Multi-Object Shape Retrieval Using Curvature Trees

by

Naif Alajlan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2006

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Naif Alajlan

Abstract

With the increasing number of images generated every day, textual annotation of images becomes impractical and inefficient. Thus, content-based image retrieval (CBIR) has received considerable interest in recent years. For comparing images, CBIR uses generic image features which are traditionally either intensity-based (color and texture) or geometry-based (shape and topology); the latter is generally less developed than the former. A common limitation of the existing geometry-based retrieval systems is not considering simultaneously both shape and topology of image objects (or components) which may reveal important properties of the scene being analyzed.

This work presents a geometry-based image retrieval approach for multi-object images. We commence with developing an effective shape matching method for closed boundaries. Then, a structured representation, called curvature tree (CT), is introduced to extend the shape matching approach to handle images containing multiple objects with possible holes. We also propose an algorithm, based on Gestalt principles, to detect and extract high-level boundaries (or envelopes), which may evolve as a result of the spatial arrangement of a group of image objects.

At first, a shape retrieval method using triangle-area representation (TAR) is presented for non-rigid shapes with closed boundaries. The TAR is a 2D matrix that utilizes the areas of the triangles formed by the boundary points to measure the convexity/concavity of each point at different scales (or triangle side lengths). This representation is effective in capturing both local and global characteristics of a shape, invariant to translation, rotation, scaling and shear, and robust against noise and moderate amounts of occlusion. For matching, two algorithms are introduced. The first algorithm matches concavity maxima points extracted from TAR image obtained by thresholding the TAR. In the second matching algorithm, dynamic space warping (DSW) is employed to search efficiently for the optimal (least cost) correspondence between the points of two shapes. Then, a dissimilarity measure is derived based on the optimal correspondence. Experimental results using the MPEG-7 CE-1 database of 1400 shapes show the superiority of our method over

other recent methods in the literature.

Then, a geometry-based image retrieval system is developed for multi-object images. We model both shape and topology of image objects including holes using a structured representation called curvature tree (CT). The hierarchy of the CT reflects the inclusion relationships between the objects and holes. To facilitate shape-based matching, the TAR of each object and hole is stored at the corresponding node in the CT. The similarity between two CTs is measured based on the maximum similarity subtree isomorphism (MSSI) where a one-to-one correspondence is established between the nodes of the two trees. Our matching scheme agrees with many recent findings in psychology about the human perception of multi-object images. Two algorithms are introduced to solve the MSSI problem: an approximate and an exact. The approximate algorithm follows a continuous optimization approach for maximal clique detection of an auxiliary graph derived from the two CTs where the obtained solution most likely corresponds to a MSSI. In the exact algorithm, a recursive procedure searches directly for the MSSI from all possible isomorphisms. The search space is drastically reduced due to the nonnegative property of the employed node similarity measure. Both algorithms have polynomial-time computational complexity and use the DSW as the similarity measure between the attributed nodes. Experiments on a database of 13500 real and synthesized medical images and a database of 1580 logo images have shown the effectiveness of the proposed method.

The purpose of the last part of this thesis is to allow for high-level shape retrieval in multi-object images by detecting and extracting the envelope of high-level object groupings in the image. Motivated by studies in Gestalt theory, a new algorithm for the envelope detection and extraction is proposed that works in two stages. The first stage detects the envelope (if exists) and groups its objects based on their proximity, shape similarity and orientation using hierarchical clustering techniques. In the second stage, each grouping is merged using morphological operations and then further refined using concavity tree reconstruction to eliminate odd concavities in the extracted envelope. Experiment on a set of 110 logo images demonstrates the feasibility of our approach.

Acknowledgments

I would like to thank my supervisors: Prof. Mohamed S. Kamel and Prof. George H. Freeman for their guidance, support and encouragement throughout the course of my studies and preparing this thesis. I also would like to thank Profs. Djemel Ziou, Daniel Stashuk, Otman Basir and Mohamed-Yahia Dabbagh for serving on my examining committee, carefully reading my thesis and providing me with helpful feedback and suggestions.

I would like to thank the Saudi Arabian Cultural Bureau in Canada for financial support and encouragement. In particular many thanks to Prof. Ghazy Al-Makky, Prof. Ahmed Mitwalli and Dr. Yeihya Al Khazraj. All support is gratefully acknowledged and appreciated.

My sincere gratitude goes also to the supporting staff of the Electrical and Computer Engineering Department, everyone was always very patient and helpful. In particular many thanks to Wendy Boles and Fernando Rivero Hernandez. Many thanks also go to the PAMI lab members and colleagues.

Finally, I am forever indebted to my mother and wife for their understanding, endless patience and prayers. I am also grateful to my brother Adel for his encouragement and support.

Dedication

To my mother, my wife and my son.

Contents

List of Tables

List of Figures

List of Algorithms

List of Symbols and Abbreviations

AR	Aspect ratio.
ARG	Attributed Relational Graph.
ARTISAN	Automatic Retrieval of Trademark Images by Shape ANalysis.
BAS	Beam Angle Statistics.
CBIR	Content-Based Image Retrieval.
CED	Curve Edit Distance.
CSS	Curvature Scale Space.
CT	Curvature Tree.
$\mathcal{D}$	Database shape.
δ	Attributes function of CT nodes.
DP	Dynamic Programming.
DSW	Dynamic Space Warping.
DTW	Dynamic Time Warping.
E	Eccentricity.
$\mathcal{E}$	A set of CT edges.
GIS	Geographical Information Systems.
IDSC	Inner-Distance Shape Context.
MAO	Major Axis Orientation.
MCC	Multi-scale Convexity Concavity.
MCS	Maximum Common Subgraph.
MOCH	Multi-Object Convex Hull.
MRI	Magnetic Resonance Imaging.

MSSI	Maximum Similarity Subtree Isomorphism.
MWB	Multi-scale Wavelet-Based.
N	Number of boundary points.
n	Boundary points index.
Ω	Total weight of matching two CTs.
ω	Similarity function between two CT nodes.
ϕ	An isomorphism between the nodes of two CTs.
PR	Precision-Recall.
$\mathcal{Q}$	Query shape.
QBIC	Query By Image Content.
RAG	Region Adjacency Graph.
RSS	Retrieval by Spatial Similarity.
S	Solidity.
SAD	Sum of Absolute Difference.
SC	Shape Complexity.
σ	Standard deviation of a Gaussian kernel.
STAR	System for Trademark Archival and Registration.
TAR	Triangle Area Representation.
TAR-CI	TAR-Concavity Image.
T_s	Number of triangle side lengths.
t_s	Triangle side length or scale level.
$\mathcal{V}$	A set of CT nodes.
VP	Visual Parts.
w	Diagonal width of the DSW search table.
WTAG	Weighted Tree Association Graph.
ZMM	Zernike Moment Magnitude.

Chapter 1

Introduction

Shape is a concept which is widely understood yet difficult to define formally. The human perception of shapes is a high-level concept whereas mathematical definitions tend to describe shape with low-level features. However, for 2-D objects, Marshall [73] tried to define shape as *a function of position and direction of simply connected curves within the two-dimensional field*. Shape is an important visual information that has received much attention from researchers in pattern recognition and computer vision in the past few decades. Most existing techniques for shape analysis and recognition are concerned with single-object shapes, i.e. the silhouette of an object. This thesis investigates different issues related to the matching and retrieval of multi-object shape images.

1.1 Definitions

Before going through the details of our work, we define some terminologies that will be used in the context of this thesis. We commence by defining a shape image, then, we proceed to each of object and hole, single- and multi-object shapes, an envelope, and closed and open boundaries.

Shape image: in the context of our work, a shape image is a binary image that contains one or more connected components of pixels. We assume the image objects are already segmented and their boundaries are well identified; therefore, binary images

are considered.

Objects and holes: an object is a set of connected foreground pixels. Similarly, a hole is a set of connected background pixels which are surrounded by foreground pixels, i.e., a hole is contained in an object. By this definition, each object (hole) may contain one or more holes (objects). Note that any object (hole) can not directly contain another object (hole).

Single- and multi-object shapes: we mean by a single-object shape a binary image that contains only one filled object (with no holes). On the other hand, a multi-object shape is a binary image that contains one object with at least one hole or more than one object with possible holes. Fig. 1.1 shows an illustrative example.

(a) (b)

Figure 1.1: Examples of (a) single-object and (b) multi-object shapes.

Envelopes: an envelope is a high-level virtual boundary, which reflects a meaningful shape as perceived by humans, of a group of objects (or holes) as a result of their spatial arrangement. This concept is discussed in Chapter 5.

Open and closed boundaries: an open boundary has a unique and distinct start and end points when traversed in certain direction, as shown in Fig. 1.2 (a). Conversely, a closed boundary starts and ends at the same point which can be any point on the boundary, as shown in Fig. 1.2 (b). In this thesis, unless mentioned otherwise, a boundary refers to a closed boundary.

(a) (b)

Figure 1.2: Examples of (a) an open and (b) closed boundaries.

1.2 Motivations

Due to the recent developments in digital imaging technologies, an increasing number of images are generated everyday. Millions of images are available via the internet. Therefore, there is a growing interest in finding images in large collections or from remote databases. In order to achieve this task, images have to be represented by specific features. Early attempts tried to use textual annotation of images and then search images using their annotations. Clearly, this method is not practical for large databases. In addition, the textual annotation of image content by itself is a difficult and subjective process. Therefore, searching images using generic features has received considerable attention in recent years. Fig. 1.3 shows categorization of the generic image features as geometry-based (shape and topology) or intensity-based (texture and color). Shape is considered the most promising for the identification of entities in an image. It can be argued that most real subjects are easily identified using only their silhouettes. A user survey in [64] indicated that 71% of the users were interested in retrieval by shape. This thesis focuses on the geometric information, including shape and topology, for content-based image retrieval.

In a typical image, the scene may contain many objects, each of which may include holes and other objects. Human perception of such scenes depends, not only on the shapes of individual objects and holes, but also, on how they are spatially arranged. For instance, consider the binary images in Fig. 1.4. Images (a) and (b) are similar to each other because they have identical external boundaries. At the same time, images (b) and (c) share similarity in their topological structure. Thus, both shape and topology information contribute, although in different amounts, to the overall similarity between two images.

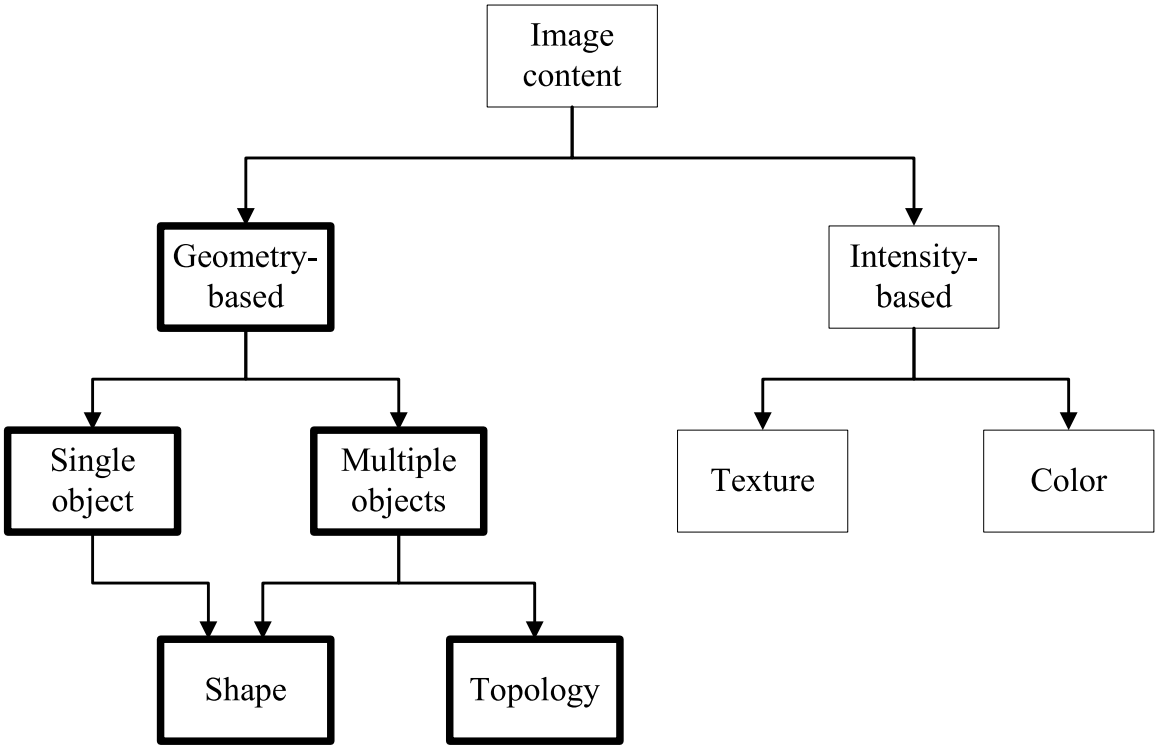


Figure 1.3: Categorization of the features for content-based image retrieval.

(a) (b) (c)

Figure 1.4: An illustration of the shape similarity concept. See text for details.

(a) (b) (c)

Figure 1.5: Illustration of the role of the spatial configuration in shape similarity. See text for details.

Fig. 1.5 shows another example of the role of the spatial arrangement on how a group of objects is perceived as a whole. Image (a) shows a trademark image which contains three triangles. However, when these triangles are arranged in a different way, as shown in image (b), the new image looks similar to image (c). Note that images (a) and (c) are not similar. We conclude that different arrangements of the same objects can lead to completely different perceived shapes.

The main motivation of this research is to establish a successful strategy for measuring the similarity between two multi-object shape images. We found support for our observations in the literature of psychology. Biederman [21] concluded that different arrangements of the same objects of one subject can easily lead to a different subject. He also argued that certain relations between objects, such as the relative size and the spatial configuration, determine the identity of a shape. Another notion by Lowe [71] states that the similarity between two groups of objects does not equal the sum of the similarities between individual objects. In conclusion, the proposed approach for measuring the similarity between two multi-object shape images (presented in Chapter 4) considers both shape and topology at once, agrees with recent studies in Psychology about the human similarity comparison process and works in polynomial-time complexity. To our best knowledge, no such technique exists in the literature so far.

1.3 Research Objectives

The main objective of this research is to develop, implement and evaluate a prototype system for multi-object shape matching and retrieval. We model both shape and topology of image objects including holes using a structured representation called curvature tree (CT). The hierarchy of the CT reflects the inclusion relationships between the objects and holes. To facilitate shape-based matching, triangle-area representation (TAR) of each object and hole is stored at the corresponding node in the CT. The similarity between two CTs is measured based on the maximum similarity subtree isomorphism (MSSI), where a one-to-one correspondence is established between the nodes of the two trees. An effective shape matching algorithm, called dynamic space warping (DSW), is developed to measure the similarity between the attributed nodes. Our matching scheme agrees with many recent findings in psychology about the human perception of multi-object images. To allow high-level shape retrieval in multi-object images, an algorithm for detecting and extracting the envelope of high-level object groupings in the image is proposed; motivated by studies in Gestalt theory [120].

Fig. 1.6 illustrates the different modules of our proposed system. The user presents a query image and the system retrieves images from the database ranked according to their similarity with the query. At first, a preprocessing stage removes noise and labels objects and holes of the image. For gray-scale images, different types of noise removal methods exist [9, 7]. Since only binary images are considered in this research, a binary noise removal method is adopted. Then, the CT representation of the image objects is constructed to encode both shape and topology. As in most retrieval systems, the computation of the CT for the database images is performed off-line whereas the query CT is computed at the time of retrieval. For measuring the similarity, attributed tree matching module matches the query with the database images, ranks them according to their similarity with the query and provides the user with a number of top ranked images according to the user preference. A shape matching algorithm is used during the matching process to measure the shape similarity between the tree nodes.

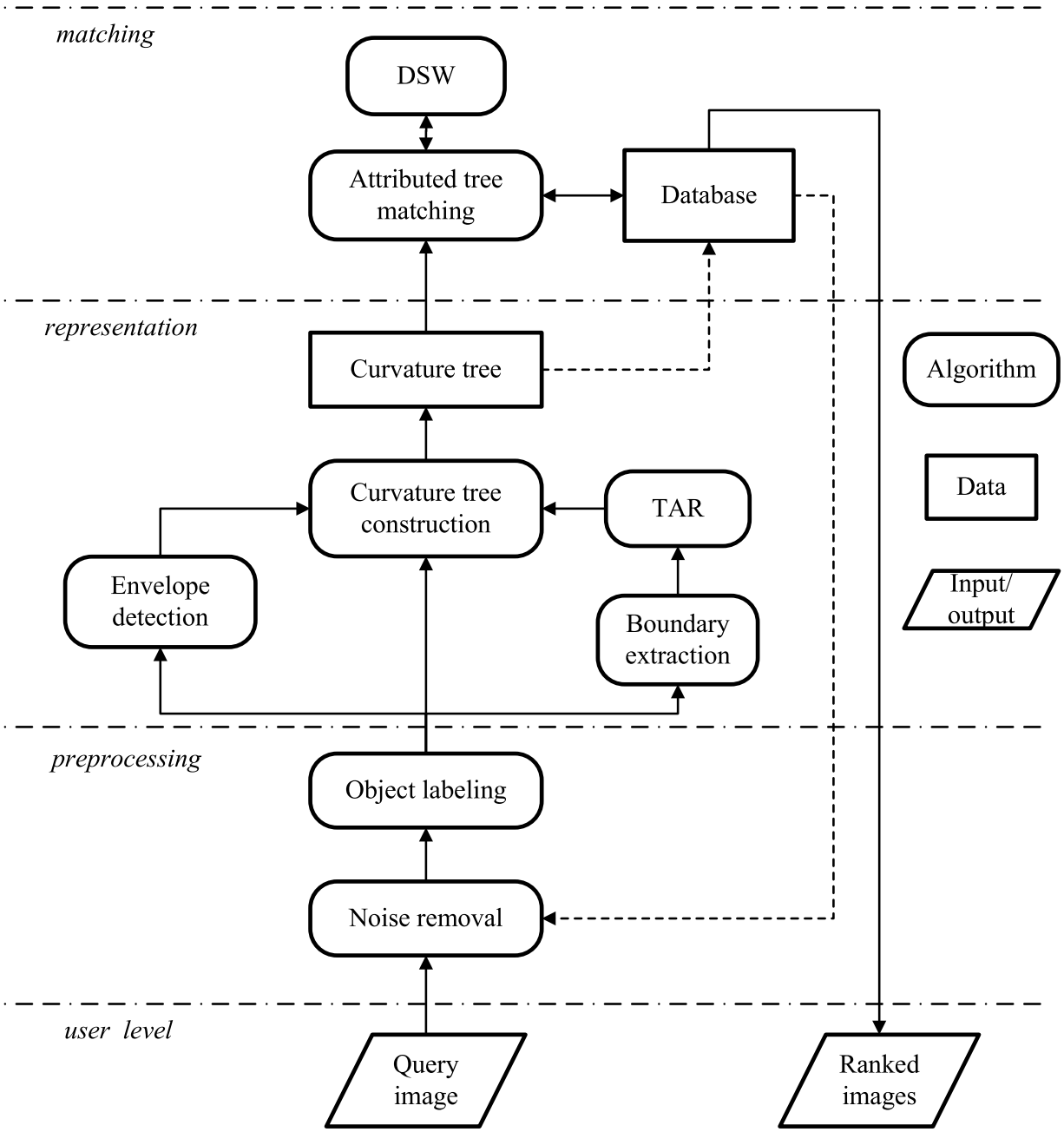


Figure 1.6: Block diagram of the proposed multi-object shape retrieval system.

1.4 Thesis Outline

The rest of this thesis is organized as follows.

Chapter 2 presents background information and review of the related work in the literature. The background includes the main methodologies for representing and describing shapes, distances for measuring the similarity between images and the performance evaluation of retrieval systems.

Chapter 3 is concerned with single-object shape representation and matching. In this chapter, two shape descriptors are derived from the triangle-area representation of the external boundary of an object. Then, two matching algorithms are developed based on each description.

Chapter 4 introduces the curvature tree (CT) as a tool for representing multi-object shapes and explains the CT properties. Two matching algorithms, an approximate and an exact, are proposed to measure the similarity between two CTs. The proposed system is employed in two application domains; namely, medical image retrieval and logo retrieval.

Chapter 5 proposes an algorithm for detecting and extracting high-level envelopes of object groupings in multi-object images based on Gestalt principles.

Chapter 6 summarizes the main contributions of this thesis and suggests areas for future work.

Chapter 2

Background and Review

This chapter presents a background about the shape representation and matching methodologies and the criteria for evaluating the retrieval performance of shape retrieval systems. The last section presents a detailed review of the shape analysis methods in the literature. The reader can refer to standard books in the literature for more information about digital image processing [47] and pattern recognition [108].

2.1 Shape Representation and Description

Shape representation means obtaining a set of features characterizing the shape in such a way that it becomes possible to reconstruct the shape from such features. Shape description is the extraction of shape features in order to quantify important properties of the shape; however, the extracted features are not necessary sufficient to reconstruct the shape.

A wide range of shape representation and description methods have been proposed and many of them have been implemented into commercial systems [70]. The effectiveness of a shape retrieval system depends mainly on the type of the shape descriptor used and the efficiency of the shape matching function. There are some criteria that should be taken into consideration when developing a shape descriptor, which include [92]:

Uniqueness : a shape needs to be represented uniquely; otherwise, the retrieved shapes may not be similar to the query shape.

Invariance : it is highly desirable for a representation to be invariant to geometric transformations such as the affine transformations which include translation, scale, rotation and skewness.

Robustness : to be suitable for practical applications, moderate amounts of noise and deformations should have no (or at least little) impact on the representation.

Scalability : the discrimination ability of a representation should be independent of, or at least slightly affected by, the number of shapes it describes.

Efficiency : a representation must be computationally efficient in order to be applicable in real-time.

Compactness : a representation must be compact for storage purposes.

There are many classifications of the shape representation techniques [31, 137] depending on the nature of their inputs (contour-based vs region-based) or the nature of their outputs (global or numeric vs structural or non-numeric). Global techniques usually compute a numerical feature vector for a shape. Then, the matching is conducted using a simple metric distance such as the Euclidean distance or the city block distance. On the other hand, structural techniques divide shapes into segments called primitives. They differ in the selection and organization of these primitives. The similarity matching in this case is usually performed using string or graph matching. Fig. 2.1 shows the hierarchy of these classifications. The remainder of this section discusses samples from the existing shape representations according to the classification in Fig. 2.1. It should be mentioned that there are techniques based on combination of global and structural representations; for example, a structural representation decomposes a shape into primitives and a global descriptor is extracted for each primitive [6].

2.1.1 Global contour-based techniques

Circularity

Circularity reflects the compactness of the contour of an object. It is defined as:

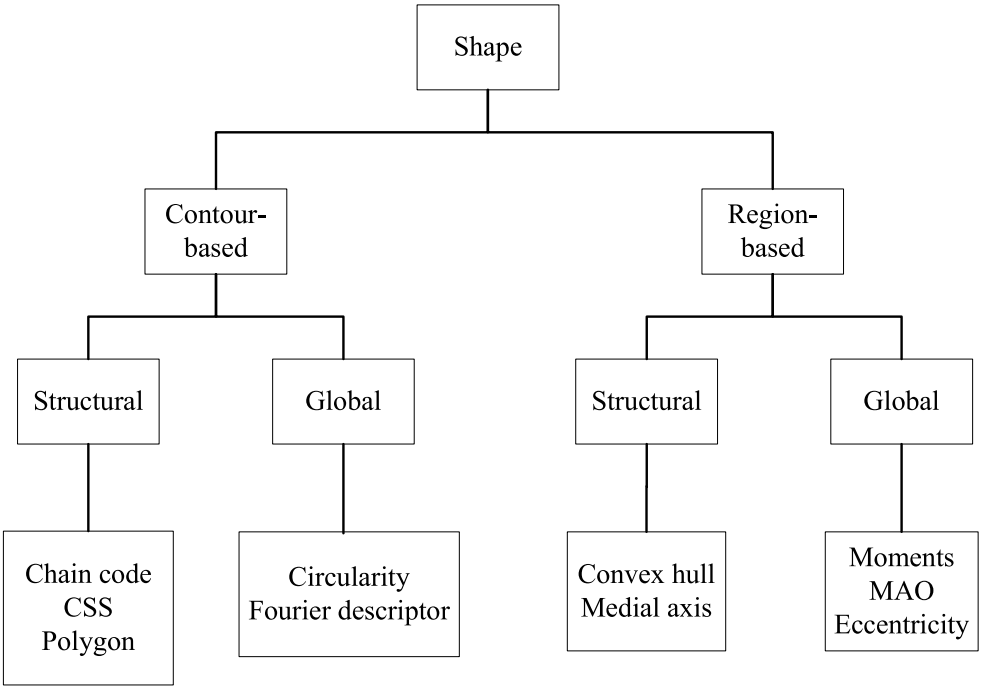


Figure 2.1: Classification of shape representation techniques.

$$Circularity = \frac{Perimeter^2}{Area} \tag{2.1}$$

Circularity has large values for elongated objects and it is roughly correlated with the complexity of the contour. It has a minimum value of 4π for a circle. It is also invariant to scaling, translation and rotation.

Fourier descriptors

Fourier descriptors are the complex coefficients of the Fourier series expansion of waveforms. Given a shape in 2D space, the horizontal $x(k)$ and the vertical $y(k)$ coordinates of the contour points can be expressed as [136]:

$$p(k) = x(k) + jy(k) \tag{2.2}$$

where $k = 0, 1, ..., N-1$ and N is the number of the contour points. The discrete Fourier transform (DFT) of $p(k)$ is:

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} p(k) \exp\left[-j2\pi uk/N\right] \tag{2.3}$$

for $u = 0, 1, ..., N-1$. The complex coefficients $a(u)$ are called the Fourier descriptors of the contour. In practice, only the first few descriptors are enough to characterize the contour. This representation has the advantage of reducing the 2D problem to two 1D problems.

2.1.2 Structural contour-based techniques

Chain codes

Chain codes were introduced by Freeman [44]. In this approach, an arbitrary curve is represented by a sequence of vectors of unit length and a set of possible directions. From a selected starting point, a chain code can be generated by using 4-directional or 8-directional chain code, as seen in Fig. 2.2. N-directional chain code is also possible. Fig. 2.3 shows an example of representing a shape using 4-directional chain code. This method suffers from being sensitive to noise and to the starting point.

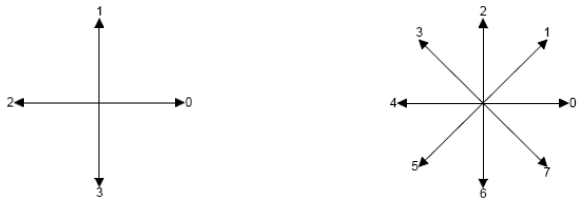


Figure 2.2: Chain codes numbering schemes: 4-directional (a) and 8-directional (b).

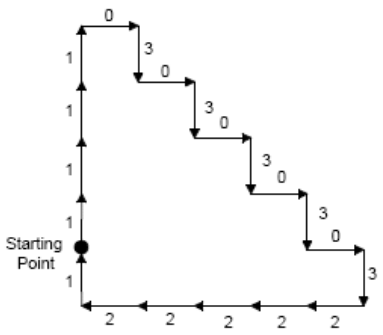
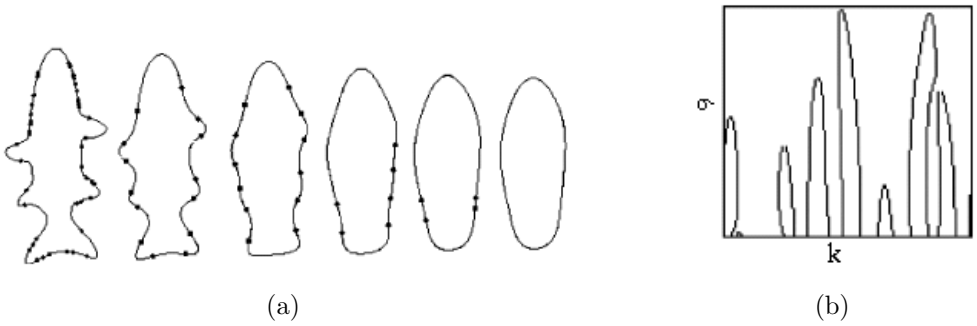


Figure 2.3: 4-directional chain code of a shape.

Curvature scale space

The curvature scale space (CSS) representation of a contour is obtained by tracking the positions of inflection points in that contour as it is gradually smoothed by low-pass Gaussian filters of variable widths (σ) [79]. As σ increases, the contour becomes smoother and less significant inflection points are eliminated, as shown in Fig. 2.4(a). The result of plotting the inflection points versus σ is known as CSS image as shown in Fig. 2.4(b). This approach is of particular interest since it shares some similarities with part of our work.



(a) (b)

Figure 2.4: Contour smoothing (a) and the resulted CSS image (b).

An efficient computation of the curvature scale space representation using B-spline wavelets was proposed by Wang et al. [132, 131]. Their method provides an alternative to the classical Gaussian-based scale space representation while being much more efficient and relying on the well-established wavelet theory.

Polygon decomposition

Here, the shape is broken down into line segments using polygon approximation [48]. Then, the polygon vertices are used as primitives. The feature vector for each primitive is a 4-element string of the internal angle, distance from the next vertex and its x and y coordinates. For efficiency, only a fixed number of sharpest vertices are considered. While this method had worked for artificial drawings, it is impractical for natural objects.

2.1.3 Global region-based techniques

Geometric moment invariants

Hu published the first significant paper on the use of moment invariants for 2D pattern recognition applications [51]. Given a binary image $f(x, y)$, moments of order $(p + q)$ are defined as:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \tag{2.4}$$

The central moments of $f(x, y)$ are defined as:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{2.5}$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$. Then, the normalized central moments are computed:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\lambda}}, \text{ s.t. } \lambda = 1 + \frac{p+q}{2} \text{ and } (p+q) \geq 2 \tag{2.6}$$

The second and third order moments, which are invariant to translation, scale and rotation, were used by Hu to derive his well-known seven moment invariants:

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right) + \\
&\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right) \\
\phi_6 &= (\eta_{20} - \eta_{02})\left((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right) + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})\left((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right) - \\
&\quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})\left(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right)
\end{aligned} \tag{2.7}$$

Major Axis Orientation

The major axis of an object is the straight line segment joining the two points farthest from each other. The major axis orientation is defined as the angle between the x-axis and the axis around which the object can be rotated with minimum inertia [47]. This feature is particularly important to find out how a group of objects are aligned. Using (2.5), this angle is given by:

$$\theta = \frac{1}{2}\tan^{-1}\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \tag{2.8}$$

Eccentricity

The minor axis of an object is perpendicular to the major axis and of length equals the width of the minimum-area rectangle that contains the object. The eccentricity is the ratio of the major axis to the minor axis. Its normalized value ranges from zero (circle) to one (line) and it is useful to eliminate highly dissimilar shapes from further matching. Using (2.5), the normalized eccentricity is given by:

$$Eccentricity = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{(\mu_{20} + \mu_{02})^2} \tag{2.9}$$

Solidity

The solidity is defined as the ratio of the shape's area to the area of its convex hull. The solidity measures the deviation of a shape from being totally convex.

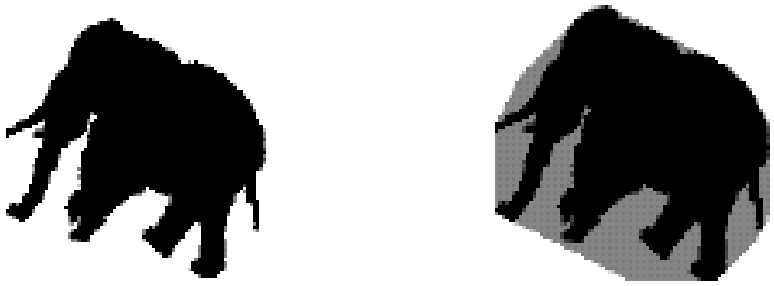
Aspect ratio

The aspect ratio is the ratio of the width to the height of the shape. It can be computed from the minimum bounding rectangle of the shape.

2.1.4 Structural region-based techniques

Convex hull and concavity trees

A region R is convex if and only if for any two points $P_1, P_2 \in R$, the whole line segment whose end points are P_1 and P_2 is also inside R. The convex hull of a region is the smallest convex region H which satisfies the condition $R \subseteq H$ [118]. This definition of the convex hull is the most general one and includes the multi-object convex hull (MOCH). Fig. 2.5 shows the convex hull of an elephant's shape.



(a) (b)

Figure 2.5: An object (a) and its convex hull (b).

The convex hull can be used to build a tree structure of region concavity called *concavity tree* [115]. A concavity tree is constructed recursively during the generation of the convex hull. This concept is illustrated in Fig. 2.6, where the convex hull of the whole region is constructed first, and convex hulls of concave residua are found next. The resulting convex hulls of concave residua of the regions from previous steps are searched until no concave residuum exists. Thus, the nodes in a concavity tree represent convex hulls and the root represents the convex hull of all objects in the image. Nodes in the first level represent the convex hulls of the concavities (or holes) and nodes in the second level represent convex hulls of meta-concavities, and so on. The resulting tree is a shape representation of the region.

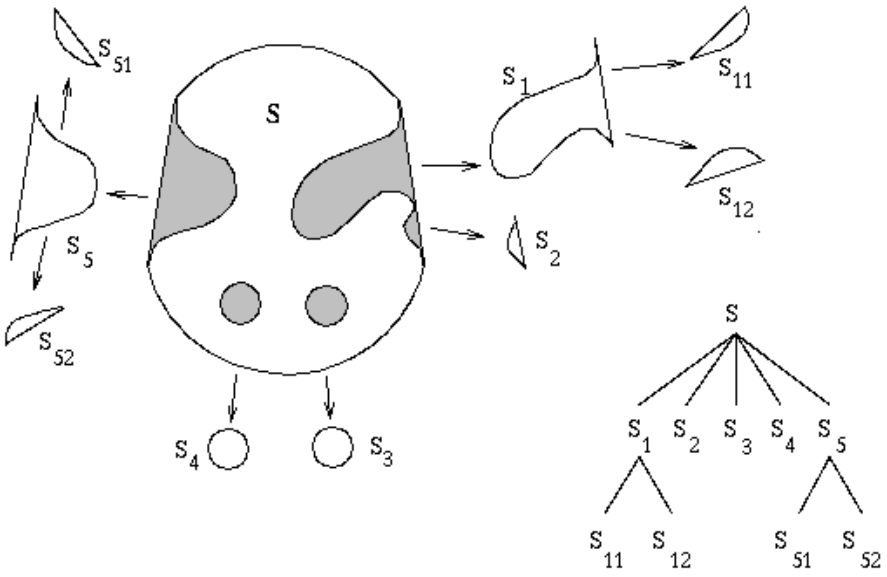


Figure 2.6: Concavity tree construction.

Medial axis transform

Like the convex hull, the skeleton of an object can be employed for shape description. The basic idea is to eliminate redundant information while retaining only the topological structure of the object. The medial axis is the locus of the centers of maximal circles that fit within the shape [70], as illustrated in Fig. 2.7. The skeleton is then segmented and represented as a graph according to certain criteria. The matching between shapes becomes graph matching problem. This method is sensitive to noise and requires high computations.

2.2 Distance Measures

A distance measure, sometimes referred to as dissimilarity measure, is a function that quantifies the dissimilarity between two patterns based on their descriptors. For image retrieval, measuring the dissimilarity between a query image and a database image is

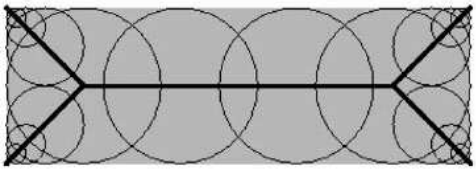


Figure 2.7: Medial axis of a rectangle.

required to rank the database images according to their similarity with the query. It is usually desirable for a distance to be a metric. Given a set of patterns $S = \{x, y, z\}$ and a distance d, such that $d : S \times S \rightarrow \mathcal{R}$, d is a metric if it satisfies the following conditions:

1. $d(x, y) = 0 \Leftrightarrow x = y$ (self identity).
2. $d(x, y) > 0$ if $x \neq y$ (positivity).
3. $d(x, y) = d(y, x)$ (symmetry).
4. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality).

A distance that satisfies the first three conditions and violates the triangle inequality is called semi-metric. Although the metric property is useful in many applications such as image database indexing, many studies in psychology concluded that human similarity judgements do not obey the metric properties. In particular, the human judgement violates the triangle inequality and is not symmetric [124, 11]. This fact is illustrated in Fig. 2.8, where shapes a and b are similar, i.e., $d(a, b)$ is small. Similarly, $d(b, c)$ is small. Whereas shapes a and c are very different, i.e., $d(a, c)$ is large. So, $d(a, b) + d(b, c) < d(a, c)$, which violates the triangle inequality.

Many researchers in the shape analysis community have reported that a metric distance is not suitable for meaningful shape retrieval [18, 127]. In order to make use of the metric advantages, in [19, 6] a shape is decomposed into primitives and a metric distance is used to retrieve the primitives which are used in a final (nonmetric) similarity function. In the following, samples of the distance measures used in the shape analysis literature are reviewed. It should be emphasized that the choice of an appropriate distance measure depends mainly on the employed shape representation and description method.

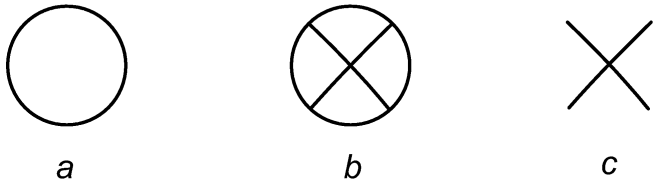


Figure 2.8: An example shows the triangle inequality failure in the human similarity judgement. See text for explanation.

2.2.1 Minkowski distance (L_p)

A large number of statistical shape analysis methods use the L_p distance as a similarity measure. It works for vectors of a fixed size. Let $x, y \in \mathcal{R}^n$, L_p is defined as:

$$L_p(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{2.10}$$

where $p > 0$. For $p = 1$, L_p becomes the Manhattan or sum of absolute difference (SAD) distance. When $p = 2$, L_p becomes the well-known Euclidean distance. L_p is a metric.

2.2.2 Earth mover's distance

The earth mover's distance (EMD) is suitable for matching incomplete or partially-occluded patterns. The EMD measures the minimum amount of work needed to transform one pattern (the supplier) to the other (the receiver) by moving weights under certain constraints. Each point in the supplier and the receiver patterns can be viewed as a mass and hole, respectively, at the location of that point. Then, the EMD measures the minimum amount of work to fill the holes with the mass. Formally, given two weighted point patterns $A = \{A_i, W(A_i)\}$ and $B = \{B_i, W(B_i)\}$, where $i \in \langle 1, m \rangle$, $j \in \langle 1, n \rangle$ and A_i and B_j are points with associated wights $W(A_i)$ and $W(B_j)$. Let f_{ij} be the flow from location A_i to B_j, F is the matrix of elements f_{ij} and d_{ij} is a distance between A_i and B_j (for instance, the Euclidean distance). Then, the EMD is defined as [28]:

$$EMD(A,B) = \frac{\min \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}} \tag{2.11}$$

subject to $f_{ij} \geq 0$, $\sum_{i=1}^{m} f_{ij} \leq W(B_j)$, $\sum_{j=1}^{n} f_{ij} \leq W(A_i)$, and $\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min(\sum_{i=1}^{m} W(A_i), \sum_{j=1}^{n} W(B_j))$. These constraints prevents a supplier point from giving more weight than it has and a receiver point from getting more weight than it needs. The EMD has been applied for image retrieval based on color [105] and shape [45, 46].

2.2.3 Graph matching

In structural image representation, graphs are usually employed where nodes represent primitives and edges describe relationships between these primitives. Therefore, graph theory has attracted many researchers in pattern recognition and computer vision. Graph theory is a huge area and many different kinds of graphs exist in the literature with enormous number of applications in various areas such as chemistry, geography and engineering. More details about the classes of graphs can be found in standard textbooks of graph theory, such as [15].

The comparison between two graphs is performed using some form of graph matching. Graph matching is the process of finding a correspondence between the nodes and the edges of two graphs that satisfies some constraints ensuring that substructures in one graph are mapped to similar substructures in the other. A good recent survey paper about the graph matching in pattern recognition can be found in [30].

Graph matching methods can be classified into two broad categories. The first contains exact graph matching methods that require strict correspondence among graphs. The second category defines inexact matching, which can handle minor differences between graphs. The last category is of great interest to pattern recognition applications where patterns are subjected to noise and deformations. In both categories, the matching can be performed in different ways, depending on the application. A correspondence between

two graphs that preserves the adjacency between nodes is called *subgraph isomorphism*. If this correspondence is one-to-one, then it is called *graph isomorphism*. Another interesting type of matching maps a subgraph of the first graph to an isomorphic subgraph of the second one. Since such a mapping is not uniquely defined, usually the goal is to find the largest subgraph for which such mapping exists. This problem is known in the literature as the *maximum common subgraph* (MCS) of the two graphs.

Another way of graph matching is to introduce a set of graph edit operations including node insertion, node deletion and node substitution. Once each operation is assigned a cost, the lowest cost sequence of operations needed to transform one graph into the other is computed. This cost is called the *graph edit cost*, which can be considered as the cost of the matching.

2.3 Performance evaluation

Evaluation of retrieval performance is a crucial problem in content-based image retrieval, mainly due to the subjectivity of the human similarity judgement. The evaluation of a shape retrieval system depends on the application domain [25]. However, many different methods for measuring the performance of a system have been created and used by researchers. Perhaps the most widely used measure, for retrieval effectiveness, in the literature is the precision and recall (PR) graph [82]. Precision measures the accuracy of retrieval and recall measures the ability of retrieving relevant images from the database. Higher precision and recall values mean better performance. PR graphs offer a convenient way to assess and compare the performances of different systems in one graph. Precision and recall are given by:

$$Precision = \frac{number\,of\,relevant\,retrieved\,images}{number\,of\,retrieved\,images} \tag{2.12}$$

$$Recall = \frac{number\,of\,relevant\,retrieved\,images}{number\,of\,relevant\,images} \tag{2.13}$$

Computing the recall requires the ground truth to be provided for all database images. When the ground truth is unknown, some researchers plot the precision as a function of

the number of retrieved images. Another effectiveness measure is the F-measure, which is usually used by information retrieval community [97]. The F-measure provides a single number between zero (worst) to one (best), rather than a graph, and is given by:

$$\mathrm{F} = \frac{2}{\mathrm{n}} \sum_{i=1}^{n} \frac{Precision(i) \times Recall(i)}{Precision(i) + Recall(i)} \tag{2.14}$$

where n is the number of precision-recall pairs.

It is also important to evaluate the system robustness against conditions that images may undergo in practice such as discretization noise, affine transformations (translation, scale, rotation and skewness) and occlusion. To perform such evaluation, the retrieval accuracy, at a particular recall, is monitored as the influence of each condition increases.

To measure the retrieval efficiency, the time complexity measure is usually used which reflects the running time of an algorithm on various machines [24]. In order to make the complexity measure sufficiently universal, it is usually assumed that computations are performed as elementary operations such as arithmetic and comparison operations. Since we are interested only in the asymptotic of the execution time, the number of elementary operations will be considered as its time complexity. In our case, the time complexity is a function of the input.

2.4 Related work

In this section, we present a literature review of some of the existing shape matching techniques that can, either partially or completely, handle multi-object shapes. There are many ways to categorize shape representation techniques; for instance, region-based versus boundary-based and global versus structural. From object multiplicity point of view, we categorize the existing techniques as single-object versus multi-object. The latter is more generic and it includes our proposed method. The reader can refer to this section for more details about methods mentioned during the performance comparisons in the following chapters of this thesis.

2.4.1 Single-object shape matching methods

They are also called silhouette-based. The majority of the existing techniques for shape representation and matching fall into this category. An obvious limitation of all these techniques is either they can not describe a multi-object shape or their extension to the multi-object case is nontrivial. However, there are two main motivations to study these methods. Firstly, a single-object shape is a particular case of a multi-object shape. Secondly, many issues and problems related to both categories are by far better studied and researched in the case of single-object shapes. These issues include descriptor compactness, robustness to noise, scalability, invariance to affine transformations, and efficiency and effectiveness of the matching.

The multi-scale approach for shape representation and matching is considered the most promising. It can be argued that human perception of shapes is a multi-scale by nature. In addition, many interesting shape properties are revealed at different scale levels. Another advantage includes its invariance to moderate amounts of deformations and noise. In the following, recent shape matching methods are reviewed.

Curvature Scale Space

One of the most well-researched single-object shape representations is the curvature scale space (CSS) method proposed by Mokhtarian and Mackworth [78, 79], which has been selected for MPEG-7 standard [84]. It works on the external boundary of an object. At first, a parameterized representation of the contour points, in terms of the x and y coordinates, is extracted which results in two 1-D curves. Then, each 1-D curve is repeatedly convolved with a Gaussian kernel with increasing standard deviation σ (or scale level) to gradually smooth the contour at different scale levels. At each scale, the curvature of each contour point is measured by:

$$c(u,\sigma) = \frac{\dot{x}(u,\sigma)\ddot{y}(u,\sigma) - \ddot{x}(u,\sigma)\dot{y}(u,\sigma)}{\left(\dot{x}(u,\sigma)^2 + \dot{y}(u,\sigma)^2\right)^{3/2}} \tag{2.15}$$

where c is the curvature at location u and scale σ, $\dot{x}$ and $\ddot{x}$ are the first and second derivatives of x, respectively. By setting (2.15) to zero, the inflection points (curvature zero crossings) are located at each scale. This results in a binary image, called CSS image, where

the foreground and background pixels represent concave and convex segments, respectively, as shown in Fig. 2.9. More specifically, it shows the end-points of the concave segments along the contour (the horizontal axis) at each scale level (the vertical axis). As the scale level increases, the smoothing effect increases and the number of inflection points decreases until the contour becomes totally convex. Only the maxima of the CSS images' contours are used for matching two shapes [2]. In order to achieve rotation or starting point invariance, correspondence between two sets of maxima points is established by finding the optimum horizontal shift of one CSS image with respect to the other that results in the least cost, which represents the dissimilarity function. The cost is defined in terms of the differences of the scale levels of the corresponding maxima. Many heuristics are employed to find the best correspondence efficiently. This method exhibits certain degree of robustness to affine transformation [1], where in another implementation explicit estimation of the affine parameters using the least squares approach was employed to verify the matching results [76]. The main limitations of the CSS method include its limited representation to only the concave segments and its failure to discriminate a shallow concavity from a deep one. For instance, totally convex shapes like squares and triangles have the same CSS image, which is only the background. Another disadvantage is the requirement for a large number of scales to obtain the CSS image (may exceed 400).

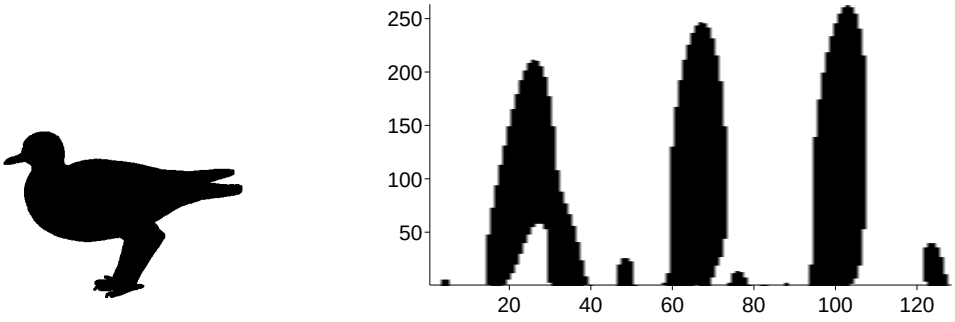


Figure 2.9: The CSS image of a bird shape.

Multi-scale Wavelet-Based

Elrube et al. [40, 39] proposed a multi-scale wavelet-based (MWB) representation for single-object shapes. In their method, a closed-contour is represented by its x and y coordinates and 1-D dyadic wavelet transform is applied separately to each coordinate sequence in order to decompose the contour into multi-scale levels. Then, invariant moments are computed for the approximation and detail coefficients at each scale level. This scheme provides coarse-to-fine matching in order to eliminate dissimilar shapes. It was demonstrated to perform well under affine transformations and small boundary deformations.

Multi-scale Convexity Concavity

In a recent work, Adamek and O'Connor proposed another multi-scale representation for closed contours that makes use of both concavities and convexities of all contour points [3]. It is called multi-scale convexity concavity (MCC) representation, which is a 2-D matrix where each column describes a contour point at different scales (rows). The contour smoothing is done in the same way as the CSS method [79]. However, a new measure for the curvature is used based on the relative displacement of a contour point with respect to its position in the preceding scale level. This idea is motivated by the observation when smoothing a closed contour, convex and concave points are moved inside and outside the contour, respectively. The amount of displacement reflects the curvature degree. Afterwards, the matching is done using a dynamic programming (DP) approach where a global optimal match is searched between two MCC representations using a DP table. In the DP table, each entry denotes the distance between two points (one from each contour). This distance is chosen to be the sum of absolute differences between the columns of each point in the MCC matrices. The search for the optimal match is restricted with many heuristics; for instance, no more than two points on one contour are matched to a single point on the other. The rotation invariance is achieved in similar manner as the CSS method, i.e., by repeating the algorithm for each contour point as the starting point. The MCC was able to achieve better average accuracy (84.9%) than the CSS method (80.5%) when they were tested on the MPEG-7 dataset. However, the MCC is computationally expensive, $O(N^3)$ where N is the number of contour points. The robustness of the MCC method to affine transformation was not demonstrated.

Dynamic Programming

Milios et al. proposed one of the earliest approaches for matching shapes using dynamic programming [75, 92]. In their approach, implicit multi-scale matching takes place through matching merged contour segments in order to avoid the cost of computing the scale space explicitly. At first, the contour is approximated into polygons using cubic B-splines and inflection points are located, which segments the contour into convex and concave segments. Then, a set of geometrical features are computed for each segment to guide the merging process. The DP algorithm examines all possible merges of small segments of one shape to match with larger segments of the other and selects the best merge, i.e., that results in the minimum cost. An upper limit K is imposed on the maximum number of merged segments to compromise speed with accuracy. The rotation invariance is achieved in similar way to the MCC method. In the case both shapes are closed, the complexity of the algorithm is $O(M^3N^2)$, where M and N are the number of inflection points of the two contours, and reduces to $O(K^2M^2N)$ when K is considered. The authors did not report superior performance over other existing methods. The main limitations of this method include the lack of robustness to the general affine transformations and the high computational complexity of the matching process.

WARP

Bartolini et al. proposed a method for shape matching and retrieval based on the Fourier descriptors that is called *WARP* [16]. They chose to use the phase of the Fourier descriptors and claimed to outperform the state-of-the-art Fourier-based methods. At first, the low-frequency coefficients are normalized in terms of translation, scale and rotation. For matching, the inverse discrete Fourier transform is used to obtain normalized versions of the original contours in the spatial domain. Then, a DP method is employed to find the similarity between the two transformed contours. Although this technique outperformed other Fourier-based methods, the authors reported less retrieval accuracy than the CSS method [2]. Besides, WARP is not invariant to the general affine transformation.

Inner Distance Shape Context

Ling and Jacobs were mainly motivated by computing a distance for articulated shapes [69]. The inner-distance, which is the length of the shortest path between two boundary points within the shape boundary, was derived to be invariant to shape articulation. The authors claim that the inner-distance is the natural replacement to the well-known Euclidean distance. In order to apply the inner-distance for shape matching and retrieval, the authors extended the shape context method [17] using this distance and called it inner-distance shape context (IDSC). Then, DP is used for matching shapes after calculating the IDSC distances. The retrieval performance on the MPEG-7 dataset is 85.4% which is the highest published performance so far. The main disadvantage of this method is the use of different parameter settings for different databases. This method has not been tested on geometrically transformed shapes.

Curve Edit Distance

Sebastian et al. proposed a curve alignment approach, which is called curve edit distance (CED), for matching open and closed curves [109]. In their method, the correspondence between the points of the two curves is controlled by the relative difference in their spatial location and their curvature. Then, a matching function is defined as the minimum cost of such correspondence. The search for the optimal correspondence is made efficient by decomposing each curve into segments, then ideally solved using DP. Moreover, merging, deletion and addition of curve segments are allowed in order to account for shape deformations. The complexity of this method is $O(N^2 \log N)$ for closed curves and the authors reported 78.2% accuracy for MPEG-7 retrieval test.

Beam Angle Statistics

Arica and Vural proposed another descriptor for closed contours based on the curvature information of all boundary points, which is called Beam Angle Statistics (BAS) [10]. In BAS, the curvature at each boundary point is viewed as a random variable that draws its values from the angles between each equally-distant neighboring points at that point. Then, few order moments are computed for the random variable at each point. For measuring

the similarity, DP is used to find the best correspondence that minimizes the Euclidean distance between the signatures of two shapes. The authors reported 82.4% accuracy using the MPEG-7 retrieval test.

Visual Parts

In another recent work, Latecki et al. presented a shape matching approach that works directly on the the closed boundaries [65, 67, 68]. It is based on visual parts (VP), where (part of) a database shape is simplified in the context of the query shape prior to their matching. The simplification process includes the elimination of particular points from the database shape such that the similarity to the query shape is maximized. The main disadvantage of this method is the high computational complexity of the matching algorithm, which is $O(N^3 \log N)$ where N is the number of the boundary points.

2.4.2 Multi-object shape matching methods

Two main approaches exist in the literature that deal with the matching of multi-object images. The first approach is based on features extracted from the whole image, i.e., global features, such as invariant moments [51], Zernike moment magnitudes (ZMMs) [59, 62], geometric features [41], and combination of invariant moments and edge directions histograms [55]. While the global features have the advantages of being compact and efficient for matching, local information and spatial configuration of image objects are lost.

In the second approach, features for each object are extracted and then combined into a single distance measure for matching. While these methods overcome some of the limitations of the global methods by including the contribution of each object in the final similarity function, they still ignore the topological structure of the image and do not account for the unmatched objects in the similarity measure. Another limitation of weighted-sum distances is that the number of parameters to be tuned is proportional to the number of objects in both images.

Zernike Moments Magnitudes

The Zernike moment magnitudes (ZMMs) is one of the most effective and widely used global shape descriptors. ZMMS has been selected as the MPEG-7 CE-2 region-based shape descriptor [56]. The Zernike moment of order n and repetition m for a digital image $f(x, y)$ is defined as [59]:

$$Z_{nm} = \frac{n+1}{\pi} \sum_{x} \sum_{y} f(x, y) V^*_{nm}(\rho, \theta), \quad x^2 + y^2 \leq 1 \tag{2.16}$$

where $n \geq 0$, $|m| \leq n$, $n - |m|$ is even, ρ and θ are the length and the angle of vector from origin to (x, y) pixel, and $V_{nm}(\rho, \theta)$ is the set of Zernike polynomials defined as:

$$V_{nm}(\rho, \theta) = R_{nm}(\rho) \exp(jm\theta) \tag{2.17}$$

The radial component $R_{nm}(\rho)$ is the defined as:

$$R_{nm}(\rho) = \sum_{k=0}^{\frac{n-|m|}{2}} (-1)^k \frac{(n-k)!}{k! \left(\frac{n+|m|}{2} - k\right)! \left(\frac{n-|m|}{2} - k\right)!} \rho^{n-2k} \tag{2.18}$$

To compute the ZMMs, the origin of the image is assumed to be at its centroid to achieve translation invariance. For scale invariance, the area of the image is normalized to a constant area. The Zernike polynomials are orthogonal; therefore, ZMMs are non-redundant. In [61], 36 ZMMs of order and repetition from 0 to 10 are used as feature vector and the SAD is used as a dissimilarity function. Instead of treating the ZMMs equally, Kim and Kim proposed the use of selected ZMMs in probabilistic framework for trademark retrieval [62]. Regardless of how many objects in the image, 90 ZMMs are computed for each image in the database, which are used to estimate the distribution model for each feature. When a query image is preprocessed, the distribution models help in selecting the most dominant features which are then matched using the Euclidean distance.

Two stage hierarchy

Jain and Vailaya proposed another global approach for trademark retrieval [126, 55]. The edge direction histogram and invariant moments of the image are used to prune the

database and select candidate images when a query image is presented. Then, these candidates are screened using a deformable template matching process that works on the edge map of the filled images.

Query By Image Content

The QBIC (Query By Image Content) system from IBM uses the generic image features, which include color, texture and shape to retrieve images in a database similar to a query image [42, 41]. QBIC uses geometric features to represent the shape of an object, which include area, circularity, eccentricity, major axis orientation and invariant moments for a total of 20 features. These features are extracted for each object in the database images. However, the query shape is restricted to include only one object. For matching, a weighted Euclidean distance is used to retrieve the image which includes the most similar object to the query object. A trademark retrieval system that is based on QBIC was developed [32]. However, there is no report for the performance of the system.

System for Trademark Archival and Registration

Wu et al. proposed STAR (System for Trademark Archival and Registration) that uses both color and shape for retrieval [135, 63]. The shape is represented in terms of combination of contour-based (Fourier descriptors) and region-based (invariant moments) features. Objects in the image are segmented manually by the user through an interactive user interface. Once an object or a group of objects is selected by the user, their features are computed and stored. The overall similarity measure is a weighted sum of the individual distances between objects in the query and database images.

Attributed Relational Graph

Petrakis et al. proposed a graph-based approach, called attributed relational graph (ARG), for multi-object image retrieval [91, 94]. The ARG is a variant of the region adjacency graph (RAG) where nodes represent objects and edges between the nodes represent their spatial relationships. In the ARG, node attributes describe the corresponding object which include area, roundness (the ratio of the smallest to the largest second moment), and orientation

(the angle between the horizontal axis and the major axis of the object). For encoding the spatial relationships, edge weights include the minimum distance between the boundaries of two objects and the angle of the line connecting their centroids. To achieve scale and rotation invariance, these features are normalized with respect to the largest object in the image. For matching two ARGs, a graph edit distance between between two ARGs is computed using the least-cost sequence of node/edge deletions and substitutions. The ARG edit distance is optimal but it has exponential computational complexity. Therefore, ARG matching is approximated as a bipartite matching between two sets of nodes which is solved in polynomial-time using the Hungarian method.

Automatic Retrieval of Trademark Images by Shape ANalysis

The ARTISAN (Automatic Retrieval of Trademark Images by Shape ANalysis) project seeks to develop and evaluate a prototype shape retrieval system for trademark images [35, 37]. The system segments a trademark image into regions based on a set of rules that uses heuristic measures such as proximity and parallelism of objects. Then, the boundaries of these regions are approximated by straight-line and circular-arc segments and a set of geometrical features are computed for each region. For matching, the dissimilarity function between the query and a database images is computed as the average of the closest distances between the objects in both images. The authors reported better performance over the whole image matching [36].

Closed contours

Pen and Chen used closed contours for trademark recognition [89]. Trademark images are first decomposed into sets of closed contours which are encoded using chain code method. A two-step string matching algorithm is then used to compute similarities between the closed contours in the query and the database images. Finally, the maximum and the average of the contour similarities are integrated as the whole trademark similarity measure.

Negative shape feature

Soffer and Samet presented a method to match logos based on positive and negative shape features [117]. A positive shape means an object and negative shape means the resulted holes after enclosing the logo in a minimum bounding box. Then, several geometric features are computed for the extracted regions. Here, the similarity function is the average of the pairwise distances between each region in one image and the closest region in the other which allows many-to-one matching between the regions in the two images.

Retrieval by Spatial Similarity

Retrieval by spatial similarity (RSS) approach, for multi-object image retrieval, focuses only on the spatial relationships between the image objects in measuring the similarity function [26, 49, 38]. Basically, spatial methods work on symbolic images[1] which limits their use in fully automated systems. Although these methods have many application domains such as geographical information systems (GIS), spatial methods ignore the shape information.

Concavity graphs

Elbadawy and Kamel proposed the concept of concavity graphs to describe multi-object shape images [13]. Their work was originally motivated by concavity trees which are more useful for single-object images and their application to multi-object images has shortcomings. The concavity graph is a directed graph with a unique root. There are five types of nodes where each node describes a region. A node is classified as an object (o), a hole (h), a concavity (c), a multiple object (mo) or a multiple hole (mh). To describe each region, a vector of its global features is stored at the corresponding node. Fig. 2.10 shows the concavity graph of a multi-object image. Matching concavity graphs is still an open problem.

[1] A symbolic image is a logical representation of the original image where the image objects are uniquely labeled with symbolic names.

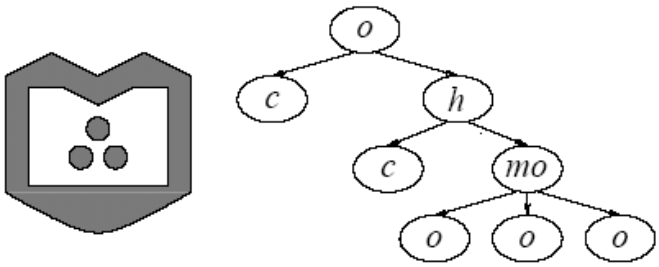


Figure 2.10: A multi-object image (left) and its concavity graph (right).

2.5 Discussions

In this chapter, we have reviewed many techniques in the literature that deal with single- and multi-object shape representation and matching. We note that the single-object shape matching problem has received much greater attention from researches than the multi-object case. In addition, most researchers in shape analysis have not devoted enough attention to the recent advances in other areas such as cognitive science and psychology and focused on the computational aspects of their algorithms. Indeed, this is part of a larger problem known as 'the semantic gap' in content-based image retrieval.

For single-object shape matching, dynamic programming has proven to be an effective approach. Most methods, which have achieved high retrieval accuracy, used dynamic programming for the matching. Although dynamic programming demands high computations, heuristics are usually employed to reduce the computations. For shape description, global descriptors such as moments are simple and fast to compute; however, they don't provide high discrimination capability and cannot describe local variations. On the other hand, structural descriptors provide high discrimination and capture fine shape details, but these descriptors demand complex matching algorithms.

Most of the multi-object shape matching methods are global, which means they handle multiple objects implicitly. The object-based methods only combine the contributions of individual objects; thus, ignoring the spatial relationships between the objects. Recently, the MPEG-7 community have selected the CSS and the ZMMs methods as the contour-based and the region-based shape description and matching standards, respectively, after

comprehensive comparative experiments with other methods in the literature. Therefore, we included these methods for comparison in our experiments throughout this thesis.

Chapter 3

Single-Object Shape Matching and Retrieval

Single-object shape matching constitutes the kernel of the multi-object shape matching method proposed in this thesis. As will be explained in Chapter 4, the shape of each individual object is essential in comparing two multi-object images. In this chapter, two single-object shape matching methods are introduced based on triangle-area representation (TAR) of closed boundaries; a representation computed from the area of triangles formed by equidistant triplet points on the shape boundary [104, 5, 103].

Shape matching is the process of associating primitives of two shapes, generally, in a point-by-point manner. For shape retrieval, shape matching is employed to measure the dissimilarity between a query shape and stored shapes in a database in order to rank the stored shapes according to their similarity with the query shape. The remainder of this chapter is organized as follows. Section 3.1 gives an introduction to the subject. TAR is introduced in Section 3.2. Then, the two matching algorithms are explained in Section 3.3. Experimental results are shown in Section 3.4. Finally, a summary is given in Section 3.5 with discussions.

3.1 Introduction

Shape matching is a critical step in shape retrieval systems. The performance and the complexity of a shape matching method is largely dependent on the invariance, robustness, stability, and uniqueness of the applied shape representation method. A literature survey of shape representation and matching methods is given in Section 2.4.

The multi-scale approach for shape representation and matching is considered the most promising. It can be argues that the human perception of shapes is a multi-scale process by nature. As a human perceives a shape, global features of the shape dominate the human's attention. Then, finer details of the shape become more apparent. Global shape descriptors are generally robust to moderate amounts of noise; however, global descriptors face major difficulties in capturing fine details of shape boundaries. On the other hand, local shape descriptors are superior in describing fine details, but local descriptors are usually sensitive to noise. Therefore, in our opinion, the two requirements, namely, the robustness to noise and the discrimination of fine details, conflict with each other and the choice between them is context-dependent unless another semantic-based measure is employed to distinguish between noise and fine details information.

The representation of shapes requires a number of criteria to be satisfied for reliable shape matching and retrieval. It should be invariant to geometrical transformations such as rotation, scale, translation, and skew. This requirement arises from the problem of projecting the 3D real world objects into 2D images. In addition, a shape representation should satisfy the following criteria: high discrimination capability, computational efficiency, robustness to distortion and noise, compactness, generality of the application, and handling large image databases without heavy degradation in the performance. These criteria are also required by the MPEG-7 standard for measuring the similarity between shapes [74].

Fig. 3.1 shows a block diagram of the two shape matching methods proposed in this chapter. After the boundary of an input image is extracted, the triangle-area representation (TAR) is computed from the parameterized boundary points. In the first method, TAR-concavity image (TAR-CI) is obtained by thresholding TAR signatures. Since the characteristics of TAR-CI are similar to those of the curvature scale-space (CSS) image, a modified version of the CSS matching algorithm [2] is developed for matching concavity peaks of two TAR-CIs. Our experiments show that this matching scheme achieves slightly

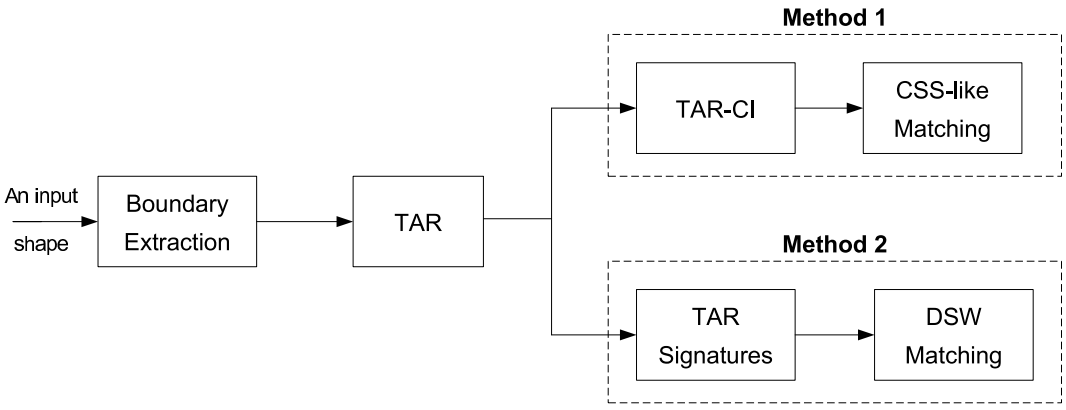


Figure 3.1: Block diagram of the two proposed shape matching methods.

better accuracy than the CSS method [77], which has been selected for MPEG-7 standard after comprehensive comparative experiments with many other methods [84], with considerable reduction in the computational complexity. In the second method, normalized TAR signatures are matched directly in a point-by-point manner using an efficient dynamic programming algorithm called dynamic space warping (DSW) that efficiently searches for the optimal (least cost) correspondence between the points of two shapes. Unlike concavity-based methods such as the first method and the CSS method, the convexity/concavity of each boundary point is utilized in the matching. The main contribution of this method is the higher retrieval accuracy than all published shape retrieval methods so far by a good margin based on the MPEG-7 CE-shape-1 part B retrieval test which constitutes the most comprehensive shape retrieval test in the literature until now.

3.2 Triangle-Area Representation of Closed Boundaries

The triangle-area representation (TAR) is computed from the area of the virtual triangles formed by the points of the shape boundary. In the following, the framework for obtaining TAR of an arbitrary single-object shape is presented. From the computed TAR, two shape descriptors are derived; namely, TAR-concavity image (TAR-CI) and TAR signatures.

Given a single-object binary image, the shape is extracted using the bug-following technique [95]. Each boundary point is represented by its x and y coordinates and separated parameterized contour sequences x_n and y_n are obtained and re-sampled to N points. In this thesis, $N = 128$ unless mentioned otherwise. Then, the curvature of each point is measured using the triangle-area representation (TAR) as follows. For each three consecutive points (x_{n-t_s}, y_{n-t_s}), (x_n, y_n), and (x_{n+t_s}, y_{n+t_s}), where $n \in \langle 1, N \rangle$ and $t_s \in \langle 1, T_s \rangle$ is the triangle side length[1], the signed area of the triangle formed by these points is given by:

$$TAR(n, t_s) = \frac{1}{2} det \begin{bmatrix} x_{n-t_s} & y_{n-t_s} & 1 \\ x_n & y_n & 1 \\ x_{n+t_s} & y_{n+t_s} & 1 \end{bmatrix} \tag{3.1}$$

When the boundary is traversed in counter clock-wise direction, positive, negative and zero values of TAR mean convex, concave and straight-line points, respectively. Fig. 3.2 demonstrates these three types of the triangle area. The triangles at the edge points are formed by considering the periodicity of the closed boundary. The complete TAR signature for the hammer shape is also shown in Fig. 3.2. By Increasing the length of the triangle sides, i.e., considering farther points, the function of (3.1) will represent longer variations along the boundary.

The choice of the number of scales, i.e., triangle side lengths (T_s), is constrained by the implied periodicity of the closed boundaries. More specifically, for a closed contour of N points:

$$TAR(n, t_s) = \begin{cases} -TAR(n, N + 1 - t_s) & t_s = 1 \ldots \lfloor \frac{N-1}{2} \rfloor \\ 0 & \text{at } t_s = \frac{N}{2} \text{ and } N \text{ is even} \\ \text{does not exist} & \text{at } t_s = \frac{N}{2} \text{ and } N \text{ is odd} \end{cases} \tag{3.2}$$

where $\lfloor \frac{N-1}{2} \rfloor$ is the floor value of $\frac{N-1}{2}$. The first line in (3.2) shows the odd symmetry property of the triangle area versus the triangle side length t_s. Also, at t_s equals the middle point of the boundary, the value of the triangle area depends on N, the total

[1]Here, t_s refers to the separation between the indices of the current point and any of its equally-separated neighbors (representing the triangle vertices) in the parameterized boundary sequence.

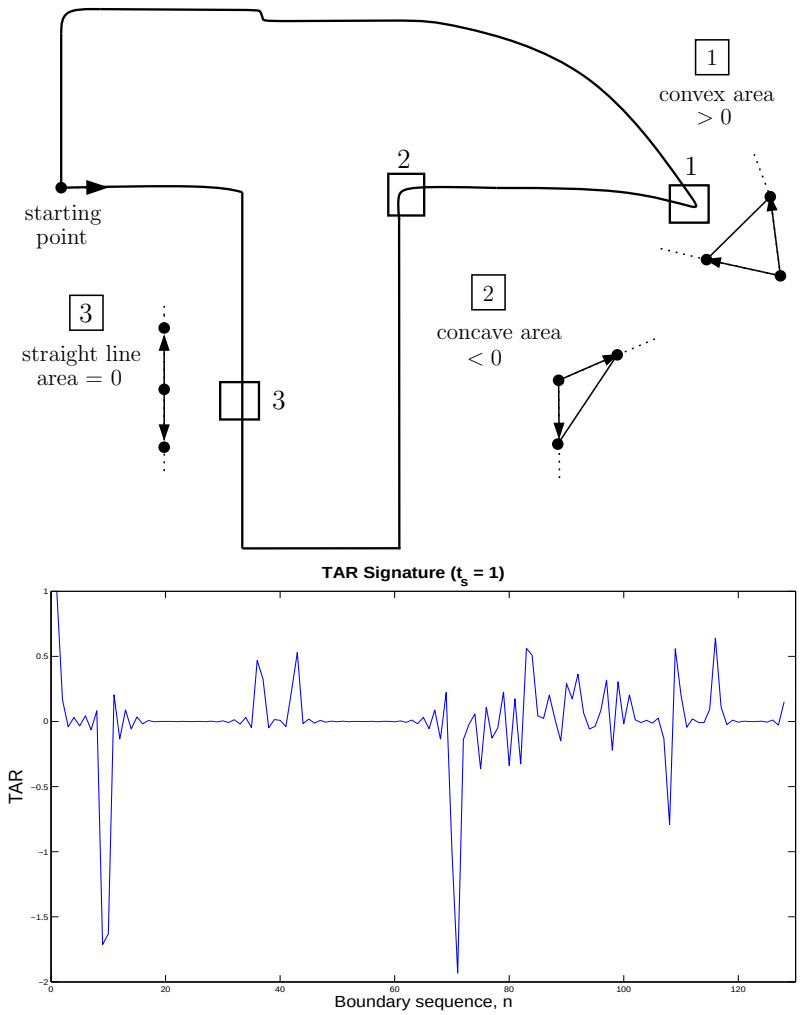


Figure 3.2: Three different types of the triangle-area values and TAR signature for the hammer shape.

number of points on that boundary. If N is odd, then there will be no zero-crossing inflection points on the area curve. Usually, researchers tend to use an even number of points on the shape boundary. In this case, the inflection (zero-crossing) point exists at $t_s = \frac{N}{2}$, where $TAR(n, \frac{N}{2}) = 0$. Fig. 3.3 illustrates the odd symmetry property of TAR

signature that is computed for only one point on the shape boundary of the Misk shape.

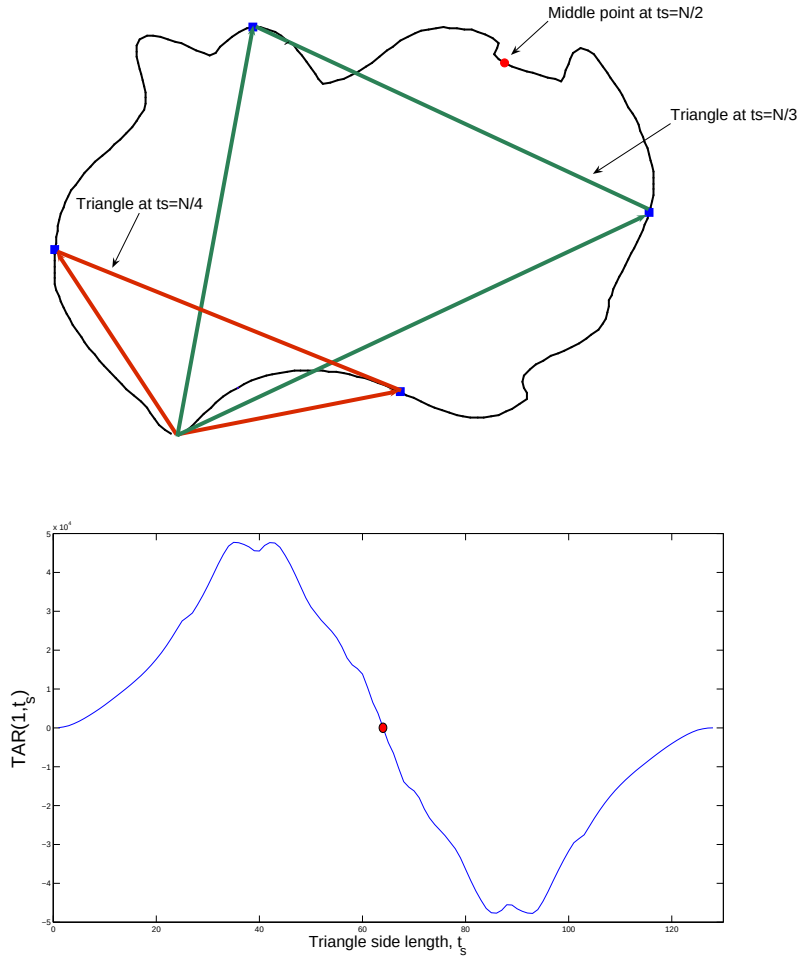


Figure 3.3: Illustration of the odd symmetry of the triangle-area signatures. The upper part shows two virtual triangles for computing TAR signature shown in the bottom part. The circled point on the boundary is the middle-point, which is also considered as the inflection point for TAR signature.

3.2.1 TAR under general affine transformations

For a 2D shape, represented by its boundary sequences, x_n and y_n, and subjected to an affine transformation, the relation between the original and the distorted sequences is given by:

$$\begin{bmatrix} \hat{x}_n \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{3.3}$$

where $\hat{x}$ and $\hat{y}$ are the affine distorted sequences, e and f represent translation, and a, b, c and d reflect scale, rotation and shear. The effect of the translation parameters is easily eliminated by normalizing the shape boundary with respect to its centroid. This normalization is achieved by subtracting from each boundary sequence its mean value. By substituting (3.3) into (3.1), we obtain:

$$\hat{TAR}(n, t_s) = (ad - bc)\, TAR(n, t_s) \tag{3.4}$$

where $\hat{TAR}$ is the affine transformed version of TAR. It is clear that $\hat{TAR}$ is relatively invariant to the affine transformations. Absolute invariance can be achieved by dividing $\hat{TAR}$ by its maximum value. A complete affine test is given in the experimental results section of this chapter.

3.2.2 TAR-Concavity Image

TAR-concavity image (TAR-CI) is a binary 2D plot that shows the indices of concave segments along the boundary at different scale levels, i.e., triangle side lengths [104, 103, 102]. To obtain TAR-CI of a given shape, the function of (3.1) is computed at gradually increasing values of the triangle side length (t_s) and, at each t_s, the indices of the negative-valued (or concave) points are located in TAR-CI. The value of t_s starts from 1 to a maximum value of $\lfloor \frac{N-1}{2} \rfloor$ or until the computed values of (3.1) at all boundary points become positive. Thus, the horizontal and vertical axes represent the boundary point index and the triangle side length (t_s), respectively. Fig. 3.4 illustrates TAR-CI for a camel shape with samples of its TAR computed at $t_s = 5$, 10, 15, 20, and 30. The thick dots on the horizontal axis of each TAR sequence indicate the indices of concave points.

Note that TAR sequence at $t_s = 30$ is all positive; therefore, it does not contain any concave point.

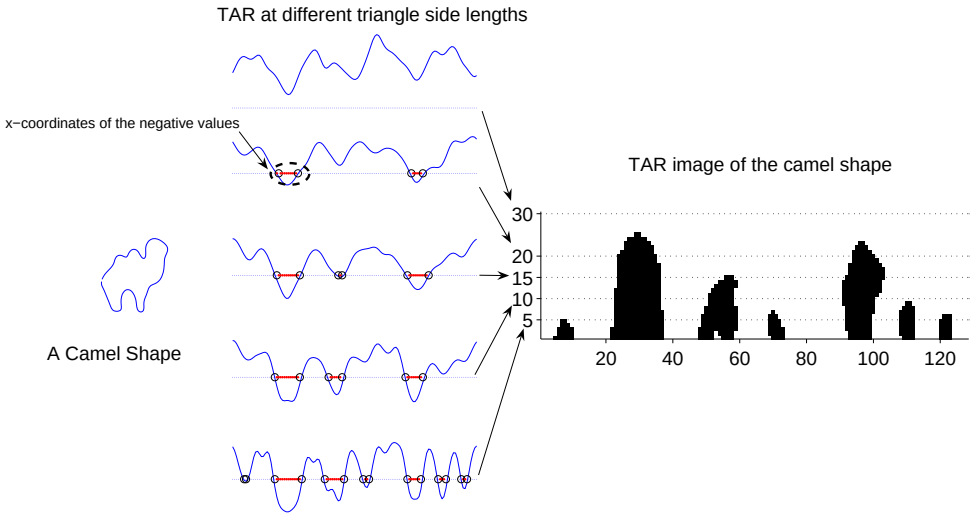


Figure 3.4: Illustration of the computation of TAR-CI. The second column shows TAR signatures computed at specific triangle-side lengths. TAR-CI (third column) is obtained using all the triangle-side lengths from 1 to 30.

In comparison with the CSS image [76], [77] (see Section 2.4 for details), both the CSS image and TAR-CI describe concavities along the boundary at different scale levels. Fig. 3.5 shows TAR-CI and the CSS image of the pentagon shape (device0-18 image in the MPEG-7 database). It is clear that both images represent the major five concavities of the shape. However, there are two major differences between the two images. Firstly, they use different functions to measure the curvature; TAR-CI is derived from the affine invariant function of (3.1) whereas only the zero crossings of the CSS function of (2.15) are affine invariant [76]. Secondly, the scale parameter (or the vertical axis) of TAR-CI is the triangle-side length (t_s) while in the CSS image the standard deviation of the Gaussian kernel (σ) is the scale parameter as the boundary is gradually smoothed. This difference has great implication on the computational complexity of each method. TAR-CI requires simple computations of (3.1) with a number of iterations upper-bounded by $N/2$, where

N is the number of the boundary points, due to the implied periodicity of TAR function (see Fig. 3.3); therefore, the computational complexity of TAR-CI is $O(N^2/2)$ or $O(N^2)$. On the other hand, the CSS image requires convolving the boundary's curvature signature (2.15) with Gaussian kernel of different width at each iteration. Besides the demanding convolution operations, the number of iterations in the CSS image is not upper-bounded and depends on the depth of a shape's concavities. Fig. 3.6 shows TAR-CI (of 60 iterations) and the CSS image (of 438 iterations) of the bat shape. Table 3.1 shows some statistics of the number of iterations required for computing TAR-CIs and the CSS images of the 1400 shapes of the MPEG-7 database. In terms of computational complexity, it is obvious that TAR-CI is significantly less demanding and better characterized than the CSS image.

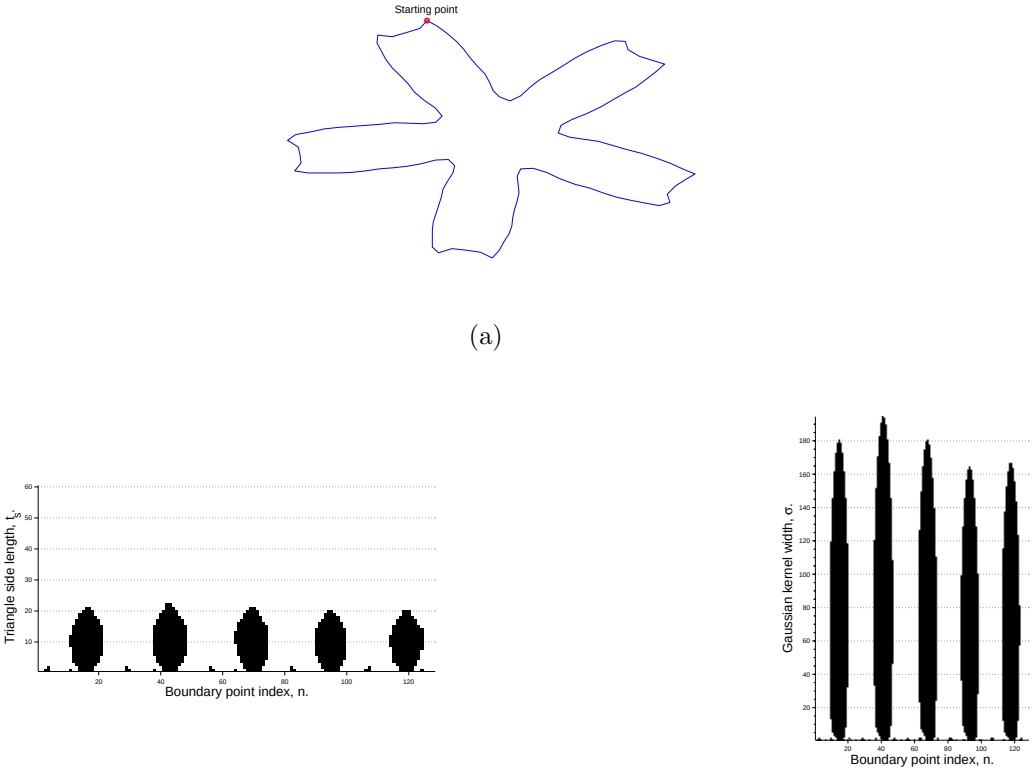
Table 3.1: Statistics of the number of iterations required for computing each of TAR-CI and the CSS image over 1400 shapes of the MPEG-7 database.

Statistic	CSS image	TAR-CI
Mean	278.4	60
Standard deviation	144.4	0
Maximum	500	60

3.2.3 TAR signatures

Here, TAR signatures at different scales are normalized and used directly for the matching. Therefore, TAR signatures form a 2-dimensional surface reflecting the value of TAR function of (3.1) for each boundary point at each scale or triangle-side length, t_s (see Fig. 3.7). Unlike TAR-CI which only indicates whether a boundary point is convex or concave, TAR signature represents the degree of convexity/concavity of each boundary point at a given scale and distinguishes straight-line points as well. Therefore, TAR signatures matrix is regarded as a more informative representation than TAR-CI.

Many researchers have used the area of the triangle formed by the boundary points as the basis for shape representations [98, 52, 113, 112, 101, 39]. These methods use a global measure, i.e., the total sum of the signature, for the signature normalization. In this



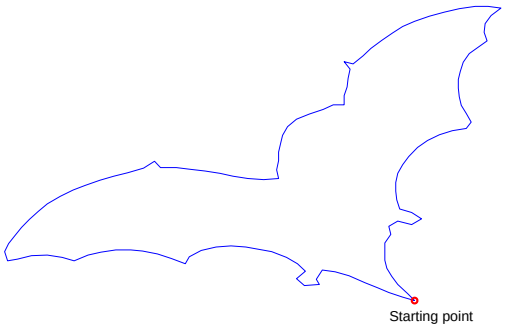
(a)

(b)

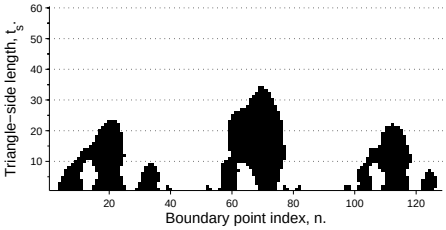
(c)

Figure 3.5: An example of (a) a pentagon shape, (b) its TAR-CI and (c) its CSS image.

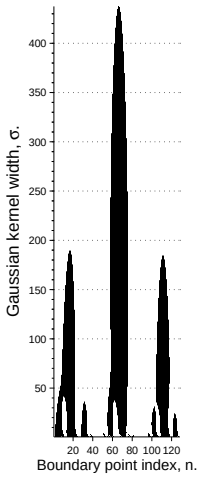
thesis, all possible scale levels are systematically used for computing TAR, whereas the method of [113] uses selected scale levels. Besides, the normalization of TAR signatures is performed locally by dividing TAR signature at each scale, t_s, by the maximum TAR's absolute value at that scale, whereas the normalization is made globally in [113]. Fig. 3.7 demonstrates the difference between the local and the global normalization of the signatures. The local normalization ensures equal contribution of different scale levels during the matching process and prevents the domination of scale levels yielding large areas (the signature values at t_s close to $N/4$, see Fig. 3.3).



(a)

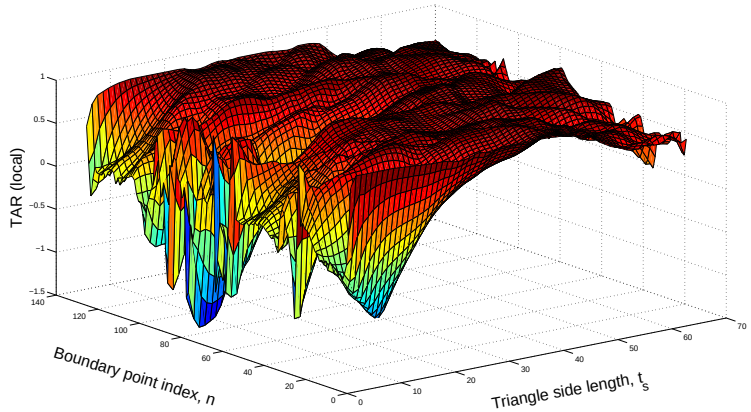


(b)

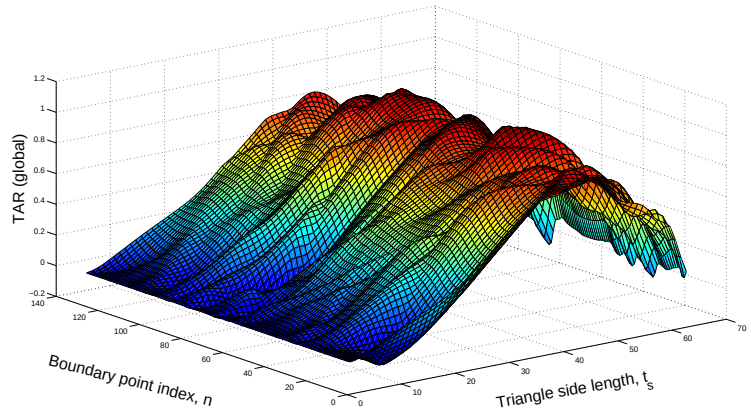


(c)

Figure 3.6: Aa example of (a) a bat shape, (b) its TAR-CI and (c) its CSS image.



(a)



(b)

Figure 3.7: 3D plots of TAR signatures ($t_s = 1$ to 63 and $N = 128$) for the shape in Fig. 3.3. The two plots differ only in the normalization method of the signatures. TAR signatures normalized: (a) locally per scale, and (b) globally according to [113].

3.3 Shape Matching

In this section, two shape matching algorithms are introduced. The first algorithm matches TAR-CI maxima points of two shapes (explained in Subsection 3.2.2). The algorithm developed in this thesis is a modified version of the CSS matching algorithm [2] to accommodate TAR-CI properties. In the second algorithm, TAR signatures (Subsection 3.2.3) are matched directly via an efficient dynamic programming approach called dynamic space warping (DSW). In both algorithms, simple geometric features are added to the matching function to further increase the accuracy and to provide means for indexing.

3.3.1 TAR-CI maxima matching

TAR-CI descriptor is extracted from the maxima points of TAR-CI regions. Since TAR-CI is a binary image, the maxima points are easily extracted using connected-component labeling technique. Fig. 3.8 shows two examples of maxima points extraction from TAR-CIs of the apple and the horse shapes. Intuitively, these maxima points represent the depth and the location of concave segments and convex segments contained in larger concavities. For matching, only the maxima of concavity regions starting at the initial scale ($t_s = 1$) are considered. Besides, a lower bound is set for a maximum to be at least 10% of the largest maximum in TAR-CI.

Now, the algorithm for matching two sets of TAR-CI maxima points is presented. It is a modified version of the CSS matching approach [2] adapted to accommodate the properties of TAR-CI. The main idea behind the CSS matching strategy is to establish a correspondence between the maxima points of a query shape, $\mathcal{Q}$, and those of a stored shape in the database, $\mathcal{D}$. This correspondence maps each query maximum to at most one maximum in the database shape to obtain a meaningful mapping that is used to measure the dissimilarity between the two shapes. Since the maxima locations are sensitive to the starting point of the extracted boundary (or the rotation angle), several most likely initial correspondences between the maxima points of two shapes are explored. In [2], only one initial correspondence is fully extended and considered in the dissimilarity function. However, a robust matching should take into account several possible rotation angles or starting points. Our implementation of TAR-CI maxima matching has two main modifications to the CSS

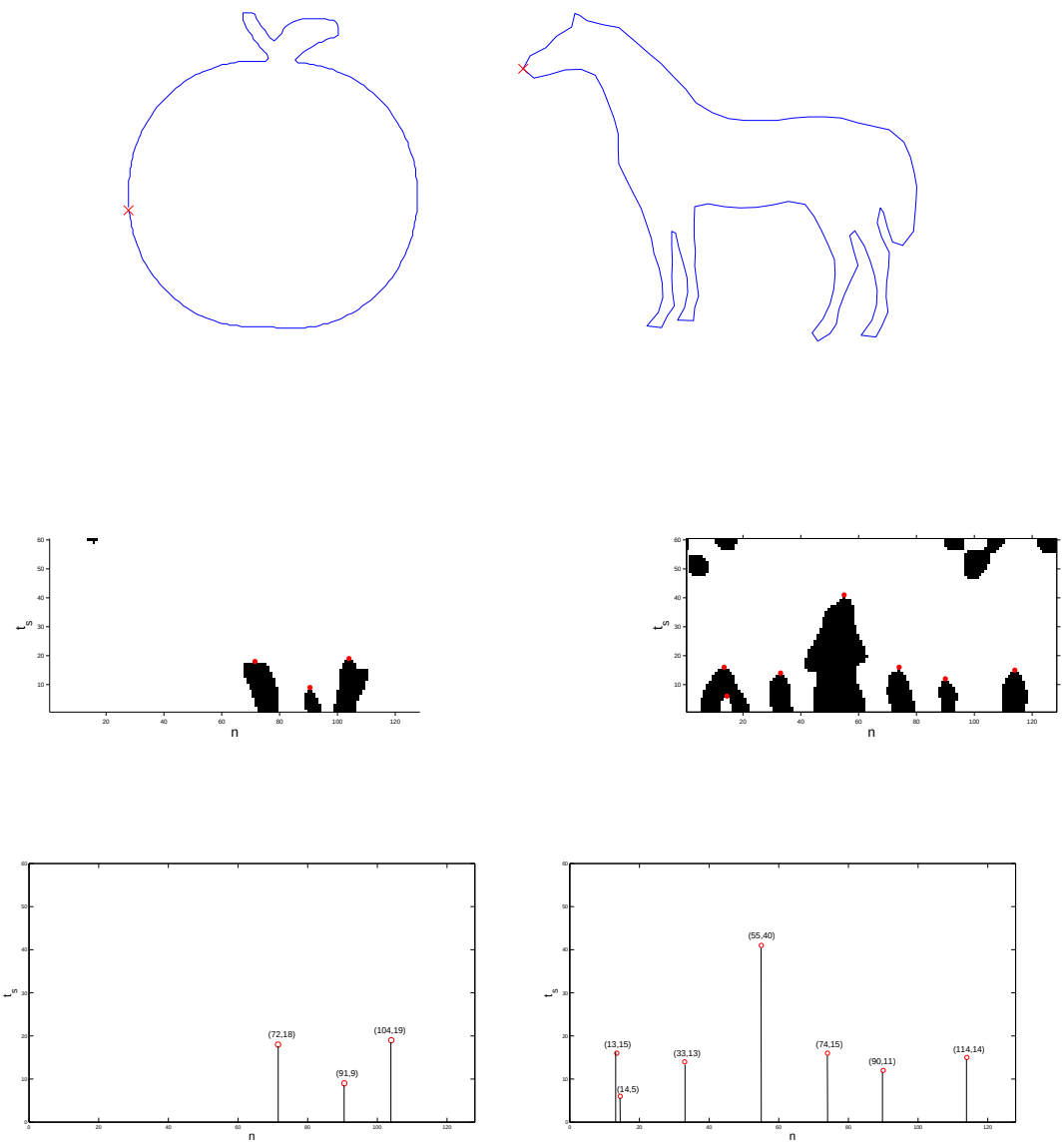


Figure 3.8: Illustration of (up) two shapes, (middle) their TAR-CIs and (bottom) the maxima points extraction.

matching approach introduced in [2]. Firstly, more than one initial correspondence are fully explored to increase the robustness to the rotation angle of the shapes. Our experiments have shown that considering the two least-cost initial correspondences increases the matching accuracy with minor increase in the computational complexity of the matching process. The second modification is to reduce the lower bound parameter that controls the selection of the initial correspondences. This parameter requires a query maximum to be at least 80% of the selected database maximum. In our implementation, this parameter is set to 50% to increase the number of initial correspondences. The reason behind this selection is based on the the observation that TAR-CI usually has a significantly lower number of scales than the CSS image of the same shape (see Subsection 3.2.2); thus, the variance of TAR-CI maxima values is significantly lower than that of the CSS maxima values and exploring more initial correspondences becomes desirable to increase the robustness to the rotation angle.

A pseudo-code of TAR-CI maxima matching algorithm is detailed in Algorithm 3.1. The main algorithm, $MaxMatch(.)$, is based on $match(.)$ function given in Algorithm 3.2. $MaxMatch(.)$ looks at four alternatives for computing the $match(.)$ function. The first two alternatives ensures the symmetry of the matching function by switching the positions of the query and the database maxima sets whereas the other two alternatives makes the matching function invariant to the flipping transformation of the shapes. The dissimilarity measure returned by the algorithm is the sum of the Euclidean distances between the matched TAR-CI maxima points and the values of any unmatched TAR-CI maxima points.

3.3.2 Dynamic Space Warping

In the following, a brief review about the origin of Dynamic Time Warping[2] (DTW) and how some researchers applied it to the shape matching problem is presented. Then, the description of our DSW algorithm to measure the similarity between two shapes based on their TARs is given followed by the definition of the dissimilarity function.

[2]In this thesis, the analogous terminology Dynamic Space Warping (DSW) is used instead of DTW since still images are space-variant as opposed to speech which is time-variant.

Algorithm 3.1 TAR-CI maxima matching (main algorithm): $d = MaxMatch(M^q, M^d)$

Notation:

M^d is a set of maxima in a database TAR-CI.

M^q is a set of maxima in a query TAR-CI.

n_d is the number of maxima in M^d.

n_q is the number of maxima in M^q.

$y^d(i)$ and $x^d(i)$ are the scale and the location of $M^d(i)$, respectively.

$y^q(j)$ and $x^q(j)$ are the scale and the location of $M^q(j)$, respectively.

Γ_{ij} is a candidate match initialized as $\Gamma_{ij}(1) = (M^d(i), M^q(j))$.

c_{ij} is the cost associated to Γ_{ij}.

τ_{ij} is the location shift of Γ_{ij}.

$M^d_{ij} \subset M^d$ and $M^q_{ij} \subset M^q$ associated to Γ_{ij} and Γ_{ij}.

$y^d_{ij}(k)$ and $x^d_{ij}(k)$ are the scale and the location of $M^d_{ij}(k)$, respectively.

$y^q_{ij}(l)$ and $x^q_{ij}(l)$ are the scale and the location of $M^q_{ij}(l)$, respectively.

$\emptyset$ is the empty set.

Require: $y^d(i) \geq y^d(i+1)$, $y^d(n_d) \geq y^d(1)/10$, $y^q(j) \geq y^q(j+1)$, and $y^q(n_q) \geq y^q(1)/10$

1: $c_1 \leftarrow match(M^d, M^q)$

2: $c_2 \leftarrow match(M^q, M^d)$

3: **for** $i = 1$ to n_q {To account for flipping the query} **do**

4: $\quad x^q(i) \leftarrow N - x^q(i)$

5: **end for**

6: $c_3 \leftarrow match(M^d, M^q)$

7: $c_4 \leftarrow match(M^q, M^d)$

8: **return** $d = min(c_1, c_2, c_3, c_4)$

Algorithm 3.2 TAR-CI matching function: $c = match(\ M^d,\ M^q)$

1: Initialization step.

$\Gamma_{11} \leftarrow \{(M^d(1), M^q(1))\}$

$c_{11} \leftarrow abs(y^d(1) - y^q(1))$

$\tau_{11} \leftarrow x^d(1) - x^q(1)$

$M^d_{11} \leftarrow M^d - \{M^d(1)\}$ and $M^q_{11} \leftarrow M^q - \{M^q(1)\}$

2: **for all** $M^d(k)$ s. t. $y^d(k) \geq 0.5 * y^q(1)$ and $k > 1$ **do**

$\Gamma_{k1} \leftarrow \{(M^d(k), M^q(1))\}$

$c_{k1} \leftarrow abs(y^d(k) - y^q(1))$

$\tau_{k1} \leftarrow x^d(k) - x^q(1)$

$M^d_{k1} \leftarrow M^d - \{M^d(k)\}$ and $M^q_{k1} \leftarrow M^q - \{M^q(1)\}$

end for

3: repeat 2 for all $M^d(l)$ s. t. $y^d(l) \geq 0.5y^q(2)$ and $l \geq 1$ to get Γ_{l2}, c_{l2}, τ_{l2}, M^d_{l2}, M^q_{l2}.

4: **for all** Γ_{ij} **do** {expand all candidate matches}

if $M^q_{ij} = \emptyset$ **then**

$c_{ij} \leftarrow c_{ij} + y^d_{ij}(1)$

$M^d_{ij} \leftarrow M^d_{ij} - \{M^d_{ij}(1)\}$

else

locate $M^d_{ij}(k)$ closest to $M^q_{ij}(1)$ s. t. $abs(x^d_{ij}(k) - x^q_{ij}(1)) \leq 0.1 * N$

if $M^d_{ij}(k)$ exists **then**

$$c_{ij} \leftarrow c_{ij} + \sqrt{\left(y^d_{ij}(k) - y^q_{ij}(1)\right)^2 + \left(x^d_{ij}(k) - x^q_{ij}(1)\right)^2}$$

$\Gamma_{ij} \leftarrow \Gamma_{ij} \cup \left\{\left(M^d_{ij}(k), M^q_{ij}(1)\right)\right\}$

$M^d_{ij} \leftarrow M^d_{ij} - \left\{M^d_{ij}(k)\right\}$

$M^q_{ij} \leftarrow M^q_{ij} - \left\{M^q_{ij}(1)\right\}$

else

$c_{ij} \leftarrow c_{ij} + y^q_{ij}(1)$

$M^q_{ij} \leftarrow M^q_{ij} - \{M^q_{ij}(1)\}$

end if

end if

end for

5: repeat 4 for Γ_{kl} and Γ_{mn} with lowest costs until $M^q_{kl} = M^d_{kl} = M^q_{mn} = M^d_{mn} = \emptyset$

return $c = min(c_{kl}, c_{mn})$

DTW in shape matching

The idea of using dynamic programming for matching 1D sequences originally came from the speech recognition community [53, 106, 33], where optimal alignment between two 1D sequences is searched via a dynamic programming approach called dynamic time warping (DTW). In the past few years, several researchers adopted DTW for 1D sequences alignment and matching [16, 130]. Unlike the Euclidean distance, which provides one-to-one alignment, nonlinear alignment can be achieved by the DTW where one point on the sequence can be aligned to one or more points on another sequence.

Recently, many researchers have applied DTW in the 2D shape matching problem. In [92], a DP table is used to find the least cost match between segments of two curves. Merging of segments is allowed during the matching to facilitate a more meaningful correspondence between segments. However, this increases the complexity of matching. In the MCC method [3], the DP algorithm searches for the optimal correspondence between the N-points boundaries. A window, which limits the optimal path to be around the diagonal, is used to make the search more efficient. Another constraint that limits a single point of one contour to correspond to a maximum of two points on the other contour is enforced, which limits the generality of the method and demands more computations. In an attempt to reduce the size of the DP search space, the WARP method [16] applies DTW on normalized points after applying the inverse discrete Fourier transform. In [109], the optimal path in the DP table is used to define an edit distance metric that transforms one shape into the other.

Finding the minimum cost distance using DSW

Now, the DSW algorithm that is used to compute the dissimilarity between two closed boundaries based on their TAR signature matrices is described [5]. At first, it is necessary to define the distance between two individual boundary points. Let $TAR_{\mathcal{Q}}(n, t_s)$ and $TAR_{\mathcal{D}}(n, t_s)$ be TAR signature matrices for a query shape $\mathcal{Q}$ and a database shape $\mathcal{D}$, where $n \in \langle 1, N \rangle$ is the index of the boundary points and $t_s \in \langle 1, T_s \rangle$ is the triangle-side length. Then, the distance between the two boundary points $n \in \mathcal{Q}$ and $m \in \mathcal{D}$ is defined as:

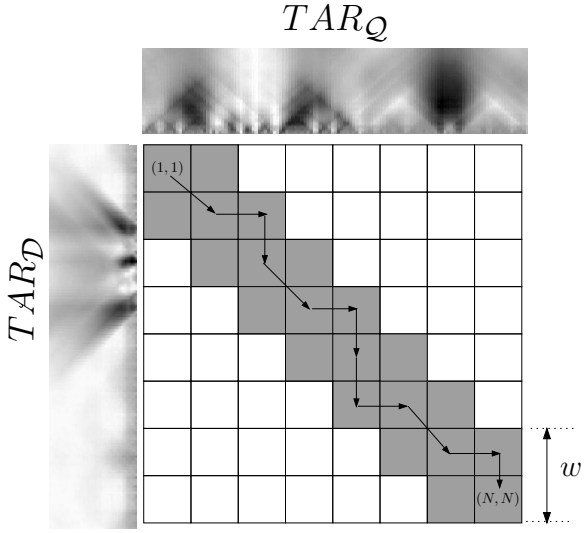


Figure 3.9: DSW table.

$$D_p(n,m) = \frac{1}{T_s}\sum_{t_s=1}^{T_s} \left| TAR_{\mathcal{Q}}\left(n,t_s\right) - TAR_{\mathcal{D}}\left(m,t_s\right)\right| \tag{3.5}$$

Then, an $N \times N$ distance table, DT, is constructed to find the optimal correspondence between the points of the two boundaries. The columns of DT represent the points of one boundary and the rows represent the points of the other. Initially, the elements of DT are set as:

$$DT_{initial}\left(n,m\right) = \begin{cases} 0, \ max\left(1, n-w+1\right) \leq m \leq min(N, n+w-1) \\ \infty, \ \text{otherwise} \end{cases} \tag{3.6}$$

where $n, m \in \langle 1, N \rangle$, w is a predefined diagonal width for DT as illustrated in Fig. 3.9, and $max(a,b)$ and $min(a,b)$ are the maximum and minimum values of a and b, respectively. Only the elements of DT that fall within the w-width diagonal are updated during the DSW search. This initialization of DT avoids computing the distances between all the points of the two boundaries and restricts the distance computation to only those points more likely corresponding to each other. Therefore, the computational complexity is largely reduced while more meaningful correspondences are obtained.

Starting at an arbitrary TAR point for both boundaries of a query shape $\mathcal{Q}$ and a database shape $\mathcal{D}$, the distance table DT is searched, through the diagonal window of width w, left-to-right and up-to-bottom starting from the upper-left element, as shown in Fig. 3.9. The first row and first column elements are initialized as the distance between the corresponding points using (3.5). Then, the rest of the zero-valued elements of DT are updated as:

$$DT(n,m) = D_p(n,m) + \min \begin{cases} DT(n-1,m) \\ DT(n-1,m-1) \\ DT(n,m-1) \end{cases} \tag{3.7}$$

The least cost path through the distance table is the value of element $DT(N,N)$ corresponding to the best matching between the two TAR points according to the selected starting points. However, it is clear that the established correspondence is sensitive to the starting point of each TAR. In order to achieve starting point (or rotation) invariance, it is sufficient to fix the starting point of one TAR and try all N starting points of the other TAR. Moreover, invariant to the mirror transformation can be obtained by flipping the points of one TAR and repeating the search for the N starting points again. The final least cost correspondence, denoted by DT_{min}, is taken as the minimum value of $DT(N,N)$ among all $2N$ runs of the DSW table search.

The dissimilarity measure

Following the approach presented in [3], the dissimilarity function (D_{dis}) is chosen as the minimum cost distance DT_{min} normalized by the *shape complexity* (SC) of each boundary. The motivation behind this normalization is based on the observation that the sensitivity of the human perception to the boundary variations reduces as the shape complexity increases. Here, the shape complexity is considered as the average, over all boundary points, of the absolute differences between the maximum and minimum TAR signature values at all scale levels (or triangle side lengths):

$$SC = \frac{1}{N} \sum_{n=1}^{N} \left| \max_{1 \le t_s \le T_s} \{TAR(n,t_s)\} - \min_{1 \le t_s \le T_s} \{TAR(n,t_s)\} \right| \tag{3.8}$$

Then, the dissimilarity function between a query shape $\mathcal{Q}$ and a database shape $\mathcal{D}$ is given by:

$$D_{dis}\left(\mathcal{Q},\mathcal{D}\right) = \frac{DT_{min}\left(\mathcal{Q},\mathcal{D}\right)}{K + SC_{\mathcal{Q}} + SC_{\mathcal{D}}} \tag{3.9}$$

where $DT_{min}(\mathcal{Q},\mathcal{D})$ is the minimum cost distance between the two shapes $\mathcal{Q}$ and $\mathcal{D}$ computed using the DSW table search described earlier in this section and $SC_{\mathcal{Q}}$ and $SC_{\mathcal{D}}$ are the complexities of shapes $\mathcal{Q}$ and $\mathcal{D}$, respectively. A constant K is added to prevent the domination of the denominator when the complexities are very small. In our experiments, K is set to 1. Fig. 3.10 shows two examples of similar shapes with small and large shape complexities, respectively.

In many practical applications, normalizing the shape dissimilarity function within a dynamic range, for instance between 0 and 1, is a fundamental requirement or, at least, highly desirable. Obviously, this normalization allows computing the similarity function directly from the dissimilarity (by subtracting the latter from its upper bound). In this thesis, the dissimilarity normalization is essential to facilitate combining the contributions of different single-object shape similarities of two multi-object shape images into one similarity function. Without such normalization, a single similarity value between two shapes in the two multi-object images may dominate the other similarity values just because its magnitude is large. In conclusion, normalizing the shape similarity function ensures equal emphasis of each individual distance within the overall similarity function (as detailed in Chapter 4). Here, the function of (3.9) has a lower bound of 0 (when $\mathcal{Q} = \mathcal{D}$) since it represents a sum of absolute differences (or positive values); therefore, it is sufficient to maintain an upper bound on its value. For this purpose, (3.9) is divided by $L_{\mathcal{QD}}$, the length of the least-cost path through the DSW table (see Fig. 3.9), to give the final DSW distance:

$$DSW\left(\mathcal{Q},\mathcal{D}\right) = \frac{D_{dis}\left(\mathcal{Q},\mathcal{D}\right)}{2L_{\mathcal{QD}}} \tag{3.10}$$

Since (3.10) is a normalized dissimilarity function, the analogous similarity function is given as :

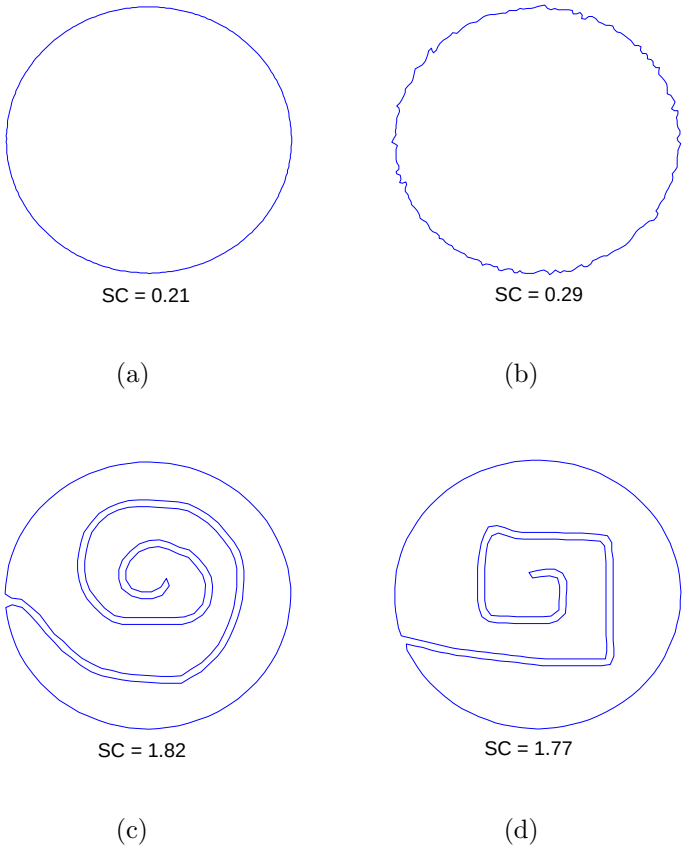


Figure 3.10: Two examples of similar shapes with small shape complexities (panels a and b) and with large shape complexities (panels c and d).

$$DSW_{sim}\left(\mathcal{Q},\mathcal{D}\right)=1-\frac{D_{dis}\left(\mathcal{Q},\mathcal{D}\right)}{2L_{\mathcal{QD}}} \tag{3.11}$$

Selecting the search window width (w)

Restricting the DSW table search to be within a w-diagonal window has three main advantages. Firstly, it avoids matching one point in the first shape with a farther point in the other; thus, more meaningful correspondences are obtained. Secondly, it limits the

maximum number of points of one shape that can be matched to a single point in the other shape. Thirdly, the computational complexity is significantly reduced.

The limitations on the search window have been studied in many fields. In speech processing, where the DTW was firstly originated, the optimal window is selected around 10% of the original sequence width. Recently, Ratanamahatana and Keogh [96] proved that the 10% width, widely selected by the data mining community, is too large for data mining applications and suggested using 4% as an optimal width for such applications. In this thesis, our experiments show that selecting $w = 3$ as the DSW width achieves good accuracy (which is approximately 2.5% of $N = 128$) and larger values increase the computations without improving the performance. While there is no solid study about the window selection in the shape analysis community (due to the recent adaptation of dynamic programming), our selection agrees with Adamek and O'Connor [3] selection for the same application. In their work, the window width is selected about 5%; however, they imposed another restriction that limits the maximum number of points in one shape corresponding to a single point in the other shape which is intuitively equivalent to reducing the window width by a magnitude.

3.3.3 Geometric features

In many practical applications, it is highly desirable that the shape descriptor provides means for indexing in order to organize the database efficiently. Abbasi et al. [2] used a set of global features, i.e., circularity, eccentricity, and aspect ratio, at an initial stage to discard very dissimilar shapes and increase the discrimination power of the descriptor. Jain and Vailaya [55] used invariant moments and histograms of edge directions for fast pruning of the database.

In this work, a set of simple geometric features are used to further increase the discrimination ability of the dissimilarity function and to allow indexing the database shapes. These features include aspect ratio (AR), eccentricity (E) and solidity (S). These features contain considerable information about the global properties of a shape. However, since some dissimilar shapes have comparable global features, the indexing using the global features comes at the price of the accuracy. Therefore, the final dissimilarity function between a query shape $\mathcal{Q}$ and a database shape $\mathcal{D}$ using TAR-CI matching (described in Algorithm

3.1) is given as:

$$\widehat{MaxMatch}(\mathcal{Q},\mathcal{D}) = \alpha_{AR}\,|AR_{\mathcal{Q}} - AR_{\mathcal{D}}| + \alpha_E\,|E_{\mathcal{Q}} - E_{\mathcal{D}}| + \alpha_S\,|S_{\mathcal{Q}} - S_{\mathcal{D}}| + MaxMatch(\mathcal{Q},\mathcal{D}) \tag{3.12}$$

where $AR_{\mathcal{Q}}$, $E_{\mathcal{Q}}$, and $S_{\mathcal{Q}}$ are the aspect ratio, eccentricity, and solidity of the query shape $\mathcal{Q}$ (same for the database shape $\mathcal{D}$), and α_{AR}, α_E and α_S are the associated weights. Similarly, the final dissimilarity function between a query shape $\mathcal{Q}$ and a database shape $\mathcal{D}$ using the DSW matching, given by (3.10) is given as:

$$\widehat{DSW}(\mathcal{Q},\mathcal{D}) = \alpha_{AR}\,|AR_{\mathcal{Q}} - AR_{\mathcal{D}}| + \alpha_E\,|E_{\mathcal{Q}} - E_{\mathcal{D}}| + \alpha_S\,|S_{\mathcal{Q}} - S_{\mathcal{D}}| + DSW(\mathcal{Q},\mathcal{D}) \tag{3.13}$$

Our experiments show that both (3.12) and (3.13) performs effectively under a wide range of the weight values, which supports the generality of our approach.

3.4 Computational Complexity

The complexity of each of TAR and the two matching algorithms is evaluated separately. Note that both matching algorithms presented in Section 3.3 are based on the same representation. It should also be noted that the complexity of the matching stage is more critical since the representation of the database images can be computed prior to the time of the matching whereas the matching usually takes place between the query image and most (if not all) the database images.

3.4.1 TAR

TAR computation involves calculating the triangle area, according to (3.1), at each of the N points of the boundary. In addition, at each boundary point, the triangle area is calculated at different scales (or triangle side lengths). Typically, there are $\frac{N-1}{2}$ scales (see Section 3.2). Therefore, the computational complexity of TAR stage is $O\left(N.\frac{N-1}{2}\right)$ or $O\left(N^2\right)$. The extraction of TAR-CI maxima is performed using a simple connected-component labeling of the binary TAR-CI which has a resolution of $\frac{N-1}{2} \times N$. For TAR signatures, they are used directly for the matching and no feature extraction is required.

3.4.2 TAR-CI maxima matching

The complexity of the function $match(.)$ (given by Algorithm 3.2) depends on the number of concavities and sub-convexities of both shapes which is by far less than the number of the boundary points (N); therefore, it is reasonable to assume the number of maxima points to be $O\left(log(N)\right)$. The function $match(.)$ creates at most $O\left(2log(N)\right)$ initial correspondences of which the best two correspondences are fully extended. Since expanding a correspondence requires comparing each maximum in the query set with all maxima in the database set, the correspondence expansion has $O\left(log(N).log(N)\right)$ or $O\left(log^2(N)\right)$ complexity and the full expansion of the best two correspondences requires $O\left(2log^2(N)\right)$. Thus, the complexity of the function $match(.)$ becomes $O\left(2log(N) + 2log^2(N)\right)$. The main function $MaxMatch(.)$ (given by Algorithm 3.1) uses the function $match(.)$ four times; thus, the overall complexity becomes $O\left(4(2log(N) + 2log^2(N))\right)$ or $O\left(log^2(N)\right)$. Including the geometric features in the matching function of (3.12) does not affect the complexity of the algorithm since this addition requires only three computations of the absolute difference operation.

3.4.3 DSW matching

For the DSW matching function of (3.10), the length of the least-cost path L is computed during the table search for computing D_{dis}; thus, the latter governs the complexity of the algorithm. Each of the each of the shape complexity terms $SC_{\mathcal{Q}}$ and $SC_{\mathcal{D}}$ in the denominator of (3.9) requires $O\left(N\right)$ complexity as given by (3.8). For the minimum cost distance term DT_{min} given by (3.7), the DSW table search is restricted within the diagonal w-width window; thus, the DSW table search complexity is $O\left(wN\right)$ (usually $w \ll N$). Since the DSW search is repeated for N starting points, the complexity becomes $O\left(wN^2\right)$. Finally, by considering the flipping operation, the total complexity of the matching stage turns out to be $O\left(2wN^2\right)$ or $O\left(N^2\right)$ (for $N = 128$, our experiments show that $w = 3$ is good enough and larger w doesn't achieve better results).

3.5 Experimental Results

In this section, the performance of each of the proposed matching algorithms (Section 3.3) is evaluated and compared with existing methods in the literature using five standard experiments on two benchmark shape databases. The first three experiments investigate the robustness of the matching methods to the affine transformation distortions; namely, the robustness to scale (MPEG-7 CE-shape-1 part A1 test), the robustness to rotation (MPEG-7 CE-shape-1 part A2), and the robustness against skew. Then, the results of MPEG-7 CE-shape-1 part B retrieval test, which is the most comprehensive shape retrieval test in the literature so far, are presented. Finally, the robustness to partial occlusion is tested.

Two standard shape databases are used. The first is the well-known MPEG-7 CE-Shape-1 database [66] which consists of 1400 images semantically classified into 70 classes. The shapes of this database are derived from natural objects, man-made objects, objects extracted from cartoons, and manually-drawn objects under various rigid and non-rigid deformations (a sample of the database is shown in Fig. 3.11). The importance of this database is due to the fact that it is *the only set that is used to objectively evaluate the performance of various shape descriptors* [67]. The other database is the Kimia's database [110] which contains 99 images for 9 categories as shown in Fig. 3.12. There are 11 images for each category and most of the images are partially occluded.

3.5.1 Robustness to scaling and rotation

Here, the results of our method, according to the MPEG-7 Core Experiment CE-Shape-1 part A1 (for scaling) and part A2 (for rotation) tests, are presented. The database used in part A1 includes 420 shapes, 70 basic shapes from the MPEG-7 CE-Shape-1 database (one shape per class) and 5 derived shapes from each basic shape by scaling the images with factors 2, 0.3, 0.25, 0.2, and 0.1. Each of the 420 shapes is used as a query and the number of correct matches among the first 6 retrieved shapes represents the accuracy of that query. The overall accuracy is the average of the accuracy values of all the queries.

Similarly, the database used for part A2 test consists of 420 shapes as in part A1, but the derived images are obtained by rotating each basic image with angles 9, 36, 45, 90,

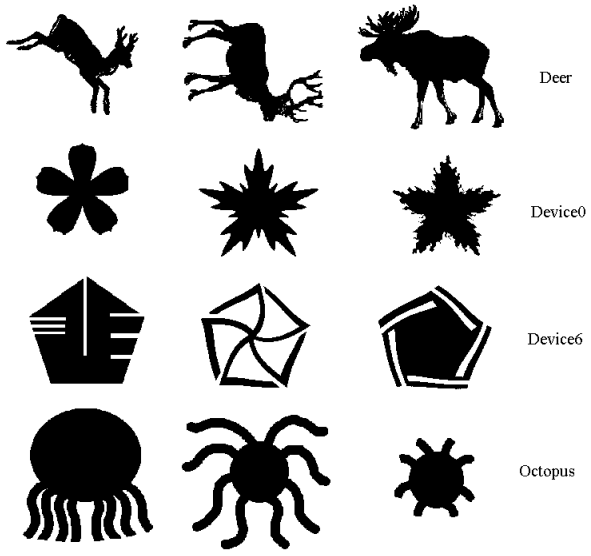


Figure 3.11: Sample of the MPEG-7 CE-shape-1 database.

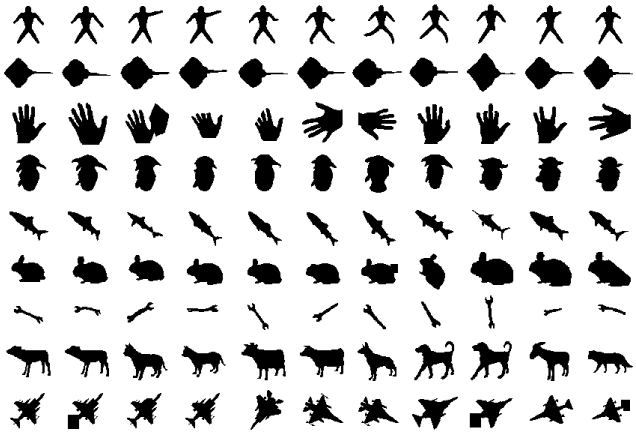


Figure 3.12: Kimia's database.

and 150 degrees. The overall accuracy is evaluated as in part A1.

Table 3.2 summarizes the results of both tests for our two matching methods along with four other methods from the literature. The Curvature Scale Space (CSS) method is due to Mokhtarian et al. [77, 2] and has been selected as the MPEG-7 standard for the boundary-based shape descriptor after comprehensive experiments. The other three methods are the Beam Angle Statistics (BAS) [10], the Zernike moment magnitudes (ZMMs) [61], and the Visual Parts (VP)[65] (see Section 2.4 for details). The results in the table clearly shows that the DSW matching, given by (3.10), performs better than the others even without using the global parameters of the shapes. Using DSW with the global features, given by (3.13), further improves the performance. Regarding the MaxMatch function of (3.12), its performance is comparable to that of the CSS matching and slightly lower than the ZMMs according to these tests.

Table 3.2: Comparison of the results of different methods on the MPEG-7 CE-shape-1 part A test.

Test	BAS [10]	CSS [77]	ZMMs [61]	VP [65]	$\widehat{\text{MaxMatch}}$	DSW	$\widehat{\text{DSW}}$
Part A1	90.87%	92.86%	93.15%	88.65%	92.93%	96.1%	98.7%
Part A2	100%	100%	100%	100%	100%	100%	100%

3.5.2 Robustness to skew

In this experiment, the robustness of each of the MaxMatch and the DSW matching algorithms to the skew distortion is tested and compared to that of the CSS. The distorted shapes are obtained by transforming the original 70 shapes representing the MPEG-7 dataset groups using equation (3.3). The parameters used to obtain these shapes are $b = [0, 0.4, 0.8, 1.5, 2, 3, 4, 6]$. A sample of these distorted shapes are shown in Fig. 3.13.

Fig. 3.14 shows the precision-recall curves of the CSS method [77], $\widehat{\text{MaxMatch}}$ function of (3.12), and $\widehat{\text{DSW}}$ function of (3.13). The plots demonstrate that the MaxMatch method achieves higher accuracy than the CSS at all recalls and the DSW method is the best according to this test.

Figure 3.13: Sample of the skew distorted shapes.

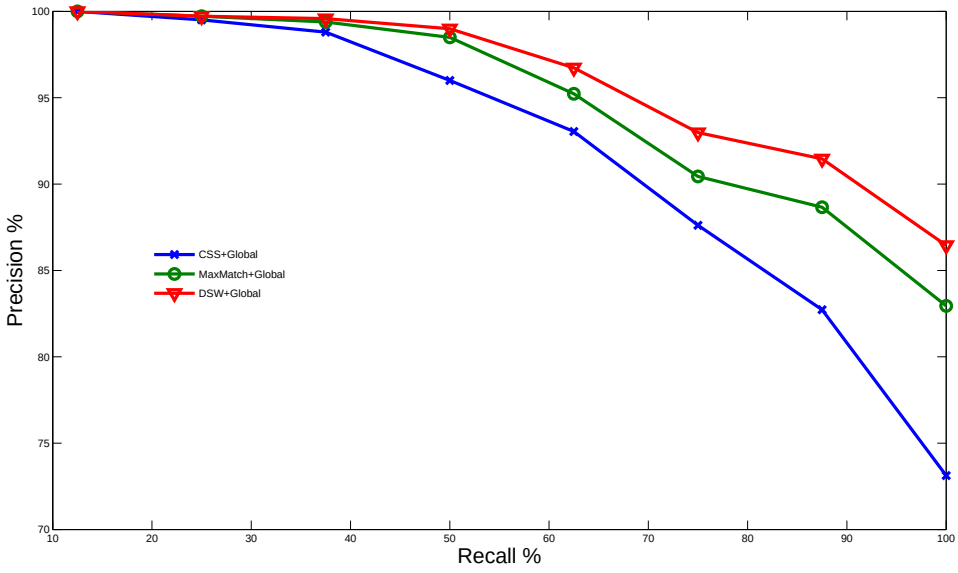


Figure 3.14: Skew test precision-recall curves for the CSS, the $\widehat{\text{MaxMatch}}$, and the $\widehat{\text{DSW}}$ methods.

Table 3.3: Comparison of the results of different methods on the MPEG-7 CE-shape-1 part B (bulls-eye) test.

Test	BAS [10]	CSS [77]	ZMMs [61]	VP [65]	MCC [3]	WARP [16]
Part B	82.37%	81.12%	70.63%	76.45%	84.93%	58.50%
Test	CED [109]	IDSC [69]	MaxMatch	$\widehat{\text{MaxMatch}}$	DSW	$\widehat{\text{DSW}}$
Part B	78.17%	85.40%	77.20%	81.30%	87.13%	87.75%

3.5.3 Similarity retrieval test

The retrieval effectiveness of our two matching methods are evaluated using the MPEG-7 Core Experiment CE-Shape-1 part B test (also known as *bulls-eye* test), which is the main part of CE-Shape-1. All the 1400 images of the MPEG-7 database were used here. Each image is used as a query; then, the number of correct matches was counted in the first 40 retrieved shapes. As stated in [66], a 100% retrieval rate in this case is not possible using only the shape information since many classes contain very different objects. In our opinion, this shape retrieval test is the most challenging in the literature so far.

Table 3.3 shows the results of the MaxMatch and the DSW methods and many recent methods; namely, BAS [10], CSS [77], ZMMs [61], VP [65], MCC [3], WARP [16], CED [109], and IDSC [69] (see Section 2.4 for details). The best performance based on the bulls-eye test was reported as 85.4% [69]. Our DSW method outperforms all existing methods in the retrieval accuracy (87.13%) even without using the global parameters. Fig. 3.15 shows the breakdown of the total retrieval rate into the retrieval rates for each class for both the $\widehat{\text{DSW}}$ and the MCC [3]. To get better insight about the retrieval performance of our methods, Fig. 3.16 show the precision-recall curves of our two matching methods and the CSS. Clearly, the DSW outperforms both the CSS and the MaxMatch with a good margin at all recalls even without using the global parameters.

3.5.4 Retrieval using the Kimia's database

In this test, the Kimia's database [110] is used. As shown in Fig. 3.12, partial occlusion is the main factor of variation among shapes of the same category. Each shape in the database is considered as a query and the first 10 retrieved shapes, excluding the query,

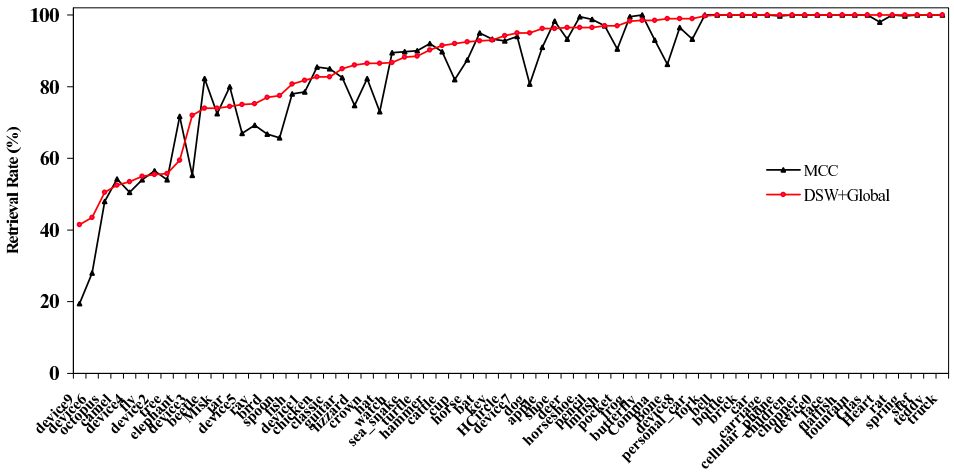


Figure 3.15: Results of the MPEG-7 CE-shape-1 part B test for each class for both the $\widehat{\text{DSW}}$ and the MCC methods.

are determined. Then, the correct retrievals for each ranking, over all 99 shapes, are counted. Table 3.4 summarizes these results for the DSW and three other methods. Note that the maximum number of possible correct retrievals in each case is 99. The shock graph edit (SGE) method [110] breaks down a shape's skeleton into parts (or shocks) and represents them as graph nodes and their relations as graph edges; thus, it handles partial occlusions explicitly. The SGE outperforms the shape context method [17] in this test. However, the performance of the SGE method on the MPEG-7 part B test was not reported. In contrast, the IDSC method [69] slightly outperforms the SGE method. Table 3.4 also shows the results of $\widehat{\text{MaxMatch}}$ given by (3.12), DSW given by (3.10), and $\widehat{\text{DSW}}$ given by (3.13). Our latter method achieves comparable performance to the IDSC method according to this test.

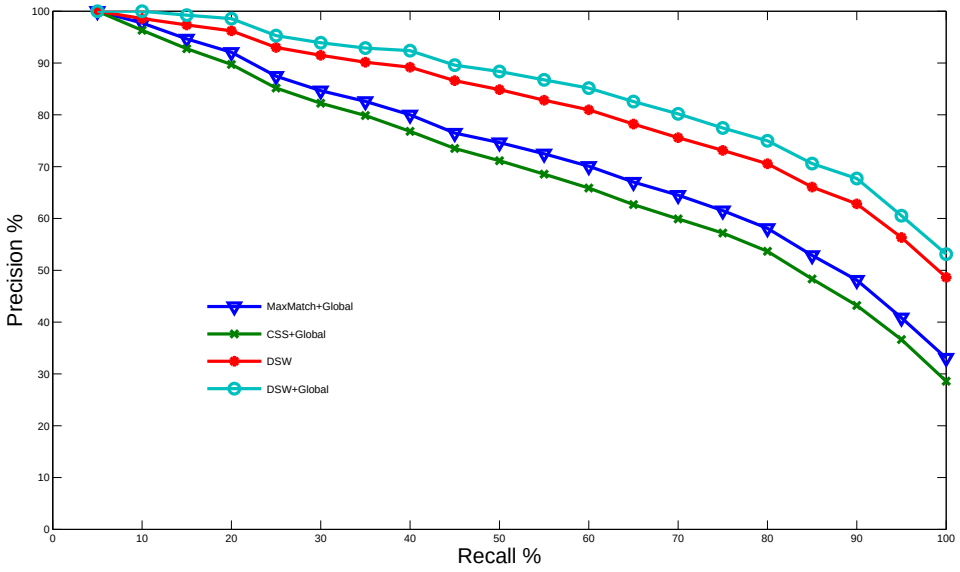


Figure 3.16: Retrieval precision-recall curves for the CSS, the $\widehat{\text{MaxMatch}}$,the DSW, and the $\widehat{\text{DSW}}$ methods.

Table 3.4: Results on Kimia database of 99 shapes. The table shows the number of correct retrievals, over all 99 shapes, at different rankings. See text for details.

Method	Ranking of the retrieved shape									
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Shape Context [17]	97	91	88	85	84	77	75	66	56	37
SGE [110]	99	99	99	98	98	97	96	95	93	82
IDSC [69]	99	99	99	98	98	97	97	98	94	79
$\widehat{\text{MaxMatch}}$	99	97	94	91	88	84	76	67	57	40
DSW	99	99	96	97	96	94	91	84	70	45
$\widehat{\text{DSW}}$	99	99	99	98	98	97	98	95	93	80

3.6 Conclusions and Discussions

In this chapter, two single-object shape matching algorithms are introduced. For shape description, both algorithms employ the triangle-area representation (TAR) of a shape's boundary which measures the convexity/concavity of each boundary point using the area of the triangle formed by equidistant boundary points at different scales. The first matching algorithm, called MaxMatch, searches for the best correspondence between the maxima points of two TAR-concavity images, i.e., binary images of concavity regions obtained by thresholding TAR of each boundary. In the second matching algorithm, the optimal correspondence between the points of two boundaries is searched efficiently using dynamic space warping (DSW). Based on the established correspondence in both matching methods, a dissimilarity function is derived. Global shape features, including aspect ratio, eccentricity and solidity, are incorporated in the dissimilarity function to further increase the discrimination ability and to facilitate indexing in large shape databases. Both techniques are invariant to translation, rotation, scaling and skew. Our experiments show that MaxMatch outperforms the curvature scale-space (CSS) method [1, 77], which has been selected for MPEG-7 standard after comprehensive comparative experiments with many other methods [84]. For the MPEG-7 CE-shape-1 part B test, which is considered the most comprehensive shape retrieval test yet, the DSW method outperforms all existing methods by a good margin.

Table 3.5 shows a comparison between the MaxMatch and the DSW methods. It is clear that DSW achieves higher accuracy than MaxMatch. Regarding computational complexity, MaxMatch requires less computations; however, it requires extracting the maxima points whereas DSW works directly on TAR signatures without any feature extraction. The processing times reported in the table are per query shape and obtained using Matlab ©(version 7.0) program running on Pentium IV 3.0 GHz PC. It should be noted that the codes are not optimized and better speeds can be obtained using other programming languages such as C and C++. Another desirable property of DSW is that it can be easily used for measuring the similarity between two shapes since its value is normalized between 0 and 1. For these reasons, the DSW matching is used in the multi-object shape matching described in Chapter 4.

Table 3.5: Comparison between MaxMatch and DSW.

Method	MaxMatch	DSW
Complexity	$O\left(log^2(N)\right)$	$O\left(N^2\right)$
Accuracy (bullseye test)	81.30%	87.75%
Compactness	$log(N)$ points	$N \times T_s$ matrix
Average representation time	16 msec	16 msec
Average feature extraction time	36 msec	N/A
Average matching time	3 msec	15 msec
Normalized	No	Yes

Chapter 4

Multi-Object Shape Representation and Matching

In this chapter, curvature tree (CT) is introduced as a tool for representing and matching multi-object shape images [8]. The main aim is to provide a unified framework for geometry-based image retrieval that includes the shape and the topology of objects and holes comprising a multi-object image[1]. In the following, an introduction to the subject is given in Section 4.1. Then, a review of recent psychological findings, related to the human inference about comparing multi-object images, is presented in Section 4.2. The CT is introduced in Section 4.3; Section 4.4 explains an approximate and an exact CT matching algorithms. Section 4.5 discusses the computational complexity followed by experimental results in Section 4.6. Finally, we conclude our work in Section 4.7.

4.1 Introduction

With the ever-increasing number of digital images generated everyday, textual annotation of images becomes impractical and inefficient for image description and retrieval. This trend is mainly motivated by the rapid advances in imaging technologies and the availability of the internet access virtually everywhere. Thus, content-based image retrieval (CBIR) has

[1]In this thesis, we assume the objects in an image are already segmented and their boundaries are well identified. Therefore, binary images are considered.

received considerable attention in recent years from researchers in various fields [116, 128]. For comparing images, CBIR uses generic image features which are traditionally either intensity-based (color and texture) or geometry-based (shape and topology). Many systems have been proposed for CBIR; among the most popular ones are QBIC from IBM [42], Virage [12], and Photobook from MIT [90]. Geometry-based image retrieval is generally less developed than the intensity-based retrieval; for example, the QBIC system is more successful in the intensity-based than in the geometry-based search [127]. Currently, multi-object image representation and matching is an active area for research.

A common limitation of the existing geometry-based retrieval systems is not considering simultaneously both shape and topology of image objects (or components) which may reveal important properties of the scene being analyzed. In a recent study from the visual cognition community, Markman and Gentner [72] concluded that structural relations between image components play a central role in the human similarity comparison process. An earlier finding by Lowe [71] states that the similarity between two groups of objects does not equal the sum of similarities between the individual objects.

In this chapter, the CT is introduced as a tool for multi-object image representation and matching. The aim is to provide a unified framework for geometry-based image retrieval that includes the shape and topology of objects and holes composing an image. More specifically, the CT is a hierarchical data structure that reflects the inclusion relationships among objects and holes. It consists of nodes and edges where each node stores shape descriptor of the closed boundary corresponding to an object or hole. For measuring the similarity between two multi-object images, their CTs are matched based on the notion of maximum similarity subtree isomorphism (MSSI). This matching scheme is highly desirable since it handles both shape and topology at once. The matching scheme comes in accordance with many recent findings from the visual cognition community about comparing multi-object images (see Section 4.2). To the author's best knowledge, the derivation of a similarity measure between two multi-object shapes based on structural isomorphism is novel. Two attributed tree matching algorithms are introduced to solve the MSSI problem: an approximate and an exact. The approximate algorithm follows a continuous optimization approach that transforms the MSSI problem, which is a discrete combinatorial optimization problem, to the continuous domain, uses continuous optimization techniques

to solve the problem, and then transforms the solution back to the discrete domain. In this algorithm, the continuous optimization method, proposed by Pelillo et al. [88, 87, 86], is adapted with two modifications. The motivations for these modifications are to avoid spurious solutions (which cannot be mapped back to the discrete domain) when the optimum solution is not unique and to limit the number of computations for solving the MSSI problem (as explained in Section 4.4). In the other algorithm, an exact solution to the MSSI problem is searched by recursively computing all possible subtree isomorphisms between two CTs and then selecting the one yielding maximum similarity. Although the MSSI problem is NP-hard for unordered trees such as our CT, the nonnegative property of the similarity function between the nodes of two CTs drastically reduces the search space for the optimum solution and enables the algorithm to work in polynomial time.

The choice between the approximate and the exact algorithms is a tradeoff between accuracy and speed. As will be shown theoretically and experimentally, the approximate algorithm achieves acceptable accuracy compared to the exact algorithm with considerable improvement in the computational complexity. The performance of the proposed approach is evaluated using two application domains: logo retrieval from a database of 1580 logo images and retrieval of medical images from a database of 13500 Magnetic Resonance Images (MRI). In both applications, comparisons with the state-of-the-art methods from the literature are presented. The experimental results prove the superiority of the proposed approach for obtaining more meaningful retrieval results as compared to the existing methods in both applications.

4.2 Image Comparison in Psychology

Advances in signal processing and applied mathematics have enabled researchers to develop sophisticated techniques to analyze the geometric content of images. Although such methods are effective in capturing low-level image features, the gap between these low-level features and high-level semantics is still wide [34]. Humans perform image comparisons efficiently and effortlessly; however, it is not the case for machines. In the following, we review recent studies in the psychological literature about how humans perceive and compare multi-object images. This is particularly important in developing computational models

that satisfy the user requirements in retrieving similar images.

Gestalt theory suggests four principles known as Gestalt principles of perception [120] that include proximity, size and shape similarity, continuity, and closure. Although the mechanisms behind these principles are still unclear, they provide a strong evidence that humans do not perceive multi-object images as purely the sum of the individual objects, which also has been noted by Lowe [71]. Biederman [21] concluded that different arrangement of the same objects of an image can easily lead to a completely different perceived image.

According to our literature review in visual cognition, the most relevant work to representing and comparing multi-object images is due to Markman and Gentner [72]. Their study provides more details about the comparison process in the human cognition system. They concluded that carrying out similarity comparisons involves structured representations and such comparisons work to align these structures. In addition, the similarity comparison possesses the following characteristics:

- *Consistency*: the similarity comparison is constrained by a one-to-one correspondence, that is, each element in one representation can be matched to at most one element in the other representation.

- *Systematicity*: the similarity comparison is driven by a search for correspondence that preserves connections between the representation elements. In other words, matching connected elements is preferred over matching isolated elements. In relation with our work, the connectivity refers to the object level, i.e, the topological relations between objects and holes, and does not refer to the pixel connectivity.

- *Subjectivity*: different people might generate different interpretations of the similarity between the same images, so as the same person does at different times. This property justifies the difficulty in evaluating content-based image retrieval systems.

As explained in the rest of this chapter, these findings constitute the main motivations for the proposed similarity function between two multi-object images. We believe that any computational model for measuring the image similarity should benefit from the advances in related fields such as psychology.

4.3 Curvature Trees

The curvature tree (CT) is a structured representation of a multi-object binary image that describes both shape and topology of objects and holes comprising the image[2]. Formally, the CT is a rooted, directed, unordered, and acyclic graph $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges. The CT has a single root node at level 0 representing the background, the external contours of the primary objects are stored at the first level nodes, and the contours of possible holes are at the second level nodes, and so on. Therefore, the tree hierarchy reflects the inclusion relationships between the objects and holes. To facilitate shape-based matching, triangle-area representation (TAR) of each closed boundary of an object or hole is stored at the corresponding node (Subsection 3.2.3).

Figures 4.1 and 4.2 show illustrative examples of the CTs of two simple multi-object images. By analyzing the curvature tree of Fig. 4.1 alone, one can deduce that the image contains one primary object with a primary hole, three secondary objects are contained in that hole, and two secondary holes are contained in each secondary object. The shape information of each object and hole is stored at the corresponding node as its TAR signature.

Many researchers have suggested different tree representations for gray-scale images. For example, Salembier and Garrido [107] proposed binary partition tree (BPT), where initial image partitions are merged following a sequential, pairwise fashion. Therefore, the BPT hierarchy reflects the merging sequence and the topological relationships between the regions cannot be easily identified. In another work, Monasse and Guichard [80, 81] proposed tree representation of gray-scale images that employs a region-growing algorithm to detect connected components and then organizes them based on their inclusion relationships. In fact, our CT exhibits the same structure as the inclusion tree in [80] for binary images. However, the CT differs in the stored information in the nodes and its application to measure the similarity between images.

[2]In the context of this thesis, an object (hole) is a connected set of foreground (background) pixels.

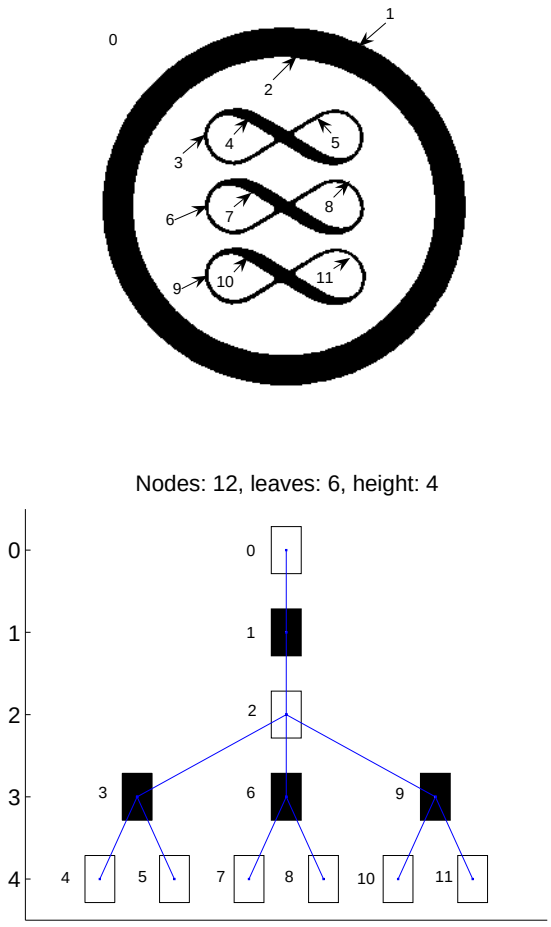


Figure 4.1: An example of a multi-object image and its CT.

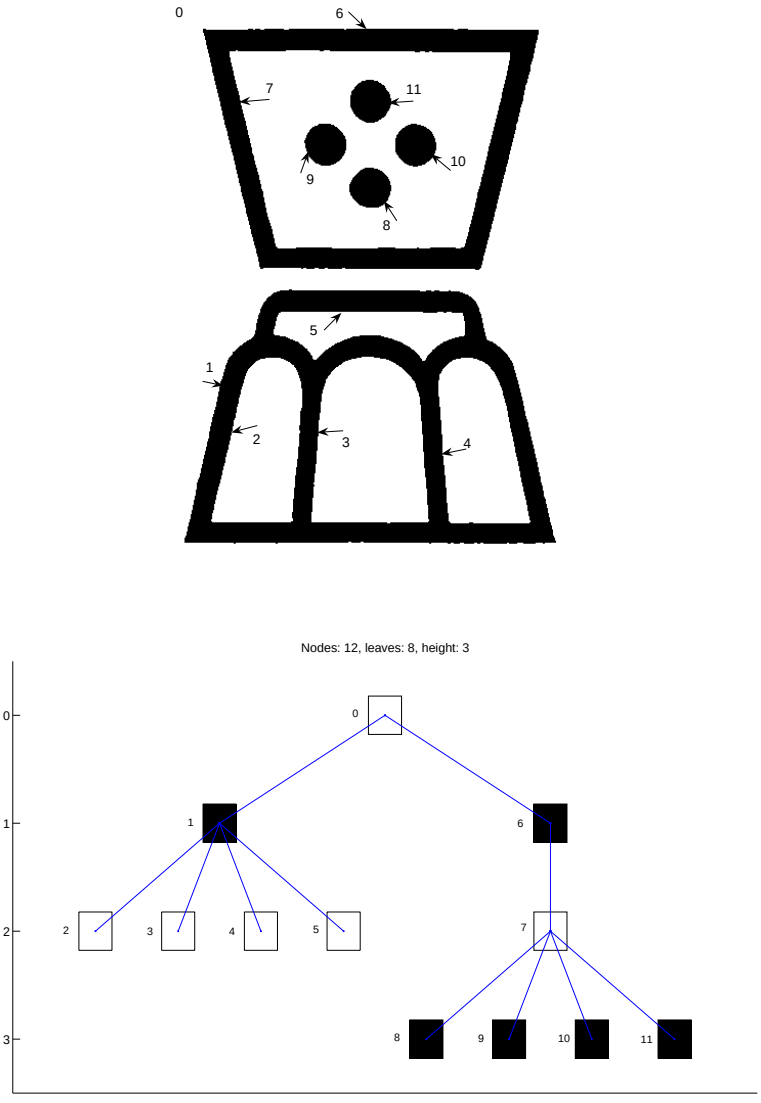


Figure 4.2: A multi-object image and its CT.

4.3.1 Curvature tree construction

Given a multi-object shape image, we want to represent it using its curvature tree. The process of curvature tree construction is simple and efficient. In addition, it describes a shape image uniquely and robustly. Each tree node stores TAR signatures of the corresponding object/hole. The edges between nodes can be weighted to reflect a spatial relationship between a parent node and its child node. The choice of such spatial relationship depends on the application. For instance, the medical images used in our experiments, which contain manually-segmented objects, allow overlapping to occur between the objects; therefore, the edges are weighted to reflect the percentage of inclusion. All curvature trees have a single root node at level 0 representing the background. Primary objects in the image are at the first level, possible holes in the primary objects are at the second level and objects inside these holes are at the third level, and so on. Thus, the inclusion relationships between objects and holes can be clearly followed. The algorithm that builds the curvature tree from a shape image is summarized as follows:

1. Insert the root node at level 0 (denotes the background).
2. Label all isolated objects. For each object:
   (a) Insert a node in the next tree level.
   (b) Store TAR signatures and the global features of the corresponding object/hole.
   (c) Compute the edge's weight (if applicable).
   (d) Search for internal holes. For each hole (if any):
      i. Insert a node in the next tree level.
      ii. Repeat 2 (b, c).
      iii. Repeat step 2 for the objects contained in the hole (if any).
3. Stop when all objects and holes have been visited.

In our implementation, the CT of a given multi-object shape image is stored in a hierarchical data structure called *cell array*. A desirable feature of the cell array is the provision to store data of different types and sizes; for instance, a cell array can store

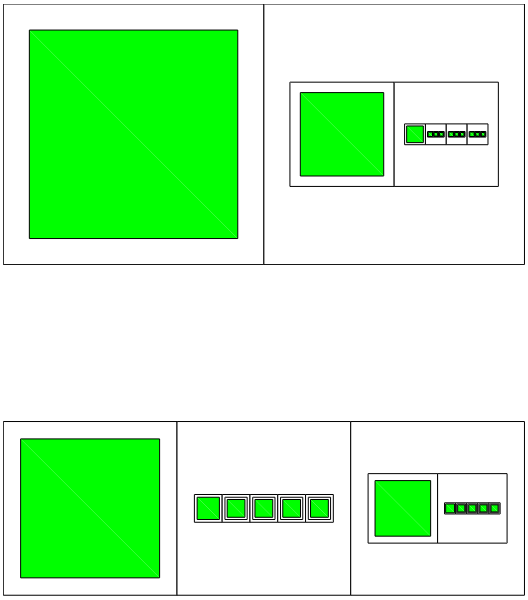


Figure 4.3: The cell arrays of the CTs of (top) Fig. 4.1 and (bottom) Fig. 4.2.

vectors or matrices of real numbers, strings of alphabets, and/or another cell array. The first element of the CT's cell array stores information pertaining to the root node; the other elements are cell arrays of the primary objects. Similarly, the first element in each cell array stores information of the corresponding node and the other elements are cell arrays of the children nodes. Thus, each subtree in the CT is stored in a cell array. Fig. 4.3 shows the plots of the cell arrays of the CTs in Figures 4.1 and 4.2. The data stored at each node, which are used during the matching process, include TAR signatures of the corresponding shape, a set of global features (area, aspect ratio, eccentricity, and solidity), the node index, the indices of the children nodes, and the level of the node (the length of the shortest path joining the node with the root node).

4.3.2 Curvature tree properties

Curvature tree possesses many properties that distinguish it from other trees. These properties are critical for developing an effective matching method. The properties include the following:

- The CT is a rooted, directed, weighted, unordered, and attributed tree.
- The number of nodes of the CT equals the total number of objects and holes in the image. This means that the order and the size of the tree grow linearly with the complexity of the image, which is desirable for both representing and matching multi-object images.
- The hierarchy of the CT reflects the inclusion relationships between objects and holes.
- If applicable, the weight of each edge represents a spatial relationship between the parent and the child nodes (for example, the percentage of the included area or the spatial distance).
- The incoming degree of all nodes equals one except for the root node, which has an incoming degree of zero. This property is highly desirable for the matching since it allows including the weights of the edges in the node similarity function.
- The CT is invariant to the inversion (i.e., taking the negative) of the image. Inverting the background from black (white) to white (black) causes insertion (deletion) of a child node of the root node whose boundary follows the frame of the image. The image inversion causes a defect to statistical image descriptors such as moments since they are computed from the foreground pixels.
- The CT is a scale-space representation of the image where the scale is directly the shape area. Large scale objects appear near the root node whereas small scale objects appear near the leaf nodes.

From these properties, it can be deduced that the CT overcomes some limitations of the region adjacency graph (RAG) [99], which encodes the neighborhood relationships between

objects and does not follow a hierarchical structure. In addition, the tree matching is a much easier problem than the graph matching. There are polynomial-time algorithms for attributed tree matching where such algorithm does not exist for attributed graph matching [30].

4.4 Attributed Tree Matching

The main aim is to measure the similarity between two multi-object images using their CTs. As discussed in Section 4.2, interesting recent findings in psychology strongly suggest that such similarity should follow a one-to-one correspondence between structured representations of the two images [72]. Therefore, we propose a similarity function based on the maximum similarity subtree isomorphism (MSSI) between the two CTs. This matching scheme is desirable since it allows matching whole or part of one image with whole or part of the other image. In the following, an approximate and an exact algorithms are introduced to solve the MSSI problem; then, the similarity function is proposed which is based on the MSSI solution.

4.4.1 Notations and definitions

We first introduce some graph theoretic notations and definitions. More details can be found in the graph theory literature such as [50, 15]. An attributed graph is a triple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \delta)$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, and δ is a function that assigns attributes $\delta(u)$ to each node $u \in \mathcal{V}$. The order of $\mathcal{G}$ is the number of nodes and its size is the number of edges. Two nodes $u, v \in \mathcal{V}$ are said to be adjacent, denoted by $u \sim v$, if they are connected by an edge. A path is a sequence of distinct nodes $u_0 u_1 ... u_n$ such that $u_{i-1} \sim u_i$, for $i = 1...n$. If $u_0 = u_n$, the path is called a cycle. A graph is said to be connected if there is a path between any two nodes. The path distance between u and v, $d(u, v)$, is the length of the shortest path joining them. A subgraph $\mathcal{F}$ of $\mathcal{G}$ is the graph having a set of nodes $\mathcal{H} \subset \mathcal{V}$ and any two nodes in $\mathcal{F}$ are adjacent if and only if they are adjacent in $\mathcal{G}$.

A tree is a connected graph with no cycles. An attributed tree has attributes stored at its nodes. A weighted tree is the one with weights assigned to its edges. A rooted tree has a

distinguished node called the root. The level of a node u in a rooted tree, denoted by $l(u)$, is the length of the path connecting the root node to u. Note that the rooted tree implies a multi-level organization of the tree nodes. Moreover, if $u \sim v$ and $l(u) - l(v) = -1$, then u is the parent of v and, conversely, v is a child of u. Note that any node in a rooted tree has exactly one parent node, except the root, whereas the node can have any number of children; a node that does not have any child node is called a leaf. An ordered tree respects the order of the children nodes at any tree level whereas, in an unordered tree, the children nodes do not follow any order.

Let $\mathcal{T}_1 = (\mathcal{V}_1, \mathcal{E}_1, \delta_1)$ and $\mathcal{T}_2 = (\mathcal{V}_2, \mathcal{E}_2, \delta_2)$ be two rooted and attributed trees, $\mathcal{H}_1 \subseteq \mathcal{V}_1$ and $\mathcal{H}_2 \subseteq \mathcal{V}_2$. A subtree isomorphism is a mapping $\phi : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ that preserves both adjacency and hierarchical relations between the nodes of each subtree. In matching attributed trees, the isomorphism should also pair similar nodes. Let $\omega(\delta_1(u), \delta_2(v))$ be a similarity function between nodes $u \in \mathcal{T}_1$ and $v \in \mathcal{T}_2$ based on their attributes (their TAR signatures in our case). Then, the overall similarity Ω between the subtrees $\mathcal{H}_1$ and $\mathcal{H}_2$ induced by ϕ can be defined as follows:

$$\Omega(\phi) = \sum\nolimits_{u \in \mathcal{H}_1} \omega\left(\delta_1(u), \delta_2(\phi(u))\right) \tag{4.1}$$

The isomorphism ϕ is maximal similarity subtree isomorphism if there is no other subtree isomorphism $\phi' : \mathcal{H}_1{}' \rightarrow \mathcal{H}_2{}'$ such that $\mathcal{H}_1 \subset \mathcal{H}_1{}'$ and $\Omega(\phi) < \Omega(\phi')$. It is called maximum similarity subtree isomorphism if $\Omega(\phi)$ is largest among all possible isomorphisms.

4.4.2 Approximate tree matching

Approximate solutions to the MSSI problem demand less computations than exact solutions which is a desirable property in most pattern recognition and computer vision applications. In this thesis, a continuous optimization method proposed recently by Pelillo et al. [88, 87, 86] is adapted with two modifications. As explained in the rest of this section, the motivations for these modifications are to improve the accuracy by avoiding spurious solutions of (4.5) when the global minimum is not unique and to limit the number of computations for solving the MSSI problem [8]. The main steps of the algorithm are summarized as follow:

1. Derive an auxiliary graph, called weighted tree association graph (WTAG), from the two CTs to be matched.

2. Compute the maximal clique of the derived WTAG using continuous optimization techniques.

3. The maximal clique corresponds to maximal subtree isomorphism and the clique's weight is the solution of the MSSI problem.

In the following, an explanation of each of these steps is provided with more details.

Association graphs and the MSSI problem

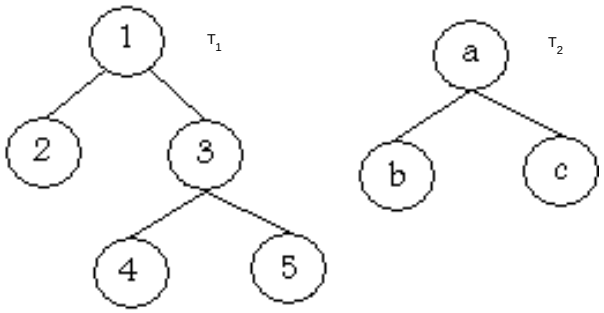
At first, an auxiliary graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ is derived from the two trees $\mathcal{T}_1$ and $\mathcal{T}_2$ that is called weighted tree association graph (WTAG), where $\mathcal{V} = \mathcal{V}_1 \times \mathcal{V}_2$, ω is a similarity function that assigns positive weights to the nodes in $\mathcal{V}$ (the DSW similarity function of (3.11) in our application), and $\mathcal{E}$ is defined as follows: for any two nodes (u_1, u_2) and (v_1, v_2) in $\mathcal{V}$, where $u_1, v_1 \in \mathcal{V}_1$ and $u_2, v_2 \in \mathcal{V}_2$, we have $(u_1, u_2) \sim (v_1, v_2) \Leftrightarrow d(u_1, v_1) = d(u_2, v_2)$ and $l(u_1) - l(v_1) = l(u_2) - l(v_2)$.

Fig. 4.4 illustrates the WTAG construction procedure. Nodes 1 and 2 in the first tree have the same structural relation as nodes a and b in the other tree; therefore, there is an edge between nodes $(1, a)$ and $(2, b)$ in the corresponding WTAG. Intuitively, the hierarchical relations between the tree nodes are encoded in the flat structure of the WTAG; enabling the use of powerful continuous optimization techniques to solve the MSSI problem.

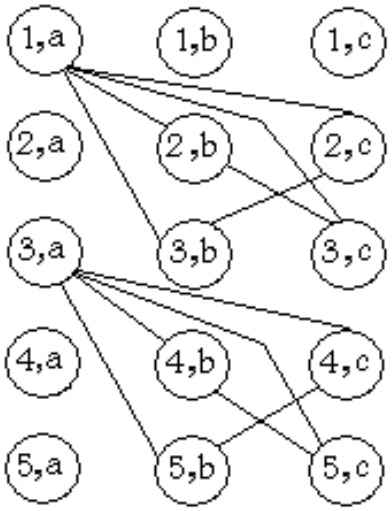
A clique in $\mathcal{G}$ is a fully connected subgraph. A maximal weight clique in $\mathcal{G}$ is not contained in any other larger weight clique and the maximum weight clique has largest total weight. The following theorem is essential to solve the MSSI problem (see [22] for proof).

Theorem 5.1

Any maximal (maximum) similarity subtree isomorphism between two attributed trees induces a maximal (maximum) weight clique in the corresponding WTAG, and vice versa.



(a)



(b)

Figure 4.4: Two trees (a) and their WTAG (b).

Based on this theorem, solving the MSSI between two attributed trees and the maximum clique detection of their WTAG are equivalent. In the example of Fig. 4.4, there are four maximum subtree isomorphisms between the two trees in panel (a); namely, $\phi_1 : \{1 \rightarrow a, 2 \rightarrow b, 3 \rightarrow c\}$, $\phi_2 : \{1 \rightarrow a, 2 \rightarrow c, 3 \rightarrow b\}$, $\phi_3 : \{3 \rightarrow a, 4 \rightarrow b, 5 \rightarrow c\}$, and $\phi_4 : \{3 \rightarrow a, 5 \rightarrow b, 4 \rightarrow c\}$ (note that the trees are unordered and the nodes are assumed to have the same attributes). Each of these isomorphisms has an order of three. On the other hand, the WTAG of panel (b) has four maximum cliques having a cardinality of three; namely, $\{(1, a), (2, b), (3, c)\}$, $\{(1, a), (2, c), (3, b)\}$, $\{(3, a), (4, b), (5, c)\}$, and $\{(3, a), (5, b), (4, c)\}$. These four cliques are equivalent to the four isomorphisms between the two trees. In the following, an approximate solution of the maximum clique detection problem is presented based on a continuous optimization formulation.

Approximating the maximum clique using continuous optimization

To solve the maximum weight clique problem using continuous optimization, let $\hat{z}$ be a characteristic vector of any subset of nodes $\mathcal{C} \subseteq \mathcal{V}$ defined as :

$$\hat{z}_i = \begin{cases} \omega(u_i)/\Omega(\mathcal{C}) & \text{if} \quad u_i \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

where $i \in \langle 1, |\mathcal{V}| \rangle$ ($|\mathcal{V}|$ is the number of nodes in $\mathcal{V}$) and $\Omega(\mathcal{C}) = \sum_{u_j \in \mathcal{C}} \omega(u_j)$ is the total weight of $\mathcal{C}$. Also, consider the following quadratic function:

$$f(z) = z^T \mathcal{W} z \tag{4.3}$$

where z is $|\mathcal{V}| \times 1$ vector that satisfies $e^T z = 1$ and $z_i \geq 0$, e is the unit vector with $i \in \langle 1, |\mathcal{V}| \rangle$, and $\mathcal{W} = \lambda e e^T - \mathcal{B}$ is a symmetric real matrix where $\mathcal{B} = (b_{ij})$ is a $|\mathcal{V}| \times |\mathcal{V}|$ matrix defined as:

$$b_{ij} = \begin{cases} \frac{1}{2\omega(u_i)} & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \,\text{and}\, u_i \sim u_j \\ \frac{1}{2\omega(u_i)} + \frac{1}{2\omega(u_j)} & \text{otherwise} \end{cases} \tag{4.4}$$

and $\lambda = max(b_{ij})$.

Bomze [22, 23] proved that global (local) maxima of $z^T \mathcal{B} z$ (and hence minima of f) correspond to maximum (maximal) cliques of the WTAG. The following dynamical system, which is called replicator equation, is used to find the minimum value of (4.3):

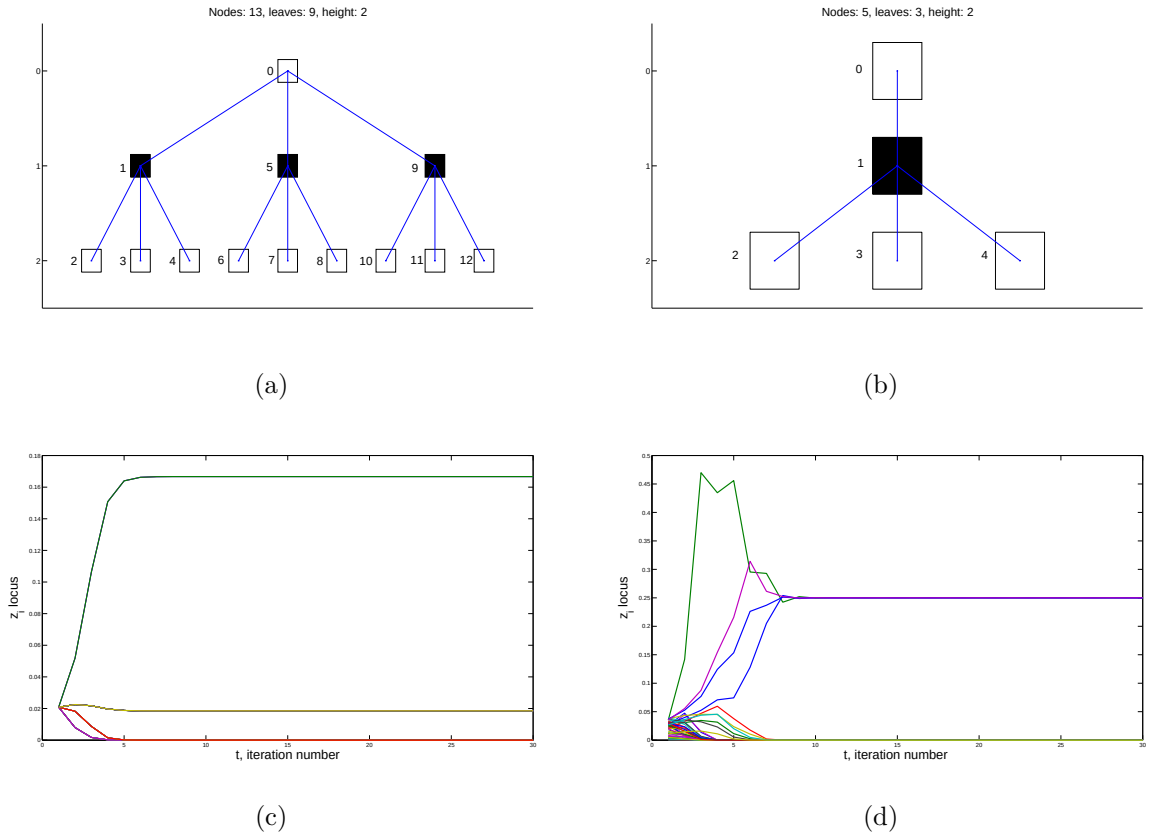
$$z_i(t+1) = z_i(t) \frac{(\mathcal{W} z(t))_i}{z(t)^T \mathcal{W} z(t)} \tag{4.5}$$

where t is the iteration number and z_i is the i-th element of the vector z. An interesting property of (4.5) is the asymptotic convergence to a stable local minimizer of f that corresponds to the characteristic vector of a maximal weight clique of the WTAG. As reported in [88], the basins of attraction of the global minimizer of (4.3) are quite large; consequently, the solution obtained by (4.5) is most likely global and corresponds to the maximum weight clique which approximately solves our original MSSI problem.

In [88], the initialization of (4.5) is made with an equal-weight vector, i. e., $z_i(0) = 1/|\mathcal{V}|$ for all i. However, a major problem with this initialization occurs particularly when the global solution is not unique. In this case, a spurious solution is obtained that does not correspond to a characteristic vector and thus the solution cannot be mapped to the discrete domain. In our implementation, (4.5) is initialized with uniformly distributed random numbers between zero and one and then normalized to ensure their sum equals one. Our experiments show that this initialization effectively avoids the spurious solutions and achieves faster convergence. Fig. 4.5 shows the locus of each element in the vector z during the computation of (4.5) when matching the two CTs in the figure. The nodes are assumed to have the same attributes; thus, the nodes of the WTAG have similar weights. In this example, there are 18 MSSI solutions since node 1 in the CT of panel (b) can be matched to any of the nodes 1, 5, or 9 in the CT of panel (a) and, in each case, there are 6 possible combinations for matching the three children of each node (because the children are not ordered). The uniform initialization in [88] returns a spurious solution that cannot be mapped back to the discrete domain because z is not in the characteristic form whereas the random initialization in this thesis converges to a global solution. Note that there is overlapping between more than one loci of z_i. Fig. 4.6 shows the convergence values of z_i for the uniform and random initializations and the loci of f in each case. Clearly, the uniform initialization returns a local minimum ($f = 0.75$) while the random initialization

returns the global minimum ($f = 0.125$).

In our implementation, the number of iterations in (4.5) is limited to be equal to $|\mathcal{V}|$ (the number of nodes in the WTAG) which achieves robust performance and gives better assessment of the computational complexity of our algorithm. As shown in the example of Fig.4.5, fast convergence of (4.5) is obtained in less than 10 iterations (here, $|\mathcal{V}| = 48$).



(a) (b)

(c) (d)

Figure 4.5: Illustration of the dynamics of (4.5) when matching the two trees in (a) and (b); with uniform initialization as in [88] (c) and with random initialization (d).

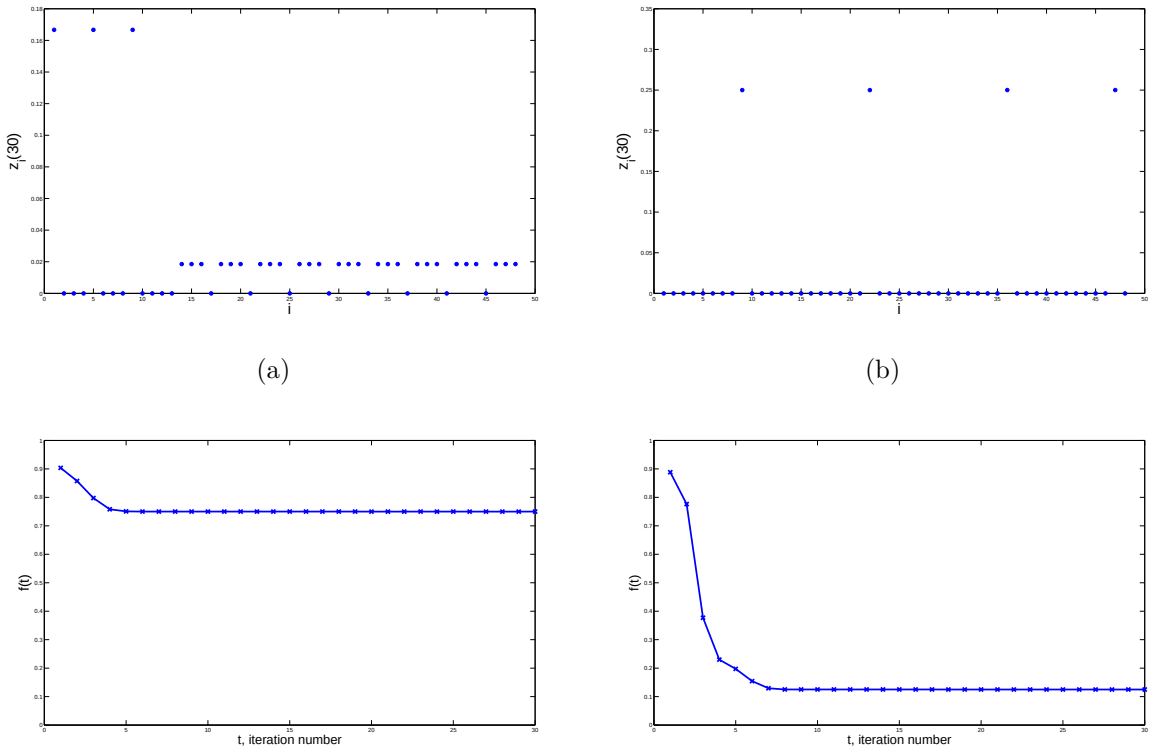


Figure 4.6: The final values of z_i and the loci of f for matching the two trees in Fig. 4.5 with uniform (a) and (c) and random (b) and (d) initializations of (4.5).

The MSSI solution

Once the replicator equation of (4.5) converges to a stable solution, an approximation of the MSSI between the two CTs is directly detected. Each non-zero element in the characteristic vector z corresponds to a node in a maximal clique of the WTAG which in turn induces a mapping between two nodes in the CTs. Therefore, the mapping from the continuous domain to the discrete domain is straightforward. In the example of Fig. 4.6(b), the four non-zero elements of the characteristic vector corresponds to matching nodes 9, 10, 12, and 11 of the CT of Fig. 4.5(a) with nodes 1, 2, 3, and 4 of the CT of Fig. 4.5(b), respectively.

4.4.3 Exact tree matching

In this method, an exact solution to the MSSI problem is computed. Unlike the previous method, where the obtained isomorphism is maximal, the obtained isomorphism here is guaranteed to be maximum. The exact algorithm works directly on two CTs and does not require any auxiliary structure. More specifically, the algorithm recursively searches for all possible isomorphisms between two trees and selects the one yielding the maximum similarity. Such an exhaustive search is computationally expensive; for two trees having n and m nodes, respectively, there are nm possible isomorphisms to be searched between subtrees rooted at each two nodes in the two trees. Fortunately, due to the nonnegative property of the employed similarity function between the attributed nodes, i.e., the DSW similarity function given by (3.11), without loss of generality, the search space reduces to $n + m$ possible isomorphisms. The reason for this reduction is as follows. Suppose an isomorphism $\tilde{\phi}(\tilde{u}, \tilde{v})$ is induced between two subtrees rooted at two arbitrary non-root nodes $\tilde{u} \in \mathcal{T}_1$ and $\tilde{v} \in \mathcal{T}_2$, with parent nodes u and v, respectively. Since the node similarity function is nonnegative, adding u and v to the isomorphism $\tilde{\phi}$ increases its similarity weight or, at least, does not change it, i.e., $\Omega\left(\phi\left(u, v\right)\right) \geq \Omega(\tilde{\phi}\left(\tilde{u}, \tilde{v}\right))$. To achieve maximum similarity, an isomorphism has to include the parent nodes of the subtrees' root nodes until the root of either tree is included in the isomorphism. Therefore, the MSSI must include at least one of the root nodes of the two trees. In this case, the number of possible isomorphisms, which are candidates for MSSI, reduces to $n + m$.

Algorithm 4.1 is a pseudo-code of our exact tree matching algorithm. It accepts two CTs as inputs and returns their MSSI along with its weight. The algorithm examines all subtree isomorphisms, between the nodes of the first CT and the root of the other CT and vice versa, and returns the isomorphism yielding maximum similarity. The main function, *ExactMatch*, uses the function *TreeMatch* of Algorithm 4.2 to find the MSSI between two subtrees rooted at $u \in \mathcal{T}_1$ and $v \in \mathcal{T}_2$. *TreeMatch* initializes the isomorphism with $u \rightarrow v$ and its weight with $\omega\left(u, v\right)$. If both u and v are non-leaf nodes (both have at least a child node), then an optimal assignment procedure assigns the child node(s) of u to the child node(s) of v such that the sum of the pairwise similarities is maximum. This assignment problem is known as bipartite matching; for two sets of n_1 and n_2 elements and $n_1 \geq n_2$, there are $n_1!/(n_1 - n_2)!$ possible mappings between the elements of the two

sets ($n_1!$ is the factorial of n_1). For this purpose, the Hungarian method is employed which has polynomial-time complexity of $O(n_1 n_2^2)$ [83, 85]. *TreeMatch* recursively matches the nodes of the subtrees rooted at u and v at different tree levels. This algorithm is similar to the algorithm in [122, 123] except that the latter returns only the weight of the maximum isomorphism and does not return the isomorphism itself.

Algorithm 4.1 Exact tree matching (main algorithm): $[\phi, \Omega] = ExactMatch(\mathcal{T}_1, \mathcal{T}_2)$

Notation:

$\mathcal{T}_1, \mathcal{T}_2$ are two CTs to be matched.
ϕ is a MSSI between the two trees.
Ω is the total weight of ϕ.

1: $\Omega \leftarrow 0$
2: **for** each node $u_i \in \mathcal{T}_1$ **do**
3: $[map, sim] \leftarrow TreeMatch(u_i, root(\mathcal{T}_2))$
4: **if** $sim > \Omega$ **then**
5: $\Omega \leftarrow sim$
6: $\phi \leftarrow map$
7: **end if**
8: **end for**
9: **for** each node $v_j \in \mathcal{T}_2$ **do**
10: $[map, sim] \leftarrow TreeMatch(root(\mathcal{T}_1), v_j)$
11: **if** $sim > \Omega$ **then**
12: $\Omega \leftarrow sim$
13: $\phi \leftarrow map$
14: **end if**
15: **end for**
16: **return** ϕ, Ω

4.4.4 The similarity function

The decision on the similarity function between two multi-object images depends on the application under consideration according to the user requirements. Our method can

Algorithm 4.2 Exact subtree matching: $[map, sim] = TreeMatch(u, v)$

Notation:

n_u is the number of children of u.

n_v is the number of children of v.

$\delta(u)$ is TAR signatures of node u (as described in Subsection 3.2.3).

ω is the similarity function DSW_{sim} given by (3.11).

W is the similarity matrix between the child nodes of u and v.

$hungarian(A)$ is an optimal assignment function that assigns the rows to the columns of the cost matrix A.

1: $sim \leftarrow \omega(u, v)$
2: $map \leftarrow \{(u, v)\}$
3: **if** $n_u \neq 0$ and $n_v \neq 0$ **then**
4: **for** each child node c_i of u **do**
5: **for** each child node c_j of v **do**
6: $[M(i, j), W(i, j)] = TreeMatch(c_i, c_j)$
7: **end for**
8: **end for**
9: $[assign, cost] = hungarian(max(W) * ones(n_u, n_v) - W)$
10: $sim \leftarrow sim + max(W) * min(n_u, n_v) - cost$
11: $map \leftarrow map \cup assign$
12: **end if**
13: **return** map, sim

handle various types of queries. For example, in some applications the user is interested in the objects of a database image similar to the objects in the query image; thus, the remainder portion of the database image is insignificant. In this case, the similarity function can be considered as the weight of the MSSI, Ω. Another example is an application where the user presents a query of single-object shape which means a linear search of the CT nodes of a database image. In the most general case, we propose a similarity function that accounts for the matched objects of two multi-object images as well as the unmatched objects. Once the weight of the MSSI, Ω, is obtained, the final similarity function between two CTs is defined as:

$$D_{CT} = \frac{\Omega}{2}\left(\frac{1}{|\mathcal{T}_1|} + \frac{1}{|\mathcal{T}_2|}\right) \tag{4.6}$$

Note that D_{CT} equals Ω normalized by the number of nodes in each CT. This normalization accounts for the unmatched parts in the two CTs.

4.5 Computational Complexity

4.5.1 CT representation

The CT of a multi-object image of n shapes has n nodes; therefore, the order of the CT grows linearly with the number of objects and holes in the image. Each node stores TAR of the corresponding shape. Since TAR computation has complexity of $O(N^2)$, where N is number of the boundary points (see Section 3.4 for details), the overall complexity of the CT construction is $O(nN^2)$.

4.5.2 Approximate CT matching

Matching a CT of n nodes with another CT of m nodes requires the construction of their WTAG of nm nodes and the weight of each node is computed using the DSW matching of complexity $O(N^2)$ (as described in Section 3.4). The search for the MSSI solution using (4.5) has complexity $O(nm)$, since the number of iterations equals the cardinality of the WTAG. Therefore, the complexity of the WTAG construction is $O(nmN^2)$ which is also

the complexity of the approximate CT matching. Note that the complexity of the matching is invariant to the topological parameters of the CT such as the number of tree levels.

4.5.3 Exact CT matching

The computational complexity of our exact matching algorithm heavily depends on the number of CT levels and the branching factor (i.e., the number of children of a CT node); therefore, characterizing the algorithm's complexity is difficult. Let b be the maximum branching factor. For matching two CTs of n and m nodes, the main algorithm *ExactMatch* (of Algorithm 4.1) actually computes at most nm bipartite matchings: one for each possible pair of nodes from the two trees. The bipartite matching under nodes $u_i \in \mathcal{T}_1$ with p_i children and $v_j \in \mathcal{T}_2$ with q_j children, using the Hungarian procedure, has complexity $O\left(max(p_i, q_j)p_iq_j\right) \leq O\left(bp_iq_j\right)$. Summing over all possible matches, we have $\sum_{i=1}^{n}\sum_{j=1}^{m} bp_iq_j = bnm$. As in the approximate matching algorithm, the node similarity is measured using the DSW algorithm of complexity $O(N^2)$, where N is the number of the boundary points. Therefore, the complexity of our exact CT matching algorithm is $O\left(bnmN^2\right)$.

4.6 Experimental Results

We demonstrate the performance of our method using two application domains of image retrieval. The first application is retrieval of medical images from a database of 13500 real and synthesized medical images. In the second application, retrieval of logos is conducted on a database of 1580 logo images. In both applications, the shape of individual objects as well as their topological structure are essential to determine the image identity. In the following, detailed explanations of the experimental setups, evaluation methods, database images and comparisons with relevant methods in the literature are presented.

4.6.1 Retrieval of medical images

Here, a database of 13500 real and synthesized medical images is used [93]. The database consists of 124 real MRI (Magnetic Resonance Imaging) images manually segmented by

tracing the boundaries of the image objects. Fig. 4.7 shows an example MRI image and its segmented image. The rest of the database images were produced from the original images by allowing the objects in the original images to rotate, scale, and translate by certain amounts controlled by a random number generator. In addition to these transformations, objects having random sizes, shapes and positions are added to the derived images. All images contain between 4 to 8 objects.

The evaluation is based on human relevance judgements as reported in [91]. Two images are considered similar if they contain similar objects in similar spatial relationships. Besides, the similarity is judged based on the query image, that is, a database image may contain extra objects without affecting the similarity but not the query. The following performance measures are used to evaluate the retrieval performance:

Precision which is the ratio of the number of relevant retrieved images to the total number of retrieved images.

Recall which is the ratio of the number of relevant retrieved images to the number of relevant images in the database. To compute recall, all database images relevant to the query have to be known. However, comparing every query with all 13500 database images by a human referee is practically impossible. To overcome this problem, the sampling method, known as pooling method in text retrieval community [129], is employed which results in labeling part of the database images as explained in the rest of this section.

Ranking quality (R_q) which measures the goodness of the retrieval ranking, that is, the method retrieves relevant images before the irrelevant images. The higher the value of R_q the better the ranking quality of a method and vice versa. Note that R_q is independent from the accuracy. R_q is computed as follows:

1. Each retrieved image is assigned a rank number according to its order and judged as relevant or irrelevant with respect to the query image.
2. Pairs of the retrieved images are taken such that each pair contains a relevant and irrelevant images and the relevant one is first and the irrelevant is second.

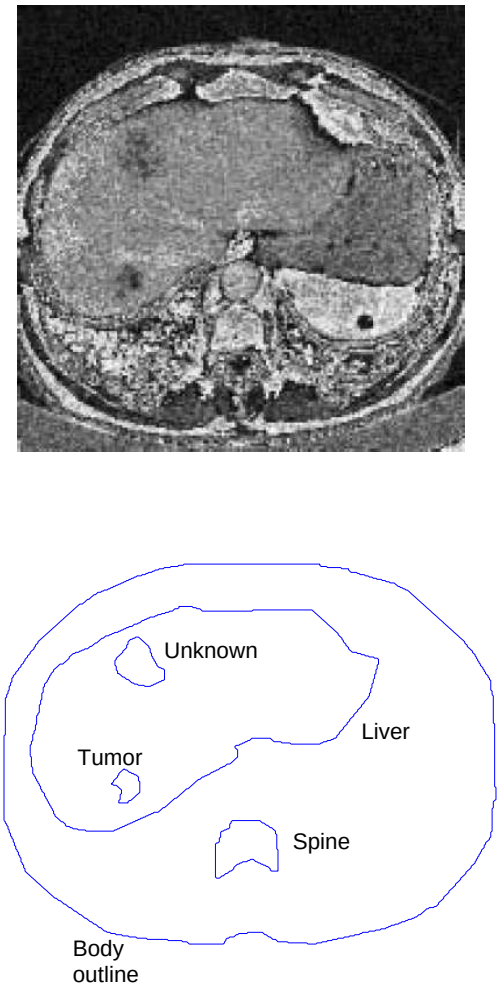


Figure 4.7: A medical image and its manually segmented image.

3. Let S^+ be the number of pairs with the relevant image's ranking better than the ranking of the irrelevant image, S^- is the number of pairs with the irrelevant image's ranking better, and $S = S^+ + S^-$. R_q is computed as:

$$R_q = \begin{cases} \frac{1}{2}\left(1 + \frac{S^+ - S^-}{S}\right) & \text{if } S > 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.7}$$

Since the objects of the images in this database are manually segmented, overlapping between the objects can occur which makes the inclusion relationships not clearly identified. Therefore, the CT construction and matching are modified accordingly. In this application, the CT edges are weighted to reflect the amount of inclusion of a child node in its parent node. To resolve the inclusion ambiguity caused by the overlapping, let object u with area A_u and object v with are A_v have an overlapping with area A_{uv} such that $A_u > A_v$. Then, node v is inserted in the CT as a child to the node u and the weight of the edge $\vec{uv}$ equals A_{uv}/A_v. For the approximate and the exact CT matching algorithms, since the incoming degree to any node in the CT equals one (except the root), only the node similarity function (ω) is modified to include the edge weights as follows. Let $u_1, v_1 \in \mathcal{T}_1$, $u_2, v_2 \in \mathcal{T}_2$, and ζ_1 and ζ_2 the weights of the edges $u_1\vec{v}_1$ and $u_2\vec{v}_2$, respectively. Then, the similarity between nodes v_1 and v_2 is redefined as:

$$\omega(v_1, v_2) = 1 - \left(\alpha_1 \left|\zeta_1 - \zeta_2\right| + \alpha_2 |\hat{A}_{v_1} - \hat{A}_{v_2}| + \alpha_3 DSW\left(\delta(v_1), \delta(v_2)\right)\right) \tag{4.8}$$

where $\hat{A}_{v_1}$ is the normalized area of v_1 obtained by dividing A_{v_1} by the area of the largest object in $\mathcal{T}_1$ in order to achieve scale invariance (same for $\hat{A}_{v_2}$), DSW is the dissimilarity function given by (3.10), and α_i are positive weights such that $\sum_{i=1}^{3} \alpha_i = 1$ to maintain $0 \leq \omega \leq 1$. In our experiments, $\alpha_1 = 0.15$, $\alpha_2 = 0.35$, and $\alpha_3 = 0.5$ (these values are not optimized).

In the retrieval test, precision-recall pairs are computed as follows. Twenty query images (shown in Fig. 4.8) are presented and, for each query, the system retrieves the best 50 matches out of the 13500 database images. In [91], the results of 5 methods and 19 of their variants are reported which demanded 24000 comparisons by human referees (each method required $20 \times 50 = 1000$ comparisons). Based on our estimate, these comparisons

resulted in labeling 42 % of the database images as relevant/irrelevant to any of the 20 queries. For consistency, we used the labeled database images as templates to classify the unlabeled retrievals for each query. Then, the ground truth is updated and the precision-recall curves are computed as shown in Fig. 4.9. Each point in the figure is the average over the 20 query images; thus, each curve consists of 50 points. The figure shows the results of the proposed method with the approximate and the exact matching algorithms (as described in Section 4.4) and the ARG method [91] with two matching approaches: the graph edit distance and the Hungarian method (see Section 2.4 for details). Clearly, the proposed method outperforms the ARG method which performed better than many other methods [91].

Another test is performed based on the ground truth provided in [91] alone. For each query, the system retrieves the best 50 matches and the number of relevant images, n_r, and the number of irrelevant images, n_i, are counted. Note that both the relevant and irrelevant images are provided by the ground truth given in [91]; thus, our judgement in the evaluation is not considered in this test. Then, the accuracy for each query is computed as $n_r/(n_r + n_i)$. Table 4.1 shows the accuracy of our method is significantly higher than that of the ARG method.

Table 4.1: The accuracy of different methods on the medical images database based on the ground truth provided in [91].

Method	ARG+Hung.	ARG+ED	Proposed (approx.)	Proposed (exact)
Accuracy	24%	29.60%	39.60%	40.91%

Regarding the quality of retrieval quality (R_q), Table 4.2 shows that the proposed method also outperforms the ARG method. Figs. 4.10 and 4.11 demonstrate two example queries and the first 6 matches retrieved by the exact CT matching algorithm (Algorithm 4.1). For clarity, the isomorphism returned by the algorithm is shown by objects with same labels in the query and each retrieved image. The identification number in the database is shown above each image.

Figure 4.8: The 20 query medical images used in the our experiment.

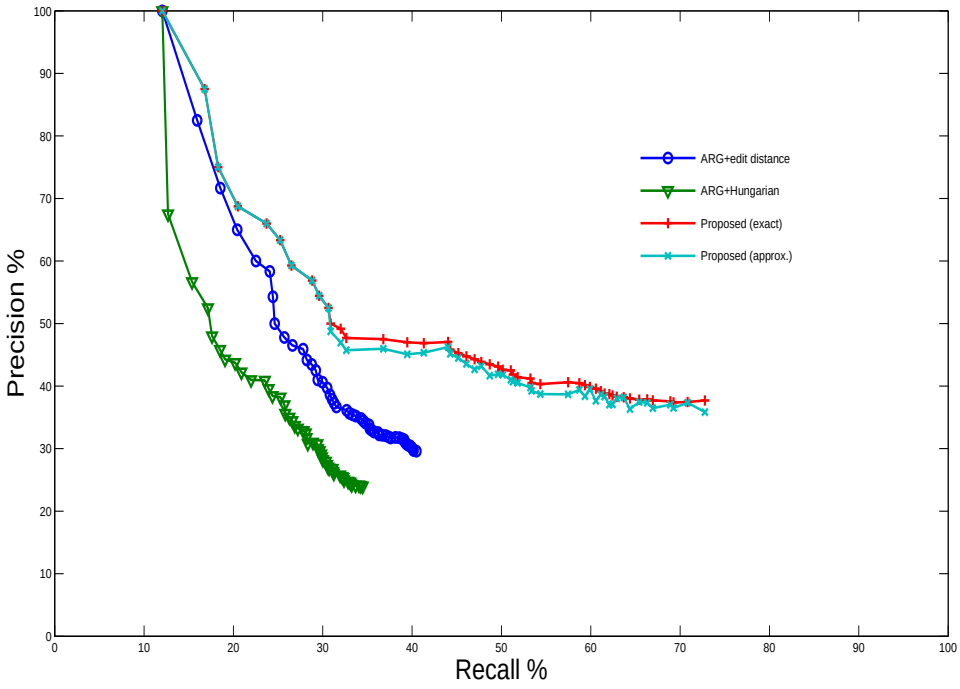


Figure 4.9: Precision-recall curves of the proposed method with approximate and exact matching, and the ARG method with edit distance and Hungarian method [91].

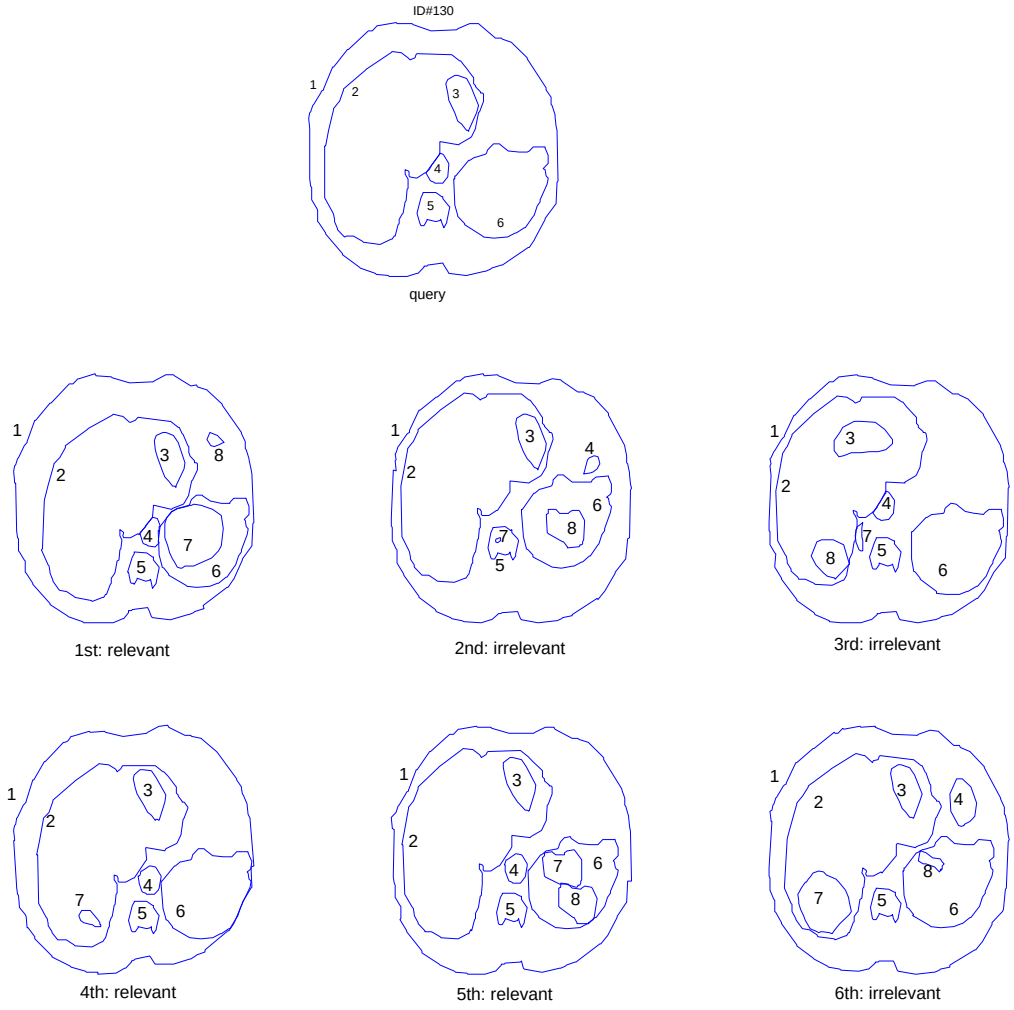


Figure 4.10: A query image and the first 6 matches using the exact CT matching algorithm. The object labels denote the isomorphism returned by the algorithm.

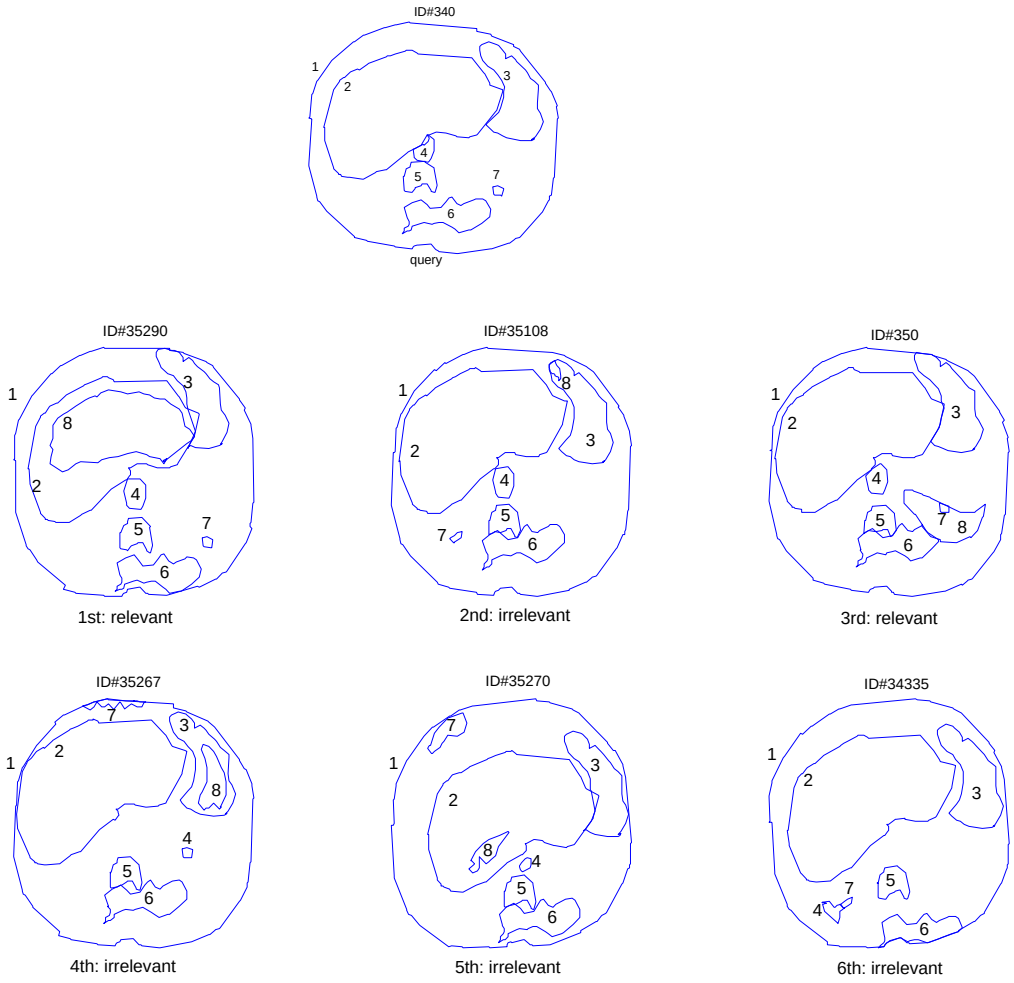


Figure 4.11: Another query image and the first 6 matches using the exact CT matching algorithm. The object labels denote the isomorphism returned by the algorithm.

Table 4.2: The ranking quality of different methods on the medical images database.

Method	ARG+Hung.	ARG+ED	Proposed (approx.)	Proposed (exact)
R_q	0.0890	0.2390	0.5316	0.5431

4.6.2 Logo retrieval

A trademark is either a word, phrase, symbol or design, or combination of words, phrases, symbols or designs, which identifies and distinguishes the source of goods or services of one party from those of others [125]. A logo (or trademark) image is a good example of multi-object image where the shape of the individual objects as well as their topological structure are essential to determine the image identity. There are millions of registered trademarks worldwide; for example, there are more than one million in the U.S. [125] and more than 300,000 [27] in Canada. Logo retrieval is an extremely challenging task. At present, the most widely used method for organizing trademark image databases is the Vienna classification originated by the World Intellectual Property Organization [133]. This method assigns manual codes to trademarks such that similar images are given the same codes. However, manual classification is both time-consuming and inaccurate. With the growing number of registered trademarks, automated logo retrieval can significantly simplify the manual retrieval task.

A database of 1580 logo images is used to evaluate our proposed method. Part of these images are taken from the MPEG-7 CE-2 database of logo images [60] (images containing texture or purely text are excluded here). The rest of the database images are obtained from the database of 1100 logo images in [55]. Since the original images are provided in gray-scale format, preprocessing is required to transform them into binary format. At first, a simple thresholding operation transforms the images into the binary format which introduces binary impulse noise. Then, the binary noise removal algorithm in [111] is used to remove the noise.

Unlabeled logo images

A retrieval test is conducted on the database of 1580 logo images. Since the original images are not labeled and the the similarity between logos is subjective, it is recommended to include the human judgement in the evaluation. For this purpose, 24 query logos were used. For each query, the system measures its similarity with all database logos according to the similarity function given by (4.6) and provides a ranked list of logos. Then, we asked five persons to determine whether the top-five retrieved logos are similar to the query or not and decided the similarity based on the majority vote (in [92], the evaluation is based on the judgments by four persons). Therefore, each person performed 120 comparisons.

We compared the performance of our method with two approaches. The first method is based on Zernike moment magnitudes (ZMMs) as global features and the sum of absolute difference (SAD) distance to measure the similarity [62, 61]. This method has been selected as the MPEG-7 CE-2 region-based shape descriptor [56]. The second approach is an object-based method that uses similarity function as in [117] (see Section 2.4), but based on our DSW as the similarity measure between the objects. Fig. 4.12 plots the average accuracy over the 24 queries versus the retrieved logo ranking for the three methods. An example that shows a query and the top three retrieved logos by each method is shown in Fig. 4.13. Note that the object-based method was able to retrieve logos of similar objects regardless of their topology; all retrieved images have object of rectangular shape. On the other hand, the ZMMs method retrieved dissimilar logos as expected from most global methods [55]. Clearly, our proposed approach provides more meaningful retrievals based on this test with the exact matching algorithm slightly outperforms the approximate matching algorithm.

Labeled logo images

To compute recall, the database images have to be classified into groups according to their similarity. However, labeling all 1580 images is an extremely difficult task. Here, 246 logo images from the logo database used in the previous test are grouped into 17 classes. The number of logo images (m_i) for the i-th class ranges from 7 to 28. The manually grouped images in [61, 62] constitute about 40% of these images and are used as templates to manually group the rest of the images. Samples of these images are shown in Fig. 4.14 (for classes 1 to 9) and Fig. 4.15 (for classes 10 to 17). Each logo image is presented as

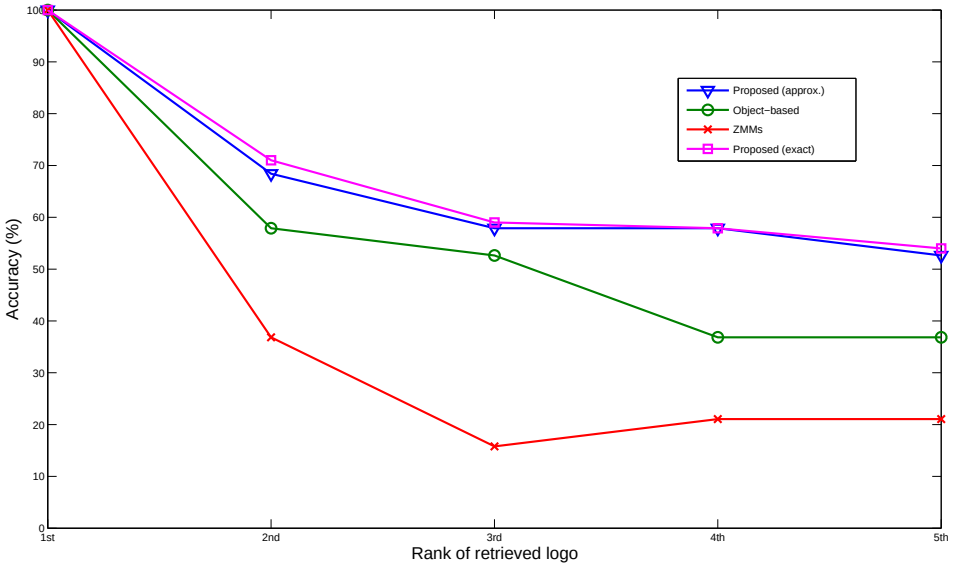


Figure 4.12: Results of the logo retrieval test using 24 queries and database of 1580 logos for the proposed (exact and approximate), the object-based and the ZMMs methods.

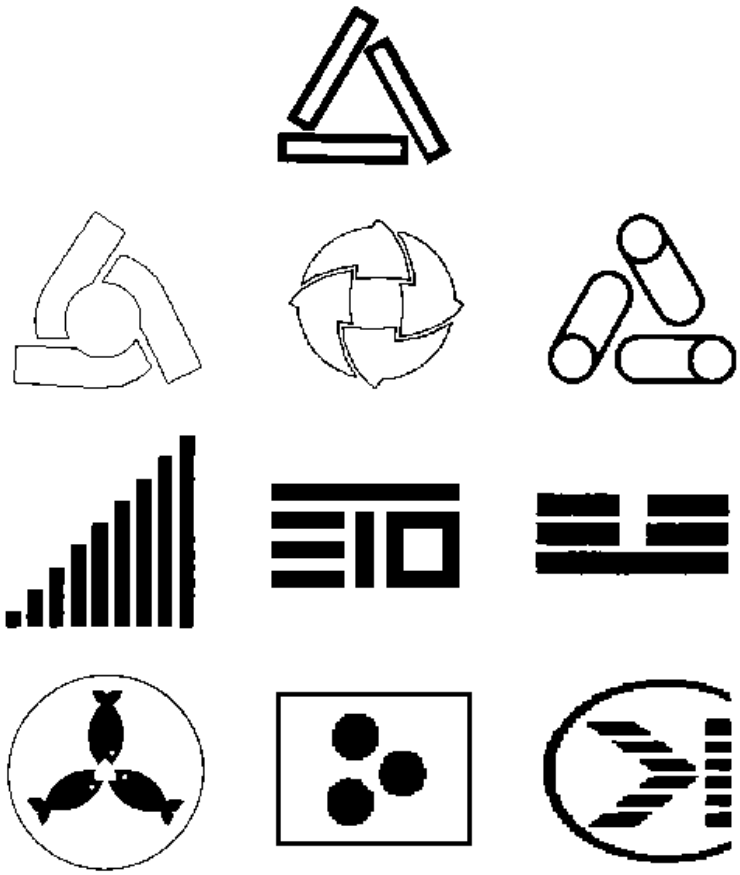


Figure 4.13: An example of (top) a query image, and the top-three retrieved logos of (2nd row) the proposed, (3rd row) the object-based and (last row) the ZMMs methods.

a query and the system sorts the rest of the images according to their similarity with the query. The accuracy for each query, belonging to class i, is computed as the number of the relevant images among the first m_i retrieved images. Fig. 4.16 shows the average accuracy for each class. Obviously, our proposed method outperforms the other two methods based on this test with a good margin.

Another test is based on the precision and the recall for each query. Since the number of relevant images is not fixed for all query images, computing the exact average precision-recall is not possible. Therefore, another effectiveness measure used by the information retrieval community, the F-measure [97], is computed for each query (see Section 2.3). Fig. 4.17 shows the average F-measure for each class in the database. The results of this test comes in accordance with those of the previous tests which confirms the advantage of our method over the others.

4.7 Conclusions and Discussions

In this chapter, the CT is introduced as a structured representation of multi-object shape images that encodes both shape and topology. Motivated by recent findings in psychology, the matching of two CTs is formulated as a maximum similarity subtree isomorphism (MSSI) problem. Two matching algorithms are developed which work in polynomial time. The first follows a continuous optimization approach that approximately solves the MSSI problem efficiently. The second algorithm guarantees finding an exact solution to the MSSI problem. We applied our method in two image retrieval application domains: retrieval of medical images and logo retrieval. Experiments with a database of 13500 medical images and a database of 1580 logos show the superiority of the proposed method over other methods in the literature.

The main strengths and limitations of the proposed approach for retrieving multi-object images are summarized in the following points:

- **Accuracy**: the proposed method has the advantage of achieving higher retrieval accuracy than many other methods in the literature based on medical imaging and logo retrieval applications.

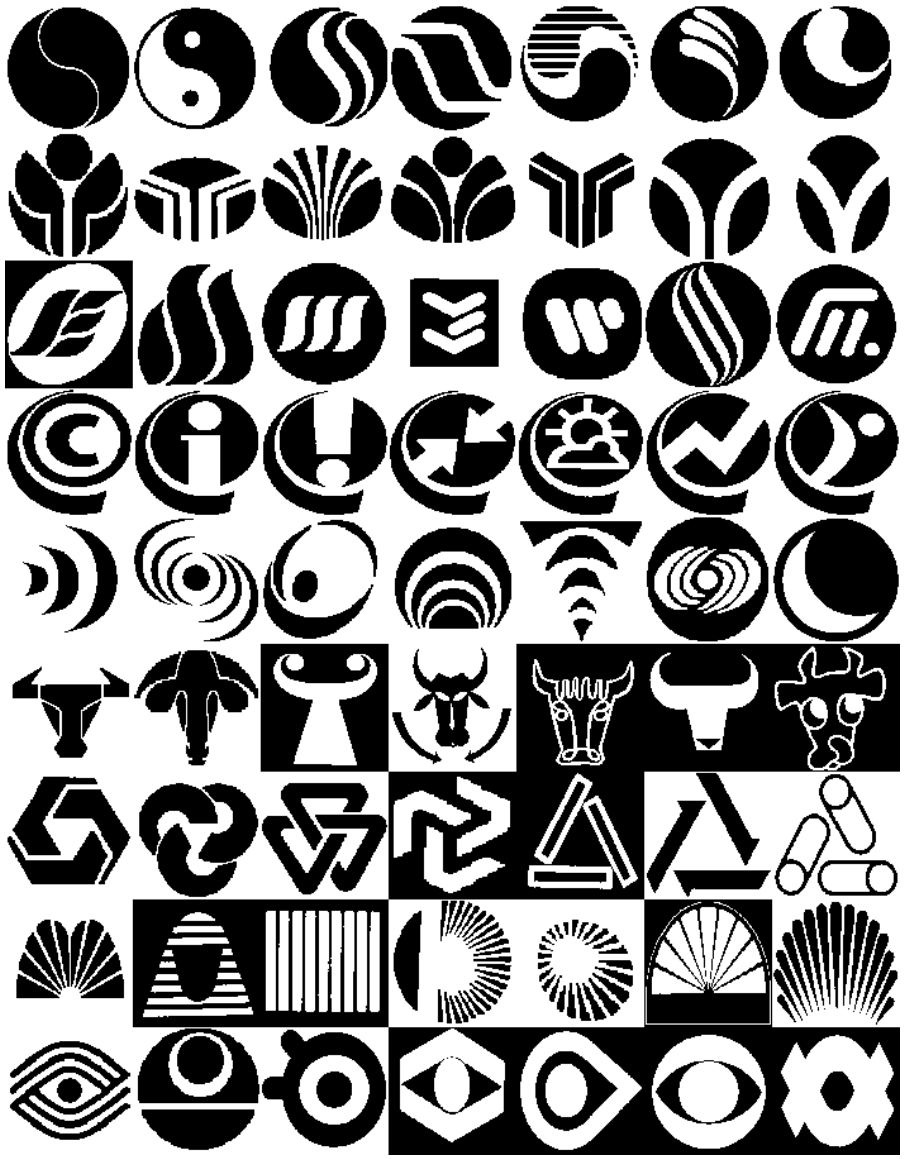


Figure 4.14: Sample of the labeled logo database images of class 1 (top row) to class 9 (bottom row).

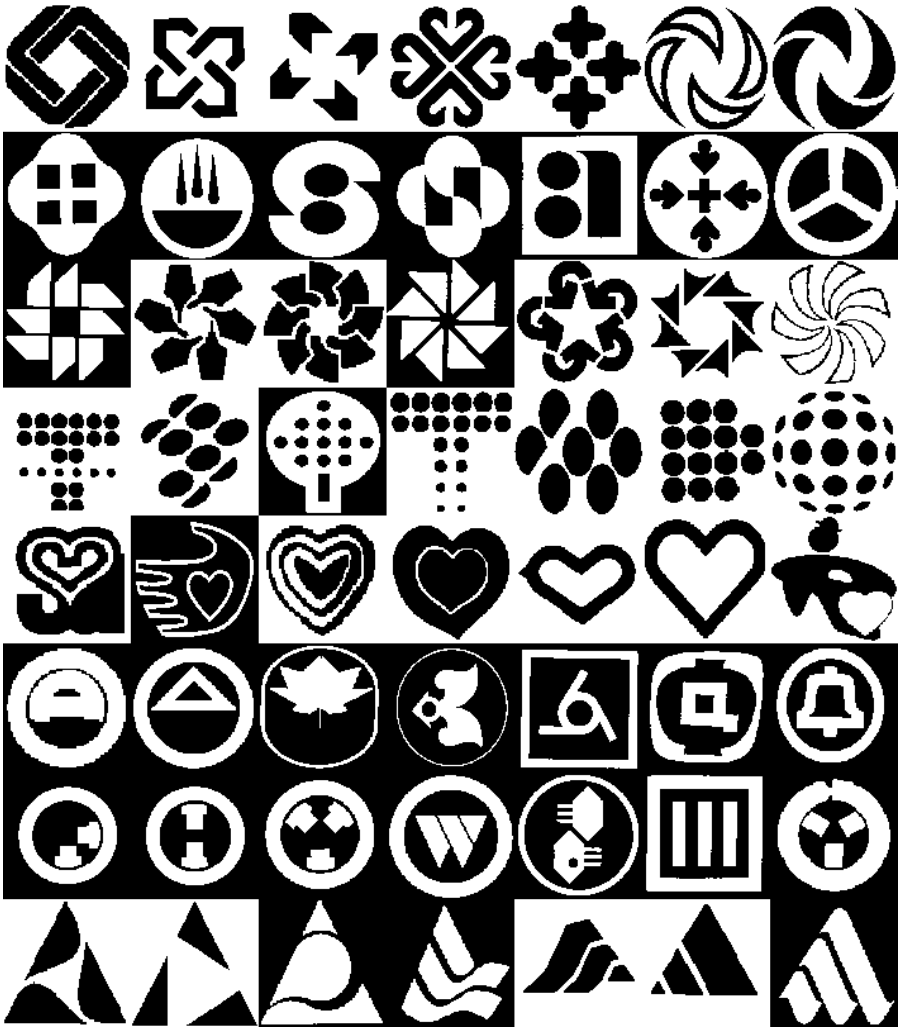


Figure 4.15: Sample of the labeled logo database images of class 10 (top row) to class 17 (bottom row).

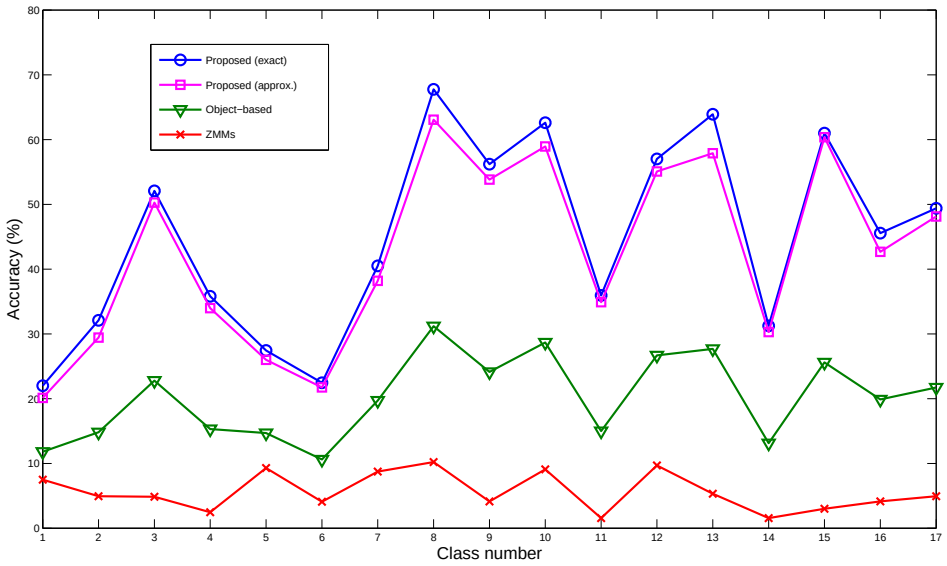


Figure 4.16: The average accuracy for each class of the labeled logo database of 246 images for the proposed (exact and approximate), the object-based and the ZMMs methods.

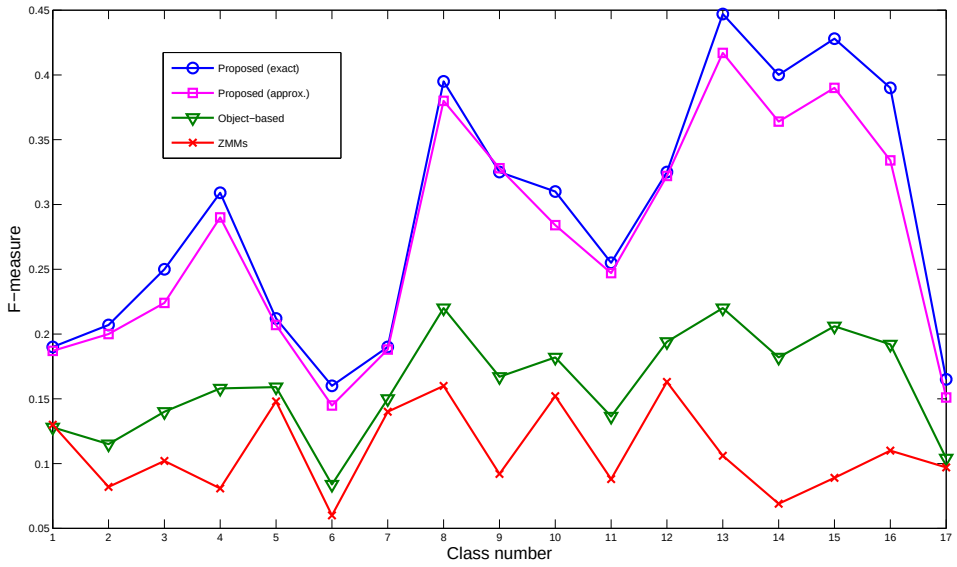


Figure 4.17: The average F-measure for each class of the labeled logo database of 246 images for the proposed (exact and approximate), the object-based and the ZMMs methods.

- **Flexibility**: the proposed method handles various types of queries. The user can search a database for images containing a particular object or multiple objects with specific structural relationships; taking into consideration the shape of the objects, their topology, or both shape and topology at once.

- **Invariance**: the rotation of (part of) the image objects can change the order of the sibling nodes in the corresponding CT; therefore, the employed tree matching method is invariant to the rotation angle of the image since it works for unordered trees. Regarding scale, the CT is clearly invariant to the scale of the image. In addition, the CT representation is invariant to negative transformation of the binary image; changing the background/foreground from 0/1 to 1/0 only adds/deletes a child node to the root node.

- **Complexity**: the proposed method has polynomial-time complexity; a property highly recommended in most retrieval applications. In our implementation using

Matlab ©(version 7.0) program running on Pentium IV 3.0 GHz PC, the average times for matching two images are approximately 0.5 seconds for the approximate matching and 0.6 seconds for the exact matching. It should be noted that the codes are not optimized and better speeds can be obtained using other programming languages such as C and C++. The ZMMs method is much faster (takes less than 0.1 milliseconds for matching two images). One possible way to speed-up the retrieval time is to use a hybrid design of two-level hierarchy, similar to that in [55], where the ZMMs method is used in the first stage to prune the database (by excluding very dissimilar images) and the CT matching is used in the second stage to refine the matching.

- **Robustness**: in this thesis, the segmentation is assumed to be already done and the object boundary is clearly identified. In practice, noise and partial occlusion can drastically change the topology of the CT. One way to overcome this limitation is to use mathematical morphology in the segmentation stage which is the basis for our future work in this area. For instance, if a dilation or erosion operation changes the topology of an image's CT, then both CTs are used for representing the image.

Chapter 5

Envelope Detection of Multi-Object Shapes

In Chapter 3, the aim is to measure the similarity between two single-object shape images. Then, Chapter 4 provides a solution to measure the similarity between images, containing multiple objects with holes, based on both shape and topology. However, the previous two chapters do not address the problem of detecting high-level boundaries resulting from certain arrangements of a group of objects such as in Fig. 5.1. The purpose of the work presented in this chapter is to allow for high-level shape representation and matching in multi-object images by detecting and extracting the envelope of high-level object groupings in the image. A new approach for the envelope detection and extraction is proposed which is mainly motivated by studies in Gestalt theory. The proposed approach works in two main stages. The first stage detects the envelope (if exists) and groups its objects using clustering techniques. Two algorithms are proposed for the object grouping. In the second stage, the high-level boundary of the envelope is extracted using morphological operations and then further refined using concavity tree reconstruction to eliminate odd concavities in the extracted envelope. Experiment on a set of logo images demonstrates the effectiveness of our approach.

The remainder of this chapter is organized as follows. Section 5.1 gives an introduction to the subject. Then, Section 5.2 reviews the main Gestalt principles. The proposed approach is explained in Section 5.3 and the experimental results are presented in Section

5.4. Finally, Section 5.5 concludes the work of this chapter.

5.1 Introduction

Semantic image retrieval has recently emerged as a result of the fact that most users do not require to retrieve images based on only their low-level features [34]. Existing techniques fall behind that target. The ability of an image retrieval system to extract all or most relevant information from an image is a necessary first step for the understanding of its content. As an example, multi-object images contain more shape information than the mere sum of the shape information of the individual components. A group of objects can be spatially arranged such that their envelope has a semantically high-level shape. Fig. 5.1 shows two multi-object images containing the same objects with different spatial arrangements. Clearly, this difference significantly changes the way we perceive the images. The objects of the image in the left of the figure form a star-shaped envelope whereas the objects of the other image form a rectangular envelope. Our aim is to detect and extract such envelopes which is useful as a mean for higher level interpretation of the shape information in the scene being analyzed (it may as well be an end by itself such as in the logo retrieval application).

The envelope concept was first originated by Eakins et al. in the context of their ARTISAN system for trademark retrieval [37]. They suggested grouping boundary segments based on their proximity and shape features as part of an ongoing research; however, no detailed method was reported. In [57], an adaptive selection scheme between Zernike moments and geometric primitives, obtained using the Hough transform, is used for trademark feature extraction. The authors claim that these features capture Gestalt-based features which include symmetry, continuity, proximity, parallelism and closure. However, this method does not detect high-level envelopes.

In this chapter, an approach for envelope detection and extraction in multi-object shapes is proposed. It consists of two main stages. In the first, a hierarchical clustering algorithm is used to group objects based on both their "physical" proximity, as well as their shape similarity. Two approaches are implemented to decide the final grouping of the objects: a heuristic-based approach and an approach based on evidence accumulation

Figure 5.1: Two multi-object images containing the same object and with totally different perceived envelopes.

proposed in [43]. In the second stage, morphological operations are used to merge the components in each of the grouping identified in the first stage without changing their size. The envelope is then extracted by reconstructing the merged component using its concavity tree to eliminate odd concavities. Segmentation of the objects in an image is an important process for successful automatic image retrieval. For natural images, this process is very challenging and beyond the scope of this thesis. Therefore, we assume that objects are already segmented and their boundaries are well identified.

5.2 Perceptual Grouping

Although the grouping task is made effortlessly by humans, it is not the case for machines. Mathematical models tend to describe low-level features effectively but fail in high-level interpretation. Our approach for objects grouping is based on the perceptual grouping principles where low-level image features and hierarchical clustering are employed to make decisions about the proximity, shape similarity and orientation of a group of objects.

A study by Biederman [21] suggests that the human visual system quickly assumes

and uses collinearity, curvature, parallelism, and adjacency of a group of objects in order to perceive them as a whole. Another study by Witkin and Tenenbaum [134] defined what is so-called non-accidental properties that the brain uses for inferring a scene. These properties include collinearity, co-curvilinearity, symmetry, parallelism, and convergence.

Gestalt theory provides an interpretation of the human perception of visually similar objects into groups [120]. Gestalt theory provides four main principles about the human perception which include proximity, similarity, continuity and closure. Fig. 5.2 illustrates these principles. The objects in panel (a) are grouped into one square due to their proximity whereas the objects in panel (b) are grouped into two groups due to their similarity (i.e. circles and triangles). The four objects in panel (c) form a cross symbol because the human perception assumes continuity in this case. Although the four objects in panel (d) are separated from each other, they are perceived as a square since our perception assumes closure. While these principles were derived from observations, the mechanisms behind them are still unclear. Our approach models the first three principles; the fourth principle is very difficult to model.

5.3 The Proposed method

Given a multi-object binary image, the aim is to detect and extract any envelope formed by a group of objects. This is achieved in two steps. At first, hierarchical clustering is used to group the objects forming the envelope based on their proximity, shape similarity and orientation. Secondly, morphological operations are used to merge the objects in each grouping and the envelope is then extracted by reconstructing the merged object from its concavity tree; thus, removing any artifacts along the envelope boundary.

5.3.1 Object Grouping

Brain research showed experimentally that the processing of proximity and other features are performed separately [120]. Therefore, in our approach, objects are grouped separately based on their proximity, shape similarity, and orientation using hierarchical clustering. Then, the final grouping is decided using two approaches. The first is based on a set of heuristic rules while the second approach is based on a recent method for combining

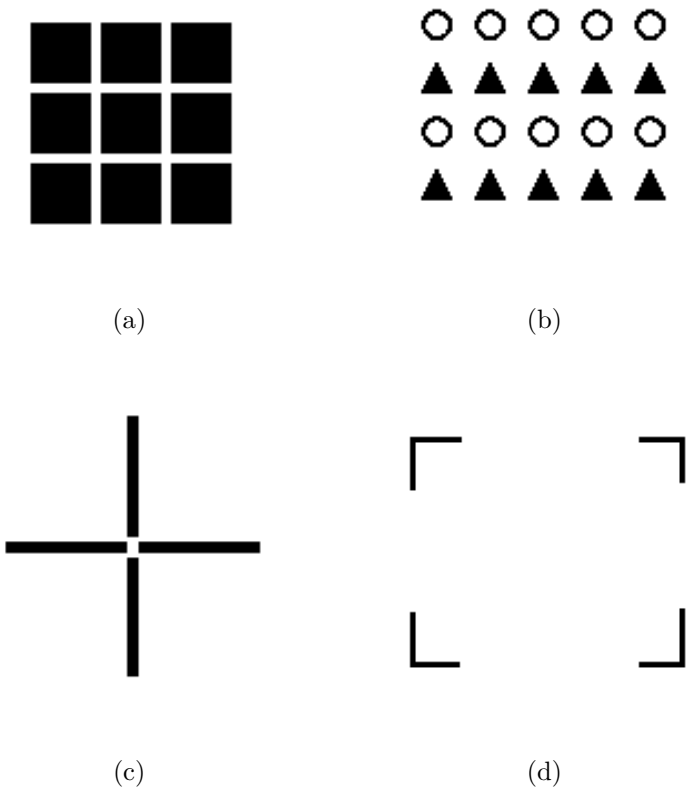


Figure 5.2: Illustration of the Gestalt principles of human perception: (a) proximity, (b) similarity, (c) continuity and (d) closure.

multiple clusterings called evidence accumulation [43]. In the following, we describe the feature extraction process for the proximity, the shape similarity and the orientation.

Proximity: the definition of a spatial distance between two objects that reflects the human judgment of such distance is not yet clear. Obviously, the shortest distance is totally independent of their shape. However, a desirable property of the distance is to be sensitive to all points in both objects. Therefore, the Hausdorff distance is adopted [100]. The Hausdorff distance between two sets of points is the maximum distance of a set to the nearest point in the other set. More formally, the Hausdorff distance $h(X, Y)$ between two objects X and Y is defined as:

$$h(X, Y) = \max_{x \in X} \left\{ \min_{y \in Y} \{d(x, y)\} \right\} \tag{5.1}$$

where x and y are points of objects X and Y, respectively, and d is the Euclidean distance. In order to make the distance function symmetric, a more general definition of Hausdorff distance would be:

$$H(X, Y) = \min \{h(X, Y), h(Y, X)\} \tag{5.2}$$

It is sufficient to consider only the boundary points of the two objects. Furthermore, we consider only the vertices of their convex hulls. This reduces the computations dramatically with minor effect on the performance.

Shape similarity: based on the observation that objects forming an envelope do not have complex boundaries, global shape descriptors are expected to describe the objects effectively. Another reason for choosing global descriptors is their compactness, which allows efficient computing of the distances between them. Here, the shape of an object is described by its area, eccentricity and solidity [31].

Orientation: the orientation of an object is taken as the orientation of its major axis, which is the straight line segment joining the two points farthest from each other. The major axis orientation is defined as the angle between the horizontal axis and the axis around which the object can be rotated with minimum inertia [31]. This

feature is particularly important because objects may form an envelope, although they are not similar in shape, when they are aligned in parallel or in series.

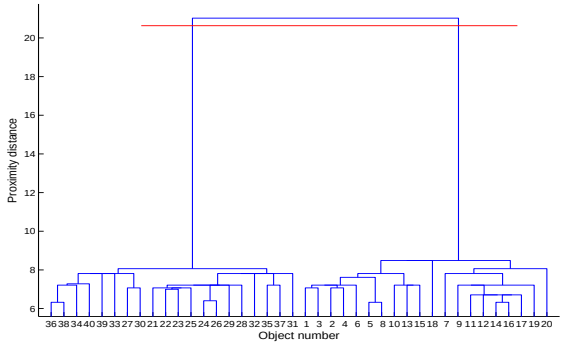
Here, a hierarchical clustering algorithm [54] is applied to the distance matrices of proximity, shape similarity, and orientation. Srivastava et al. applied hierarchical clustering to group similar silhouettes to search shape databases efficiently [119]. The result is a hierarchical tree, called dendrogram, which is not a single set of clusters, but rather a multi-level hierarchy where clusters at one level are joined as clusters at the next higher level as shown in Fig. 5.3. In our application, clusters are defined when there is a clear cut in the dendrogram. In this case, the compactness of a cluster is defined by how similar its members are. For proximity and shape similarity groupings, deciding to cut the dendrogram based on the maximum lifetime[1] has a major drawback, that is, a single large distance can dominate the decision when the inter-class or intra-class variation is large. Here, the cutting decision is based on statistics, including the mean and the standard deviation, computed from the distances under each node in the dendrogram. To achieve scale invariance, the standard deviations of all nodes are normalized to have zero mean and unity variance. Intuitively, a node with small normalized standard deviation most likely groups proximal (or similar objects), and vice versa. Therefore, a threshold is set for cutting the dendrogram based on the normalized standard deviation (which is evaluated experimentally to be equal to one). For orientation-based grouping, the dendrogram cut is made directly at the desired angle, which is considered to be ten degrees.

An illustrative example is shown in Fig. 5.3. Panel (a) shows the input multi-object shape. The results of hierarchical clustering based on proximity, shape similarity, and orientation are shown in panels (b), (d), and (f), respectively. The horizontal line in each dendrogram shows the location of deciding the groups. Grouping based on proximity, shape similarity and orientation result in two groups, four groups and one group, respectively.

The final grouping of the image objects is made based on the initial mentioned groupings. For this purpose, two approaches are presented as follows.

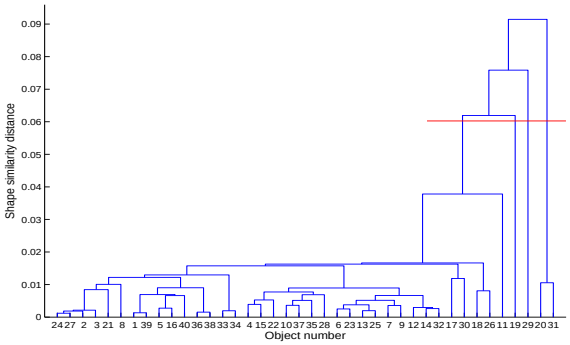
[1]The lifetime is the difference between the distances at two successive nodes.

(a)

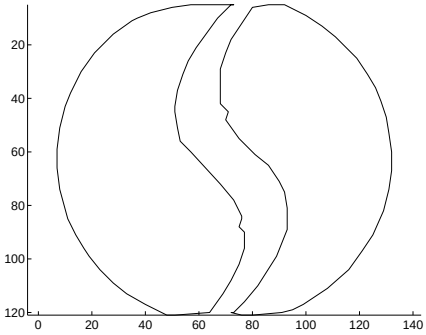


(b)

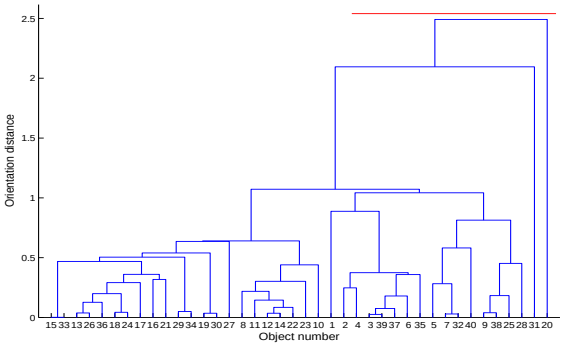
(c)



(d)



(e)



(f)

Figure 5.3: An example of object grouping and envelope extraction. See text for details.

Heuristic-based grouping

In this approach, the final grouping of the image objects is decided using heuristic rules [4]. As shown in the block diagram of Fig. 5.4, two alternative groupings are considered based on the initial groupings. In the first, objects that belong to the same proximal and shape similarity groups are judged as one group. The second alternative regards objects of the same proximal and orientation groups as one group. The alternative that results in lower number of groups is considered as the final grouping. For example, the grouping based on the proximity and the shape similarity of the image objects in Fig. 5.3 results in 6 groups whereas the grouping based on the proximity and the orientation results in 2 groups; therefore, the latter is considered as the final grouping as shown in panel (c).

Grouping based on evidence accumulation

A systematic approach for deciding the final grouping is obtained using a recent method in the literature about combining multiple clusterings using evidence accumulation [43]. Fig. 5.5 shows a block diagram of the object grouping using the evidence accumulation. Here, the outcomes of four initial groupings are considered as evidences which are accumulated in a new distance matrix. Note that, unlike the previous approach, the area is considered separately from the other shape features. Then, hierarchical clustering is applied on this matrix to decide the final grouping. In the following, we describe the application of this method on our grouping problem.

Assume there are M initial groupings (here, $M = 4$) of m image objects. The grouping ensemble is defined as:

$$G^i = \left\{C_1^i, C_2^i, ..., C_{k_i}^i\right\} \tag{5.3}$$

where C_j^i is the jth cluster in the grouping G^i with k_i clusters and $i \in \langle 1, M \rangle$. Then, a voting scheme combines the outcomes of the initial groupings using a co-occurrence matrix, D, of object pairs defined as:

$$D(p, q) = \frac{m_{pq}}{M} \tag{5.4}$$

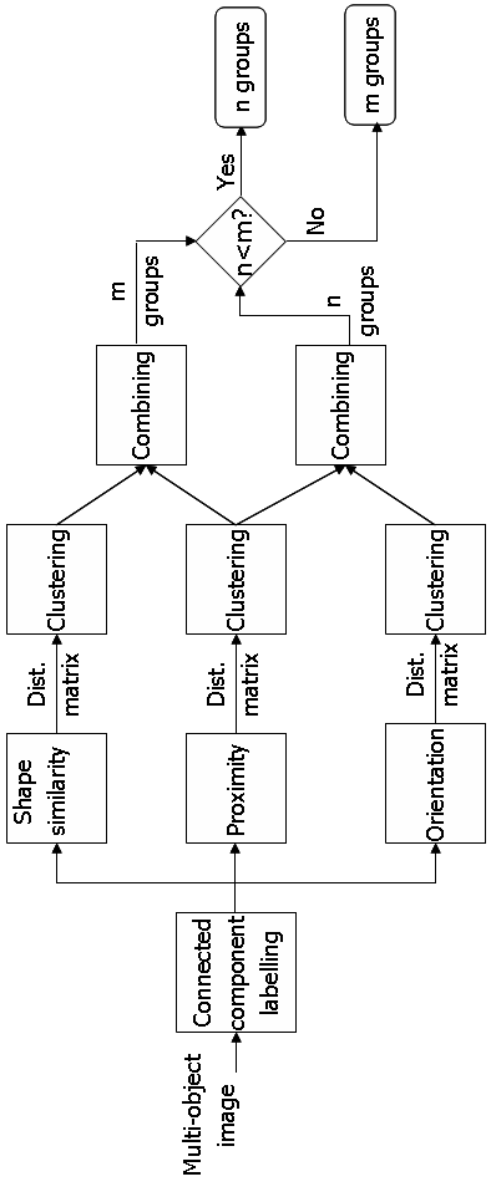


Figure 5.4: Block diagram of the heuristic-based object grouping.

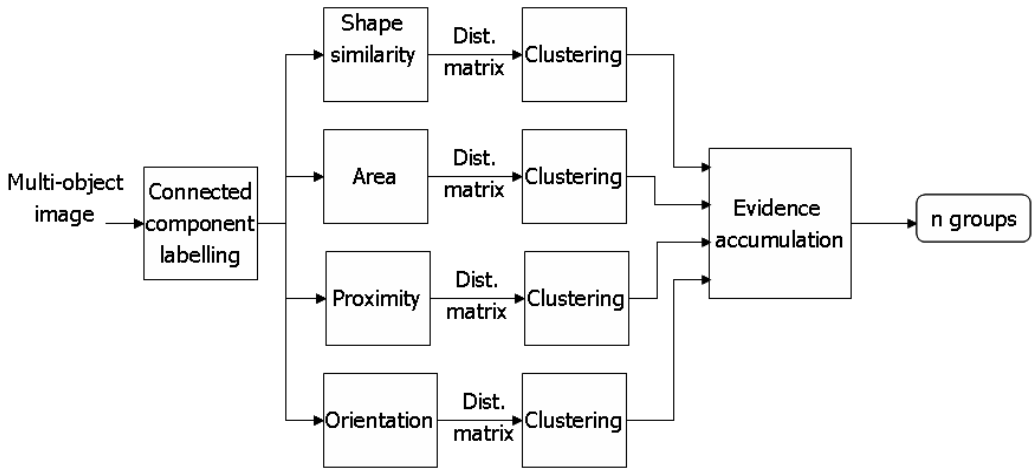


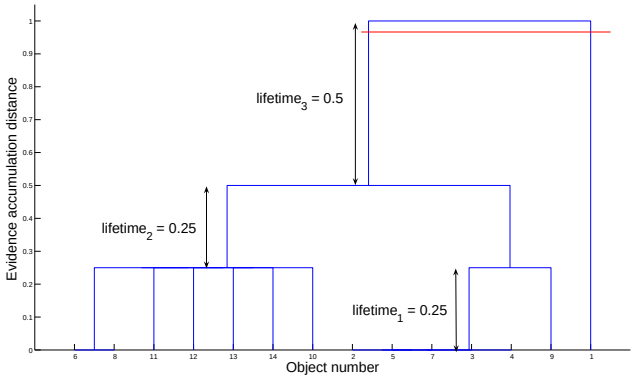
Figure 5.5: Block diagram of the object grouping based evidence accumulation.

where m_{pq} is the number of times the objects p and q are assigned to the same cluster C_j^i among the M initial groupings. To decide the final grouping, hierarchical clustering is applied on D and the dendrogram cutting is done based on the maximum lifetime. An example illustrating this process is shown in Fig. 5.6. Clearly, the maximum lifetime in the co-occurrence matrix dendrogram, of panel (b), partitions the image objects into two clusters as shown in panel (c).

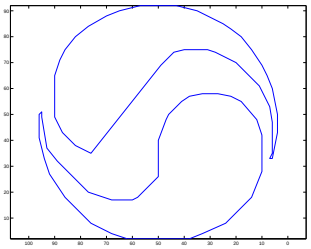
5.3.2 Envelope Extraction

This is the second stage towards the extraction of the semantic envelope. The input to this stage is the output of the object grouping stage; specifically, a labeled matrix of the object groupings in the image where objects belonging to the same group are assigned the same label. There are two sub-steps in this stage: the first is to morphologically merge the objects in each grouping; the second is to extract the envelope of the merged objects using a contour-based concavity tree reconstruction algorithm. A requirement of the second step is that it is passed an image with a single component.

(a)



(b)



(c) (d)

Figure 5.6: Illustration of the object grouping using evidence accumulation; (a) input image, (b) dendrogram of the co-occurrence matrix D, (c) the final grouping and (d) the extracted envelope.

Merging the groups

For each grouping identified in stage one, the constituent objects are repeatedly dilated using a 3×3 structuring element until the resulting grouping has only one component. If the dilation operation was performed n times, and the envelope is extracted at this stage, it would be n pixels larger than it should (because of the dilation). We need then to shrink the merged component n pixels, but without splitting it. The shrinking can be done using an erosion operation with a $(2n+1) \times (2n+1)$ structuring element. However, the erosion might (or might not) split the merged component. A splitting will occur if the (square) structuring element cannot slip through the neck joining pairs of (original) components. To get around this problem, we morphologically close the merged component with a diamond shaped structuring element with a main-diagonal of $2n+1$ pixels. This will always guarantee that the subsequent erosion will not split the merged component, as the square structuring element will now be always guaranteed to pass through the necks. If we proceed with the envelope extraction at this stage, the resulting envelope will have small odd concavities resulting from the morphological operations.

Extracting the envelope

The merged component identified in the above stage can now be used to extract the envelope. The contour of the merged component could be used as the envelope at this stage; however, it needs to be smoothed. This task is delegated to a contour-based concavity tree extraction algorithm [14] that will eliminate concavities smaller than a given threshold. This threshold varies with the gaps in between the original components. It is currently set to four times the area of the structuring element used in the erosion step. Fig. 5.7 illustrates the merging and envelope extraction processes. Note that only two tree nodes (including the root) were used for the envelope reconstruction from the corresponding concavity tree in each of the two envelopes.

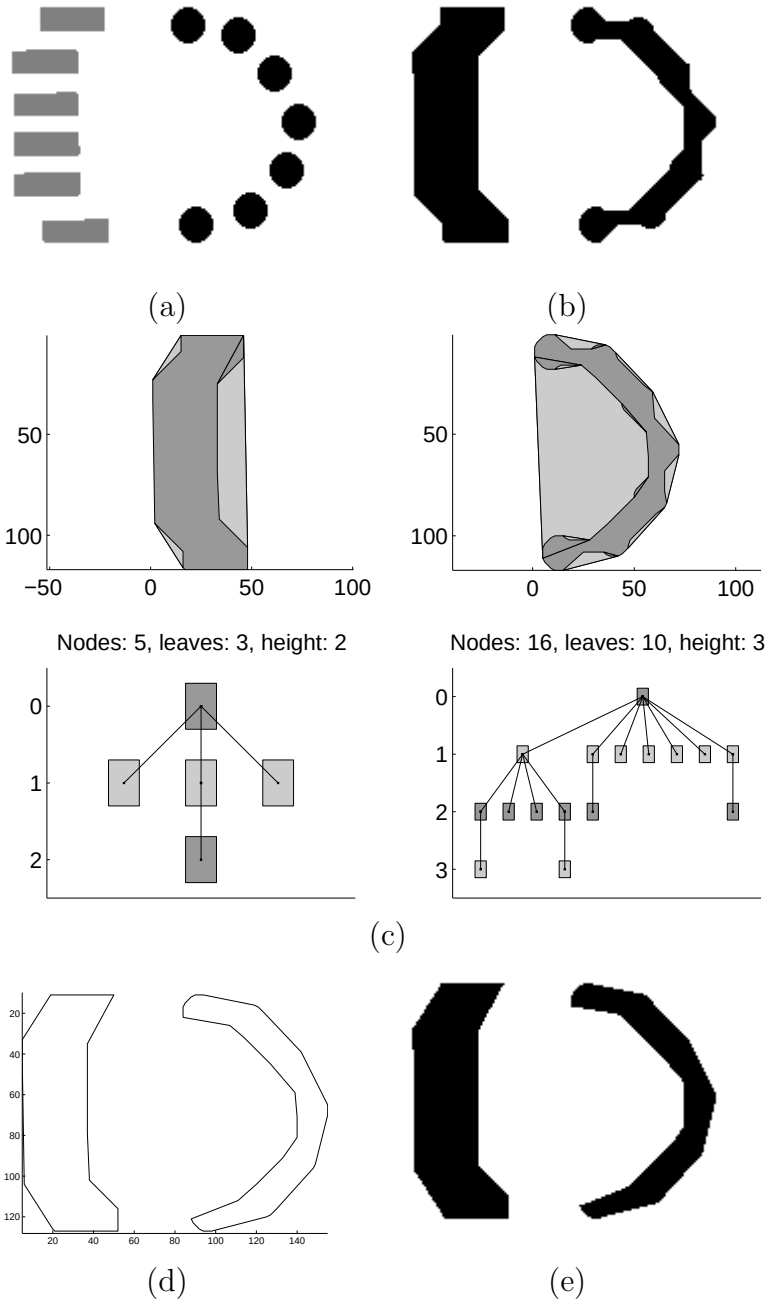


Figure 5.7: Illustration of the merging and envelope extraction process; (a) input image with two groupings, (b) result of merging, (c) concavity trees used to extract the envelope, (d) the resulting envelope and (e) the output image.

5.4 Experimental Results

The proposed approach was implemented and its performance was evaluated on a set of 110 trademark images containing a varying number of objects with various orientations and shapes. The outcomes of the object grouping stage were subjectively correct for 79 images using the heuristic-based approach and 93 images using the evidence accumulation approach. Moreover, when the envelope extraction algorithm was applied to each of the correctly grouped images, a subjectively correct envelope was extracted. Fig. 5.8 and Fig. 5.9 show samples of correct grouping and envelope extraction.

5.5 Conclusions and Discussions

Envelope extraction is a very important stage towards high-level shape representation and similarity matching. In this chapter, an approach for object grouping and envelope detection is proposed. The proposed approach utilizes the proximity and shape similarity between objects and their orientations for grouping them. Hierarchical clustering allows such utilization. The fusion of the outcomes of the initial groupings is achieved using two approaches. The first is based on heuristics and the second is based on evidence accumulation. The latter is more systematic and provided better performance. Then, the envelope of each group of objects is approximated by means of morphological operations. A contour-based approach for concavity tree reconstruction is employed to smooth the extracted envelope.

The envelope detection approach provides a tool for exploring more shape information than that of the individual image objects. Our approach detects envelopes in multi-object images with variable number of clusters. The extension to shape-based image retrieval can be easily made by applying shape matching techniques, as in Chapter 3, on the extracted envelope to obtain more meaningful retrievals.

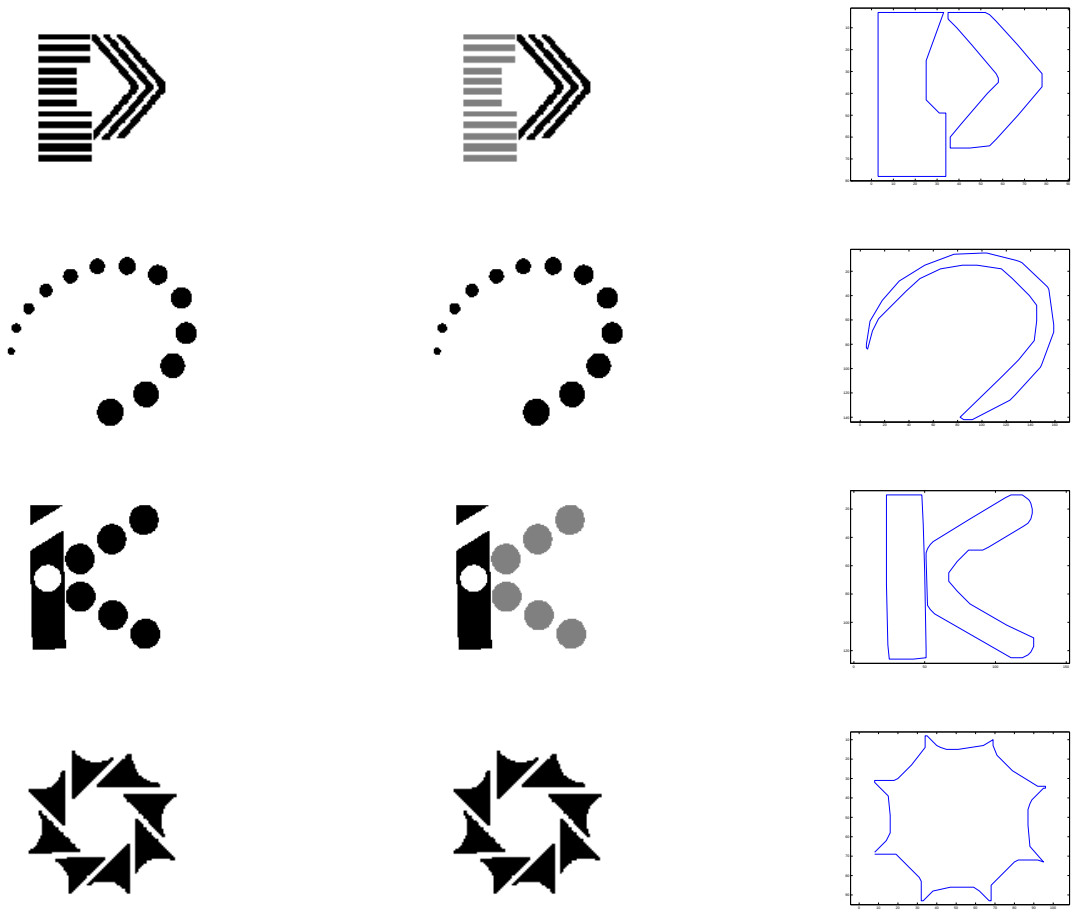


Figure 5.8: Samples of correct object grouping and envelope extraction; (left column) input images, (middle column) the grouping results and (right column) the extracted envelopes.

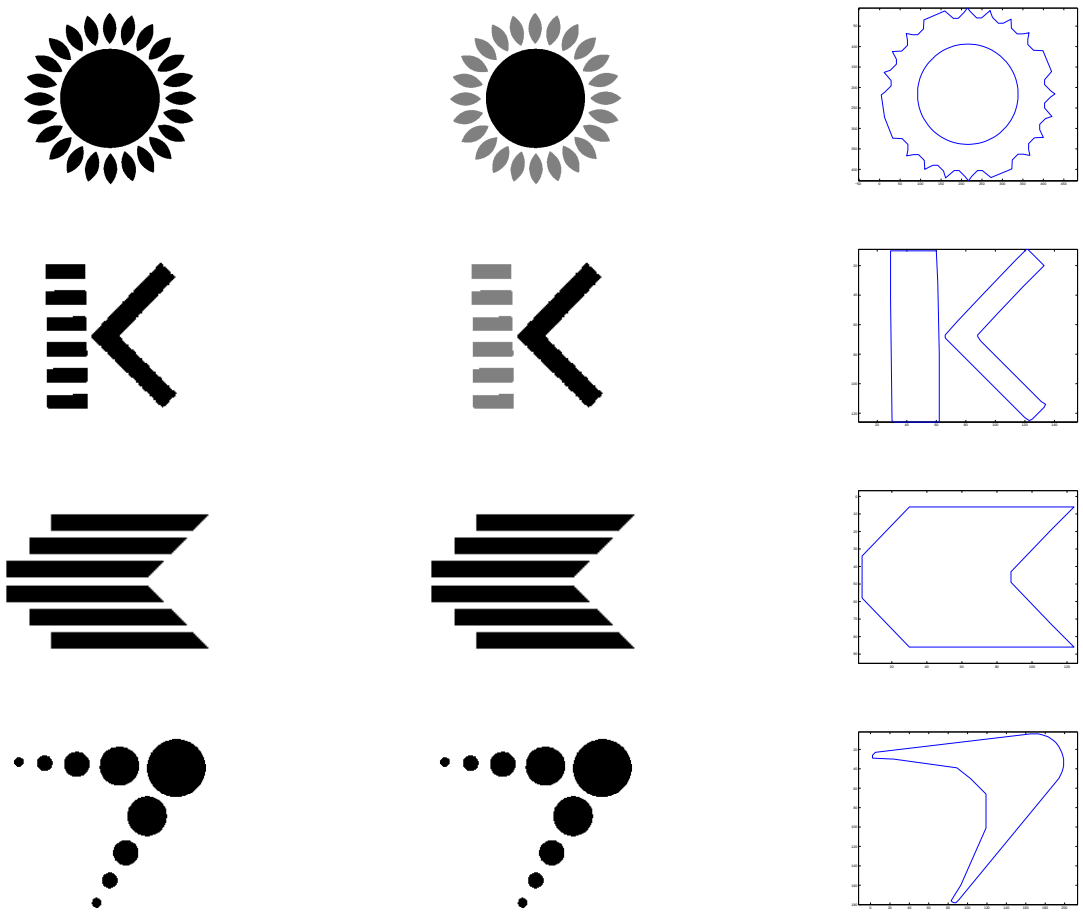


Figure 5.9: Samples of correct object grouping and envelope extraction; (left column) input images, (middle column) the grouping results and (right column) the extracted envelopes.

Chapter 6

Conclusions and Future Work

6.1 Contributions

- TAR-CI maxima matching is invariant to the affine transformations and outperforms the CSS method, which has been selected as the MPEG-7 CE-1 standard for boundary-based shape description and matching, in terms of both retrieval accuracy and computational complexity.

- The dynamic space warping algorithm for shape matching achieves the best retrieval performance, compared to all published methods in the literature so far, based on the MPEG-7 CE-1 part B retrieval test. Although the DSW is more complex than TAR-CI matching, the former has quadratic complexity in terms of the number of boundary points.

- The curvature tree encodes both shape and topology of multi-object shape images, is easy to construct and grows linearly with the number of objects in an image.

- Two algorithms are developed for curvature tree matching: an approximate and an exact. The exact (approximate) algorithm returns an exact (approximate) solution to the maximum similarity subtree isomorphism. The solution includes the weight of the isomorphism and the isomorphism itself. Both algorithms have polynomial-time complexity.

- The similarity function between two multi-object images is based on the maximum similarity subtree isomorphism between their curvature trees and takes into consideration the unmatched shapes in the images, which comes in accordance with recent findings in psychology. Besides, the similarity function is normalized between zero and one. This property is particularly important when incorporating other image features such as color or texture in a unified similarity function for retrieval.

- The curvature tree matching (either approximate or exact) achieves better retrieval performance than the state-of-the-art methods in two application domains. In medical image retrieval, the proposed method outperforms the ARG method using a database of 13500 real and synthesized medical images. The second application is logo retrieval where the proposed method outperforms the ZMMs, which has been selected as the MPEG-7 CE-2 standard for region-based shape description and matching.

- The proposed multi-object shape matching method handles various types of queries. The user can search a database for images containing a particular object or multiple objects with specific geometrical relationships; taking into consideration the shape of the objects, their topology, or both shape and topology at once.

- A new algorithm for detecting and extracting high-level envelopes of object groupings in multi-object images is developed based on Gestalt principles. This algorithm helps in obtaining more meaningful shape retrieval.

- A standard database of 1580 multi-object logo images is constructed, which include diverse types of real and synthesized shapes, for evaluation purposes of object-based shape retrieval systems. All images are composed of disjoint objects and, up to the author's best knowledge, there is no such database that is publicly available. A subset of 246 images from this database is manually classified into 17 groups and another subset of 110 images includes high-level envelopes.

6.2 Future Work

- Although the MPEG-7 and Kimia's shape databases used in our experiments include partially occluded shapes, the problem of partial occlusion has not been studied deeply in this thesis. Partial occlusion occurs in different forms; for instance, it can be modeled as missing parts of the image or as another unknown object occluding (part of) the image. Theoretically speaking, the curvature tree matching approach establishes an object-based correspondence; therefore, it is expected to perform well under missing objects up to certain extent. However, in practice, noise and partial occlusion can drastically change the topology of the curvature tree. One way to overcome this limitation is to use mathematical morphology in the segmentation stage. For instance, if a dilation or erosion operation changes the topology of an image's curvature tree, then both trees are used for representing the image. In this thesis, the segmentation is assumed already done.

- Database indexing is very critical for fast image retrieval especially in large image databases. An indexing scheme reduces the number of database images for the comparison with the query at the time of retrieval. Indexing methods are either exact or approximate. Many indexing methods for graph structures exist in the literature [114, 121]. Another alternative is to use two-level hierarchy similar to [55]. Global features, such as Zernike moments magnitudes (ZMMs) and/or shape complexity (SC), can be used for indexing and fast pruning of the database images. Then, candidate images are further refined using the curvature tree matching.

- Relevance feedback in content-based image retrieval has received great attention in recent years. Including human judgments in the similarity function improves the quality of retrieval and reduces the semantic gap problem. An important future direction of our work is to devise an architecture that incorporates the user judgement of the retrieved images to improve the quality of future retrievals. For instance, the user can directly adjusts some parameters of the system. Another recent direction in this area is to include positive and negative examples in a probabilistic framework [58], that is, the user can give examples that are very dissimilar to the query which the system should avoid during the retrieval.

- Directional relations between image objects (for example, object A is above object B) are particularly important in some application domains such as geographical information systems (GIS). A possible way to encode the directional relations in the curvature tree representation is to assign weights reflecting these relations to the tree edges. The relations can be defined with respect to a reference point in the image; for instance, the centroid of the largest object or the center point of the image.

- Another research direction is to extend the curvature tree matching approach to gray-scale and/or color images using automatic segmentation based on texture and/or color. In this case, texture and/or color features of the image objects are included in the node attributes for matching and retrieval.

- In Chapter 4, the continuous optimization approach for solving the maximum similarity subtree isomorphism problem returns an approximate solution. A possible future direction is to formulate the minimization of the quadratic function of (4.3) as a constrained optimization problem on the variables of the function [29]. For the exact curvature tree matching algorithm, the bipartite matching procedure constitutes the kernel of the algorithm. In this thesis, the Hungarian method is employed. However, more efficient procedures can be used [20].

6.3 List of Publications

6.3.1 Journal papers

1. N. Alajlan, M. S. Kamel, and E. Jernigan. Detail preserving impulse noise removal. *Signal Processing: Image Communication*, 19(10):993-1003, November 2004.

2. I. El Rube, N. Alajlan, M. S. Kamel, M. Ahmed, and G. Freeman. MTAR: a robust 2D shape representation. *International Journal of Image and Graphics (IJIG)*, 6(3):421–443, July 2006.

3. N. Alajlan, M. S. Kamel, and G. Freeman. Multi-Object image retrieval based on shape and topology. *Signal Processing: Image Communication*, 21(10):904–918, November 2006.

4. N. Alajlan, I. El Rube, M. S. Kamel, and G. Freeman. Shape retrieval using triangle-area representation and dynamic space warping. *Pattern Recognition*. Accepted in November 2006.

5. N. Alajlan, M. S. Kamel, and G. Freeman. Geometry-based image retrieval in image databases. Submitted to IEEE Transactions on Knowledge and Data Engineering.

6. N. Alajlan, M. S. Kamel, and G. Freeman. Detection and extraction of high-level envelopes in multi-object shapes. Submitted to Pattern Recognition.

6.3.2 Conference papers

7. N. Alajlan, P. Fieguth, and M. S. Kamel. Robust shape retrieval using maximum likelihood theory. In *1st International Conference on Image Analysis and Recognition (ICIAR 2004)*, volume 3211, pages 745-752, Porto, Portugal, September 2004.

8. N. Alajlan and E. Jernigan. An effective detail preserving filter for impulse noise removal. In *1st International Conference on Image Analysis and Recognition (ICIAR 2004)*, volume 3211, pages 139-146, Porto, Portugal, September 2004.

9. I. El Rube, N. Alajlan, M. S. Kamel, M. Ahmed, and G. Freeman. Robust multiscale triangle-area representation for 2D shapes. In *IEEE International Conference on Image Processing (ICIP 2005)*, pages 545-548, Genoa, Italy, September 2005.

10. I. El Rube, N. Alajlan, M. S. Kamel, M. Ahmed, and G. Freeman. Efficient mul tiscale shape-based representation and retrieval. In *2nd International Conference on Image Analysis and Recognition (ICIAR 2005)*, volume 3656, pages 415-422, Toronto, Canada, September 2005.

11. N. Alajlan, O. El Badawy, M. S. Kamel, and G. H. Freeman. Envelope detection of multi-object shapes. In *2nd International Conference on Image Analysis and Recognition (ICIAR 2005)*, volume 3656, pages 399-406, Toronto, Canada, September 2005.

Bibliography

[1] S. Abbasi and F. Mokhtarian. Affine-similar shape retrieval: application to multi-view 3-d object recognition. *IEEE Transactions on Image Processing*, 10(1):131–139, January 2001.

[2] S. Abbasi, F. Mokhtarian, and J. Kittle. Curvature scale space image in shape similarity retrieval. *MultiMedia Systems*, 7(6):467–476, 1999.

[3] T. Adamek and N.E. O'Connor. A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Transactions on Circuits and Systems for Video Tech.*, 14(5):742–753, May 2004.

[4] N. Alajlan, O. El Badawy, M. S. Kamel, and G. H. Freeman. Envelope detection of multi-object shapes. In *2nd International Conference on Image Analysis and Recognition (ICIAR 2005)*, volume 3656, pages 399–406, Toronto, Canada, September 2005.

[5] N. Alajlan, I. Elrube, M. S. Kamel, and G. Freeman. Shape retrieval using triangle-area representation and dynamic space warping. *Pattern Recognition*, Accepted in November 2006.

[6] N. Alajlan, P. Fieguth, and M. S. Kamel. Robust shape retrieval using maximum likelihood theory. In *1st International Conference on Image Analysis and Recognition (ICIAR 2004)*, volume 3211, pages 745–752, Porto, Portugal, September 2004.

[7] N. Alajlan and E. Jernigan. An effective detail preserving filter for impulse noise removal. In *1st International Conference on Image Analysis and Recognition (ICIAR 2004)*, volume 3211, pages 139–146, Porto, Portugal, September 2004.

[8] N. Alajlan, M. S. Kamel, and G. Freeman. Multi-object image retrieval based on shape and topology. *Signal Processing: Image Communication*, 21(10):904–918, November 2006.

[9] N. Alajlan, M. S. Kamel, and E. Jernigan. Detail preserving impulse noise removal. *Signal Processing: Image Communication*, 19(10):993–1003, November 2004.

[10] N. Arica and F. Vural. Bas: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639, 2003.

[11] F. Ashby and R. Gott. Decision rules in the perception and categorization of multidimensional stimuli. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14(1):33–53, 1988.

[12] J. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. Shu. Virage image search engine: An open framework for image management. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 76–87, February 1996.

[13] O. El Badawy and M. S. Kamel. Shape representation using concavity graphs. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 461–464, 2002.

[14] O. El Badawy and M. S. Kamel. Hierarchical representation of 2-D shapes using convex polygons: a contour-based approach. *Pattern Recognition Letters*, 26(7):865–877, 2005.

[15] J Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms, and Applications*. Springer, 2001.

[16] I. Bartolini, P. Ciaccia, and M. Patella. Warp: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):142–147, 2005.

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

[18] S. Berretti, A. D. Bimbo, and P. Pala. Retrieval by shape using multidimensional indexing structures. In *the 10th International Conference on Image Analysis and Processing (ICIAP 1999)*, pages 945–950, 1999.

[19] S. Berretti, A. D. Bimbo, and P. Pala. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on Multimedia*, 2(4):225–239, 2000.

[20] D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, 1981.

[21] I. Biederman. Recognition by components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

[22] I. M. Bomze. Evolution towards the maximum clique. *Journal of Global Optimization*, 10(2):143–164, 1997.

[23] I. M. Bomze, M. Pelillo, and V. Stix. Approximating the maximum weight clique using replicator dynamics. *IEEE Transactions on Neural Networks*, 11(6):1228–1241, 2000.

[24] G. Brassard and P. Bratley. *Fundamentals of Algorithms*. Prentice-Hall, Englewood Cliffs, NJ, 1996.

[25] M. Carlin. Measuring the performance of shape similarity retrieval methods. *Computer Vision and Image Understanding*, 84(1):44–61, 2001.

[26] C. C. Chang and S. Y. Lee. Retrieval of similar pictures on pictorial databases. *Pattern Recognition*, 24(7):675–681, 1991.

[27] CIPO. The canadian intellectual property office. *www.strategis.gc.ca/sc_mrksv/cipo*.

[28] S. D. Cohen and L. J. Guibas. The earth mover's distance under transformation sets. In *the 7th IEEE International Conference on Computer Vision (ICCV 1999)*, pages 1076–1083, September 1999.

[29] T. F. Coleman and Y. Li. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058, 1996.

[30] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.

[31] L. F. Costa and R. M. Cesar. *Shape Analysis and Classification: Theory and Practice*. CRC Press, Inc., 2000.

[32] IBM Prototype Trademark Database. *www.qbic.almaden.ibm.com/tmdemo*.

[33] J. Deller, J. Hansen, and J. Proakis. *Discrete-Time Processing of Speech Signals*. Wiley-IEEE Press; Reprint edition, 1999.

[34] J. P. Eakins. Towards intelligent image retrieval. *Pattern Recognition*, 35(1):3–14, 2002.

[35] J. P. Eakins, J. M. Boardman, and M. E. Graham. Similarity retrieval of trademark images. *IEEE MultiMedia*, 5(2):53–63, 1998.

[36] J. P. Eakins, K. J. Riley, and J. D. Edwards. Shape feature matching for trademark image retrieval. In *International Conference on Image and Video Retrieval*, volume 2728, pages 28–37. LNCS, Springer, August 2003.

[37] J. P. Eakins, K. Shields, and J. M. Boardman. Artisan: a shape retrieval system based on boundary family indexing. In *Storage and Retrieval for Still Image and Video Databases IV. Proceedings SPIE 2670*, pages 17–28, 1996.

[38] E. A. El-Kwae and M. R. Kabuka. A robust framework for content-based retrieval by spatial similarity in image databases. *ACM Transactions on Information Systems*, 17(2):174–198, 1999.

[39] I. Elrube, M. Ahmed, and M. S. Kamel. Coarse-to-fine multiscale affine invariant shape matching and classification. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004)*, volume 2, pages 163–166, Cambridge, UK, August 2004.

[40] I. Elrube, M. S. Kamel, and M. Ahmed. 2-d shape matching using asymmetric wavelet-based dissimilarity measure. In *1st International Conference on Image Analysis and Recognition (ICIAR 2004)*, volume 3211, pages 368–375, Porto, Portugal, September 2004.

[41] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[42] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *IEEE Computer*, 28(9):23–32, September 1995.

[43] A. L. N. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

[44] H. Freeman and A. Saaghri. Generalized chain codes for planar curves. In *International Conference on Pattern Recognition (ICPR78)*, pages 701–703, 1978.

[45] D. Fry. *Shape recognition using metrics on the space of shapes*. PhD thesis, Harvard University, 1993.

[46] P. Giannopoulos and R. C. Veltkamp. A pseudo-metric for weighted point sets. In *the 7th European Conference on Computer Vision (ECCV 2002)*, pages 715–730, 2002.

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

[48] W. I. Grosky, P. Neo, and R. Mehrotra. A pictorial index mechanism for model-based matching. *Data and Knowledge Engineering*, 8:309–327, 1992.

[49] V. Gudivada and V. Raghavan. Design and evaluation of algorithms for image retrieval by spatial similarity. *ACM Transactions on Information Systems*, 13(2):115–144, 1995.

[50] F. Harary. *Graph Theory*. Addison-Wesley, 1969.

[51] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IT-8:179–197, 1962.

[52] H. H. S. Ip and D. G. Shen. An affine-invariant active contour model (ai-snake) for model-based segmentation. *Image and Vision Computing*, 16(2):135–146, 1998.

[53] F. Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23:52–72, 1975.

[54] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, September 1999.

[55] A. K. Jain and A. Vailaya. Shape-based retrieval: a case study with trademark image databases. *Pattern Recognition*, 31(9):1369–1390, September 1998.

[56] S. Jeannin. Mpeg-7 visual part of experimental model version 2.0. Technical Report ISO/IEC JTC1/SC29/WG11/N2822, July 1999.

[57] H. Jiang, C. W. Ngo, and H. K. Tan. Gestalt-based feature similarity measure in trademark database. *Pattern Recognition*, 39(5):988–1001, May 2006.

[58] M. Kherfi and D. Ziou. Relevance feedback for cbir: a new approach based on probabilistic feature weighting with positive and negative examples. *IEEE Transactions on Image Processing*, 15(4):1017–1030, 2006.

[59] A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.

[60] H. K. Kim and J. D. Kim. Region-based shape descriptor invariant to rotation, scale and translation. *Signal Processing: Image Communication*, 16:87–93, 2000.

[61] W. Y. Kim and Y. S. Kim. A region-based shape descriptor using zernike moments. *Signal Processing: Image Communication*, 16:95–102, 2000.

[62] Y. S. Kim and W. Y. Kim. Content-based trademark retrieval system using a visually salient feature. *Image and Vision Computing*, 16(12-13):931–939, August 1998.

[63] C. P. Lam, J. K. Wu, and B. Mehtre. Star – a system for trademark archival and retrieval. In *the 2nd Asian Conference on Computer Vision (ACCV 1995)*, volume 3, Singapore, 214-217 1995.

[64] S. Lambert, E. de Leau, and L. Vuurpijl. Using pen-based outlines for object-based annotation and image-based queries. In *3rd International Conference on Visual Information and Information Systems*, pages 585–592, Amsterdam, The Netherlands, June 1999.

[65] L. J. Latecki and R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.

[66] L. J. Latecki, R. Lakamper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 424–429, 2000.

[67] L. J. Latecki, R. Lakamper, and D. Wolter. Shape similarity and visual parts. In *International Conference on Discrete Geometry for Computer Imagery*, pages 34–51, November 2003.

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

[69] H. Ling and D. Jacobs. Using the inner distance for classification of articulated shapes. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 719–726, San Diego, CA, USA, 20-26 June 2005.

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

[71] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.

[72] A. Markman and D. Gentner. Structure mapping in the comparison process. *American Journal of Psychology*, 113(4):501–38, 2000.

[73] S. Marshall. Review of shape coding techniques. *Image and Vision Computing*, 7(4):281–294, 1989.

[74] J. M. Martinez. Mpeg-7 overview (version 9). Technical Report ISO/IEC JTC1/SC29/WG11/N5525, International Organisation for Standardisation, Coding of Moving Pictures and Audio, March 2003.

[75] E. Milios and E.G.M. Petrakis. Shape retrieval based on dynamic programming. *IEEE Transactions on Image Processing*, 9(1):141–147, 2000.

[76] F. Mokhtarian. Silhouette-based isolated object recognition through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):539–544, 1995.

[77] F. Mokhtarian and M. Z. Bober. *Curvature Scale Space Representation: Theory, Applications, and Mpeg-7 Standardization*. Kluwer Academic Publishers, August 2003.

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

[79] F. Mokhtarian and A. Mackworth. A theory of multi-scale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992.

[80] P. Monasse and F. Guichard. Fast computation of a contrast-invariant image representation. *IEEE Transactions on Image Processing*, 9(5):860–872, 2000.

[81] P. Monasse and F. Guichard. Scale-space from a level lines tree. *Journal of Visual Communication and Image Representation*, 11(2):224–236, 2000.

[82] H. Muller, W. Muller, D. M. Squire, S. Marchand-Maillet, and T. Pun. Performance evaluation in content-based image retrieval: overview and proposals. *Pattern Recogn. Lett.*, 22(5):593–601, 2001.

[83] J. Munkres. Algorithms for assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[84] The MPEG Home Page. *www.chiariglione.org/mpeg/index.htm.*

[85] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[86] M. Pelillo. Relaxation labeling networks for the maximum clique problem. *Journal of Artificial Neural Networks*, 2(4):313–328, 1995.

[87] M. Pelillo. A unifying framework for relational structure matching. In *Proceedings of the 14th International Conference on Pattern Recognition*, volume 2, pages 1316–1319, August 1998.

[88] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1120, November 1999.

[89] Hsiao-Lin Peng and Shu-Yuan Chen. Trademark shape recognition using closed contours. *Pattern Recognition Letters*, 18(8):791–803, August 1997.

[90] A. Pentland, R. Picard, and S. Sclaroff. Photobook: content-based manipulation of image databases. *International Journal of Comput. Vision*, 18(3):233–254, 1996.

[91] E. G. M. Petrakis. Design and evaluation of spatial similarity approaches for image retrieval. *Image and Vision Computing*, 20(1):59–76, January 2002.

[92] E. G. M. Petrakis, A. Diplaros, and E. Milios. Matching and retrieval of distorted and occluded shapes using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1501–1516, November 2002.

[93] E. G. M. Petrakis and C. Faloutsos. Similarity searching in medical image databases. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):435–447, May/June 1997.

[94] E. G. M. Petrakis, C. Faloutsos, and K. Lin. Imagemap: An image indexing method based on spatial similarity. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):979–987, September/October 2002.

[95] W. K. Pratt. *Digital Image Processing*. John Wiley and sons Inc, second edition, 1991.

[96] C. Ratanamahatana and E. Keogh. Three myths about dynamic time warping data mining. In *International Conference on Data Mining (SDM '05)*, pages 506–510, April 2005.

[97] C. J. Rijsbergen. *Information Retrieval*. London: Butterworth, 1979.

[98] K. Roh and I. Kweon. 2-d object recognition using invariant contour descriptor and projective refinement. *Pattern Recognition*, 31(4):441–445, 1998.

[99] A. Rosenfeld. Adjacency in digital pictures. *Information and Control*, 26, 1974.

[100] G. Rote. Computing the minimum hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, 38(3):123–127, 1991.

[101] I. El Rube, M. Ahmed, and M. Kamel. Affine invariant multiscale wavelet-based shape matching algorithm. In *1st Canadian Conference on Computer and Robot Vision (CRV)*, pages 217–224, London, Ontario, Canada, 2004.

[102] I. El Rube, N. Alajlan, M. S. Kamel, M. Ahmed, , and G. Freeman. Mtar: a robust 2d shape representation. *International Journal of Image and Graphics (IJIG)*, 6(3):421–443, July 2006.

[103] I. El Rube, N. Alajlan, M. S. Kamel, M. Ahmed, and G. Freeman. Efficient multiscale shape-based representation and retrieval. In *2nd International Conference on Image Analysis and Recognition (ICIAR 2005)*, volume 3656, pages 415–422, Toronto, Canada, September 2005.

[104] I. El Rube, N. Alajlan, M. S. Kamel, M. Ahmed, and G. Freeman. Robust multiscale triangle-area representation for 2d shapes. In *IEEE International Conference on Image Processing (ICIP 2005)*, pages 545–548, Genoa, Italy, September 2005.

[105] Y. Rubner. *Perceptual Metrics For Image Database Navigation*. PhD thesis, Stanford University, 1999.

[106] H. Sakoe and S.Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26:43–49, 1978.

[107] P. Salembier and L Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing*, 9(4):561–576, 2000.

[108] R. J. Schalkoff. *Pattern recognition: statistical, structural and neural approaches*. John Wiley & Sons, Inc., New York, NY, USA, 1991.

[109] T. Sebastian, P. Klein, and B. Kimia. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–124, 2003.

[110] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571, 2004.

[111] M. Seul, L. O'Gorman, and M. J. Sammon. *Practical algorithms for image analysis: description, examples, and code*. Cambridge University Press, New York, NY, USA, 2000.

[112] D. G. Shen, H. H. S. Ip, and E. K. Teoh. Affine invariant detection of perceptually parallel 3d planar curves. *Pattern Recognition*, 33(11):1909–1918, 2000.

[113] D. G. Shen, W. Wong, and H. H. S. Ip. Affine invariant image retrieval by correspondence matching of shapes. *Image and Vision Computing*, 17(7):489–499, 1999.

[114] A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker. Indexing using a spectral encoding of topological structure. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2491–2497, 1999.

[115] J. Sklansky. Measuring concavity on a rectangular mosaic. *IEEE Transactions on Computers*, 21(12):1355–1364, 1972.

[116] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[117] A. Soffer and H. Samet. Using negative shape features for logo similarity matching. In *Proceedings of the 14th International Conference on Pattern Recognition*, volume 1, pages 571–573, Australia, August 1998.

[118] M. Sonka, V. Hiavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. PWS Publishing, 1999.

[119] A. Srivastava, S. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):590–602, April 2005.

[120] K. Thorisson. Simulated perceptual grouping: An application to human computer interaction. In *Proceedings of the 16th Annual Conference of Cognitive Science Society*, pages 876–881, Atlanta, GA, August 1994.

[121] S. Tirthapura, D. Sharvit, P. Klein, and B. B. Kimia. Indexing based on edit-distance matching of shape graphs. In *SPIE International Symposium on Voice, Video, and Data Communications*, pages 25–36, 1998.

[122] A. Torsello, D. Hidovic, and M. Pelillo. Four metrics for efficiently comparing attributed trees. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, pages 467–470, August 2004.

[123] A. Torsello, D. Hidovic, and M. Pelillo. Polynomial-time metrics for attributed trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1087–1099, July 2005.

[124] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[125] USPTO. The united states patent and trademark office. *www.uspto.gov*.

[126] A. Vailaya, Y. Zong, and A. K. Jain. A hierarchical system for efficient image retrieval. In *13th International Conference on Pattern Recognition (ICPR 1996)*, pages 356–360, 1996.

[127] R. Veltkamp and M. Hagedoorn. *Principles of visual information retrieval*, chapter State-of-the-art in shape matching, pages 87–119. Springer-Verlag, London, UK, 2001.

[128] R. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical report, Institute of ICS, Utrecht University, 2000.

[129] E. M. Voorhees and D. K. Harmann, editors. *NIST Special Publication 500-242: The Seventh Text REtrieval Conference*. National Institute of Standards and Technology, 1998.

[130] K. Wang and T. Gasser. Alignment of curves by dynamic time warping. *Annals of Statistics*, 25(3):1251–1276, 1997.

[131] Yu-Ping Wang and S. L. Lee. Scale-space derived from b-splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1040–1055, October 1998.

[132] Yu-Ping Wang, S. L. Lee, and K. Toraichi. Multiscale curvature-based shape representation using b-spline wavelets. *IEEE Transactions on Image Processing*, 8(11):1586–1592, November 1999.

[133] WIPO. The world intellectual property organization. *www.wipo.int*.

[134] A. P. Witkin and J. M. Tenenbaum. *Human and machine vision*, chapter On the role of structure in vision, pages 481–543. Prentice-Hall, 1983.

[135] J. K. Wu, C. P. Lam, B. M. Mehtre, Y. J. Gao, and A. D. Narasimhalu. Content-based retrieval for trademark registration. *Multimedia Tools and Applications*, 3(3):245–267, November 1996.

[136] D. Zhang and G. Lu. Shape based image retrieval using generic fourier descriptors. *Signal Processing: Image Communication*, 17(10):825–848, 2002.

[137] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.