misc3
0.057	0.109	0.161	0.186	0.161	0.154	0.075	0.191	0.183	0.181
frog	frog	sea turtle	misc5	sea turtle	dumbbell	flatfish	frog	dumbbell	frog
0.029	0.026	0.029	0.022	0.026	0.024	0.034	0.023	0.023	0.023
sea turtle	sea turtle	misc7	crown	misc5	misc7	tulip	elephant	frog	dumbbell
0.028	0.025	0.023	0.022	0.023	0.024	0.027	0.023	0.023	0.023
									
cat	cat	elephant	cat	cat	palm	palm	palm	palm	palm
0.122	0.058	0.072	0.081	0.168	0.169	0.170	0.077	0.200	0.080
sea turtle	camel	cat	sea turtle	elephant	starfish	giraffe	starfish	starfish	starfish
0.036	0.035	0.050	0.040	0.042	0.024	0.031	0.033	0.022	0.051
cattle	horse	horse	cattle	cattle	giraffe	frog	dumbbell	giraffe	frog
0.034	0.033	0.034	0.030	0.028	0.023	0.022	0.025	0.022	0.025
									
hand	hand	hand	hand	hand	papership	papership	papership	papership	papership
0.178	0.188	0.177	0.171	0.155	0.171	0.072	0.153	0.125	0.157
teddybear	teddybear	camel	helicopter	helicopter	dog	car2	tank	dog	tank
0.025	0.025	0.023	0.027	0.028	0.025	0.045	0.026	0.031	0.026
club	frog	teddybear	frog	frog	dino	dog	frog	tank	frog
0.023	0.023	0.023	0.023	0.024	0.024	0.031	0.024	0.027	0.024

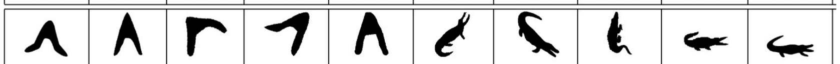
Table D.1: *continued*



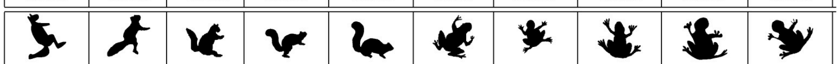
misc4 0.170	misc4 0.194	misc4 0.151	misc4 0.062	misc4 0.129	tortoise 0.125	tortoise 0.146	tortoise 0.123	tortoise 0.142	tank 0.046
tomb 0.026	fish 0.024	fish 0.031	car2 0.044	seahorse 0.027	fish 0.037	shoe 0.026	car2 0.034	tank 0.029	face 0.033
car2 0.024	frog 0.023	crocodile 0.024	fish 0.038	car2 0.026	rabbit 0.025	misc6 0.025	fish 0.026	sea turtle 0.026	cattle 0.029



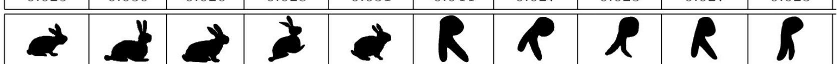
club 0.128	club 0.219	club 0.139	club 0.148	club 0.110	tulip 0.061	tulip 0.190	tulip 0.173	tulip 0.117	tulip 0.121
tulip 0.031	dino 0.022	flatfish 0.056	seahorse 0.031	giraffe 0.029	flatfish 0.034	misc3 0.035	misc3 0.036	misc3 0.032	key 0.037
seahorse 0.026	elephant 0.022	frog 0.023	tomb 0.025	tulip 0.029	club 0.027	papership 0.026	papership 0.025	starfish 0.029	bell 0.027



misc5 0.189	misc5 0.114	misc5 0.131	misc5 0.188	misc5 0.147	crocodile 0.258	crocodile 0.183	seahorse 0.052	crocodile 0.194	crocodile 0.209
frog 0.022	seahorse 0.031	seahorse 0.030	frog 0.023	giraffe 0.024	frog 0.022	misc7 0.025	squirrel 0.039	frog 0.024	squirrel 0.027
cellphone 0.022	giraffe 0.026	frog 0.024	cellphone 0.022	frog 0.023	dumbbell 0.022	rabbit 0.023	crocodile 0.034	dumbbell 0.023	baby 0.024



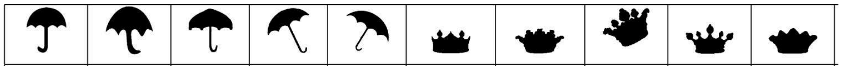
squirrel 0.161	squirrel 0.123	squirrel 0.119	squirrel 0.105	bottle 0.056	camel 0.054	frog 0.118	frog 0.159	frog 0.141	frog 0.186
baby 0.027	dog 0.045	dog 0.030	tortoise 0.034	seahorse 0.033	frog 0.052	sea turtle 0.029	tank 0.024	camel 0.028	dumbbell 0.023
umbrella 0.025	dino 0.030	umbrella 0.026	club 0.028	elephant 0.031	horse 0.044	hand 0.027	tomb 0.023	sea turtle 0.027	misc5 0.023



rabbit 0.119	rabbit 0.175	rabbit 0.079	rabbit 0.150	rabbit 0.118	misc6 0.193	misc6 0.116	misc6 0.159	misc6 0.195	misc6 0.142
shoe 0.027	tomb 0.024	fish 0.036	tomb 0.024	helicopter 0.030	frog 0.023	bottle 0.029	brick 0.026	car1 0.023	rabbit 0.024
helicopter 0.026	crocodile 0.022	tomb 0.033	helicopter 0.024	shoe 0.030	misc7 0.023	misc7 0.025	frog 0.024	teddybear 0.023	helicopter 0.024

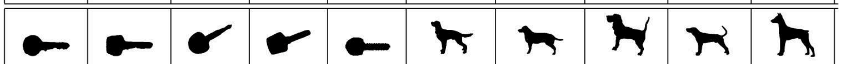
misc7 0.083	misc7 0.177	misc7 0.135	misc7 0.125	misc7 0.173	horse 0.098	horse 0.195	horse 0.109	horse 0.082	horse 0.087
butterfly 0.038	butterfly 0.024	butterfly 0.028	butterfly 0.029	butterfly 0.024	cat 0.050	misc7 0.022	giraffe 0.029	cat 0.065	cat 0.054
fish 0.027	crown 0.022	cattle 0.024	cattle 0.024	hand 0.022	frog 0.024	misc5 0.022	frog 0.025	misc5 0.024	elephant 0.028

Table D.1: *continued*



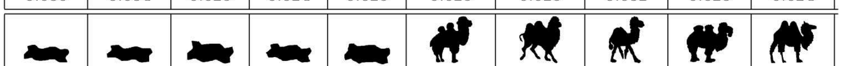
umbrella	umbrella	umbrella	umbrella	umbrella	crown	crown	crown	crown	crown
0.201	0.200	0.089	0.201	0.080	0.055	0.148	0.063	0.131	0.145
elephant	elephant	fish	car2	elephant	tank	car1	camel	camel	tank
0.023	0.023	0.030	0.031	0.034	0.048	0.026	0.040	0.027	0.024
frog	frog	rabbit	bell	bell	cattle	tank	hand	hand	face
0.023	0.023	0.028	0.027	0.032	0.039	0.025	0.032	0.025	0.024

dino	dino	dino	dino	dino	flatfish	flatfish	flatfish	flatfish	flatfish
0.081	0.086	0.069	0.140	0.113	0.088	0.085	0.190	0.134	0.185
crocodile	tulip	crown	horse	giraffe	bell	key	club	club	misc3
0.046	0.032	0.035	0.026	0.028	0.034	0.040	0.025	0.031	0.023
camel	giraffe	camel	giraffe	horse	club	misc1	frog	bell	frog
0.032	0.028	0.033	0.025	0.025	0.026	0.029	0.023	0.025	0.023



seahorse	key	flatfish	key	key	dog	dog	dog	dog	dog
0.052	0.100	0.077	0.245	0.247	0.152	0.123	0.166	0.142	0.041
key	tulip	key	tulip	frog	giraffe	squirrel	club	umbrella	tomb
0.046	0.049	0.044	0.040	0.021	0.025	0.029	0.029	0.027	0.034
watch	car1	bell	frog	misc7	misc5	misc6	frog	giraffe	brick
0.027	0.025	0.026	0.021	0.021	0.025	0.025	0.024	0.026	0.033

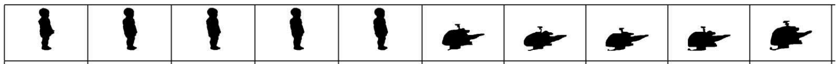
bell	bell	bell	bell	bell	bottle	bottle	fish	bottle	bottle
0.096	0.045	0.114	0.159	0.100	0.167	0.154	0.036	0.185	0.164
tomb	shoe	misc4	car2	frog	watch	car2	key	frog	frog
0.037	0.035	0.026	0.027	0.025	0.026	0.029	0.033	0.024	0.024
umbrella	car2	frog	frog	car2	misc6	frog	misc6	misc7	dumbbell
0.030	0.034	0.025	0.024	0.025	0.025	0.025	0.032	0.023	0.024



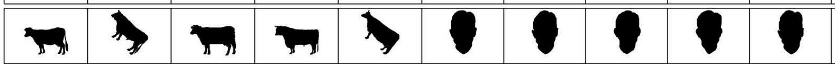
brick	cellphone	brick	brick	brick	camel	camel	camel	camel	camel
0.105	0.085	0.177	0.186	0.166	0.130	0.204	0.110	0.067	0.136
tortoise	shoe	tank	misc7	tank	teddybear	hand	horse	misc1	giraffe
0.028	0.039	0.033	0.023	0.026	0.028	0.026	0.036	0.035	0.028
cellphone	brick	misc7	misc5	misc7	butterfly	teddybear	elephant	human	butterfly
0.024	0.032	0.023	0.023	0.023	0.027	0.022	0.036	0.033	0.025

car1	car1	car1	car1	car1	cellphone	cellphone	cellphone	cellphone	cellphone
0.080	0.175	0.081	0.151	0.117	0.128	0.131	0.086	0.170	0.075
cattle	misc7	tank	bell	bell	shoe	shoe	brick	frog	shoe
0.033	0.023	0.034	0.025	0.033	0.036	0.032	0.029	0.024	0.046
crown	frog	camel	misc7	crown	frog	frog	fish	dumbbell	tortoise
0.029	0.022	0.034	0.024	0.026	0.025	0.025	0.025	0.024	0.043

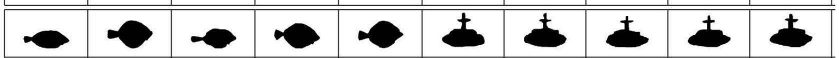
Table D.1: *continued*



baby	baby	baby	baby	baby	helicopter	camel	helicopter	helicopter	helicopter
0.069	0.176	0.173	0.177	0.147	0.193	0.033	0.150	0.137	0.212
papership	watch	frog	frog	car2	elephant	helicopter	elephant	hand	crown
0.047	0.026	0.024	0.024	0.027	0.023	0.030	0.030	0.031	0.023
squirrel	frog	dumbbell	dumbbell	bottle	tank	car1	misc7	elephant	frog
0.034	0.024	0.024	0.024	0.025	0.023	0.030	0.023	0.027	0.022

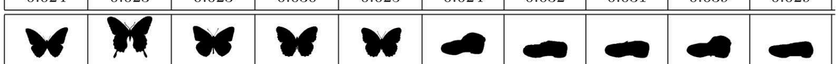


cattle	cattle	cattle	cattle	tortoise	shoe	face	face	face	face
0.062	0.161	0.084	0.060	0.052	0.088	0.169	0.172	0.055	0.149
frog	helicopter	cat	cat	cattle	face	cellphone	shoe	car2	starfish
0.031	0.025	0.032	0.043	0.042	0.046	0.024	0.024	0.045	0.024
tank	horse	tank	horse	crocodile	helicopter	umbrella	frog	shoe	misc3
0.029	0.023	0.032	0.028	0.033	0.031	0.024	0.023	0.041	0.024

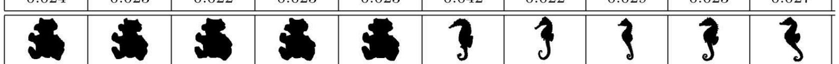


fish	key	car2	fish	key	tomb	umbrella	tomb	tomb	tomb
0.115	0.050	0.045	0.138	0.063	0.082	0.100	0.211	0.163	0.175
bottle	car2	bell	car2	car2	watch	tomb	dog	bell	papership
0.042	0.040	0.044	0.060	0.038	0.029	0.041	0.023	0.025	0.026
car2	papership	key	frog	fish	umbrella	rabbit	frog	car1	frog
0.027	0.031	0.036	0.024	0.032	0.028	0.028	0.022	0.024	0.023

giraffe	giraffe	giraffe	palm	giraffe	car2	brick	tank	car2	car2
0.128	0.171	0.154	0.064	0.137	0.220	0.050	0.053	0.126	0.167
frog	dino	frog	giraffe	horse	shoe	car2	car2	shoe	tortoise
0.024	0.028	0.023	0.059	0.028	0.028	0.040	0.051	0.041	0.031
misc7	tomb	misc7	camel	tomb	face	shoe	elephant	brick	brick
0.024	0.023	0.023	0.030	0.025	0.024	0.032	0.031	0.039	0.029



butterfly	butterfly	butterfly	butterfly	butterfly	car2	shoe	shoe	shoe	brick
0.068	0.141	0.168	0.138	0.104	0.045	0.204	0.112	0.165	0.092
dumbbell	misc7	hand	misc7	dumbbell	key	cellphone	cellphone	fish	car2
0.060	0.026	0.022	0.023	0.034	0.043	0.024	0.029	0.026	0.038
frog	frog	starfish	misc3	misc3	misc4	frog	tortoise	bell	cattle
0.024	0.023	0.022	0.023	0.023	0.042	0.022	0.029	0.023	0.027



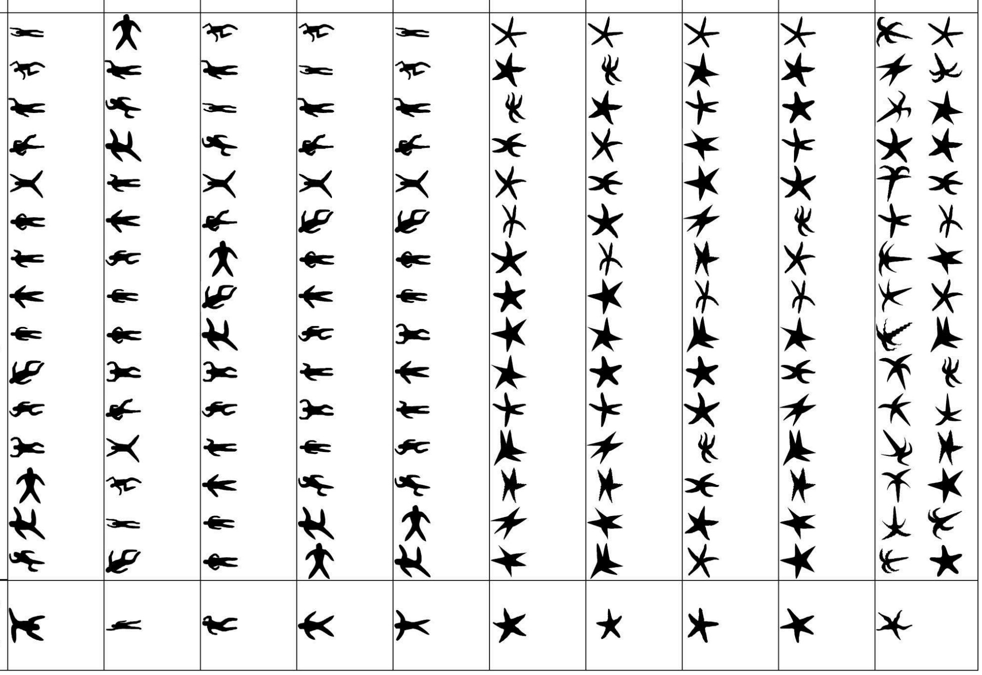
teddybear	teddybear	teddybear	teddybear	teddybear	watch	seahorse	watch	seahorse	seahorse
0.151	0.156	0.198	0.151	0.196	0.082	0.188	0.147	0.078	0.054
hand	camel	misc5	frog	car1	seahorse	key	seahorse	car2	watch
0.025	0.030	0.022	0.024	0.022	0.055	0.025	0.052	0.028	0.045
camel	hand	frog	crown	frog	frog	club	crocodile	misc4	giraffe
0.023	0.023	0.022	0.023	0.022	0.025	0.024	0.026	0.026	0.026

tank	tank	tank	tank	tank	watch	watch	watch	watch	watch
0.110	0.121	0.096	0.163	0.073	0.105	0.231	0.167	0.158	0.086
car1	shoe	helicopter	car1	shoe	seahorse	giraffe	key	seahorse	misc4
0.032	0.039	0.030	0.040	0.033	0.052	0.022	0.027	0.033	0.031
car2	tortoise	car1	helicopter	misc1	bottle	frog	papership	squirrel	fish
0.028	0.029	0.029	0.026	0.031	0.028	0.021	0.023	0.024	0.026

APPENDIX E

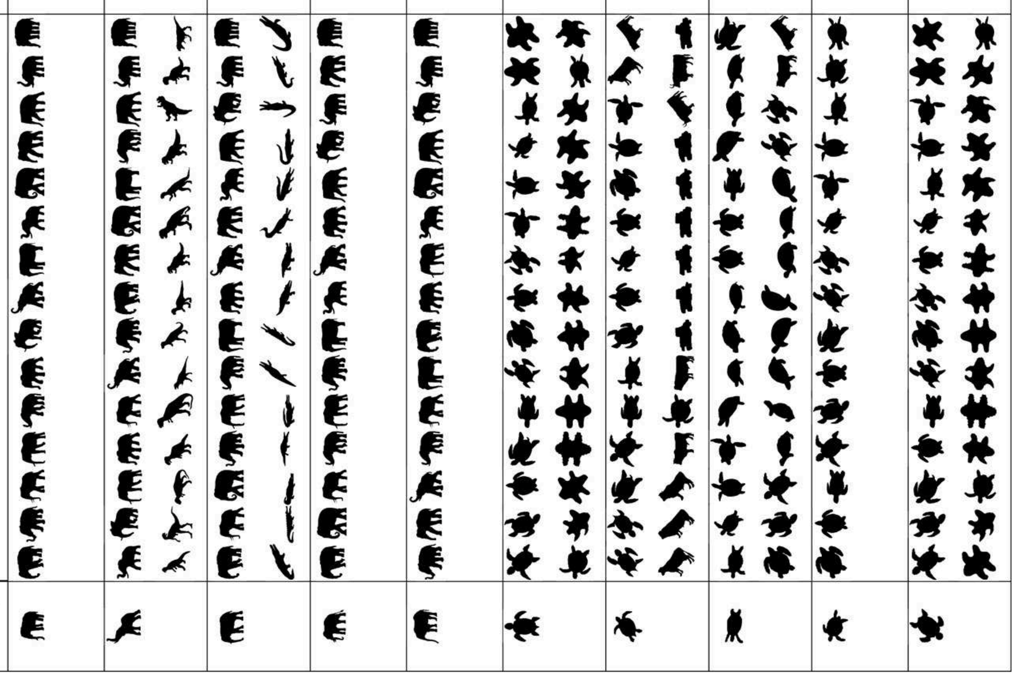
RETRIEVAL RESULTS OF CATEGORY-INFLUENCED MATCHING AFTER CLASSIFICATION

Table E.1: Retrieval results of category-influenced matching after classification.

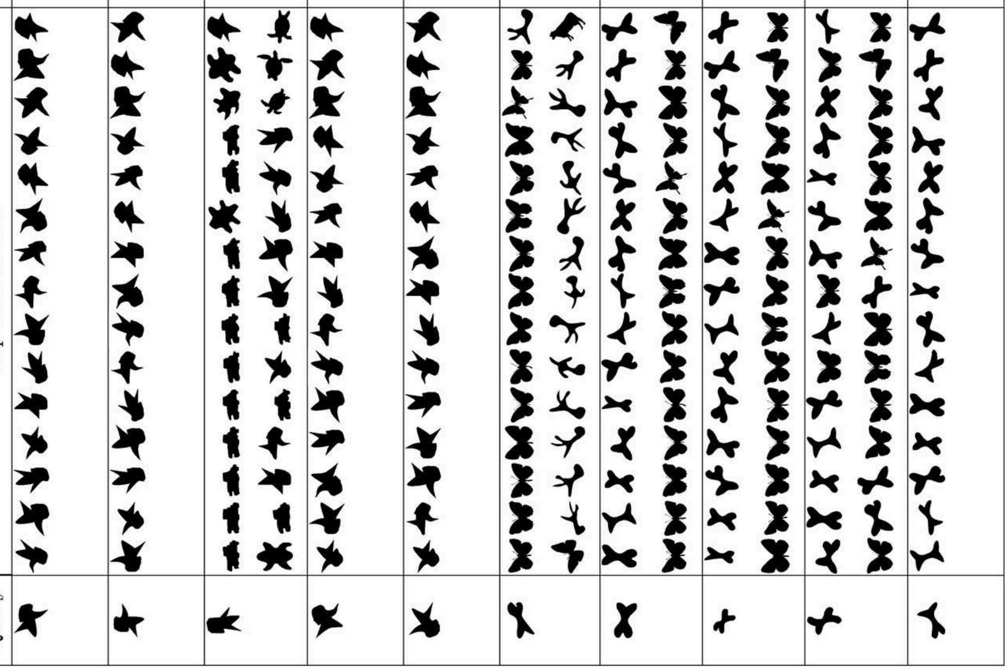


Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table E.1: *continued*



Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		14/15
		13/15
		15/15
		15/15



Query	Top 30 Matches	Correct
		15/15
		15/15
		9/15
		15/15
		15/15
		0/15
		15/15
		15/15
		15/15
		15/15

Table E.1: *continued*

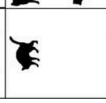
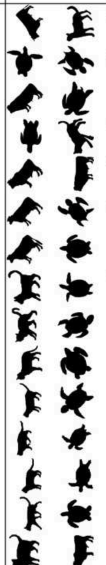
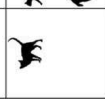
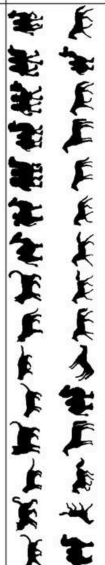
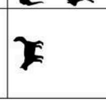
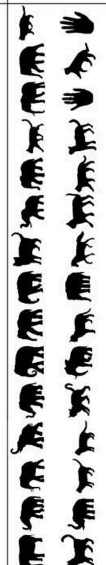
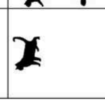
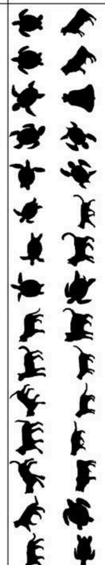
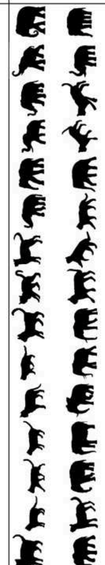
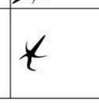
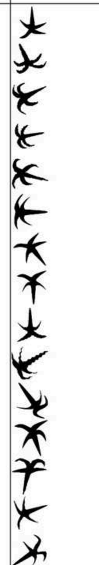
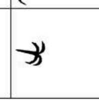
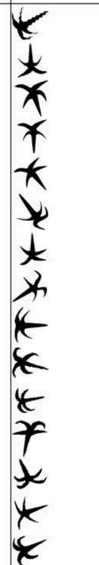
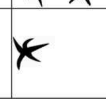
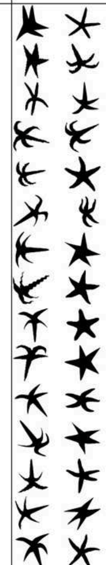
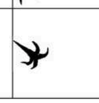
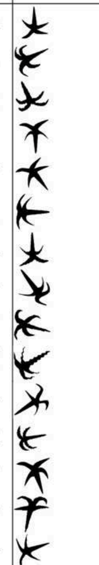
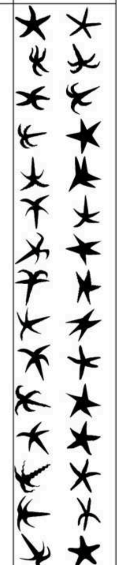
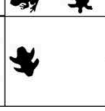
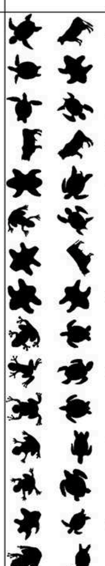
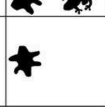
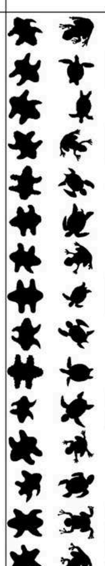
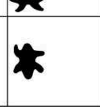
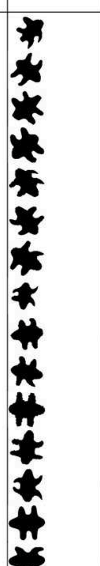
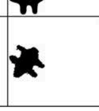
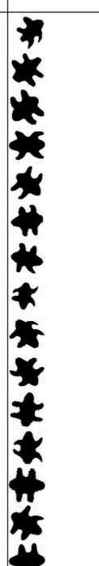
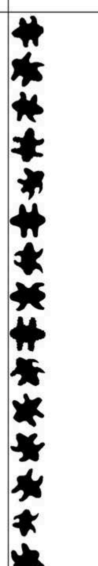
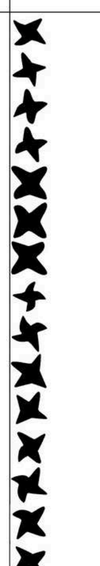
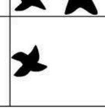
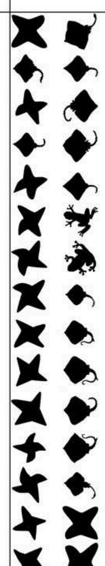
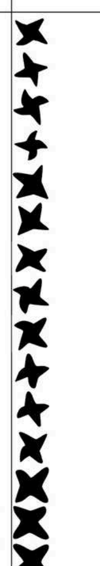
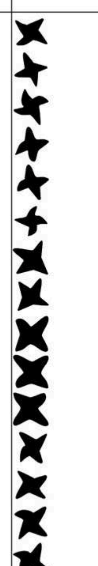
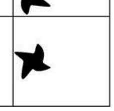
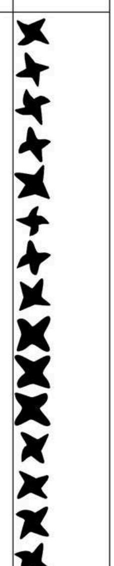
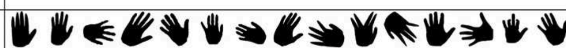
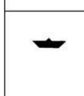
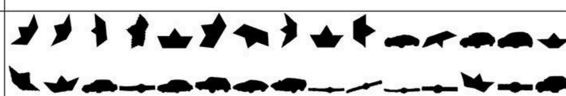
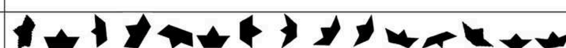
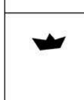
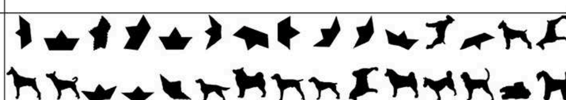
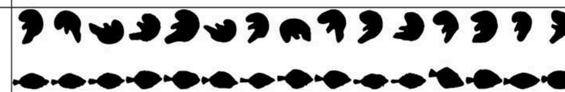
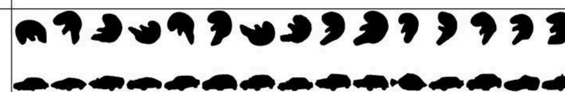
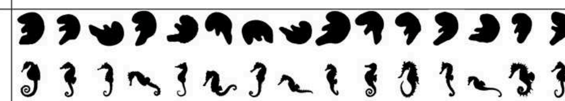
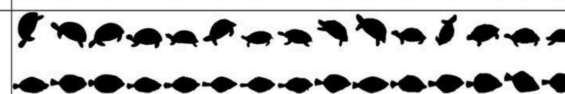
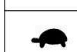
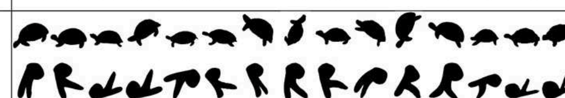
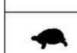
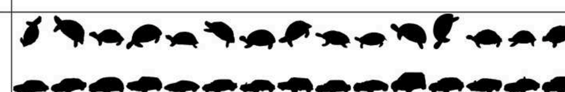
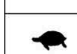
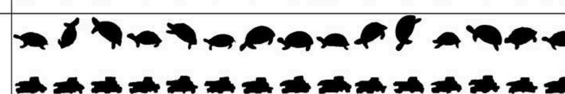
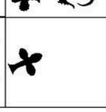
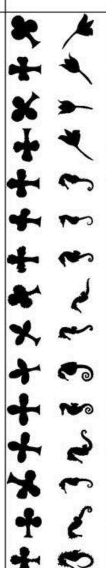
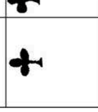
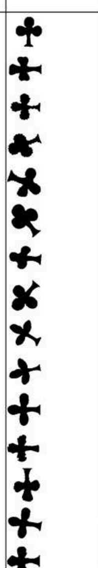
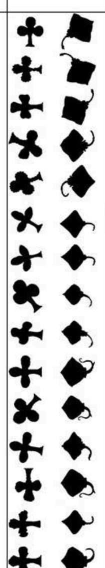
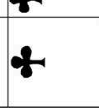
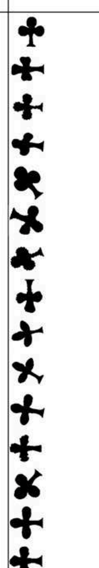
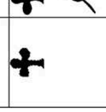
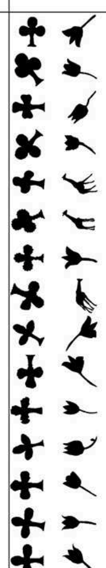
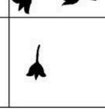
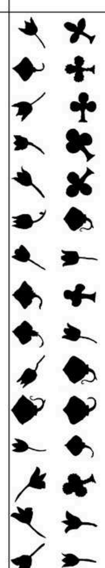
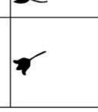
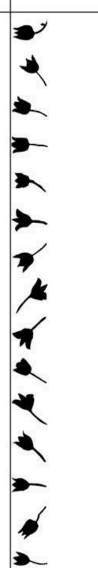
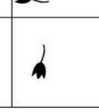
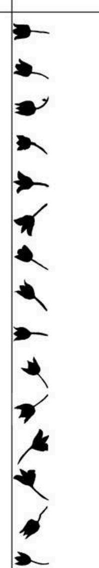
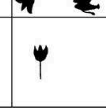
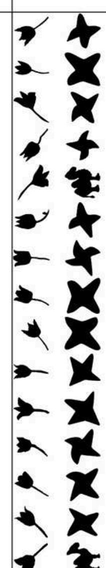
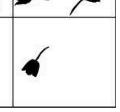
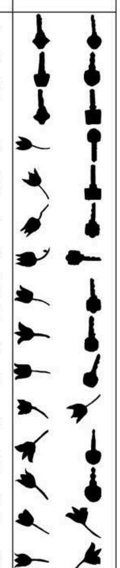
Query	Top 30 Matches	Correct
		10/15
		8/15
		12/15
		13/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		6/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table E.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		0/15

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

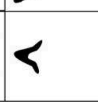
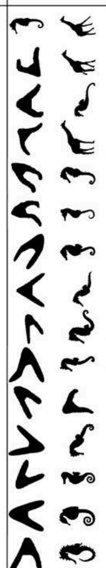
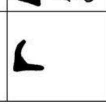
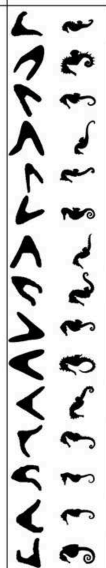
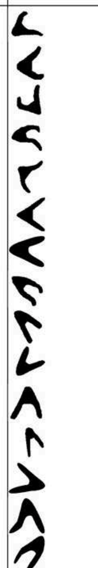
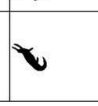
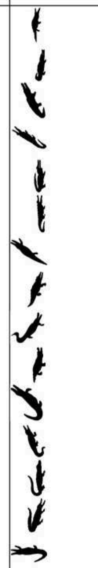
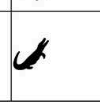
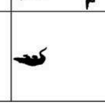
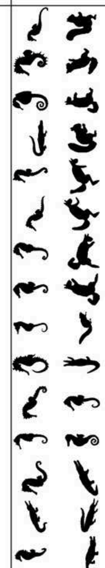
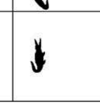
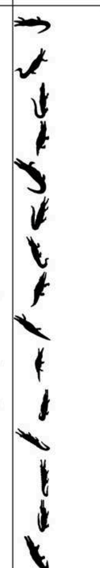
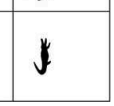
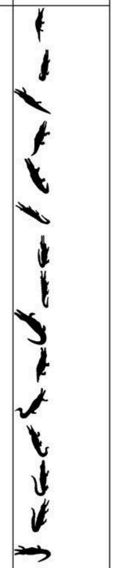
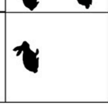
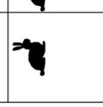
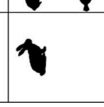
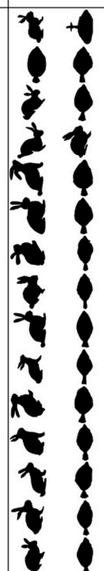
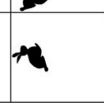
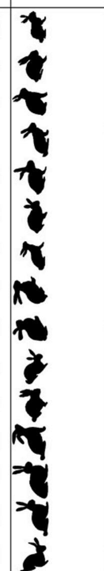
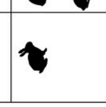
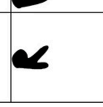
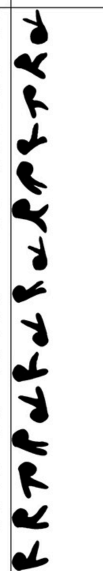
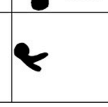
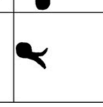
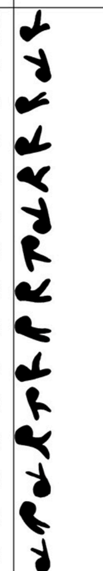
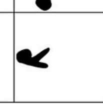
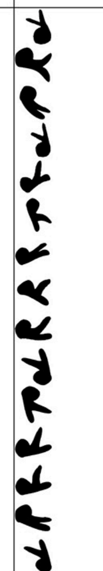
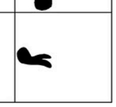
Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		6/15
		15/15
		15/15

Table E.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

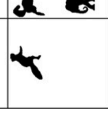
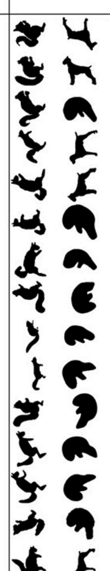
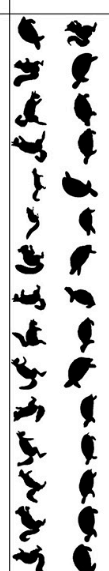
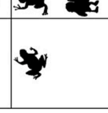
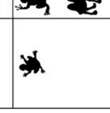
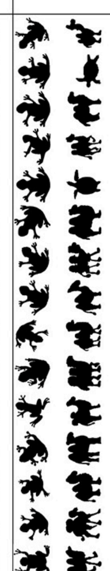
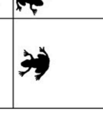
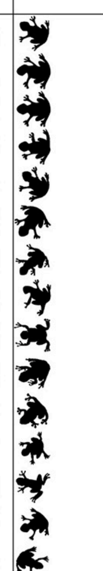
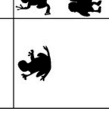
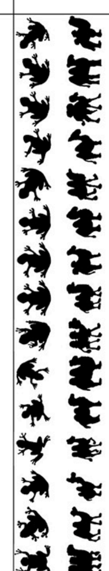
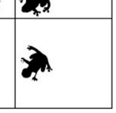
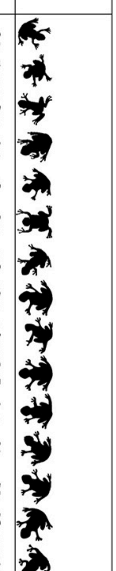
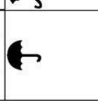
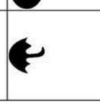
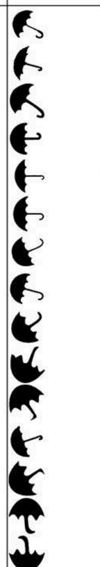
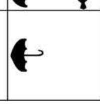
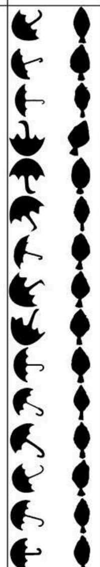
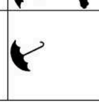
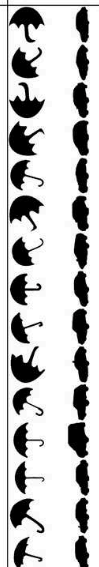
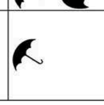
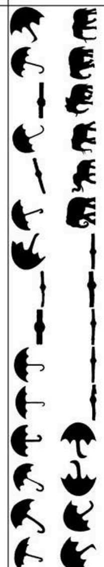
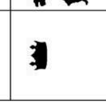
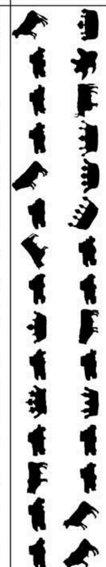
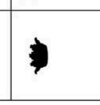
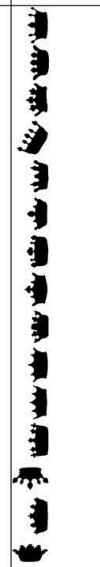
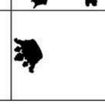
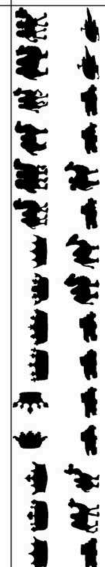
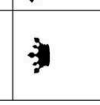
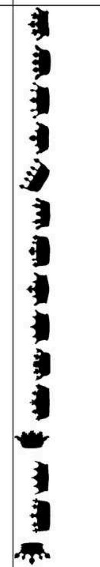
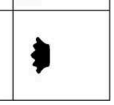
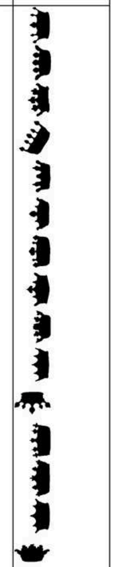
Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		0/15
		14/15
		15/15
		15/15
		15/15
		15/15

Table E.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		7/15
		15/15
		9/15
		15/15
		15/15

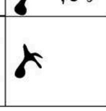
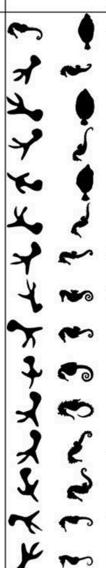
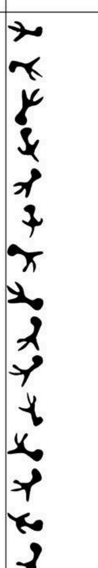
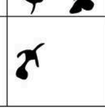
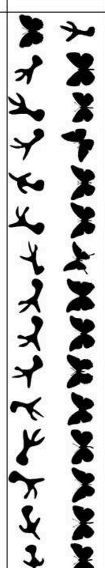
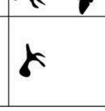
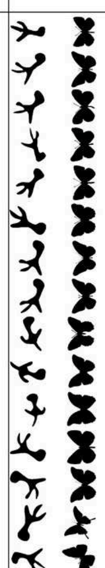
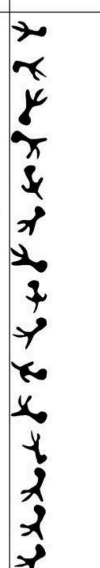
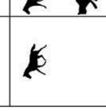
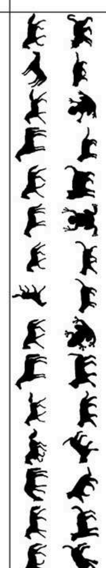
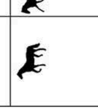
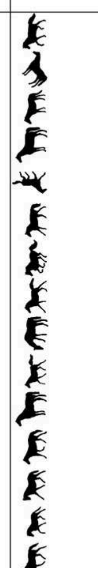
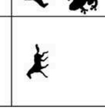
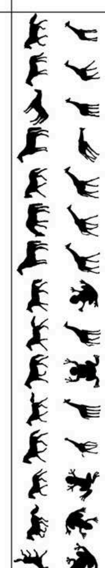
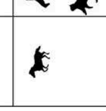
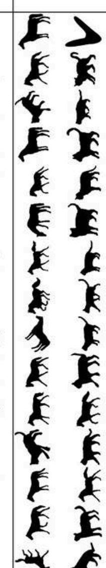
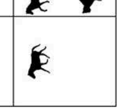
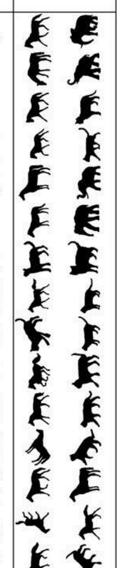
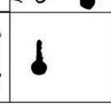
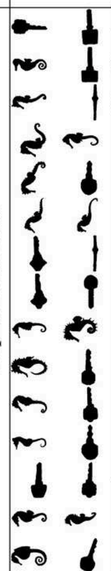
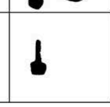
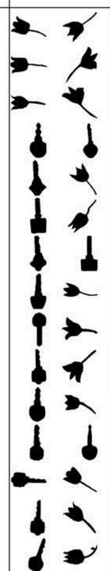
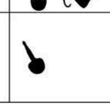
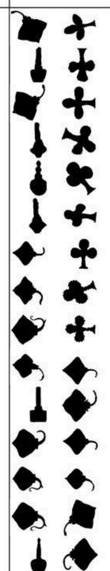
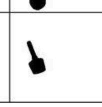
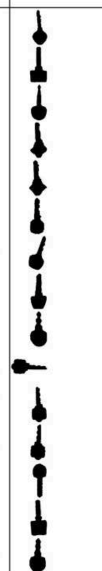
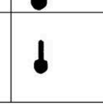
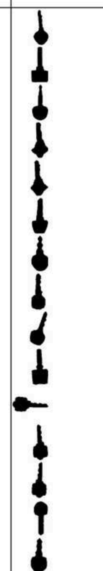
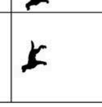
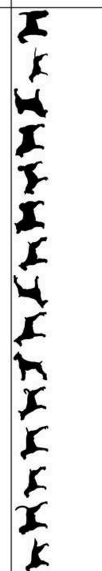
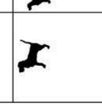
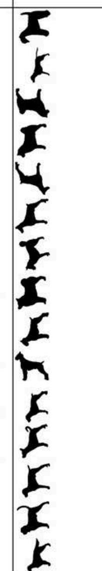
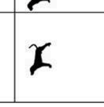
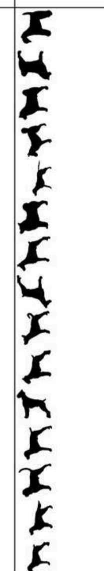
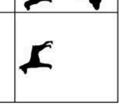
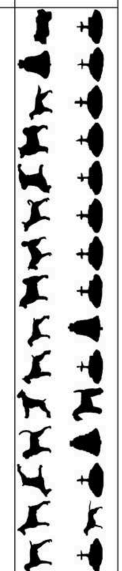
Query	Top 30 Matches	Correct
		14/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table E.1: *continued*

Query	Top 30 Matches	Correct
		13/15
		15/15
		6/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

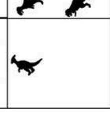
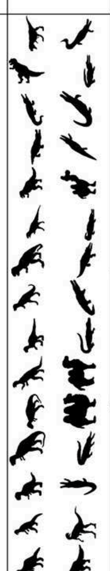
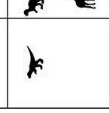
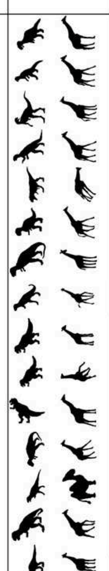
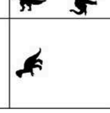
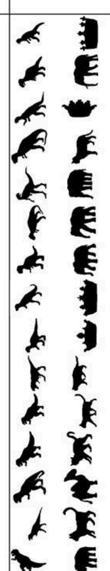
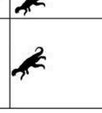
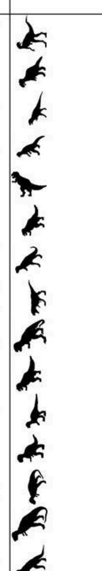
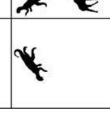
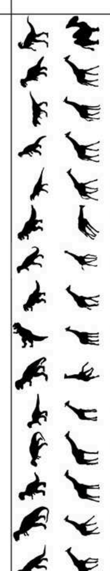
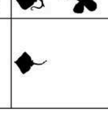
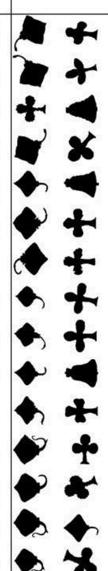
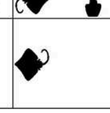
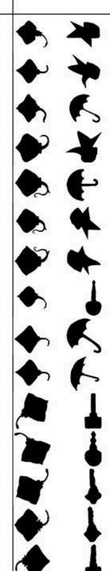
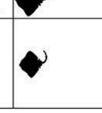
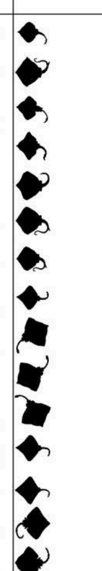
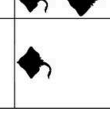
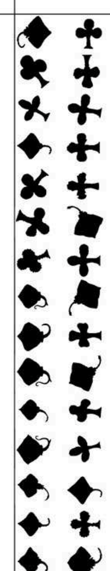
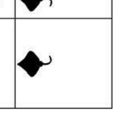
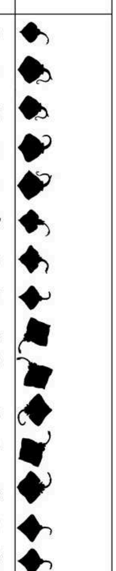
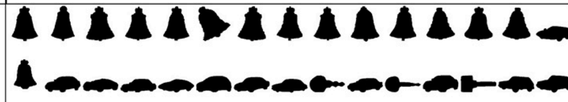
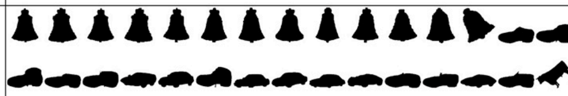
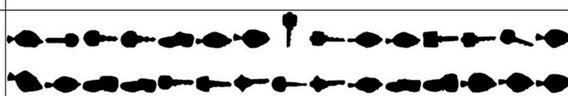
Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table E.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		13/15
		15/15
		15/15
		15/15
		15/15
		15/15
		0/15
		15/15
		15/15

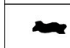
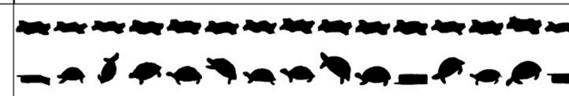
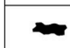
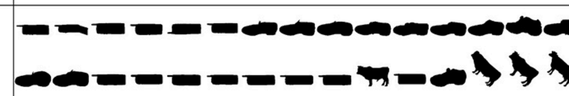
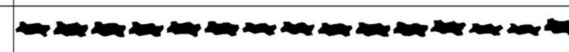
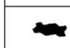
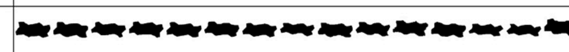
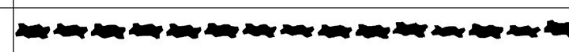
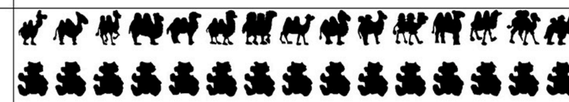
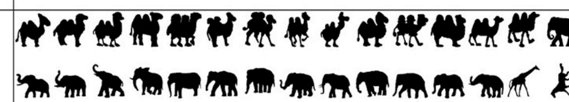
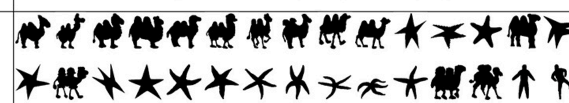
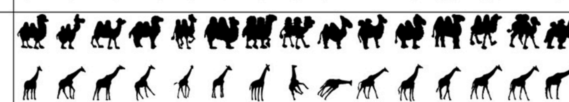
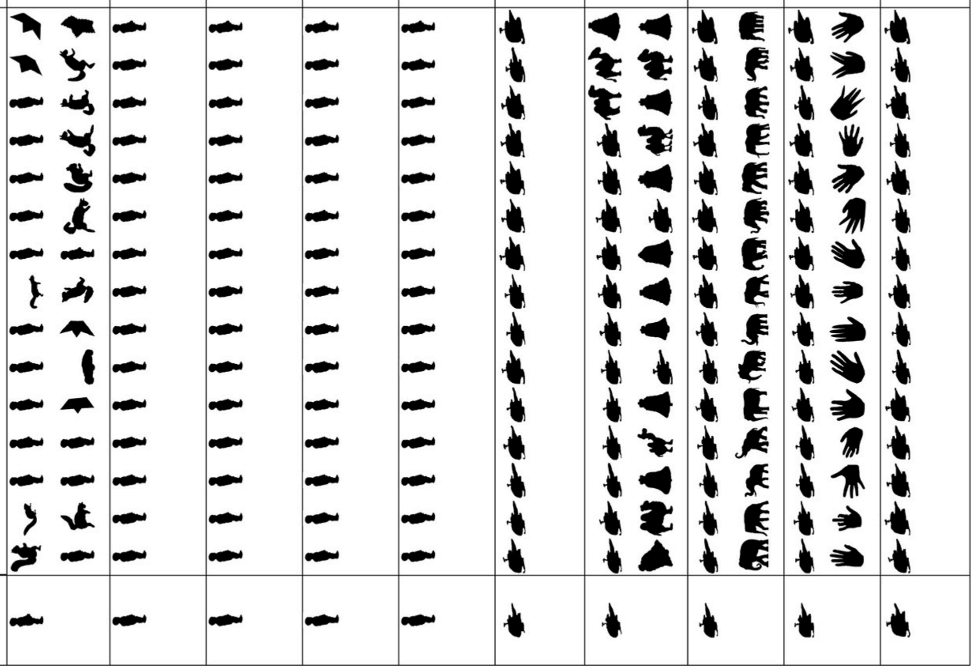
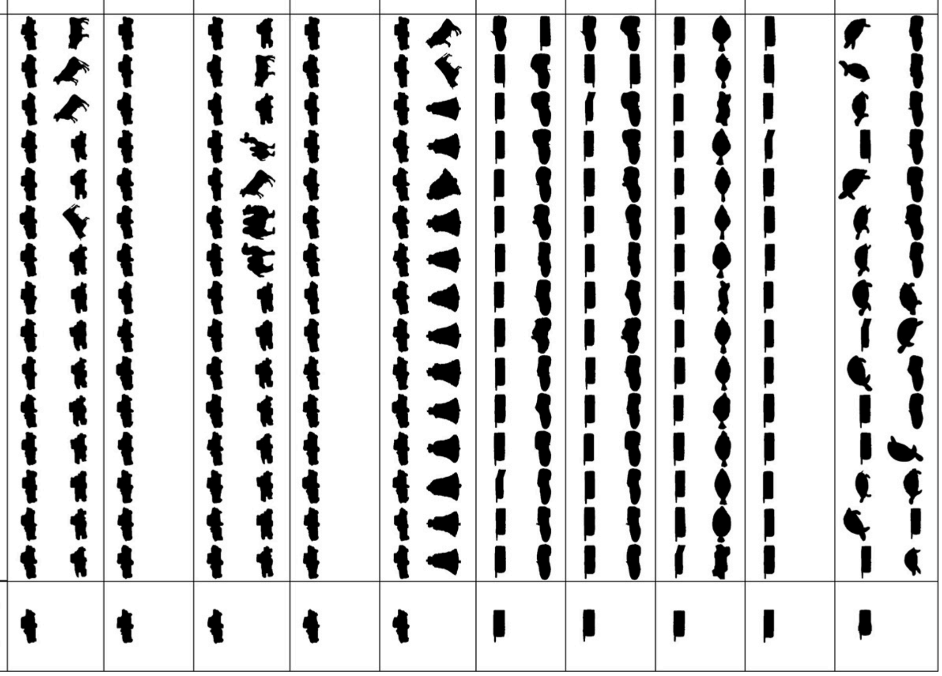
Query	Top 30 Matches	Correct
		15/15
		0/15
		15/15
		15/15
		15/15
		15/15
		15/15
		14/15
		14/15
		15/15

Table E.1: *continued*

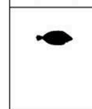
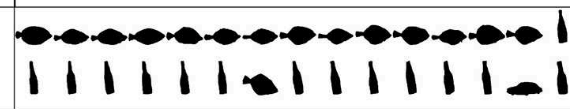
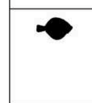
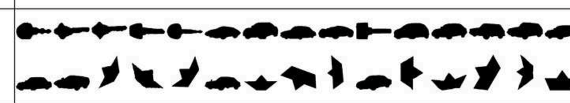
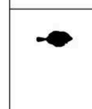
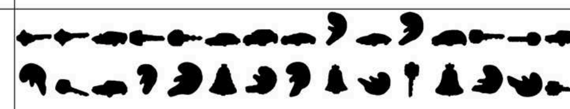
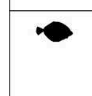
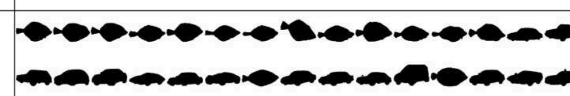
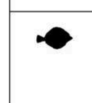
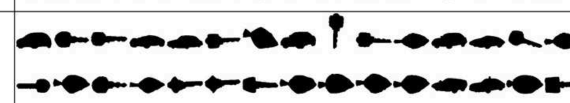
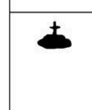
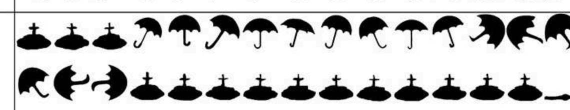
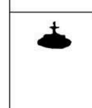
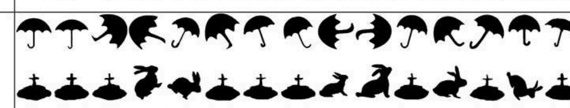
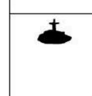
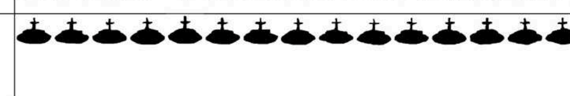
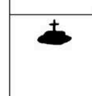
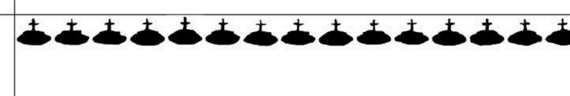
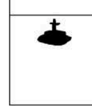
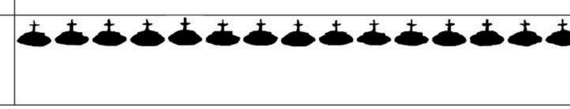


Query	Top 30 Matches	Correct
		14/15
		15/15
		15/15
		15/15
		15/15
		15/15
		14/15
		15/15
		15/15
		15/15



Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		6/15

Table E.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		0/15
		0/15
		15/15
		10/15
		14/15
		9/15
		15/15
		15/15
		15/15

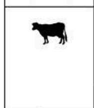
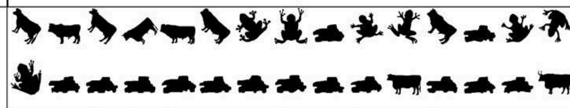
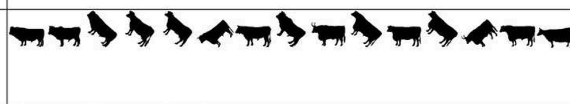
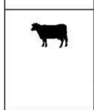
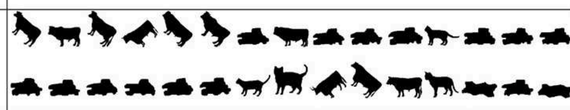
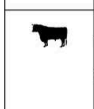
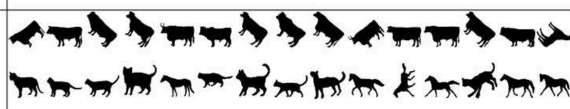
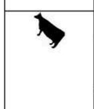
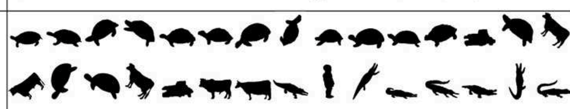
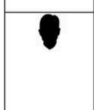
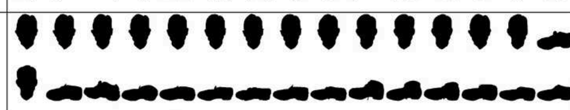
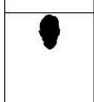
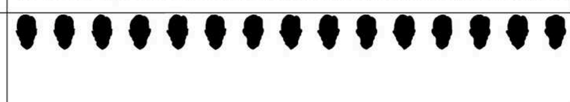
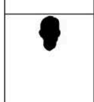
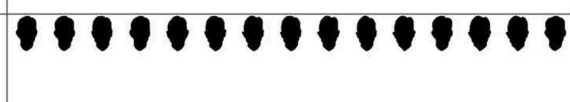
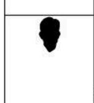
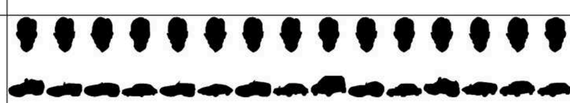
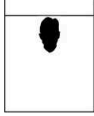
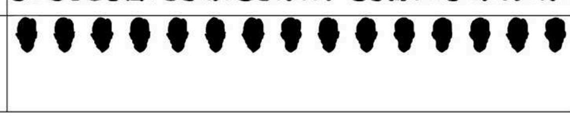
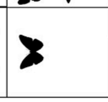
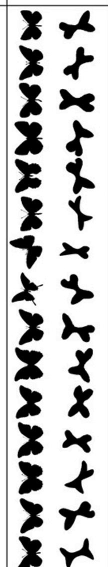
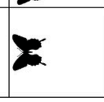
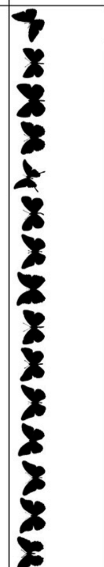
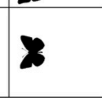
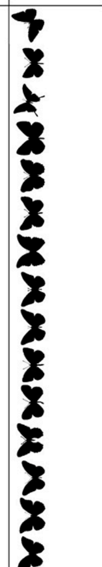
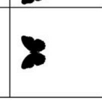
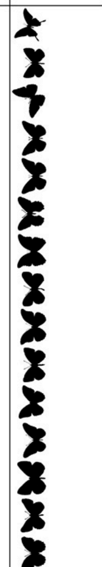
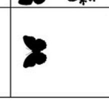
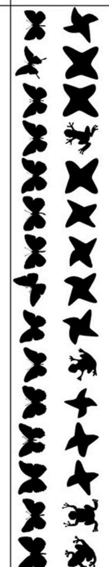
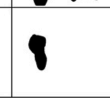
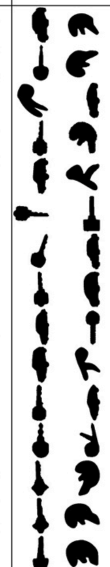
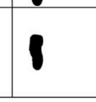
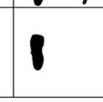
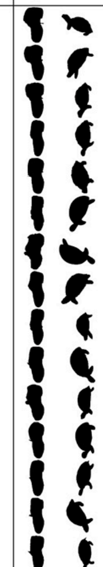
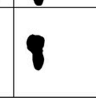
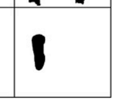
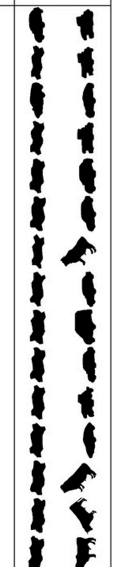
Query	Top 30 Matches	Correct
		9/15
		15/15
		10/15
		14/15
		5/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table E.1: *continued*

Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		0/15
		15/15
		15/15
		15/15
		0/15

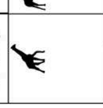
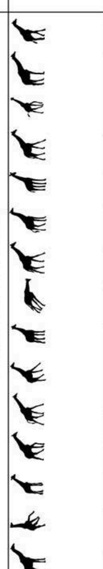
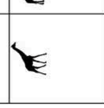
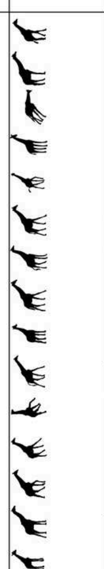
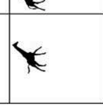
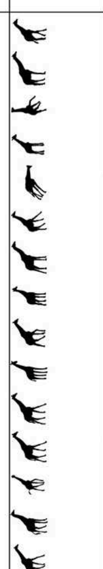
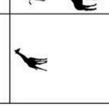
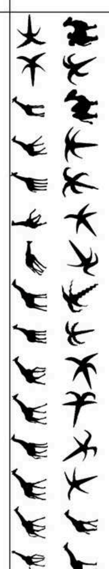
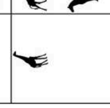
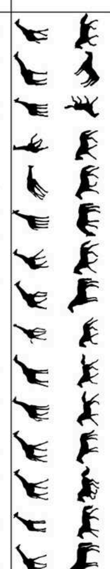
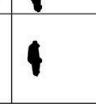
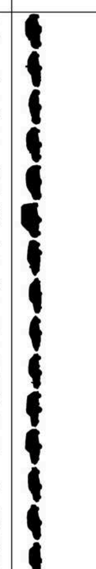
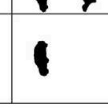
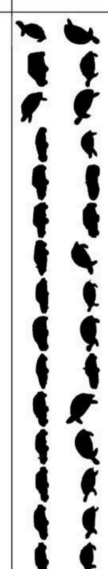
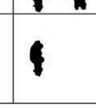
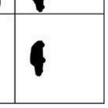
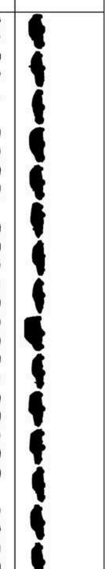
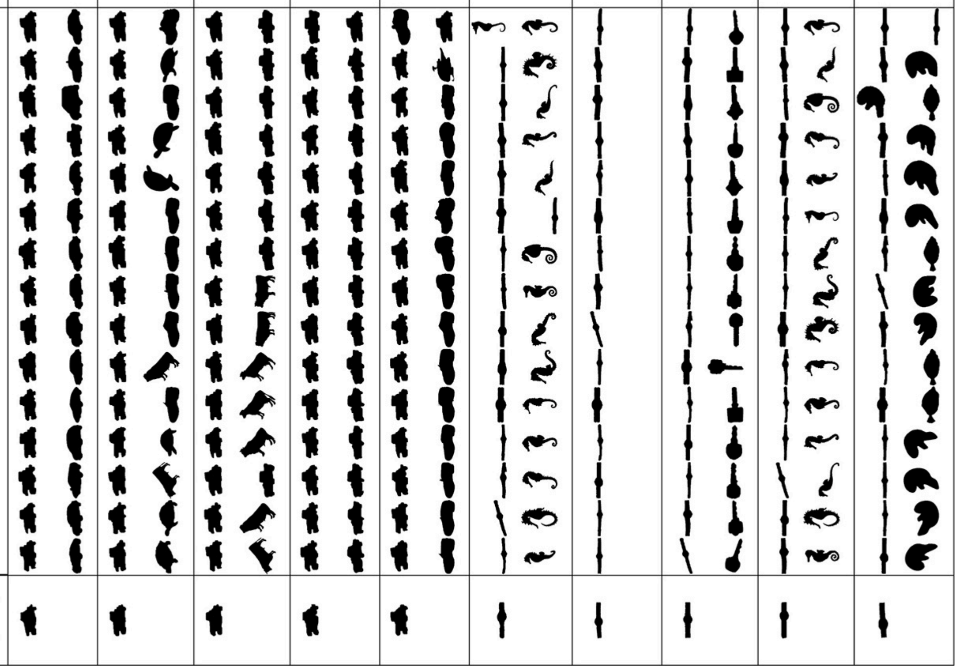
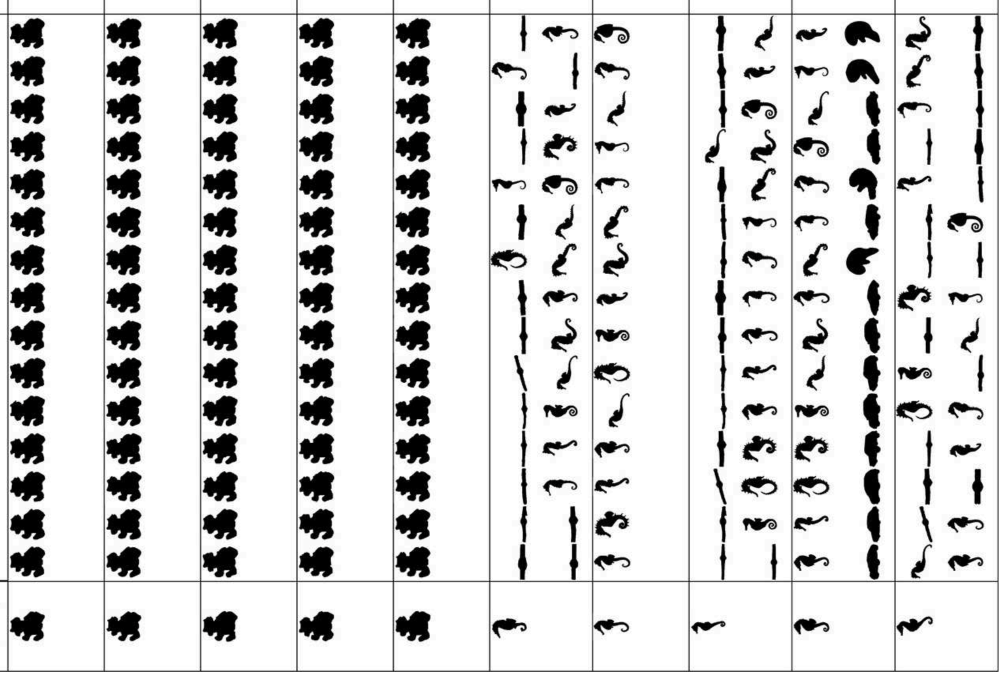
Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

Table E.1: *continued*



Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15



Query	Top 30 Matches	Correct
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15
		15/15

VITA

Personal Information

Surname, Name: Erdem, İbrahim Aykut

Date and Place of Birth: 21 March 1980, Ankara

Phone: (+90) 312 266 4577

Fax: (+90) 312 210 5544

E-mail: aykut@ceng.metu.edu.tr

Web: www.ceng.metu.edu.tr/~aykut

Research Interests

Computer vision (cognitive and computational aspects)

skeletons, shape matching, shape classification, similarity-based pattern recognition

Education

M.Sc., Dept. of Computer Engineering, METU, Turkey, July 2003

- Thesis Title: Vision-Based Human-Computer Interaction Using Laser Pointer
- Advisor: Volkan Atalay
- CGPA: 3.86 / 4.00 (High Honor)

B.Sc., Dept. of Computer Engineering, METU, Turkey, June 2001

- CGPA: 3.67 / 4.00 (High Honor)

Awards

- IEEE SIU 2004 Alper Atalay Best Student Paper Award (Third Prize), 2004
- Interpro Information and Communication Technologies Awards, Encouragement Award, 2002

Refereeing

- Additional Reviewer, ECCV 2006
- Additional Reviewer, ICCV 2005

Visiting Positions

Visiting Student **Sep 2007 - Nov 2007**

Computation Group, Architecture (visited George Stiny)

Massachusetts Institute of Technology, Cambridge, USA

Research Visit **Jul 2006**

Department of Mathematics (visited Jayant Shah)

Northeastern University, Boston, USA

Research Visit **Jul 2004 - Aug 2004**

Virginia Bioinformatics Institute (worked with Volkan Atalay and Rengul Cetin-Atalay)

Virginia Polytechnic Institute and State University, Virginia, USA

Teaching Experience

Teaching Assistant **Dec 2001 - Sep 2008**

Department of Computer Engineering,

Middle East Technical University, Ankara, TURKEY

courses include Image Processing, Dynamic Systems, Scientific Computing, Logic for CS and Computer Graphics.

Publications

International Journal Publications

E. Baseski, A. Erdem, S. Tari, Dissimilarity Between Two Skeletal Trees in a Context, *Pattern Recognition*, published online: 30 May 2008, doi:10.1016/j.patcog.2008.05.022

C. Aslan, A. Erdem, E. Erdem, S. Tari, Disconnected Skeleton: Shape at its Absolute Scale, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, published online: 18 December

2007, doi:10.1109/TPAMI.2007.70842

A. Erdem, E. Erdem, V. Atalay, A. E. Çetin, Vision-based continuous Graffiti-like text entry system, *Optical Engineering*, Vol. 43, Issue 3, pp. 553-558, March 2004

International Conference Publications

A. Erdem, E. Erdem, S. Tari, Articulation Prior in an Axial Representation, *Workshop on the Representation and Use of Prior Knowledge in Computer Vision - in conjuction with ECCV 2006*, Graz, Austria, to appear in Springer LNCS, May 2006

E. Erdem, A. Erdem, S. Tari, Edge Strength Functions as Shape Priors in Image Segmentation, *Proc. of EMMCVPR 2005*, Florida, USA, Springer LNCS, Vol. 3757, pp. 490-502, November 2005

E. Erdem, A. Erdem, V. Atalay, A. E. Çetin, Computer Vision Based Unistroke Keyboard System and Mouse for the Handicapped, *ICME 2003*, Baltimore, USA, Volume 2, pp. 765-768, July 2003

National Conference Publications

E. Erdem, A. Erdem, U. Yılmaz, V. Atalay, Üç Boyutlu Katı Nesnelerin Yansıtma Özelliklerinin Görüntülerden Çıkarılması, *IEEE Sinyal İşleme ve Uygulamaları Kurultayı (SIU 2004)*, Kuşadası, Nisan 2004

A. Erdem, E. Erdem, V. Atalay, A. E. Çetin, Engelliler İçin Bilgisayarlı Görmeye Dayalı Klavye ve Fare Sistemi,*IEEE Sinyal İşleme ve Uygulamaları Kurultayı (SIU 2003)*, İstanbul, Haziran 2003

A. Erdem, E. Erdem, V. Atalay, A. E. Çetin, Kağıt Klavye: Bilgisayarlı Görmeye Dayalı Klavye,*IEEE Sinyal İşleme ve Uygulamaları Kurultayı (SIU 2002)*, Pamukkale, Haziran 2002

A. Erdem, E. Erdem, V. Atalay, A. E. Çetin., Bilgisayarlı Görmeye Dayalı Fare Sistemi,*IEEE Sinyal İşleme ve Uygulamaları Kurultayı (SIU 2002)*, Pamukkale, Haziran 2002

Seminars

Disconnected Skeleton: Shape at its Absolute Scale, *UCLA Vision Lab Group Meetings*, Los Angeles, USA, November 2007

Disconnected Skeleton: Shape at its Absolute Scale, *UCLA Dept. of Mathematics Image Processing Seminar*, Los Angeles, USA, November 2007

International Conference Presentations

S. Tari, C. Aslan, E. Baseski, A. Erdem, Shape Scale: Representing Shapes at Their Absolute Scales, *SIAM Annual Meeting 2006*, Boston, USA, July 2006

M. Iskar, E. Erdem, A. Erdem, S. Akman, A. Dickerman, V. Atalay, R. Cetin-Atalay, CAPRIS: A Database for Cancer Gene Promoter Related Motif Search, *International Symposium on Health Informatics and Bioinformatics*, Antalya, TURKEY, November 2005

E. Erdem, A. Erdem, V. Atalay, Image-based Extraction of Material Reflectance Properties of a 3D Rigid Object, *Eurographics 2003*, Poster Session, Granada, Spain, September 2003

A. Erdem, E. Erdem, V. Atalay, A. E. Çetin, Computer based text entry for Wearable Computing, *6th IEEE International Symposium on Wearable Computers, Demonstrations*, Seattle, USA, October 2002

A. Erdem, E. Erdem, V. Atalay, A. E. Çetin, Computer Vision Based Unistroke Keyboards, *7th International Symposium On Computer and Information Sciences*, Florida, USA, October 2002

A. Erdem, E. Erdem, V. Atalay, A. E. Çetin, Computer Vision Based Mouse, *ICASSP 2002, Student Forum Session*, Florida, USA, May 2002

Technical Reports

A. Erdem, A dynamic procedure for forming shape categories from skeletal shape trees, *METU-CENG-TR-2007-04*, July 2007

Personal Interests

An active member of Serüven, which is a group of people who think, write and discuss about comics and graphic novels. For more information about Serüven, you can visit http://www.seruven.org.