Shape Analysis for Image Retrieval

by

CHOI Wai-Pak

A thesis submitted for the Degree of Doctor of Philosophy
in the Department of Electronic and Information Engineering
of the Hong Kong Polytechnic University

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University

January 2003

Abstract

Abstract of thesis entitled "Shape Analysis for Image Retrieval" submitted by CHOI Wai-Pak for the degree of Doctor of Philosophy at The Hong Kong Polytechnic University in January, 2003.

Content-based image retrieval (CBIR) system is designed to help retrieve relevant images in an image database based on their image contents. This system will allow queries on large image databases based on example images, user-constructed sketches and drawings, and other graphical information. Different image features, or descriptions, may have different significance and effectiveness in the interpretation and representation of images in different applications. The Moving Picture Experts Group (MPEG) of the International Standards Organization (ISO) initiated the MPEG-7 standard, which provides standardized core technologies that allow for the description of audiovisual data content in multimedia environments. The most challenging technical issues for a CBIR system are the effectiveness and efficiency of feature extraction and recognition algorithms for content-based image retrieval.

The objectives of this thesis are to investigate and develop efficient techniques for shape feature extraction, and to construct a content-based image retrieval system. An introduction to the general concept of image retrieval will be given in this thesis, and the recent development of the MPEG-7 standard will be described. Existing content-based image retrieval systems, and the feature extraction and recognition

techniques based on color, texture, shape and motion will be reviewed. Furthermore, more efficient and effective features will be proposed so that a reliable and practical retrieval system becomes possible. Shape descriptors, which are high level descriptions, will be emphasized in this research work.

In this research, the content-based image retrieval system developed consists of three major parts: boundary extraction, feature extraction and recognition. The first part is based on an active contour model for representing image contours. We have proposed an efficient active contour model which can represent highly irregular boundaries. The contour points can be used to form other shape descriptors such as chain code, curvature scale-space representation, skeleton, etc. After extracting the boundaries, the second part is skeletonization which is an important process that can provide a compact shape representation. We have proposed a fast, efficient and accurate skeletonization method for the extraction of a well-connected Euclidean skeleton based on the boundary information. The skeleton feature can be used as a shape descriptor, which can represent the shape more compactly, and consists of spatial and structural information. In the third part, we have proposed a robust and efficient histogram representation scheme for shape retrieval, which is based on the normalized maximal disks used to represent the shape of an object. The maximal disks are extracted by means of the fast skeletonization technique with a pruning algorithm. The logarithm of the radii of the normalized maximal disks is used to construct a histogram to represent the shape. The proposed representation scheme outperforms the other methods under affine transformation, different distortions and noise levels. Hence, these three major parts are integrated to form a complete system for content-based image retrieval. We have also devised a contour/region-based

matching algorithm has been used for retrieving relevant images containing similar shapes from a database. In the algorithm, Hausdorff distance is used to measure the similarity of two point sets. We have devised a robust line-feature-based approach for model-based recognition based on this distance measure. The proposed algorithm can achieve a good performance level in matching, even in a noisy environment or with the existence of occlusion, and can be used as a similarity measure for image retrieval.

Acknowledgements

I would like to express sincere gratitude to various bodies from The Hong Kong Polytechnic University, where I have the opportunity to work. I would like to thank my chief supervisor, Dr. K. M. Lam, and my co-supervisor, Prof. W. C. Siu, for guiding and encouraging me from the outset of my degree studies. Not only their professional advice concerning my study, but also their attitudes toward life have been precious knowledge to me. They also gave me many invaluable ideas and suggestions in writing my thesis.

I am greatly indebted to my colleagues, Mr. W. H. Wong, Mr. K. W. Wong, Mr. K. H. Lin, Mr. C. M. Lai, K. W. Sze, Mr. K. K. Yiu, Mr. K. C. Hui, Mr. W. L. Hui, Ms. K. L. Lau, Mr. K. W. Kwok, Dr. T. C. Hsung, Dr. Y. L. Chan, Dr. Bonnie Law, and Dr. Korris Chung. Their expert knowledge and friendly encouragements have helped me overcome many difficulties in my study. I would also like to thank all members of the Centre for Multimedia Signal Processing, past and present. The countless discussions I had with them have proved to be both fruitful and inspiring. Especially, I would like to thank my parents and my fiancée, Jessica Cheng, who always give me complete support in my study.

It is also my pleasure to acknowledge the Research Office of the Hong Kong Polytechnic University for providing financial support to me over the last four years through the Research Grants Council of the Hong Kong Special Administrative Region (Project No. G-V596), which made possible this research by granting me a studentship over the course of my degree program.

Finally, I would also like to take this opportunity to thank the Centre For Multimedia Signal Processing of the Department of Electronic and Information Engineering.

STATEMENT OF ORIGINALITY

The following contributions reported in this thesis are claimed to be original.

1. A new efficient active contour model for representing a highly irregular boundary. (Chapter 3, Section 3.3.1 and 3.3.2).

   This is an efficient algorithm for active contour model, which can represent highly irregular boundaries. The algorithm includes an adaptive force along the contour, and adjusts the number of points for the snake according to the desired boundary. The adaptive force is introduced at a point of the snake whenever the image forces in its neighborhood are smaller than a certain threshold. In order to achieve an accurate representation of a boundary, the distances between adjacent points of the snake are kept close to a constant. Two processes, namely deletion and insertion, are introduced to change the number of points for the snake during the energy minimization process.

2. A better terminating Criterion (Chapter 3, Section 3.3.3).

   A new terminating criterion called contour area criterion (CA-criterion) has been proposed, which makes use of the normalized total area to determine the convergence of an iteration process. The CA-criterion exhibits a smoother and faster convergence than that of the contour length criterion (CL-criterion). The CA-criterion also has better stability in convergence, as its fluctuation between successive iterations is much smaller than the CL-criterion without using any averaging. In addition, the calculation of area is simple and fast, while the CL-criterion requires computing square roots to obtain the length of each segment.

3. Criteria for extracting multiple objects (Chapter 3, Section 3.4).

   The adaptive snake model can be used to represent the contour of a single object. In order to extract multiple objects, the snake model is modified such that it can determine the corresponding contour of each object. We have proposed a criterion to determine critical points where the snake is split and

connected for multiple object representation. The criterion includes the processes of splitting and connecting. A critical point is defined as the end points of a contour segment which are adjacent to non-contour segments. During the iteration process, each of the critical points will be checked in sequence. Pairs of the critical points will be marked as connected points where the splitting and connecting operations will be performed to form two contours. Having searched the pairs of connected points, which are close to each other, the splitting and connecting operations are then performed.

4. Criteria for validating of the snake (Chapter 3, Section 3.4.3).

   With the process of splitting and connecting process, new snake may be generated. Some of the snakes will grow until they touch the edges in the image. The validity of a new snake can be identified by calculating its enclosed area. If the points of a snake collapse and meet each other, the area enclosed by the snake will be zero. If the snake grows in a reverse direction, its computed area will be negative. This signifies that an invalid snake has been formed. In addition, a snake is also considered to be invalid if its number of points is less than a certain value.

5. The criterion for a skeleton point (Chapter 4, Section 4.3).

   In real applications, the contour points and the skeleton points must be located at the pixel grids; this induces a lot of discrete problems. We have defined an ideal skeleton which has non-zero width, and all the pixels passed through by this ideal skeleton will be considered to be a skeleton pixels. Based on this ideal skeleton, we have devised a connectivity criterion for the square grid to determine whether a given pixel is a skeleton point independently. A connected Euclidean skeleton with a single pixel width can be generated without requiring a linking algorithm or iteration process.

6. A skeletonization algorithm (Chapter 4, Section 4.4).

   To determine whether a pixel point is a skeleton point in a discrete image, we have proposed a skeletonization algorithm based on the connectivity criterion and the signed sequential Euclidean distance (8SSED) map. The nearest contour points for each of the 8 neighboring points can be obtained by using

the 8SSED map. The nearest contour points of the pixel under consideration and one of its 8 neighbors then form a point pair. If any one of the point pairs satisfies the connectivity criterion, the pixel can be declared a skeleton point.

7. Maximal disk-based Histogram for Shape Retrieval (Chapter 5, Section 5.2).

   We have proposed a robust and efficient representation scheme for shape retrieval, which is based on the normalized maximal disks used to represent the shape of an object. The maximal disks are extracted by means of a fast skeletonization technique with a pruning algorithm. The logarithm of the radii of the normalized maximal disks is used to construct a histogram to represent the shape. Our proposed representation scheme outperforms the other methods, including moment invariants, Zernike moments, and curvature scale-space, under affine transformation, different distortions and noise levels.

8. Robust feature for object matching (Chapter 6, Section 6.3.3).

   The Hausdorff distance can be used to measure the similarity of two point sets. We have proposed a robust line-feature approach for model-based recognition based on the Hausdorff distance. The robust features are extracted based on the line segments formed between each point and its corresponding farthest point in a point set. These features are insensitive to noise and can find the rotation and scale between two image point sets accurately and reliably, so 2D-2D matching algorithm can be adopted.

Table of Contents

List of Figures and Tables

Author's Publications

The following technical papers have been published or submitted for publication based on the result generated from this work.

International Journal papers:

1. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "An adaptive active contour model for highly irregular boundaries", *Pattern Recognition*, vol. 34, no. 2, pp. 323-331, 2001.

2. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "Extraction of an Euclidean Skeleton based on a connectivity Criterion", *Pattern Recognition*, vol. 36, no. 3, pp. 721-729, 2003.

3. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "An Efficient and Accurate Algorithm for extracting a Skeleton based on Maximal Disks", submitted to *Journal of Electronic Imaging*.

4. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "A Robust Line-feature-based Hausdorff distance for Object Matching and its application", submitted to *IEEE Transactions on Image Processing*.

5. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "Maximal disk-based Histogram for Shape Retrieval", submitted to *IEEE Transactions on Image Processing*.

International Conference/Symposium papers:

6. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "An Adaptive Active Contour Model for a Highly Irregular Boundary", *Proceedings. 1999 International Symposium on Signal Processing and Intelligent System*, pp. 598-601, 1999.

7. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "A New Adaptive Active Contour Model for Highly Irregular Boundaries", *Proceedings. The First IEEE Pacific-Rim Conference on Multimedia*, pp. 384-387, 2000.

8. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "An Efficient and Accurate Algorithm for extracting a Skeleton", *Proceedings. 15th International Conference on Pattern Recognition*, vol. 3, pp. 750-753, 2000.

9. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "A Robust Line-Feature-Based Hausdorff distance for Shape Matching", *Proceedings. The Second IEEE Pacific-Rim Conference on Multimedia*, pp.764-771, 2001.

10. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "Robust Hausdorff distance for Shape Matching", *Proc. SPIE: Visual Communications and Image Processing 2002*, vol. 4671, pp. 793-804, 2002.

11. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "An Efficient Algorithm for the Extraction of a Euclidean Skeleton", *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 3241 –3244, 2002.

12. Wai-Pak Choi, Kin-Man Lam and Wan-Chi Siu, "Maximal disk based Histogram for Shape Retrieval", submitted to *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing.*

Chapter 1

Introduction

The objectives of this chapter are to give the motivation for content-based image retrieval, an introduction to the general concept of image retrieval, as well as the recent development of the MPEG-7 standard. An overview of the techniques for content-based image retrieval proposed and developed in this thesis will be presented. Finally, the organization of the thesis is given at the end of this chapter.

1.1 Motivation for Content-Based Retrieval from Image Databases

Many organizations have large image and video collections in digital format, available for on-line access. With the development of digital photography, on-line collections of images are growing larger and more common, and tools are needed to efficiently manage, organize, and navigate through them. In order to organize these collections into categories, an effective and efficient way of image feature extraction and recognition is imperative for real-time browsing and retrieval. A number of content-based image retrieval systems have been developed and research on content-based retrieval has emerged as an important area in computer vision and multimedia computing. In October 1996, the Moving Picture Experts Group (MPEG) of the International Standards Organization (ISO) initiated the MPEG-7 standard, which

provides standardized core technologies allowing description of audiovisual data content in multimedia environments. The descriptors, which best fulfil the requirements as defined by the experts, are selected in MPEG-7 standard. Effective and efficient feature extraction and recognition algorithms for content-based image retrieval is a challenging research topic.

1.2 Introduction to Content Based Image Retrieval System

Content-based image retrieval (CBIR) system is designed to help retrieve relevant images in an image database based on their image contents. This system will allow queries on large image databases based on example images, user-constructed sketches and drawings, and other graphical information. Different image features, or keys, may have different significance and effectiveness in the interpretation and representation of images in different applications. In other words, the features to be extracted from images should be dynamically selectable. These image content features can be combined with each other and with text-based descriptions to form a sophisticated CBIR system. The selected features from images are used to build the index for accessing image databases. Relevance feedback methods are also useful as the information to the user supplies to the system in an attempt to "guess" what are her intentions, thus making it easier to find what she wants. Relevance feedback directed queries may help to protect the user from unwanted technicalities and to find more rapidly certain categories of images in large databases. With the wide applications of multimedia systems and the ever-increasing rate of generating different types of images from civilian satellites, biomedical imaging, human face

recognition, fingerprinting, etc., it is therefore required that the CBIR system can use the information from these image repositories effectively and efficiently.

Since there are different features for image retrieval, the properties such as color percentages, color layout, and textures occurring in the images are often considered in the system. Such queries use the visual properties of images, so the descriptors such as color and texture can be matched without describing them in words. Content-based queries are also combined with text and keyword predicates to get powerful retrieval methods for image and multimedia databases. However, the descriptors such as shape and motion, which are high-level descriptions, are a challenging area for image retrieval. In this thesis, we will focus on the development of the shape descriptors which includes the feature extraction and recognition algorithms.

1.3 Our Methods for Content Based Image Retrieval System

The objectives of this research work are to investigate and develop efficient techniques for feature extraction and to construct a content-based image retrieval system. We will review existing content-based image retrieval systems, and the feature extraction and recognition techniques based on color, texture and shape, etc. Most of the content-based image retrieval systems allow users to choose different features for retrieval. These features may not be effective and efficient for real-time retrieval systems because their derivations are usually computationally expensive. These features are usually pre-computed and stored with the corresponding images in a database. In our research, different features for image retrieval have been studied, and their corresponding efficiency, effectiveness, and required computation have also been investigated. Furthermore, more efficient and effective features will be proposed

so that a reliable and practical retrieval system becomes possible. Content-based image retrieval systems may allow users to choose color, texture, shape, and other features as the query input. Due to the different signatures of the features, a weighting factor can be associated with each of the chosen features. The features can be selected and their corresponding weighting factors can be assigned by the user.

The content-based image retrieval system developed in this thesis project consists of three major parts: boundary extraction, feature extraction and recognition. The first part is based on an active contour model for representing image contours. In this part, we have proposed an efficient active contour model which can represent highly irregular boundaries. The contour points can be used to form other shape descriptors such as chain code, curvature scale-space representation, skeleton, etc. After extracting the boundaries, the second part is skeletonization which is an important process for a compact shape representation. We have proposed fast, efficient and accurate skeletonization methods for the extraction of a well-connected Euclidean skeleton based on the boundary information. The skeleton feature can be used as a shape descriptor which can represent the shape more compactly, and consists of spatial and structure information. In the third part, the contour/region-based matching algorithm has been used for retrieving relevant images containing similar shapes from the database. Since the Hausdorff distance can be used to measure the similarity of two point sets, we have proposed a robust line-feature-based approach based on the Hausdorff distance for model-based recognition, which can achieve a good performance level in matching, even in a noisy environment or with the existence of occlusion. The proposed method can be used as a similarity measure for image retrieval. Hence, these three major parts are integrated to form a complete system for content-based image retrieval.

1.4 Organization of the Thesis

The rest of this thesis will firstly give an overview of existing content-based image retrieval systems and the related techniques. Then, we will present an efficient boundary extraction technique based on the active contour model and efficient skeletonization algorithms. These algorithms can be used to extract shape features of an object. Having extracted the shape information, a new point-set matching technique based on Hausdorff distance will be presented in this thesis. Finally, conclusions and possible future work will be given in the last chapter. The detailed contents of the following chapters are briefly given in the following.

In Chapter 2, we will describe some existing CBIR systems and the image databases used for content based image retrieval, as well as the recent development of MPEG-7. We will also introduce the visual descriptors recommended in MPEG-7. Then, we will review some techniques which can be used for visual feature extraction and recognition.

In Chapter 3, the original snake model, which is used for representing image contours, will be described. Then, our proposed adaptive snake model will be presented, which can be used to extract the boundary of a highly irregular object. We apply an adaptive force along the contour, and adjust the number of points for the snake according to the desired boundary in our algorithm. A better stopping criterion based on the area of a closed contour will also be described.

In Chapter 4, we will introduce efficient skeletonization algorithms based on the maximal disk and the definition of medial axis transform. The discrete problems on the extraction of a skeleton will also be described. Then, we will establish a concept of the width of a skeleton and present our proposed connectivity criterion for

extracting a Euclidean skeleton which can be used to determine whether a given pixel is a skeleton point independently. The implementation of the skeletonization algorithm will also be illustrated and summarized.

In Chapter 5, we will give a detailed description of our proposed shape descriptor based on histogram representation. Then, experimental results will be presented and evaluated, where the performance of our proposed scheme is compared to other shape representation schemes, namely moment invariants, Zernike moments, and curvature scale-space. Finally, the robustness and accuracy of our proposed shape descriptor will also be evaluated.

In Chapter 6, we will give an introduction to the Hausdorff distances and describe an M-estimation Hausdorff distance, which can be used to determine the relative position between two shapes efficiently even if noise and occlusion exist along the two object boundaries. Then, we will present our new matching algorithm, which can accurately find the scale and orientation of a point set relative to one another. A robust line-segment Hausdorff distance will also be described for matching two point sets. Experimental results based on simulated data and a database of trademarks will be presented and evaluated.

In Chapter 7, a summary of the work completed for this thesis and a conclusion will be provided. We will also give suggestions on future directions for developing a better shape descriptor, a better similarity measure, as well as a further improved content based retrieval system.

Chapter 2

Overview of Content Based Retrieval System

2.1 Introduction

Content Based Image Retrieval (CBIR) Systems [1-6] help users retrieve relevant images based on their contents from an image database. Previous approaches to content based retrieval have taken two directions. In the first, image contents are modeled as a set of attributes extracted manually and managed within the framework of conventional database-management systems. Attribute-based representation of images entails a high level of image abstraction. The second approach depends on an integrated feature-extraction or object-recognition subsystem to overcome the limitations of attribute-based retrieval. This subsystem automates the feature-extraction and object-recognition task that occurs when the image is inserted into the database. However, automated approaches to object recognition are computationally expensive, difficult, and tend to be domain specific. This approach is advanced primarily by image-interpretation researchers. Recent CBIR research recognizes the need for synergy between these two approaches.

In this chapter, we will see some existing CBIR systems and image databases used for content based image retrieval and look at the recent development of MPEG-7

[6, 7] which consists of the descriptions of the content based image retrieval. Then, we will introduce the visual descriptors appearing in the MPEG-7 standard. Finally, we will review some methods which can be used for visual feature extraction and recognition.

2.1.1 Existing Content Based Image Retrieval Systems

Approaches to content based image retrieval differ in terms of which image features are extracted, the level of abstraction manifested in the features, and the degree of desired domain independence. There are two major categories of features: primitive and logical. Primitive, or low-level, image features such as an object's centroid and boundaries can be extracted automatically or semi-automatically. Logical features are abstract representations of images at various levels of detail. Regardless of which approach is used, generic query classes facilitate CBIR through retrieving by color, texture, sketch, shape, volume, spatial constraints, browsing, objective attributes, subjective attributes, motion, text, and domain concepts.

2.1.1.1 IBM's Query By Image Content (QBIC)

The Query By Image Content (QBIC) system [8], which can make queries of large image databases based on visual image content, has been developed by IBM. It is based on an *a priori* feature extraction approach as a comprehensive, operational CBIR system. It allows queries on large image and video databases based on example images, user-constructed sketches and drawings, selected color and texture patterns, camera and object motion, and other graphical information. Two key properties of QBIC are its use of image and video content-computable properties of color, texture, shape, and motion of images, videos, and their objects in the queries, and its graphical means. It has two main components: database population (the process of creating an

image database) and database query. During the population, image and videos are processed to extract features describing their content such as colors, textures, shapes, and camera and object motion, and the features are stored in a database. During the query, the user composes a query graphically. Features are generated from the graphical query and then input to a matching engine that finds images or videos from the database with similar features.

2.1.1.2 Advanced Multimedia Oriented Retrieval Engine (AMORE)

AMORE is a World-Wide Web image search engine. It allows the retrieval of images from the Web based on either keywords or the specification of a similar image or a combination of the two. In addition, an image on a particular web page can be specified as the search query. AMORE also allows users to control how they would like the search to be conducted. They can indicate how important color similarity should be in the results, how important shape similarity is, and whether the results should be visually or semantically similar to the search query.

2.1.1.3 IKONA

IKONA is a new architecture for building CBIR software systems, based on a client-server architecture, designed and implemented at the IMEDIA Project research team, INRIA Rocquencourt, France. The server part is written in C++ (for reasons of speed) and the client part is written in Java, so and it should normally run on any computer architecture that supports Java Runtime Environment (JRE). The client and the server communicate with each other through a protocol, which is a set of commands the server understands and a set of answers it can return to the client.

IKONA can be used to search all images in all databases and returns a list of the most visually similar images in response to the query image. It allows region-based queries and has a hybrid text-image retrieval mode. In the region based mode, the user

can select a part of an image and the system will search images or parts of images that are visually similar to the selected part.

2.1.1.4 NETRA - Image Search Using Color and Texture

NETRA is a prototype image retrieval system that is currently being developed in the UCSB Alexandria Digital Library (ADL) project. NETRA uses color, texture, shape and spatial location information in segmented image regions to search and retrieve images with similar regions from the database. A distinguishing aspect of this system is its incorporation of a robust automated image segmentation algorithm that allows object or region based search. Image segmentation significantly improves the quality of image retrieval when images contain multiple complex objects.

2.1.1.5 NETRA 2 - A Region-Based Image Retrieval System

NETRA 2 is a prototype image retrieval system that allows users to search and retrieve images in the database based on color information. One of the distinctive aspects of the system is that it allows users to localize the information and select image regions of interest as queries, thus providing a more powerful search tool than other retrieval systems that use global image features. Compared to the original NETRA system, this new version of NETRA emphasizes the latest work on color image segmentation and local color features.

2.1.1.6 FERET Database and System

An automatic face system for recognition and interactive search has been developed using the FERET face database. The system consists of a two-stage object detection and alignment stage, a contrast normalization stage, and a Karhunen-Loeve (*eigenspace*) based feature extraction stage whose output is used for both recognition and coding. This leads to a compact representation of the face that can be used for

both recognition as well as image compression. The system has been successfully tested on a database of nearly 2000 facial photographs from the ARPA FERET database with a detection rate of 97%. Recognition rates as high as 99% have been obtained on a subset of the FERET database consisting of 2 frontal views of 155 individuals.

2.1.2 Existing Image Databases

2.1.2.1 Vision Texture Database

The Vision Texture Database is a collection of texture imagery for the computer vision community. The database was created with the intention of providing a large set of high quality textures for computer vision applications. Unlike other texture collections, the images in VisTex do not conform to rigid frontal plane perspectives and studio lighting conditions. Instead, the images that make up VisTex are taken from examples of common ("real-world") photography and video. The goal of VisTex is to provide texture images that are representative of real world conditions. While VisTex can serve as a replacement for traditional texture collections, it includes examples of many non-traditional textures.

2.1.2.2 Shape Queries Using Image Databases (SQUID)

A system for Shape Queries Using Image Databases (SQUID) is the first image database retrieval system on the Internet which allows users to submit shapes as query objects. There are about 1,100 images of marine creatures in the database. Each image shows one distinct species on a uniform background. Every image is processed to recover the boundary contour, which is then represented by three global shape parameters and the maxima of the curvature zero-crossing contours in its Curvature Scale Space image.

2.2 MPEG-7 Recent Development

The Moving Picture Experts Group (MPEG) of the International Standards Organization (ISO) initiated work on MPEG-7 [9], its fourth standard, in 1997. MPEG-7, formally named "Multimedia Content Description standard", is quite different to earlier MPEG standards such as MPEG-1, MPEG-2, and MPEG-4 which address video coding. MPEG-7 is a standard for describing the multimedia content data that supports standardized Descriptors and Description Schemes for audio and video, as well as integrated multimedia content. Also standardized is a Description Definition Language that allows new Descriptors and Description Schemes to be defined. Its objective is to provide standardized core technologies allowing description of audiovisual data content in multimedia environments. Recently, the MPEG-7 standard [9] has been finalized on December 2001 and it consists of eight parts; Systems [10], Description Definition Language [11], Visual [12-15], Audio [16-17], Multimedia Descriptor Schemes [18], Reference Software, Conformance, and Extraction and use of descriptors. MPEG-7 Visual Description Tools included in the standard consist of basic structures and descriptors that cover the following basic visual features: color, texture, shape, motion, localization, and face recognition. Each category consists of elementary and sophisticated descriptors.

2.3 MPEG-7 Visual Descriptors

In order to design efficient descriptors for these features for applications such as similarity retrieval, considerable work has been done. The visual descriptors have been classified into four types: color descriptors, texture descriptors, shape descriptors and motion descriptors. Color and texture descriptors [13] are low-level descriptors while shape descriptors [14] and motion descriptors [15] are high-level descriptors.

Color and texture are among the more expressive of the visual features. Shape descriptors express the shape properties of an object's outline or the pixel distribution within the 2-D/3-D object's region. Much research has also been devoted to the properties of shape descriptors and the algorithms for recognition, retrieval, and indexing. Motion descriptors cover the range of complexity and functionality in a broad range of applications. Descriptors for human face recognition are also included, which are high-level descriptors used to retrieve face images by matching a query face image. The details of the above descriptors are given below.

2.3.1 Color Descriptors

In content based retrieval research, color is perhaps the most dominant and distinguishing visual feature. Color histograms are one of the most frequently used color descriptors that capture global color distribution in an image. Since color histograms result in large feature vectors, indexing is difficult and has high search and retrieval costs. Although spatial information is not preserved in a color histogram, several of the recently proposed color descriptors have been proposed to incorporate spatial information to varying degrees. They include the compact color moments, binary color sets, color coherence vectors, and color correlograms. The Color Descriptors can be classified into seven types: Color space, Color Quantization, Dominant Colors, Scalable Color, Color Layout, Color-Structure, and GoF/GoP Color as shown in the following.

2.3.1.1 Color space

This descriptor defines the color space to be used in a certain application, usually in combination with other descriptors such as dominant color or color histogram. In the current description, the following color spaces are supported: RGB, YCrCb, HSV,

HMMD, Monochrome, and Linear transformation matrix with reference to RGB. Most of the visual data available in computer and video applications uses RGB and YCrCb while HSV and HMMD provide better results in search and retrieval applications, since they are closely related to the human perception of color. The other color spaces based on linear transformation matrix with respect to RGB are also supported. The monochrome space is a complement which allows the more universal use in the context with other descriptors, like histogram of monochrome images.

2.3.1.2 Color Quantization

This descriptor defines a uniform quantization of a color space. The number of bins which the quantizer produces is configurable, such that great flexibility is provided for a wide range of applications. For a meaningful application in the context of MPEG-7, this descriptor has to be combined with dominant color descriptors to express the meaning of the values of dominant colors.

2.3.1.3 Dominant Color(s)

This color descriptor is most suitable for representing local features such as an object or image region where a small number of colors are enough to characterize the color information in the region of interest. Whole images are also applicable, for example, flag images or color trademark images. Color quantization is used to extract a small number of representing colors in each region or image. The percentage of each quantized color in the region is calculated correspondingly. A spatial coherency on the entire descriptor is also defined, and is used in similarity retrieval.

2.3.1.4 Scalable Color

The Scalable Color Descriptor is a Color Histogram in HSV color space, which is encoded by a Haar transform. Its binary representation is scalable in terms of bin

numbers and bit representation accuracy over a broad range of data rates. The Scalable Color Descriptor is useful for image-to-image matching and retrieval based on color features. Retrieval accuracy increases with the number of bits used in the representation.

2.3.1.5 Color Layout

This descriptor effectively represents the spatial distribution of color of visual signals in a very compact form. This compactness allows visual signal matching functionality with high retrieval efficiency at very small computational costs. It provides image-to-image matching as well as ultra high-speed sequence-to-sequence matching, which requires so many repetitions of similarity calculations. It also provides a very friendly user interface using hand-written sketch queries since this descriptor captures the layout information of color features. The sketch queries are not supported in other color descriptors.

2.3.1.6 Color-Structure Descriptor

The Color Structure Descriptor is a color feature descriptor that captures both color content and information about the structure of this content. Its main functionality is image-to-image matching and its intended use is for still-image retrieval, where an image may consist of either a single rectangular frame or arbitrarily shaped, possibly disconnected regions. The extraction method embeds color structure information into the descriptor by taking into account all colors in a structuring element of 8x8 pixels that slides over the image, instead of considering each pixel separately. Unlike the color histogram, this descriptor can distinguish between two images in which a given color is present in identical amounts but where the structure of the groups of pixels having that color is different in the two images. The Color Structure descriptor also provides additional functionality and improved

similarity-based image retrieval performance for natural images compared to the ordinary color histogram.

2.3.1.7 GoF/GoP Color

The Group of Frames (GoF) or Group of Pictures (GoP) Color Descriptor extends the Scalable Color Descriptor that is defined for a still image to the color description of a video segment or a collection of still images. Two more bits are added, which allows to define how the color histogram was calculated, before the Haar transform is applied to it. The color histogram can be computed by *average, median* or *intersection.* The average histogram, which refers to averaging the counter value of each bin across all frames or pictures, is equivalent to computing the aggregate color histogram of all frames and pictures with proper normalization. The median histogram refers to computing the median of the counter value of each bin across all frames or pictures. It is more robust to round-off errors and the presence of outliers in image intensity values compared to the average histogram. The intersection histogram refers to computing the minimum of the counter value of each bin across all frames or pictures to capture the "least common" color traits of a group of images. It is different from the histogram intersection, which is a scalar measure. The same similarity/distance measures that are used to compare Scalable Color Descriptions can be employed to compare GoF/GoP Color Descriptors.

2.3.2 Texture Descriptors

Texture is a powerful low-level descriptor for image search and retrieval applications. Three texture descriptors are considered in MPEG-7. The first one is the "Homogenous Texture Descriptor" which is based on computing the local spatial-frequency statistics of texture to provide a quantitative characterization of

homogeneous texture regions for similarity retrieval. The second one is the "Edge Histogram Descriptor" which is used when the underlying region is not homogeneous in texture properties. The last one is the "Texture Browsing Descriptor" which characterizes perceptual attributes such as directionality, regularity, and coarseness of a texture. We introduce these three texture descriptors below.

2.3.2.1 Homogenous Texture Descriptors

Homogeneous texture has emerged as an important visual primitive for searching and browsing through large collections of similar looking patterns. An image can be considered as a mosaic of homogeneous textures so that these texture features associated with the regions can be used to index the image data. To support such image retrieval, an effective representation of texture is required. The Homogeneous Texture Descriptor also provides a quantitative representation that is useful for similarity retrieval. In order to extract the feature, the image is first filtered with a bank of orientation and scale tuned filters using Gabor filters. The first and the second moments of the energy in the frequency domain in the corresponding sub-bands are then used as the components of the texture descriptor. An efficient implementation using projections and 1-D filtering operations exists for feature extraction. The Homogeneous Texture Descriptor provides a precise quantitative description of a texture that can be used for accurate search and retrieval in this respect. The computation of this descriptor is based on filtering using scale and orientation selective kernels.

2.3.2.2 Edge Histogram

The Edge Histogram Descriptor represents the spatial distribution of five types of edges, namely four directional edges and one non-directional edge. Since edges play an important role for image perception, it can retrieve images with similar semantic

meaning. Thus, it primarily targets image-to-image matching, especially for natural images with non-uniform edge distribution. In this context, the image retrieval performance can be significantly improved if the edge histogram descriptor is combined with other descriptors such as the color histogram descriptor. Besides, the best retrieval performances considering this descriptor alone are obtained by using the semi-global and the global histograms generated directly from the edge histogram descriptor as well as the local ones for the matching process.

2.3.2.3 Texture Browsing

The Texture Browsing Descriptor is useful for representing homogeneous texture for browsing type applications. It provides a perceptual characterization of texture, similar to a human characterization, in terms of regularity, coarseness and directionality. The computation of this descriptor proceeds in the same way as the Homogeneous Texture Descriptor. In order to extract the feature, the image is first filtered with a bank of orientation and scale tuned filters modeled as Gabor functions. Two dominant texture orientations are identified from the filtered outputs. The filtered image projections along the dominant orientations are analyzed to determine the regularity and coarseness. The second dominant orientation and second scale feature are optional. This descriptor, combined with the Homogeneous Texture Descriptor, provides a scalable solution to representing homogeneous texture regions in images.

2.3.3 Shape Descriptors

Much research has been devoted to the properties of shape descriptors and the algorithms for recognition, retrieval, and indexing [19-21]. The shape descriptors can be divided into three types: Contour Shape Descriptors [22], Region Shape Descriptors [23], and 3D Shape Descriptors [24, 25]. A contour-based descriptor

expresses the shape properties of an object's outline while a region-based descriptor describes the pixel distribution within the 2-D object's region. Also, 2D shape descriptors such as contour shape and region shape descriptors [25, 26] can be used to represent the visual feature of a 3D object from different viewing angles. We review these three shape descriptors: contour shape, region shape, and 3D shape, below.

2.3.3.1 Contour Shape

The Contour Shape Descriptor captures characteristic shape features of an object or region based on its contour. A Curvature Scale-Space [22, 27, 28] representation, which captures perceptually meaningful features of the shape, is used. The object contour-based shape descriptor is based on the Curvature Scale Space representation of the contour. This representation has a number of important properties. First of all, this descriptor is compact which can capture characteristic features of the shape very well, enabling similarity-based retrieval and reflects properties of the perception of human visual system and offers good generalization. Also, it is robust to non-rigid motion, partial occlusion of the shape and perspective transformations, which result from the changes of the camera parameters and are common in images and video.

2.3.3.2 Region Shape

The shape of an object may consist of either a single region or a set of regions as well as some holes in the object. Since the Region Shape Descriptor makes use of all pixels constituting the shape within a frame, it can describe any shape. Such shapes may be a simple with a single connected region or complex consisting of holes in the object or several disjoint regions. The Region Shape Descriptor not only can describe such diverse shapes efficiently in a single descriptor, but is also robust to minor deformations along the boundary of the object. The descriptor is also characterized by its small size, fast extraction time and matching. The feature extraction and matching

processes are straightforward and require a low computational complexity. The descriptor is therefore suitable for tracking shapes in video data processing.

2.3.3.3 3D Shape

Considering the continuous development of multimedia technologies, virtual worlds, and augmented reality, 3D contents become a common feature of today's information systems. Although most real-world objects are 3-D, the image and video world usually deals with 2-D projections of real-world objects. Some 2D shape descriptors can be therefore used to represent the visual feature of a 3D object from different viewing angles. One of the descriptors for 3-D shapes is based on the shape spectrum [29, 30], which can be used to represent the local convexity of a 3-D surface. Also, 3D information can be represented as polygonal meshes. MPEG-4, within the Synthetic-Natural Hybrid Coding (SNHC) subgroup, considered this issue and developed technologies for efficient 3D mesh model coding. Within the framework of the MPEG-7 standard, tools for intelligent content-based access to 3D information are needed. The main MPEG-7 applications are search, retrieval, and browsing of 3D model databases. The 3D Shape Descriptor described in detail provides an intrinsic shape description of 3D mesh models. It exploits some local attributes of the 3D surface.

2.3.4 Motion Descriptors

The motion descriptors [31-33] cover the range of complexity and functionality, such as video hyperlinking based on trajectories [34, 35], refined browsing based on motion characteristics, or refinement of table of contents [34], enabling MPEG-7 to support a broad range of applications. Simple extraction, simple matching, concise expression and effective characterization of motion features of a video sequence

provide the easiest access to its temporal dimension, and are key significance in video indexing. Using motion characteristics, camera motion-estimation techniques, trajectory-matching techniques, and aggregated motion-vector histogram-based techniques form the core of past work. The performance of similarity-based video retrieval systems [33] can significantly be improved by the motion-based indexing in combination with indexing based on still-image features such as color, texture, etc. Four motion descriptors: Camera Motion, Motion Trajectory, Parametric Motion, and Motion Activity, are described below.

2.3.4.1 Camera Motion

This descriptor characterizes 3-D camera motion parameters. Based on 3-D camera motion parameter information, the parameters can be automatically extracted or generated by capture devices. The camera motion descriptor supports the following well-known basic camera operations: fixed, panning (horizontal rotation), tracking (horizontal transverse movement, also called traveling in the film industry), tilting (vertical rotation), booming (vertical transverse movement), zooming (change of the focal length), dollying (translation along the optical axis), and rolling (rotation around the optical axis). This descriptor represents the union of the building blocks which are described by their start time, the duration, the speed of the induced image motion, the fraction of time of their duration compared with a given temporal window size, and the focus-of-expansion (FOE) or the focus-of-contraction (FOC). It has the option of describing a mixture of different camera motion types. The mixture mode captures the global information about the camera motion parameters while the non-mixture mode captures the notion of pure motion type and their union within certain time interval.

2.3.4.2 Motion Trajectory

The motion trajectory of an object is a simple, high level feature, defined as the localization, in time and space, of one representative point of this object. This descriptor is essentially a list of key points along with a set of optional interpolating functions that describe the path of the object between key points, in terms of acceleration. The properties of this representation are that it is compact, scalable, independent of the spatio-temporal resolution of the content, and it directly allows a wide variety of uses, like similarity searches, or categorization by speed (fast, slow objects), behaviour (accelerating when approaching this area) or by other high level motion characteristics. This descriptor shows usefulness for content-based retrieval in object-oriented visual databases in more specific applications and allows enhancing data interactions or manipulations to adapt the object motion to any given sequence global context such as semiautomatic multimedia editing, and trajectory stretching and shifting.

2.3.4.3 Parametric Motion

Parametric motion models have been extensively used within various related image processing and analysis areas, including motion-based segmentation and estimation, global motion estimation, mosaicing and object tracking. Parametric motion models have been already used in MPEG-4, for global motion estimation and compensation and sprite generation. Within the MPEG-7 framework, motion is a highly relevant feature, related to the spatio-temporal structure of a video and concerning several MPEG-7 specific applications, such as storage and retrieval of video databases and hyperlinking purposes. Motion is also a crucial feature for some domain specific applications that have already been considered within the MPEG-7 framework, such as sign language indexation.

2.3.4.4 Motion Activity

Video content in general spans the gamut from high to low activity, therefore we need a descriptor that enables us to accurately express the activity of a given video sequence or shot, and comprehensively covers the aforementioned gamut. The activity descriptor is useful for applications such as video re-purposing, surveillance, fast browsing, dynamic video summarization, content-based querying. A human watching a video or animation sequence perceives it as being a slow sequence, fast paced sequence, action sequence, etc. The activity descriptor captures this intuitive notion of 'intensity of action' or 'pace of action' in a video segment.

2.3.5 Localization

There are two descriptors for localization: Region locator and Spatio-temporal locator. The region locator enables localization of regions within images or frames by specifying them with a brief and scalable representation of a Box or a Polygon. The spatio-temporal locator describes spatio-temporal regions in a video sequence, such as moving object regions, and provides localization functionality.

2.3.6 Others

The Face Recognition Descriptor can be used to retrieve face images which match a query face image. This descriptor represents the projection of a face vector onto a set of basis vectors which span the space of possible face vectors. The face recognition feature set is extracted from a normalized face image. This normalized face image contains 56 lines with 46 intensity values in each line. The centers of the two eyes in each face image are located on the 24th row and the 16th and 31st column for the right and left eye, respectively. This normalized image is then used to extract the one dimensional face vector which consists of the luminance pixel values from the

normalized face image arranged into a one dimensional vector using a raster scan starting at the top-left corner of the image and finishing at the bottom-right corner of the image. The Face Recognition feature set is then calculated by projecting the one dimensional face vector onto the space defined by a set of basis vectors.

2.4 Previous Works on Visual Feature Extraction and Recognition

Most of the content-based image retrieval systems allow users to choose different features for retrieval. Some techniques for the extraction and recognition which can be applied to specific features are simply divided into three categories, such as color and texture features, contour-based shape features, and region-based shape features. The details of these three types of techniques are described below.

2.4.1 Color and Texture Features

Color and texture are visual features that are immediately perceived when looking at an image. For a content-based retrieval system, the color and texture extractions are very important. It directly affects the performance of the system which retrieves the relevant image from an image database. Color and texture analysis is also an important subject of research in imaging technology for fundamental applications like image compression, object identification and recognition in images and videos, and pattern inspection and discrimination, etc. Therefore, we will show some methods used in color and texture features extraction and recognition in the following.

2.4.1.1 Edge-Flow

EdgeFlow [36] is a technique for boundary detection and image segmentation. It utilizes a predictive coding model to identify the direction of change in color and

texture at each image location at a given scale, and then constructs an edge flow field. By projecting the edge flow vectors, the boundaries can be detected at the image locations where two opposite directions of flow are encountered in the stable state. The edge energies and the corresponding probability of the edge flow are firstly computed from the image attributes, which are the intensity/color, texture and phase, to form a single edge flow field. The direction of the edge flow vector can be estimated by maximizing the sum of the corresponding probabilities in different orientations of the half plane. After this, the edge flow vector can be determined by summation of the edge energies. After the edge flow propagation and boundary detection, the disjoint boundaries are then connected to form closed contours and result in a number of image regions. This approach requires a small number of parameters for tuning and achieves acceptable quality for the segmentation of the different images including the texture images. However, the computation of the texture feature is expensive because of the extraction of a set of Gabor texture features.

2.4.1.2 Peer Group Filtering

Noise removal and image smoothing are important to many image processing applications. For color images, a common approach to remove impulse noise is by vector median filtering (VMF) but it is typically implemented uniformly across the image and tends to modify pixels that are not corrupted by noise. Other approaches such as Teager-like operator is used to first detect the outliers so that only the noisy pixels are replaced. However, the detection process is performed on each individual color component. For the case of mixed Gaussian and impulse noise, an adaptive nonlinear multivariate filtering method has been proposed, but it may blur the edges

and the details. Therefore, peer group filtering (PGF) [36, 37], which is a nonlinear algorithm, has been proposed for noise removal in color images.

Suppose that $x_0(n)$ denotes an image pixel vector, characterizing the color information at position n centered in a $w \times w$ window and $x_i(n)$ denotes other pixels where $i = 0,\ldots,k$ ($=w^2-1$). All the pixels in the window are sorted according to their distances to $x_0(n)$ in ascending order,

$$d_0(n) \le d_1(n) \le \ldots \le d_k(n) \tag{2.1}$$

where $d_i(n) = \|x_0(n) - x_i(n)\|, \qquad i = 0,\ldots,k$.

Therefore, the peer group $P(n)$ of size $m(n)$ for $x_0(n)$ is defined as

$$P(n) = \{x_i(n), i = 0,\ldots,m(n)-1\} \tag{2.2}$$

By using 1-D distances $d_i(n)$ for Fisher's discriminant estimation, the criterion to be maximized is

$$J(i) = \frac{|a_1(i) - a_x(i)|^2}{s_1^2(i) + s_2^2(i)}, \qquad i = 1,\ldots,k \tag{2.3}$$

where

$$a_1(i) = \frac{1}{i}\sum_{j=0}^{i-1} d_j(n) \quad \text{and} \quad a_2(i) = \frac{1}{k+1-i}\sum_{j=i}^{k} d_j(n)$$

$$s_1^2(i) = \sum_{j=0}^{i-1} |d_j(n) - a_1(n)|^2 \quad \text{and} \quad s_2^2(i) = \sum_{j=i}^{k} |d_j(n) - a_2(n)|^2$$

Then, the cut-off position can be obtained by maximizing $J(i)$,

$$m(n) = \arg\max_i J(i) \tag{2.4}$$

In order to remove the effect of impulse noise, the following test is performed on the first and the last M points of $x_i(n)$, where M is half of the window size, to check if they belong to impulse noise:

$$f_i(n) = d_{i+1}(n) - d_i(n) \tag{2.5}$$

where $f_i(n) \leq \alpha$ and α is set large for highly corrupted images and small for slightly corrupted ones.

If $f_i(n)$ does not satisfy the condition, the end points $x_j(n)$ for $j \leq i$ or $j > i$ are considered as impulse noise and removed. The remaining $d_j(n)$ are used to estimate the true peer group. After the impulse noise removal and the peer group classification, the pixel $x_0(n)$ is replaced by the weighted average of its peer group members

$$x_{new}(n) = \frac{\sum_{i=0}^{m(n)-1} w_i p_i(n)}{\sum_{i=0}^{m(n)-1} w_i}, \qquad p_i(n) \in P(n) \tag{2.6}$$

where w_i are the standard Gaussian weights depending on the relative positions of $p_i(n)$ with respect to $x_0(n)$.

2.4.1.3 Color Image Quantization

Human vision perception is more sensitive to the changes in smooth regions than in detailed regions. Accordingly, colors can be more coarsely quantized in the detailed regions without affecting the perceptual quality significantly. The maximum distance for each peer group T(n) (= $d_{m(n)-1}(n)$) is obtained. The weight of each pixel $v(n)$ is calculated as follows:

$$v(n) = \exp(-T(n)), \tag{2.7}$$

and the initial number of clusters N in vector quantization is estimated as follows:

$$N = \beta T_{avg}, \tag{2.8}$$

where β is set to 2 in the experiments and T_{avg} is the average of $T(n)$ which indicates the smoothness of the entire image. In general, the higher the T_{avg}, the less smooth the image is and more clusters are needed to quantize the colors in the image.

The generalized Lloyd algorithm (GLA) [38] is used in vector quantization. The update rule is modified to incorporate the pixel weights. For color cluster C_i, its centroid c_i is calculated as follows:

$$c_i = \frac{\sum v(n)x(n)}{\sum v(n)}, \qquad x(n) \in C_i, \tag{2.9}$$

and the weighted distortion measure D_i is defined as follows:

$$D_i = \sum v(n)\|x(n) - c_i\|^2, \qquad x(n) \in C_i \tag{2.10}$$

By using the popular splitting initialization algorithm, the initial clusters for GLA are determined. For each updated process, the centroids are shifted towards points with higher weights. The algorithm is used to determine which clusters to split until the initial number of clusters N is reached. Thus, points with smaller weights will be assigned fewer clusters so that the number of color clusters in the detailed regions are suppressed. Finally, the cluster centroids are calculated without pixel weights to obtain the true cluster centers.

2.4.1.4 Gabor Wavelet

An important property of Gabor filters [39, 40] is that they have optimal joint localization, or resolution, in both the spatial and the spatial-frequency domains. In addition, they have been shown to be a good fit to the receptive field profiles of simple cells in the striate cortex. These characteristics suggest that the Gabor-filter-based features seem to be similar to features extracted by humans and, thus, may be effective in recognition and classification. Consequently, due to the above properties, there are many researches for the applications in the classification.

The Gabor Wavelet [41-44] can be used in many applications such as remote sensing and crop classification, segmentation, interpretation of medical images, and second-generation image coding techniques. Using a simple texture feature

representation for browsing and retrieval of large image databases, the Gabor wavelet features [45, 46] have been proposed for texture retrieval systems. It uses the Gabor function modulated by a wavelet to describe the characteristics of the texture. A novel set of circularly symmetric Gabor filters [47-49] has also been proposed for rotation invariant texture classification.

2.4.2 Contour-based Shape Features

For contour-based algorithms, Chang *et al.* [50] have proposed a shape recognition scheme based on relative distances between the contour points and their centroid. Other feature extraction methods include the corner point detection [51] and the dominant point detection [52] for contour representation. These important contour points can be used as features in the matching process. However, if part of a shape is missed, or an additional part is added to the boundary of the shape, there may be significant effects on the features and this will cause false matching results. The curvature scale-space (CSS) [14, 22, 27, 28] approach has been proposed as a contour-based shape descriptor used for search and retrieval in MPEG-7. The algorithms based on these shape descriptors usually analyze the contour points in computing the similarity of the shapes, forming an essential part of the retrieval and recognition systems.

For comparing the boundaries of two objects, the algorithm should be able to handle different number of boundary points used to represent the boundaries, simple to compute and responsive to arbitrary changes in orientation, position, and scale of the objects under classification. One way of finding out the similarity of the two objects is to extract their boundaries and the corresponding properties of their shapes. There are many different algorithms for this purpose. We will review the methods

which are used for boundary extraction, shape matching method and recognition below.

2.4.2.1 Chain Code

Chain code [53, 54] has been widely used for the representation and compression of contour information in object-based video applications, such as video object manipulation and content based video coding. It is used to represent a boundary by a connected sequence of straight-line segments of a specified length and direction. The direction of each segment is coded by using a numbering scheme. Typically, this representation is based on the 4- or 8-connectivity of the segments.

Chung *et al.* [53] have proposed a new conditional differential chain coding scheme which is used for lossless representation of an object contour. It uses the conditional probability of directional symbols depending on the context of contour directions. It shows that useful information about the direction can be extracted by using the conditional probability.

Bribiesca [54] has proposed a new chain code to represent the shape of an object which is composed of regular cells. The boundary chain code is based on the number of cell vertices which touch the bounding contour of the shape. It is invariant under translation and rotation, and optionally, under starting point and mirroring transformation. In addition, it is possible to relate the chain length to the contour perimeter.

2.4.2.2 Fourier Descriptors

The Fourier descriptor [55-59] is a popular curve descriptor which describes a closed planar curve by a set of Fourier coefficients. The advantage of this representation is that it reduces a 2-D representation to 1-D problem. The number of pixels on a digital contour is always different. In order to compare the boundaries of

two different objects, we can resample the points on a contour to a specified number and then describe the shape of the digital contour using the Fourier descriptor. Cheng and Yan [55] have used this descriptor as a feature vector to classify different handwritten digits based on the contour information.

Fourier descriptors are complex coefficients of the Fourier series expansion of waveforms. From the boundary trace of a shape, a pair of one-dimensional waveforms $[x(t), y(t)]$ can be generated. Suppose that N samples of a closed boundary are taken. Each sample can be represented as follows:

$$u(n) = x(n) + jy(n) \tag{2.11}$$

where $n = 0, 1, \ldots, N\text{-}1$.

Then, its discrete Fourier Transform (DFT) representation is shown as follows:

$$f(k) = \sum_{n=0}^{N-1} u(n) \exp\left(-\frac{j2\pi kn}{N}\right), \tag{2.12}$$

where $k = 0, 1, \ldots, N\text{-}1$.

The complex coefficients $f(k)$ are called the Fourier descriptors of the boundary. The Fourier descriptors are invariant to the starting point of sampling, rotation, scaling, as well as reflection.

2.4.2.3 Curvature Scale-Space

The curvature scale-space (CSS) [14, 22, 27, 28] representation of a contour has been further extended and optimized during the MPEG-7 development phase. The key modifications include the addition of global shape parameters, transformation of the feature vector in the parameters space improving retrieval performance, and a new quantization scheme supporting a compact representation of the descriptor.

The shape properties of contours are important for retrieval of semantically similar objects. This descriptor is robust to noise present in the contour and it is very

efficient in applications where high variability in the shape is expected, due to deformations in the object (rigid or non-rigid), or perspective deformations. Some important features are that it can distinguish between shapes that have similar region-shape properties but different contour-shape properties, and support searches for shapes that are semantically similar for humans. Also, it is robust to distortions in the contour due to perspective transformations.

A CSS descriptor of a contour shape can be created as N equi-distance points which are selected on the contour and grouped into two series X and Y. The contour is gradually smoothed by repetitive application of a low-pass filter with the kernel (0.25, 0.5, 0.25) to X and Y. For each smoothed contour, the zero-crossings of its curvature function are computed. A CSS image can be associated with the contour evolution process. The horizontal line represents the indices of the contour points and the vertical line represents the amount of filtering applied. The curvature zero-crossing points separate concave and convex parts of the contour. The CSS description is extracted according to the coordinate values of the prominent peaks and listed in order of decreasing value of the y coordinate, and the eccentricity and circularity of the contour. Both of them are quantized and stored in the database.

2.4.2.4 Template Matching

For shape recognition, template matching is one approach to simplifying the problem of different scales, orientations and positions. A potential-based approach [60] has been proposed. This matching process involves the minimization of a potential function with an assumption that the border of any 2D region is uniformly charged. If a shape template is small in size and can be placed inside a region whose shape is to be determined, the template will experience a repulsive force and torque arising from the potential field.

The basic idea of this approach is to achieve a better match in the shape between the template and the given region by translating and reorienting the template along the above force and torque directions, respectively, toward the configuration of the lowest potential. This approach is intrinsically invariant under translation, rotation and size changes of the shape sample. The advantages of this method are that it is simple and highly efficient. It can also match two boundaries in a noisy environment and even the boundaries consist of unconnected segments, provided that the template remains inside the shape of the sample.

2.4.2.5 Deformable Templates

An automatic image retrieval system should be able to search in a database for those images which contain objects with similar characteristics as specified by the user. It is required to locate and retrieve an object from a complex image based on its 2D shape or boundary information. In order to locate and identify an object, a process of matching using the deformable templates [61] has been proposed. The model consists of a prototype template which describes a representative shape of a class of objects, a set of parametric transformations which deform the template and a probability distribution defined on the set of deformation mappings. The prototype captures the global structure of a shape without specifying a parametric form for each class of shapes. The template has to be deformed to match objects in an image. Then, a Bayesian scheme is used to find a match between a deformable template and objects in an image.

2.4.2.6 Shape Spectrum

Shape Spectrum [14, 24, 29, 30] is defined as the histogram of the shape index, computed over the entire 3-D surface of a 3-D object which is represented as a 3-D

mesh. The shape index computed for each vertex of the mesh is invariant to affine transformations. The shape index of the 3-D surface at point p is defined as follows:

$$SI_{\Sigma} = \frac{1}{2} - \frac{1}{\pi}\arctan\frac{k_1(p)+k_2(p)}{k_1(p)-k_2(p)} \tag{2.13}$$

where $k_1(p)$ and $k_2(p)$ denote the principal curvatures at point p, and the shape index captures information about the local convexity of the 3-D surface.

The descriptor uses histograms with 100 bins and each bin is represented by 12 bits. Also, the relative area of planar surface regions of the mesh with respect to the entire area of the 3-D mesh, and the relative area of all polygonal components with respect to the entire area of the 3-D mesh are used as two additional variables.

2.4.3 Region-based Shape Features

The applications of contour-based descriptors are more specific and limited when compared to those of region-based descriptors. This is because the latter does not need to extract the contours. For example, a trademark may consist of text, images, and occasionally other media such as scent or sound. Therefore, images of a trademark retrieval system [62, 63] are usually represented by region-based descriptors. Many well-known region-based descriptors, such as moment invariants [64-67], Zernike moments [4, 68-72], Fourier coefficients [73, 74], angular radial transformation (ART) [23], etc., have been proposed and can provide invariant features with respect to the affine transformation. As the region-based shape descriptor expresses pixel distribution with a 2-D object region, it can describe complex objects consisting of multiple disconnected regions as well as simple objects with or without holes. There are many different algorithms for this purpose. We will review the methods which are used for feature extraction, point set matching method and recognition in the following.

2.4.3.1 Moment Invariants

The use of moment invariants as shape recognition and identification have been proposed by Hu [64] in 1962. Suppose that the intensity function of an image $f(x,y)$ is assumed to be piecewise continuous and with compact support, the central moments of order (p,q) are defined as follows:

$$\mu_{pq} = \iint (x-\bar{x})^p (y-\bar{y})^q f(x,y) dx dy \tag{2.14}$$

and the normalized central moments, denoted by η_{pq}, are defined as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{\gamma}} \tag{2.15}$$

where $\gamma = \frac{1}{2}(p+q)+1$ for $p+q = 2,3,....$

From the second and third order moments, a set of seven invariant moments can be derived as follows:

$$\begin{aligned}
m_1 &= \eta_{20} + \eta_{02} \\
m_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
m_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
m_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
m_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \cdot \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + \\
&\quad (3\eta_{21} - \eta_{30})(\eta_{21} + \eta_{03}) \cdot \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
m_6 &= (\eta_{20} - \eta_{02}) \cdot \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
m_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \cdot \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] - \\
&\quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \cdot \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]
\end{aligned} \tag{2.16}$$

For the purpose of comparison, the moment invariants are computed for each image in an entire collection of images. The limits of each moment invariant are determined and then normalized between 0 and 1.

2.4.3.2 Zernike moments

Zernike moments proposed for image recognition are derived from Zernike polynomials which form a complete orthogonal set over the interior of the unit circle. Let the set of these polynomials be denoted by $V_{nm}(x,y)$. The form of these polynomials is shown as follows:

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

where

n	Position integer or zero.
m	Position and negative integers subject to constraints n-$\|m\|$ even, $\|m\| \le n$.
ρ	Length of vector from origin to (x,y) pixel.
θ	Angle between vector ρ and x axis in countclockwise direction.

Radial polynomial is defined as follows:

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

The Zernike moment of order n with repetition m for a continuous image function $f(x,y)$ that vanishes outside the unit circle is shown as follows:

$$A_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2\le 1} f(x,y) V^*_{nm}(\rho,\theta)\,dx\,dy \tag{2.19}$$

To compute the Zernike moments of a given image, the center of the image is taken as the origin and pixel coordinates are mapped to the range of unit circle. Those pixels falling outside the unit circle are not used in computation.

2.4.3.3 Angular Radial Transformation (ART)

A complex 2-D Angular Radial Transformation (ART) is defined on a unit disk in polar coordinates. From each shape, a set of ART coefficients F_{nm} is extracted using the following formulas,

$$F_{nm} = \langle V_{nm}(\rho,\theta), f(\rho,\theta) \rangle = \int_0^{2\pi} \int_0 V_{nm}(\rho,\theta) f(\rho,\theta) \rho \, d\rho \, d\theta \tag{2.20}$$

The basic functions are separable along the angular and radial directions, and are defined as follows:

$$V_{nm} = \frac{1}{2\pi} \exp(jm\theta) R_n(\rho), \tag{2.21}$$

where $R_n(\rho) = \begin{cases} 1, & n = 0 \\ 2\cos(\pi n \rho), & n \neq 0 \end{cases}$

Although an object may be split into disconnected sub-regions during the process of segmentation, the above coefficients give a compact and efficient way of describing properties of multiple disjoint regions simultaneously and is also robust to segmentation noise.

2.4.3.4 Hough Transform

Hough transform (HT) [75-79] is a voting process where each point belonging to the patterns votes for all the possible patterns passing through that point. These votes are accumulated in an accumulator array, and the pattern receiving the maximum votes is recognized as the desired pattern. The disadvantages of the standard Hough transform are the heavy burden of computational complexity, massive storage requirement and inferior to the intellectual mechanism of the human visual recognition. To overcome these weaknesses of the Hough transform, many modified approaches have been proposed. The earliest and most classical probabilistic Hough transform (PHT) algorithm is the randomized Hough transform (RHT). A single parameter point can be determined uniquely with a pair, triple or generally *n* feature points from the original picture, depending on the complexity of the curves to be detected.

2.4.3.5 Distance Transformation

In most of the digital image processing applications, it is preferable to use integers to represent distance. The process, so called Chamfer distance [80] can be used to compute a good integer approximation of the Euclidean distance. It can be calculated sequentially by a two-pass algorithm. First, a distance image is created such that each boundary pixel is set to zero and each non-boundary pixel is set to infinity. The distance image can be modified after the forward and backward pass operators. Then, the distance image can be represented as the measure of distance from the boundary.

It can convert the boundary pixels of a shape into a gray-level image where all pixels have a value corresponding to the distance to the nearest boundary pixel. The goodness of the match between the two shapes can be computed from the root-mean-square value of the normalized boundary pixels of the second shape on the distance image created from the first shape.

Using every pixel of the boundary to calculate the distance works very well when the object under classification is undistorted or only a small part of it is distorted. When a large part of the object is distorted, the distance measurement scheme cannot serve the purpose of distance calculation. Liu and Srinath [81] have proposed a partial distance measurement scheme which can be used to recognize a partial shape without knowing its orientation, location, and size.

2.4.3.6 Hausdorff Distance

Hausdorff Distance [82] can be used in binary image comparison and computer vision. Suppose that A and B are the two point sets representing objects in two images.

$A = \{a_1, ..., a_p\}$, *and* $B = \{b_1, ..., b_q\}$,

Then, the Hausdorff distance *H(A,B)* is defined as follows:

$$H(A,B) = \max(h(A,B), h(B,A)), \tag{2.22}$$

where $h(A,B) = \max_{a \in A} \min_{b \in B} \|a - b\|$.

Unlike most shape comparison methods, the Hausdorff distance can be calculated without the explicit pairing of points in their respective data sets. It can be used to compare partial images with a natural allowance in simple and fast implementation. This distance can be used to determine the degree of resemblance between two objects that are superimposed on one another. However, the method is sensitive to noise and has some problems for object matching under noisy conditions. Therefore, the original definition of *h(A,B)* is redefined such that the measure of the distance is improved. The modified Hausdorff distances [83] proposed have the most desirable behavior for real-world applications. It can be used in face matching and fast screening of large facial databases.

2.4.3.7 Robust Orientated Hausdorff Similarity Measure

An oriented Hausdorff similarity (OHS) [84] measure based on the similarity concept of the Hough transform (HT) for robust object alignment has been proposed, which embedded the robust Hausdorff measure into an accumulating operation of the Hough transform and can eliminate the wrong correspondences by computing the orientation at each pixel. This algorithm is composed of edge detection, the distance transform (DT) map, and the similarity computation. The edge maps along with their DT maps, and the gradient images are necessary to compute the proposed OHS. The similarity is calculated by accumulating the distance value of distance transform map. The Oriented Hausdorff Similarity (OHS) is defined as follows:

$$H_{OHS} = \min(h_{OHS}(A_G, A_E, d_B), h_{OHS}(B_G, B_E, d_A)) \tag{2.23}$$

and the direct OHS is

$$h_{OIIS}(A_G, A_E, d_B) = \sum_{a \in A_E} s(a)\rho_T(d_B(a)), \tag{2.24}$$

where an orientation vector $O_{A_G(a)}$ represents a unit gradient vector of A at position a, $s(a) = O_{A_G(a)} \cdot O_{B_G(a)}$ denotes the dot product of two gradient vectors obtained from two images, and $\rho_T(x)$ is a symmetric threshold function.

The computational complexity of the algorithm has linear-order in computing the Hausdorff distance measure at a point. For the algorithm, fast search algorithms can be developed by sub-sampling the edge images of a model and an input. In the coarser level, the matching point is detected in the Hough space generated by sub-sampled edge pixels. In a finer level search, other edge information is used to accumulate the Hough translational space, thus yielding accurate matching point.

2.4.3.8 Principal component analysis Methods

Principal component analysis [85], also known as the Karhunen-Loeve transform [86], has been used for face recognition. The main idea is to find the vectors that best account for the distribution of face images within the entire image space. We called these eigenvectors or eigenfaces [87, 88]. These eigenfaces can be thought of as a set of features (or eigenfeature) that together characterize the variation between face images. However, some eigenfeatures may not be related to recognition. More features do not imply a better success rate for classification. For example, the illumination direction and the facial expression are factors which affect the success rate of the classification. Therefore, the fisherfaces method [89-91] has been proposed to deal with this problem. Even though the principal component analysis can describe some major variations in the class, it is not good for discriminating among classes defined by the set of samples. The fisherfaces method can weigh down the

unimportant features appropriately or discard them by using the selection process. It is an effective feature space to be used for classification and it discounts factors unrelated to classification. In addition, the fisherfaces method can be used in face recognition system [92] using local autocorrelations and multiscale integration by extracting the salient features.

2.5 Summary

In this chapter, we have described some existing CBIR systems and image databases used for content based image retrieval, and the recent development of MPEG-7. We have also introduced the visual descriptors recommended in the MPEG-7 standard. Finally, we have reviewed some techniques which can be used for visual feature extraction and recognition. We have mentioned some techniques used for the three types of features: color and texture features, contour-based shape features and region-based shape features. However, there are still a lot of different techniques proposed previously for content based retrieval. In the following chapters, we will present some algorithms used for shape feature extraction and recognition, as well as retrieval.

Chapter 3
Proposed Methods for Boundary Extraction

Snake is an active contour model for representing image contours. In this chapter, we propose an efficient active contour model which can represent highly irregular boundaries. The algorithm includes an adaptive force along the contour, and adjusts the number of points for the snake according to the desired boundary. A better stopping criterion based on the area of a closed contour is devised. Furthermore, in this method, a contour can break automatically to represent the contours of multiple objects. Experiments show that this method can extract object's boundaries accurately and efficiently.

3.1 Introduction

Locating an object's boundary is an important step for content-based retrieval systems [1]. Most of the systems use shape information for image query. An active contour model (snake) [2] has been successfully used in contour detection for object recognition, computer vision, computer graphics, and biomedical image processing. Based on the object's boundary, its contents can then be extracted. Nevertheless, this active contour model has many limitations. Firstly, the initial contour should be near to the object, provided that the contour is in the effect of the image force. Secondly,

the performance of the snake depends on the number of contour points, which is usually fixed. According to these reasons, the model is unable to represent irregular objects accurately. Besides, this method fails to extract the contours of multiple objects. In real situations, an image may contain a number of objects which are relevant objects for retrieval.

In this chapter, the original snake model is reviewed in Section 3.2. Our proposed adaptive snake model is presented in Section 3.3. Section 3.4 describes the contours representation for multiple objects by using the adaptive snake model. Experimental results are demonstrated in Section 3.5. Finally, summary is given in Section 3.6.

3.2 Original Snake Model

An active contour model (snake) [2] is an energy-minimizing spline, which can be operated under the influence of internal contour forces, image forces, and external constraint forces. A snake is represented as a parametric curve $v(s) = [x(s), y(s)]$, where the arc length s is a parameter. An energy functional of the snake is defined as

$$E^{*}_{snake} = \int_0^1 E_{internal}[v(s)] + E_{image}[v(s)] + E_{constraint}[v(s)]ds \qquad (3.1)$$

where $E_{internal}$ represents the internal energy of the contour due to bending or discontinuities, E_{image} refers to the image forces, and $E_{constraint}$ is the external constraint forces. The location of the snake corresponds to the local minima of the energy functional.

We have proposed the fast greedy algorithm [3], which is an iterative method for minimizing the energy functional. In order to locate the boundary of an object in an image, an accurate estimation of the initial position and the number of points of the snake is necessary. The initial snake is then drawn by image forces to the object's

boundary. However, the image force is effective only when the snake is placed near the edge. To attract snake from a fairly large distance away from the edges, the image force is blurred by a 2-D Gaussian smoothing operator with a large window size. The window size is chosen to be 23×23. This allows the snake to move to the blurry energy functional and the blurring is then slowly reduced when the snake is close to the image. Nevertheless, for a highly irregular boundary, part of the snake may still be too far away from the desired boundary.

Cohen [4] proposed a balloons model for snake, but it only partially resolved the problems. It is difficult to locate concave parts of an object from outside. In view of such limitation, Wong *et al.* [5] proposed a segmented snake which converts the global optimization of a closed snake curve into local optimization based on a number of open snake curves. A recursive split-and-merge procedure is then used to determine the final object contour. However, the method is computationally expensive.

In our proposed model, an adaptive force is introduced at a point of the snake whenever the image forces in its neighborhood are smaller than a threshold. In order to achieve an accurate representation of a boundary, the distances between adjacent points of the snake are kept close to a constant. Two processes, namely deletion and insertion, are introduced to change the number of points for the snake during the energy minimization process.

3.3 Adaptive Snake Model

3.3.1 Generating an adaptive force

In our method, an adaptive force will be introduced at a point if its surrounding image forces are smaller than a threshold E_{noise}. If the noise in the surroundings of a point is less than the threshold, an adaptive force is created and the effect of the noise can be overcome. The direction of the force is defined to be perpendicular to the line joining its two adjacent points, as illustrated in Fig. 3.1.

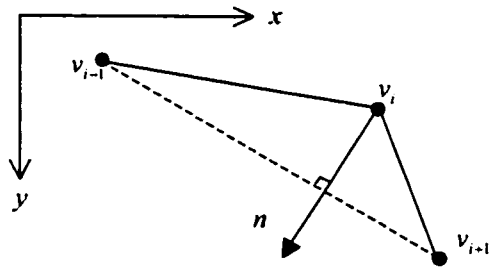


Figure 3.1: The adaptive force.

Considering a point v_i, the direction n of the force is defined as follows:

$$n \cdot v = 0 \tag{3.2}$$

and $$v = \frac{v_{i+1} - v_{i-1}}{|v_{i+1} - v_{i-1}|} \tag{3.3}$$

where n is the unit adaptive force. As the point is far away from the desired boundary, it is allowed to move with a larger step size in the direction as follows:

$$v_i' = v_i + 3 \cdot n \tag{3.4}$$

where v_i' is the new position of the point. In other words, the snake can move 3 steps in the direction of the normal force whenever the image forces are small. This can reduce the number of iterations required and speed up the whole process.

3.3.2 Overshoot and undershoot problems of the adaptive force

After determining the direction of the adaptive force, it can be observed that the points may move too close or too far away from each other. It has been shown that overshoot and undershoot problems in the concave region may occur. Wong *et al.* [5] proposed a segmented snake approach to determine the object contour, but the method is computationally expensive. Although this method can overcome the problem of overshoot and undershoot by using different normal forces in between the non-contour segments, it involves a number of calculation for each segment. Furthermore, a fixed number of snake points cannot represent the object accurately if the boundary of the object is highly irregular.

The above problems are solved by the deletion and insertion of snake points. The main idea of this method is to keep the distance between each point more constant. If the distances between adjacent points are small, a deletion process will be applied to avoid overshoot. If the distances between adjacent points are large, an insertion process will be applied such that the continuity of the snake can be maintained. The snake is constructed by using a doubly-linked list structure so that the deletion and insertion processes can be performed efficiently.

3.3.3 The Deletion and Insertion Processes

To achieve an accurate representation of a boundary and overcome the overshoot and undershoot problem, the number of points for the snake must be determined appropriately. As this number depends on the size and shape of the boundary, in the iterative process, two operations - deletion and insertion - are introduced to keep the distance between two points of the snake more constant.

In our method, if the distance of two adjacent points of a point, v_i, is less than a threshold or if the distance between the points, v_i and v_{i+1}, is less than another threshold, point v_i is deleted, and v_{i-1} and v_{i+1} will then become adjacent points. When the distance between two adjacent points is larger than a threshold, an additional point will be inserted between them.

Suppose the distances d_1 and d_2 are defined as follows:

$$d_1 = |v_{i+1} - v_i| \qquad \text{and} \qquad d_2 = |v_{i+1} - v_{i-1}|,$$

then the criteria for these two operations are described as follows:

If $d_1 < t_1$ or $d_2 < t_2$, then delete v_i

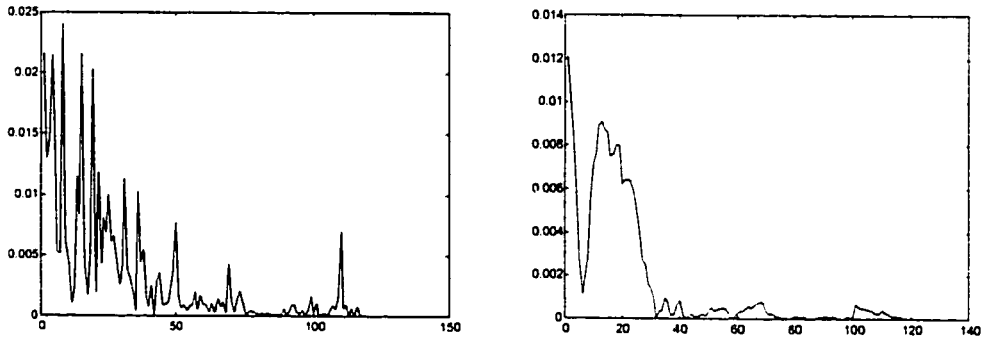
else if $d_1 > t_3$, then a new point u is inserted between v_i and v_{i+1}.

where the new point, $u = \frac{v_i + v_{i+1}}{2}$ and t_1, t_2, t_3 are threshold values.

3.3.4 Terminating Criteria

In the fast greedy algorithm, the iteration will be stopped if the number of points moved is less than a threshold or if the number of iterations has reached a certain quantity. It is observed that, sometimes, the points of the snake shift along the boundary, and the contour only moves very slightly. However, as most of the points are moved, the iteration process will be continued. Other terminating criteria include the measurement of the difference between the lengths of the contour in the current and previous iterations and the measurement of the change of the energy per unit length of the contour in successive iterations. The limitations of the criterion are the incorrect detection of image with no object, inability to locate small object, and poor stability. Wong *et al.* [6] have proposed a contour length criterion (CL-criterion) which can overcome the above limitations. The CL-criterion measures the rate of

change of the normalized total length of a curve and possesses two properties: a two-phase convergence property and an averaging property. The two-phase convergence property can solve the problem of detecting a small object or no object. The averaging property can solve the stability problem by smoothing the contour lengths computed in successive iterations. In the method, however, it requires the contour lengths to be stored over the past 10 iterations for smoothing. Otherwise, the sudden increase and decrease of the rate of change of normalized total length may cause the spikes and valleys during the iterations. The instability problem is due to the space discretization of the evolution problem which is introduced by the oscillation of the snake points between the quantized steps. The plots of CL-criterion against the iteration without averaging and with averaging are shown in Fig. 3.2. Fig. 3.2(i) illustrates the curve fluctuating seriously during iteration, while Fig. 3.2(ii) shows the improvement in the stability of the termination process when averaging is applied.



(i) (ii)

Figure 3.2: CL-criterion against the iteration (i) without averaging (ii) with averaging.

In our method, a new terminating criterion called contour area criterion (CA-criterion) is proposed, which makes use of the normalized total area to determine the convergence of the process.

Suppose that A^k is the area of the snake at the k^{th} iteration. Then, the CA-criterion, η^k, is defined as follows:

$$\eta^k = \frac{A^k - A^{k-1}}{A^k} \tag{3.5}$$

The area of the snake A^k can be calculated by using the equation as follows:

$$A^k = \frac{1}{2}\sum_{i=1}^{n}\begin{vmatrix} x_i^k & y_i^k \\ x_{i+1}^k & y_{i+1}^k \end{vmatrix} \tag{3.6}$$

where (x_{n+1}^k, y_{n+1}^k) is equal to (x_1^k, y_1^k) for a closed snake and n is the total number of snake points. The value of η^k converges quickly and is close to zero when the object boundary is located by the snake. Figure 3.3 shows the rate of convergence with the CA-criterion.

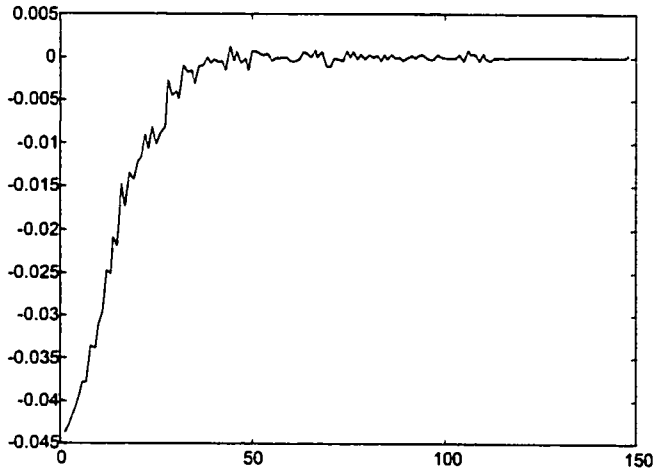


Figure 3.3: Plot of CA-criterion η^k against the iteration.

The CA-criterion exhibits a smoother and faster convergence than that of the CL-criterion. The CA-criterion also has better stability in convergence, as its fluctuation between successive iterations is much smaller than the CL-criterion without using any

averaging. In addition, the calculation of area is simple and fast, while the CL-criterion requires computing square roots to obtain the length of each segment.

Using the relations of the area and the perimeter of a closed contour, we have an inequality expressed as follows:

$$\frac{Area}{Perimeter^2} \leq \frac{1}{4\pi} \qquad \text{or} \qquad Area \leq \frac{1}{4\pi} Perimeter^2 \tag{3.7}$$

We can observe that the area of the object is bounded provided that its perimeter is convergent. If the perimeter is converged, then the area of the object will converge faster. Therefore, the use of area as a stopping criterion can give a better convergence.

3.4 Contour Representation for Multiple objects

The adaptive snake model can be used to represent the contour of a single object. However, if there are more than one object in the image, the snake model must be modified such that it can determine the corresponding contour of each object. In Section 3.4.1, we will describe the determination of critical points where the snake is split and connected for multiple object representation. In Section 3.4.2, we will demonstrate the splitting and connecting process. Finally, the validation of each contour will be verified so that invalid contours are removed.

3.4.1 Critical points for splitting and connecting

In order to determine which parts of a contour should be split to form two contours, the segments of the snake are classified into two types, contour segment and non-contour segment, which are determined by calculating the surrounding image forces along a segment. If the surrounding image forces of a point are smaller than a threshold, we define it as a non-contour point. Otherwise, it is defined as a contour

point. A sequence of contour points or non-contour points forms a contour or non-contour segment. A critical point is defined as the end points of a contour segment which are adjacent to non-contour segments. The determination of a contour point is illustrated as follows:

If $(E(v_i) > E_{noise},\ E(v_{i-1}) > E_{noise}$ and $E(v_{i+1}) < E_{noise})$

or $(E(v_i) > E_{noise},\ E(v_{i-1}) < E_{noise}$ and $E(v_{i+1}) > E_{noise})$,

then point v_i is marked as a critical point.

where $E(v_i)$ is the image forces at the surounding of v_i, and E_{noise} is a pre-defined threshold. A number of critical points are therefore identified and denoted as a sequence as follows:

$$C = \{c_i = (x, y) \mid i = 0, 1, 2, ..., n-1\}, \tag{3.8}$$

where c_i is a critical point and n is the number of critical points. The number, n, is equal to zero or an even number for a closed loop contour.

During the iteration process, each of the critical points will be checked in sequence. Pairs of the critical points will be marked as connected points where the splitting and connecting operations will be performed to form two contours.

With reference to Fig. 3.4, the procedure to identify connected points is as follows:

Step 1: Set $i = 0$ as the starting critical point c_i.
Step 2: Set $j = i + 1$ for the other critical point c_j.
Step 3: Compute the distance, d_{ij}, between the two critical points c_i and c_j.
Step 4: If d_{ij} is less than a threshold, c_i and c_j are marked as a pair of connected points.
Step 5: If $j < n$, then $j = j + 2$ and go to Step 3. Otherwise, go to Step 6.
Step 6: If $i < n$, then $i = i + 1$ and go to Step 2. Otherwise, go to Step 7.
Step 7: End.

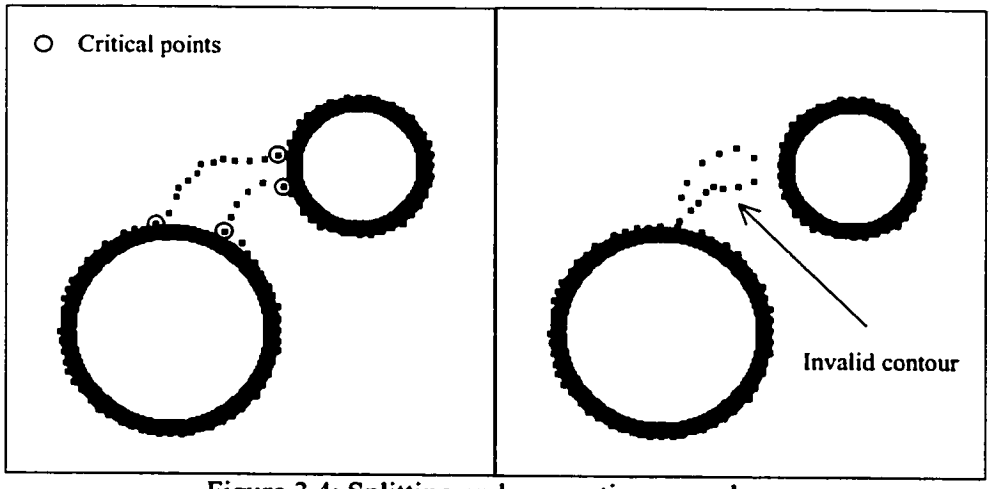


Figure 3.4: Splitting and connecting procedure.

3.4.2 Splitting and Connecting Processes

Having searched the pairs of connected points, which are close to each other, the splitting and connecting operations are then performed. Suppose that a pair of the connected points is defined as follows:

$$S = \{s_k = (v_i, v_j) \mid i < j < n, k < m\} \tag{3.9}$$

where s_k is the k^{th} pair of connected points, n is the number of snake points, and m is the number of connected pairs. The transition from contour segment to non-contour segment occurs at the point v_i, while an opposite transition happens at the point v_j. Then, the procedure for the splitting and connecting operations is shown as follows:

Step 1: Set $k = 0$.

Step 2: Splitting process: the contour is broken between the points v_i and v_{i+1}, and between the points v_j and v_{j-1}

Step 3: Connecting process: the contours between the points v_i and v_j, and the points v_{i+1} and v_{j-1} are connected. Two contours are, therefore, formed after this connection process.

Step 4: If $k < m$, then $k = k + 1$ and go to Step 2. Otherwise, go to Step 5.

Step 5: End.

3.4.3 Criteria for validation of the snake

With the above processes, new snakes may be generated. However, some of the snakes may be invalid, as illustrated in Fig. 3.4. The invalid contour will grow until it touches the edges in the image. This kind of contour can be identified by calculating its enclosed area. If the points of a snake collapse and meet each other, the area enclosed by the snake will be zero. If the snake grows in a reverse direction, its computed area will be negative. This signifies that an invalid snake has been formed. In addition, a snake is also considered to be invalid if its number of points is less than a certain value (e.g. 5). The computed area for a contour is also used in the terminating criterion in the iteration process. As described in Section 3.3.4, this is a better terminating criterion than that based on length measurement.

3.5 Experimental Results

In the experiments, we compare the performance in contour representation of our new algorithm with that of the fast greedy algorithm. The performances based on the CL- and CA-criterion are also compared. Finally, the extractions of the boundaries of multiple objects are illustrated. The experiments were conducted on a Pentium II 400MHz PC.

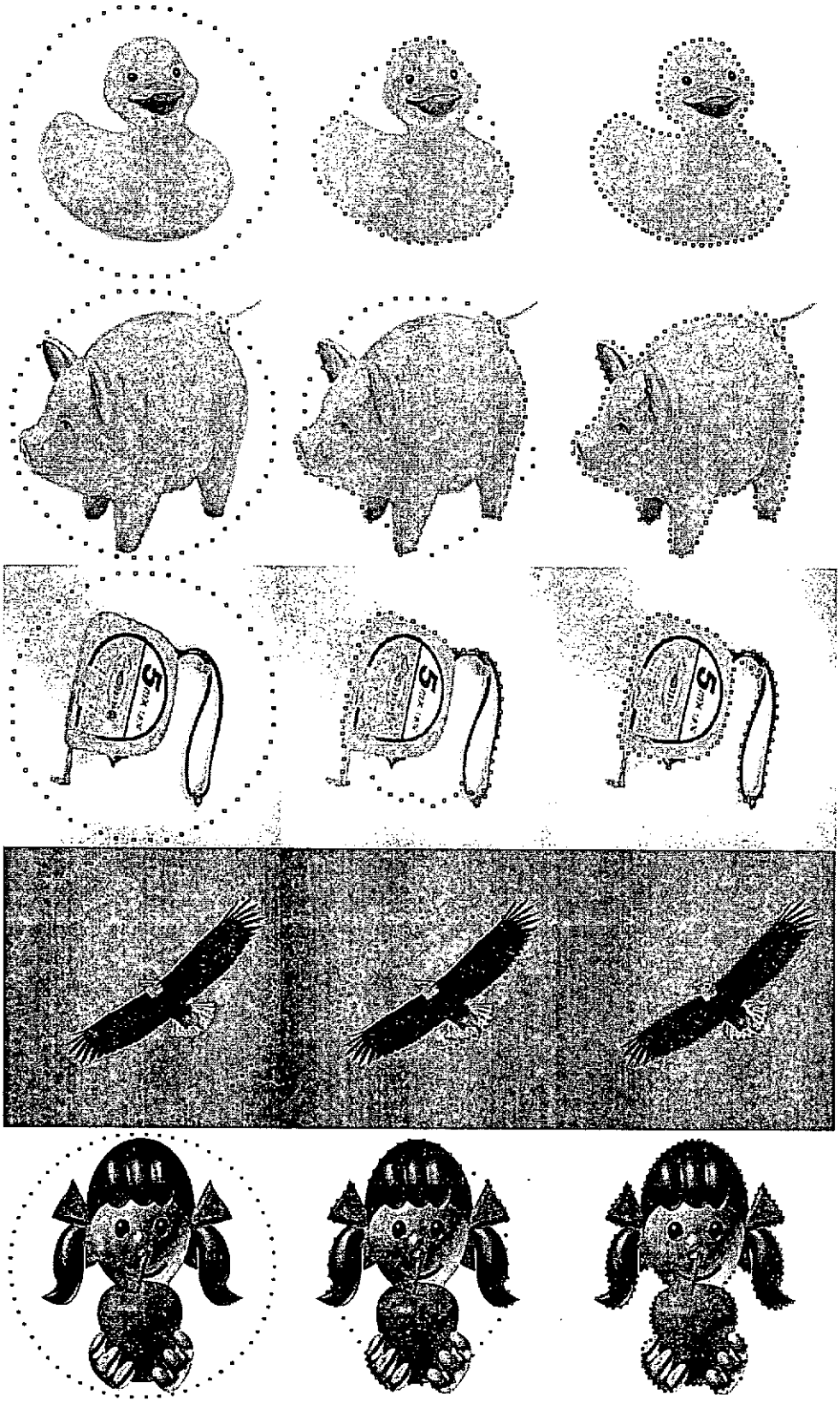
In our algorithm, the snakes are initialized as a circle or a rectangle with an arbitrary number of points surrounding the desired objects. The inter-distance between two adjacent points, and the thresholds t_1, t_2, and t_3 used in the deletion and insertion process are set at 6, 4, 8, and 12, respectively. The threshold values are basically determined by the inter-distance such that the snake points can move to the desired position smoothly without the overshoot and undershoot problems of the

adaptive force. On the other hand, the pre-defined threshold E_{noise}, which is used to determine whether to apply the adaptive force or not, is set at 30. If the surrounding image forces of the snake points are smaller than the threshold, an adaptive force is created and the effect of background noise in an image can be overcome.

Figure 3.5(a) shows the initial contours, while Fig. 3.5(b) and Fig. 3.5(c) depict the final contours using both the fast greedy algorithm and our new algorithm, respectively. Figure 3.6 shows some more examples for highly irregular objects. It is obvious that the new algorithm can achieve an accurate representation of the concave region of a boundary, and keep the distances between adjacent points close to a constant. The performances in contour representation based on the CL- and CA-criterion are very similar. However, the required number of iterations and runtimes are different, as shown in Table 3.1. The CA-criterion results in a smoother convergence and can therefore make decision on stopping iteration more accurately.

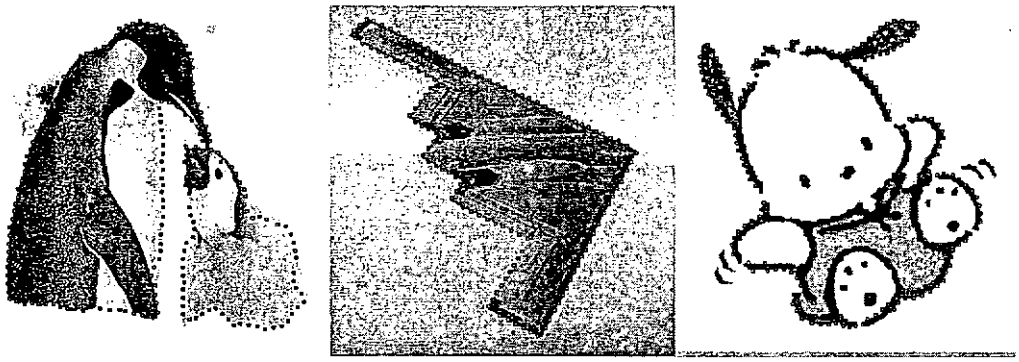
	Penguins		B-2		Pochacco	
	CL	CA	CL	CA	CL	CA
Number of iteration	75	72	47	43	41	39
Runtime (ms)	118.5	111.6	56.4	50.8	68.5	63.6
Number of snake points	142	142	109	110	146	144

Table 3.1: Comparison with CL and CA-criterion.



(a) (b) (c)

Figure 3.5: The comparison of the contour representation using snakes: (a) the initial contours of snakes, (b) using the fast greedy algorithm method, and (c) using the new algorithm.



(i)Penguins (ii)B-2 (iii)Pochacco

Figure 3.6: Some examples for highly irregular boundary.

Figure 3.7 illustrates the resulting contours for representing multiple objects during the iteration process. Due to the adaptive force, parts of the snake will move close to each other, where the snake is split into two separate contours. Some more results are illustrated in Fig. 3.8.

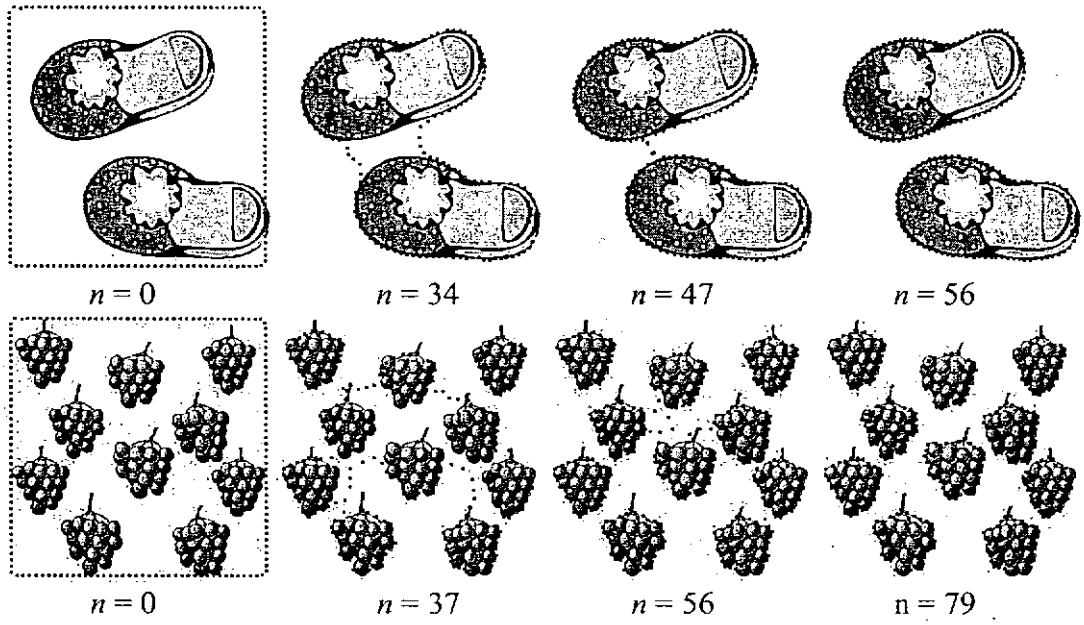


Figure 3.7: The resulting contours for multiple objects during the iteration process.

Figure 3.8: Some more examples for multiple object representation.

3.6 Summary

In this chapter, we have presented an active contour model which can locate highly irregular boundaries in an image. An adaptive force is applied when the image forces around a point are small and points on the snake can be deleted or inserted such that the distance between adjacent points can be kept more constant. An additional terminating criterion based on area is also proposed such that unnecessary iterations can be prevented. By identifying contour and non-contour segments in a snake, the algorithm can extract multiple objects in an image by the splitting and connecting processes. Experiment results show that the new algorithm can achieve a better contour representation with a required runtime similar to that of the fast greedy algorithm.

Chapter 4

Extraction of an Euclidean Skeleton based on a Connectivity Criterion

Based on the contour extracted by the adaptive snake algorithm as described in the previous chapter, a skeleton, which is obtained by using a skeletonization method, can be used as a feature to represent the original shape as it has a more compact representation. In this chapter, we propose a fast, efficient and accurate skeletonization method for the extraction of a well-connected Euclidean skeleton based on a signed sequential Euclidean distance map. A connectivity criterion is proposed, which can be used to determine whether a given pixel is a skeleton point independently. The criterion is based on a set of point pairs along the object boundary, which are the nearest contour points to the pixel under consideration and its 8 neighbors. Our proposed method generates a connected Euclidean skeleton with a single pixel width without requiring a linking algorithm or iteration process. Experiments show that the runtime of our algorithm is faster than the distance transformation and is linearly proportional to the number of pixels of an image.

4.1 Introduction

The skeleton is essential for general shape representation. It is a useful means of shape description [100] in different areas, such as content-based image retrieval systems, character recognition systems, circuit board inspection systems, as well as biomedical imagery for shape analysis. The extracted skeleton can be used as a feature to represent the original shape as it has a more compact representation. In real-time image processing, a fast skeletonization algorithm is necessary.

Due to the importance of skeletonization, many approaches for it have been proposed throughout the past decades. Most of the skeletonization algorithms can be simply classified into two essential types. The first type is referred to as thinning algorithms, such as shape thinning [101-104] and the wave front/grassfire transform [105, 106]. These algorithms iteratively remove border points, or move to the inner parts of an object in determining an object's skeleton. However, the iterative process is a time-consuming operation and requires some terminating criteria. In addition, the uniqueness of the extracted skeleton may be dependent on the initial conditions provided. The second type of algorithm is based on the medial-axis transform, as introduced by Blum [100]. Examples include the line skeleton [107-109], Voronoi skeleton [110,111], morphological transform [112-114], and maximal disk method [115-118]. As these algorithms search the set of centers and the corresponding radii of the maximal disks contained in an object, they can usually preserve all the information about the object and allow the reconstruction of the object. Although the medial axis transform is a very useful tool and yields an intuitively pleasing skeleton, its direct implementation is usually computationally prohibitive. Moreover, due to the use of discrete space, the extracted skeletons are sensitive to local variations and

noise, and there is no guarantee that the extracted medial-axis can preserve the original object's topology. In addition, these algorithms may produce redundant points on determination of the skeleton, and are memory-intensive and require a complex data structure.

In order to efficiently find the maximal disks enclosed in an object, a distance map should be generated before locating the centers of the maximal disks. Different algorithms based on different distance maps have been proposed. Approximate distance maps such as the city-block, chessboard, chamfer distance transform (CDT) [119], etc. can be used to extract the maximal disks. An exact Euclidean distance map can be obtained by using the Euclidean distance. The corresponding skeleton, namely the Euclidean skeleton, is invariant to rotation. However, the Euclidean skeleton will include much more computation due to the square root operation. This problem can be solved by means of the vector distance transform (VDT) [120]. Two sequential Euclidean distance (SED) mapping algorithms, called 4SED and 8SED, have been proposed for the efficient computation of the Euclidean distance map. The signed sequential Euclidean distance (8SSED) [121] mapping algorithm is an extension of the 8SED, which keeps track of the sign component of the vectors. Consequently, this algorithm can greatly reduce the time required to create the distance map. Not only is the shortest distance to the object's boundary determined, but the position of the corresponding point on the boundary is also provided.

Extracting the centers of the maximal disks (CMDs) using a distance map is also complicated. Many algorithms have been proposed for extracting the true CMDs by using different types of distance map. In order to reduce the required runtime, some fast algorithms [115, 117] have been proposed, but they produce some false CMDs. Ge *et al.* [118] proposed a bounding box-based algorithm followed by an exhaustive

search, which is significantly faster than simply using the exhaustive search. However, the algorithm is still rather slow for large objects. Furthermore, the centers of the maximal disks are, in general, not connected because the discrete distance map is used. Some points should be added between the centers of the maximal disks with a linking algorithm, so that the connectivity of the skeleton can be preserved. As this linking process may affect the uniqueness of the skeleton, it is limited to being used for compression purposes.

The connectivity of a skeleton is one of the important concerns for the skeletonization algorithms. Analytically, the boundary of a planar shape can be assumed to be composed of a finite number of real analytic curves. The continuity and path-connectedness of a skeleton can be proven by using the domain decomposition lemma [122]. In this lemma, a planar shape is continuously broken up into simpler pieces that are represented by the maximal disks. The centers of the disks are connected, and so is the skeleton. In the discrete case, it is assumed that the planar shape can be represented by polygons. With the use of the Voronoi diagram [110, 111], a continuous skeleton can also be obtained analytically by the relationship between the skeleton and the Voronoi diagram. In [109], it is shown that each Voronoi diagram is path-connected and the skeleton is a sub-graph of the Voronoi diagram, so the connectivity of the skeleton is preserved. Consequently, both the continuity and the path-connectedness of a skeleton in discrete and continuous representations have been proven theoretically. However, there is no connectivity criterion for the extraction of the skeleton for efficient implementation.

An abundant amount of work on skeletonization and its application has been conducted, in which a profound theoretical background of the skeleton from different aspects has been provided. The properties of the skeleton, including its thickness,

connectivity and reconstructability, have been investigated. In this paper, we propose a new skeletonization algorithm, which is fast, efficient and accurate, to extract a well-connected Euclidean skeleton based on a signed sequential Euclidean distance map. By using the connectivity criterion proposed in this paper, a skeleton point can be determined efficiently and independently by considering a set of points along the object's boundary, which are the nearest contour points to the pixel under consideration and its 8 neighbors.

This chapter is organized as follows. In Section 4.2, skeletonization based on the maximal disk is introduced and the definition of the medial axis transform is provided. Section 4.3 deals with the effect of discrete problems on the extraction of the skeleton. The concept of the width of a skeleton is established. The connectivity criterion for extracting a skeleton using the Euclidean metric is then proposed. Then, the algorithm is used to resolve different types of boundary segments. Section 4.4 illustrates and summarizes the implementation of the skeletonization algorithm. Finally, the experimental results and the summaries are given in Sections 4.5 and 4.6, respectively.

4.2 Skeleton based on the Maximal Disk

Suppose that the contour C of an object in an image is represented by a continuous closed curve. Inside the contour C, the planar shape F represents the content of the object. The corresponding skeleton S can be determined, as shown in Fig. 4.1. According to Blum's definition [100], the skeleton S of a planar shape is defined as the locus of the centers of the maximal disks contained inside the planar shape F. The medial axis transform can then be defined as follows:

Definition 1: The medial axis transform is the set of ordered pairs of the centers and radii of maximal disks in the planar shape F. That is,

$$SK(F) = \{(\boldsymbol{p},r) \in F | B(\boldsymbol{p},r) \in MaxDisk(F)\} \tag{4.1}$$

where $MaxDisk(F)$ is the set of all maximal disks in the planar shape F and $B(\boldsymbol{p},r)$ is a disk with radius r centered at the point $\boldsymbol{p}$.

It can be observed that almost all skeleton points are associated with at least two boundary points whose respective distances to the skeleton point are the shortest except the end points of the skeleton. These boundary points divide the contour into different separate segments. In Fig. 4.1, with a maximal disk centered at $\boldsymbol{p}$, the object's contour and the maximal disk touch each other at the points $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$. These two points divide the contour C into two segments A and B. If there exists at least one point along segment A and along segment B outside the maximal disk with a distance larger than a certain distance, which is called residual distance [123], the point $\boldsymbol{p}$ will be declared to be a skeleton point. The residual distance can be used to form a pruned skeleton. The magnitude of the residual distance influences the accuracy of the skeleton for the original object.

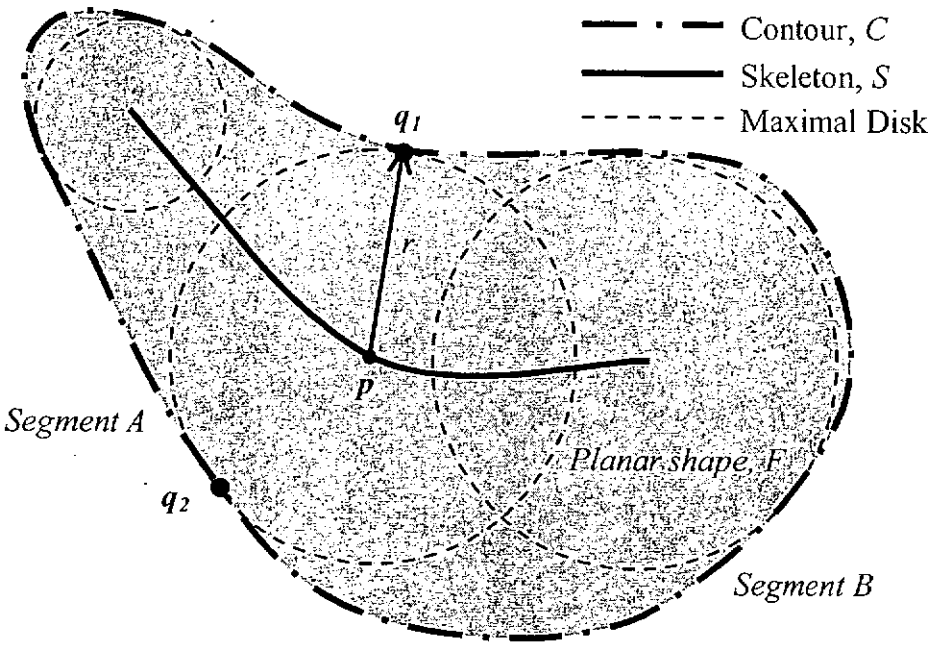


Figure 4.1: The definition of a skeleton.

4.3 The criterion for a skeleton point

4.3.1 The width of skeleton

An ideal skeleton is connected and has zero-width. A continuous boundary will produce a path-connected skeleton [122]. However, in real applications, the contour points and the skeleton points must be located at the pixel grids; this induces a lot of discrete problems. The skeleton may not pass through the pixel exactly. Hence, in practice, the skeleton has a non-zero width and all the pixels passed through by the ideal skeleton will be considered to be skeleton pixels. Consider a skeleton point $\boldsymbol{p}$, the corresponding maximal disk touches the boundary at points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$, as shown in Fig. 4.2. Points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ are the two nearest contour points with respect to point $\boldsymbol{p}$. The distance between points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ is denoted as D. Suppose that the true skeleton point $\boldsymbol{p}$ lies midway between the two adjacent points $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$. The width of the

skeleton can then be represented by a line segment $\overline{p_1 p_2}$, which is parallel to the line $\overline{q_1 q_2}$ and perpendicular to the direction of the skeleton. Due to the width of the skeleton, two values, r_1 and r_2, which are the distances of $|p_1q_1|$ or $|p_2q_2|$, and $|p_2q_1|$ or $|p_1q_2|$, can be obtained with the condition that $r_2 \geq r_1$. By using the Cosine Law and $\angle q_1q_2p_1 = \angle q_1p_2p_1$, we have:

$$\frac{D^2 + r_2^{\,2} - r_1^{\,2}}{2Dr_2} = \frac{w^2 + r_2^{\,2} - r_1^{\,2}}{2wr_2} \Rightarrow w = \frac{r_2^{\,2} - r_1^{\,2}}{D} = \frac{(r_2 - r_1)(r_2 + r_1)}{D} \qquad (4.2)$$

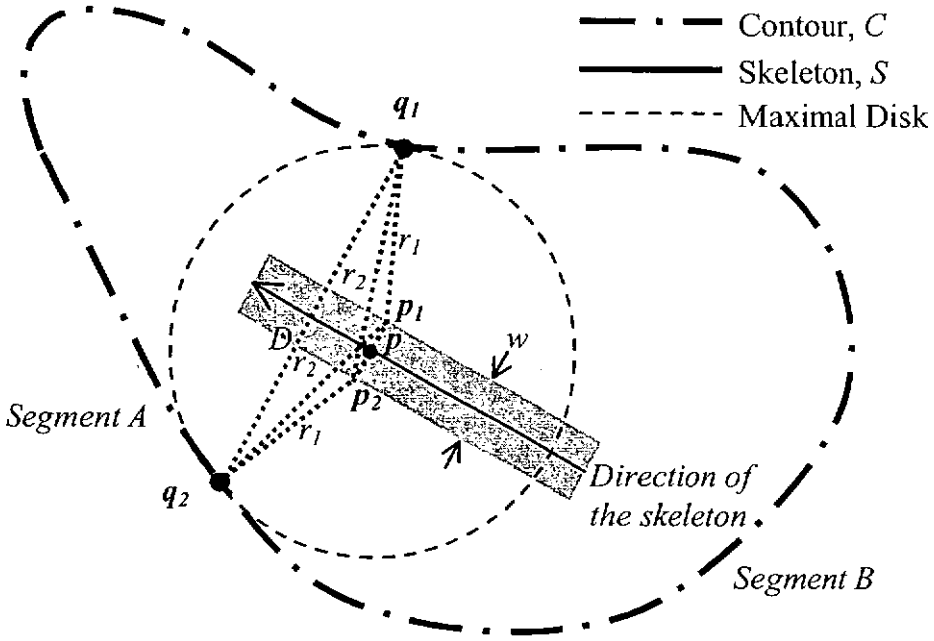


Figure 4.2: The width of a skeleton.

According to the above equation, the width of a skeleton is therefore proportional to the difference between the two local shortest distances and the radius of the maximal disk, and is inversely proportional to the value D, where D is the distance between the two nearest contour points q_1 and q_2. Consequently, a skeleton of non-zero even width can be obtained if the following criterion is satisfied:

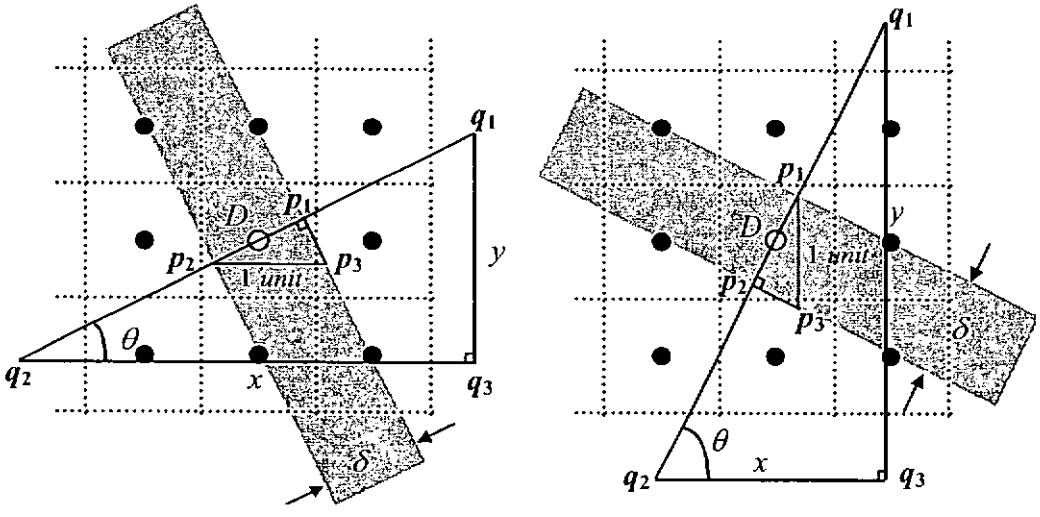
$$w = \frac{r_2^{\,2} - r_1^{\,2}}{D} \leq \delta, \qquad (4.3)$$

where δ is defined as a threshold to determine the maximum width of a skeleton and w represents the corresponding width of the skeleton at a particular point.

For the point p_1, the first shortest distance r_1 is $|p_1q_1|$ while the second shortest distance r_2 is $|p_1q_2|$. Similarly, for the point p_2, the first shortest distance r_1 is $|p_2q_2|$ while the second shortest distance r_2 is $|p_2q_1|$. For the point lying midway between the points p_1 and p_2, the two shortest distances are equal, so the value of w is equal to zero, which is the exact position of the skeleton point. Any point p along the line $\overline{p_1p_2}$ is also a skeleton point. Therefore, the criterion, as shown in equation (4.3), can be used to determine whether or not a point is a skeleton point with a given δ.

4.3.2 Connectivity of a skeleton on the square grid

The type of grid, such as square, hexagonal, etc., used for pixel position has different effect on the minimum width of the skeletons of an object under different orientations. Considering the square grid, the minimum width of a skeleton depends on its direction that is perpendicular to the line $\overline{q_1q_2}$. Figure 4.3 illustrates the effect of different orientations of a skeleton on the corresponding minimum width.



(a) $x > y$ (b) $x < y$

Figure 4.3: Effect of orientations on skeleton width.

Suppose that the coordinates of the two nearest contour points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ are (x_1, y_1) and (x_2, y_2), respectively. Since the direction of the line segment $\overline{q_1q_2}$ is perpendicular to the direction of the skeleton, the minimum width δ of the skeleton can be determined by the deviation of the line segment $\overline{q_1q_2}$. If the horizontal deviation x is greater than the vertical deviation y of the line segment $\overline{q_1q_2}$, as shown in Fig. 4.3(a), the minimum width δ of the skeleton can be obtained as x/D by considering the two similar triangles, $\Delta \boldsymbol{q_1q_2q_3}$ and $\Delta \boldsymbol{p_3p_2p_1}$. Similarly, if the horizontal deviation x is less than the vertical deviation y of the line segment $\overline{q_1q_2}$, as shown in Fig. 4.3(b), the minimum width δ of the skeleton can be obtained as y/D by considering the two similar triangles, $\Delta \boldsymbol{q_1q_2q_3}$ and $\Delta \boldsymbol{p_1p_3p_2}$. The minimum width δ of the skeleton can therefore be set as follows:

$$\delta = \frac{\max(x, y)}{D}, \tag{4.4}$$

where $x = abs(x_2\text{-}x_1)$ and $y = abs(y_2\text{-}y_1)$, and the value D is the distance between the two nearest contour points $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$.

A skeleton of non-zero width with threshold δ is illustrated in Fig. 4.4. The thick solid line represents the ideal skeleton. All these pixels passed over by this ideal skeleton are considered to be a skeleton point. Therefore, the connectivity criterion for the square grid can be obtained based on equations (4.3) and (4.4) as follows:

$$\frac{r_2^2 - r_1^2}{\max(x, y)} \le 1 \qquad \text{and} \qquad D^2 \ge \rho, \tag{4.5}$$

where $x = abs(x_2\text{-}x_1)$ and $y = abs(y_2\text{-}y_1)$, and ρ is a non-zero threshold used to determine the minimum distance between the two nearest contour points.

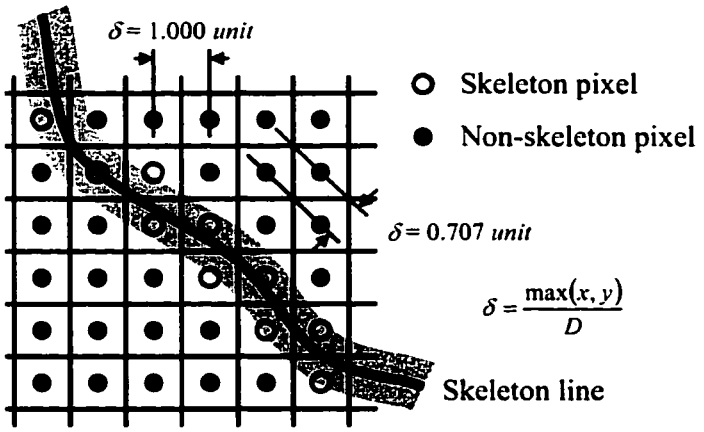


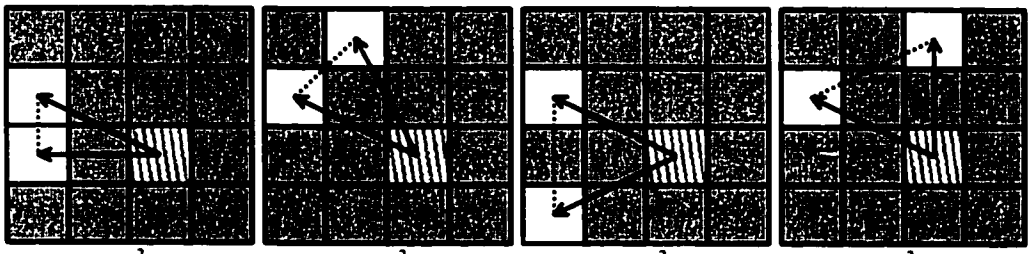
Figure 4.4: The extracted skeleton on a square grid.

According to the connectivity criterion, we can observe that only the maximum of x and y, which are the horizontal and vertical deviations of the line segment $\overline{q_1 q_2}$, and the two lengths of the line segments, $\overline{pq_1}$ and $\overline{pq_2}$, are considered in the determination of a skeleton point. This criterion can be extended to the contour surface of a 3D object when considering the minimum width of the connected skeleton of the 3D object. Based on the connectivity criterion, the minimum widths for a connected skeleton in 2D case are 0.707 and 1.000 units, respectively, in the diagonal and vertical/horizontal directions. The true skeleton line, which is represented by the dark line in Fig. 4.4, may not pass through the skeleton pixel exactly. The skeleton with non-zero width is represented by the gray regions, which include all the connected skeleton pixels satisfying the connectivity criterion.

4.3.3 Types of boundary segments

Since the skeleton points are determined based on the connectivity criterion, simple contour segments that are either in a horizontal/vertical or a diagonal direction will cause some points to be mistaken for skeleton points. These points can be removed by simply determining the corresponding segment type. For example, Fig.

4.5 illustrates the values of the square distances D^2 between the two nearest contour points. The pixel under consideration is represented by a shaded square, while the corresponding two nearest contour points are represented by white squares. With equation (4.5), only those pixels having the corresponding distance D^2 greater than the threshold ρ will be considered as skeleton points. Two adjacent 8-connected contour points have D^2 = 1 and 2 for vertical/horizontal and diagonal boundary segments, respectively, as shown in Fig. 4.5(a) and Fig. 4.5(b). In order to exclude the pixels in these two cases as skeleton points, we set ρ to be 4. Consequently, the pixels under consideration, as shown in Fig. 4.5(c) and Fig. 4.5(d), will be considered as skeleton points. Actually, if a larger value is set for ρ, the number of redundant branches in the skeleton will be reduced.



(a) D^2=1 (b) D^2=2 (c) D^2=4 (d) D^2=5

Figure 4.5: The different types of boundary segments.

4.4 The skeletonization algorithm

To determine whether a pixel point is a skeleton point, the corresponding nearest contour point for each of the 8 neighboring points will be determined, and the connectivity criterion will then be applied to this set of 8 contour point pairs. The relative positions of the nearest contour points for each pixel can be obtained by using the signed sequential Euclidean distance (8SSED) map. The nearest contour points for each of the 8-neighboring points of the point under consideration can be obtained by subtracting the relative position of its nearest contour point from the relative position

of its neighborhoods. The nearest contour points of the pixel under consideration and one of its 8 neighbors then form a point pair. If any one of the point pairs satisfies the connectivity criterion, the pixel can be declared a skeleton point. Otherwise, if all the 8 point pairs fail to fulfill the connectivity criterion, the pixel is not a skeleton point. In summary, the procedure of our proposed algorithm is described as follows:

Compute the signed sequential Euclidean distance (8SSED) map

set P to be the first pixel

do

set $Q = DM(P)$ to be the nearest contour point of the pixel P according to the distance map where $DM(.)$ represents the 8SSED map.

find the nearest contour points of its 8 neighbors P_i

$Q_i = DM(P_i) + (x_i, y_i)$,

where (x_i, y_i) is the relative position of the neighborhood i with respect to the pixel P *and* $i = 1,2,\ldots,8$ is the index of the neighborhoods.

The eight point pairs are formed by the nearest contour points of both the pixel P and its 8-neighbors, which are denoted as

(Q, Q_i) where $i = 1,2,\ldots,8$.

Apply the connectivity criterion:

$D^2 = |Q_i - Q|^2 \geq \rho$ and $|Q_i|^2 - |Q|^2 \leq \max(X(Q_i - Q), Y(Q_i - Q))$

where $X(p)$ and $X(p)$ represent the x and y coordinates of the point.

If one of the point pairs, (Q, Q_i), satisfies the connectivity criterion, **then**

the pixel P is a skeleton point.

until the last pixel of the image checked

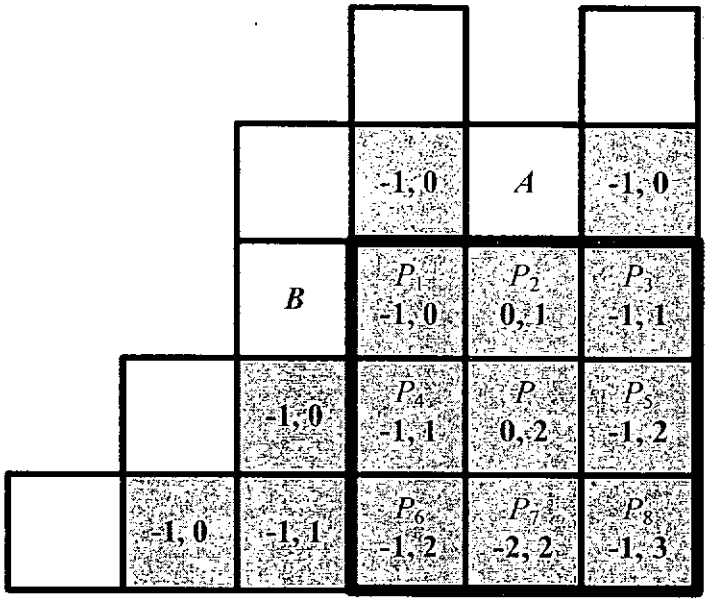


Figure 4.6: An example of the skeletonization procedure.

Figure 4.6 illustrates an example of skeletonization based on the above procedure. The white square represents the boundary points and the gray squares represent the pixels inside the boundary. The pixel under consideration is P, and its 8 neighbors are represented as P_1, P_2, P_3, P_4, P_5, P_6, P_7 and P_8. The relative positions of the nearest contour point for each pixel are obtained by using the signed sequential Euclidean distance transformation. The nearest contour point to the pixel P is located at the position A, which is (0,2) relative to its position. Similarly, the nearest contour points of its eight neighbors (P_2, P_3, P_5, P_8) and (P_1, P_4, P_6, P_7) are at positions A(0,2) and B(-2,1) relative to the pixel P, respectively. For example, the nearest contour point of P_1 is calculated as (-1,0)+(-1,1) = (-2,1), which is its position relative to the pixel P. Since the nearest contour points of some of the 8 neighbors are the same as the nearest contour point of the pixel P, the exact number of point pairs considered in this case is one, which is formed by the points A(0,2) and B(-2,1). Based on this point pair, the square distance D^2 between these two nearest contour points is $|(0,2)-(-2,1)|^2$ = 5 which is greater than the threshold $\rho = 4$. The width of the skeleton, w, is [(-2+1)-(0+2)] / max(2,1) = 0.5, which is less than one, and thus satisfies the connectivity

criterion. Consequently, the pixel P is declared an Euclidean skeleton point. As the square root operation is not required in the procedure, this algorithm is very efficient.

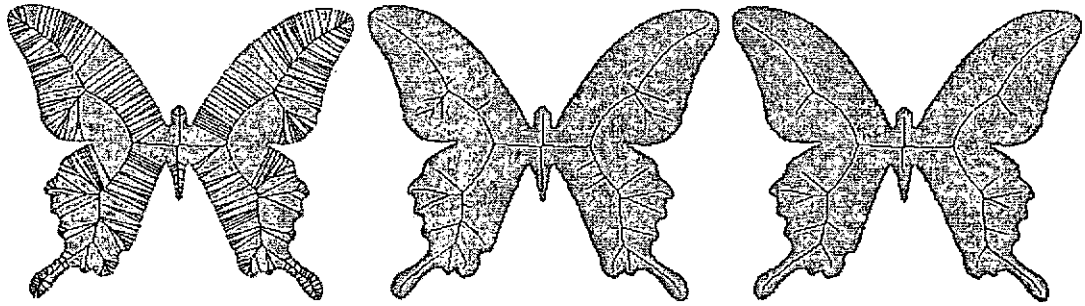
4.5 Experimental Results

In the experiment, the shapes in an image can be extracted by using a contour extraction method called the adaptive snake method [124] or any edge follower method. Based on the extracted contours, the skeletons of the shapes are extracted using our proposed algorithm. The skeletonization performance and the complexity of our proposed algorithm are evaluated in this section. The effect of the threshold values ρ on the extraction of a skeleton will be illustrated. Then, the complexity of our proposed algorithm is compared to some maximal disk extraction algorithms. The experiments were conducted on a Pentium II 400MHz PC. The signed sequential Euclidean distance map (8SSED) [121] was used in all the methods to be compared.

4.5.1 The effect of the threshold ρ

The effect of different threshold values ρ on the skeleton of an object is illustrated in Fig. 4.7. The whole skeleton extracted by using our proposed algorithm is shown in Fig. 4.7(a), which uses the threshold value $\rho = 4$. When the value of the threshold ρ increases, a smaller number of skeleton points will be extracted and as a result there will be fewer branches in the skeleton. The effect of increasing the threshold ρ is similar to that of pruning. However, this pruning procedure may cause the extracted skeleton to disconnect from some spurious points, as shown in Fig. 4.7(b), because there is no guarantee of extracting a connected skeleton for a larger value of ρ. Since many pruning methods [110, 111, 125] have been proposed, we can

apply one of them after we have applied our proposed algorithm to extract the Euclidean skeleton. Alternatively, we can use a larger threshold ρ as a simple pruning method to extract the pruned skeleton, as shown in Fig. 4.7(c).



(a) $\rho = 4$ (b) $\rho = 50$ (c) $\rho = 150$

Figure 4.7: The skeletons using different thresholds ρ with an image of size 320×240.

4.5.2 The computational requirements of the algorithms

In this section, we will investigate the computational complexity of our proposed method. As mentioned previously, there are many approaches to extracting a skeleton, and it is impossible to compare our proposed method with each of the existing methods. Therefore, we will compare our algorithm to the following existing methods, namely, method 1, a neighborhood algorithm followed by an exhaustive algorithm [118]; method 2, a modification of method 1 with the use of an exhaustive algorithm followed by a bounding box-based algorithm [118]; and method 3, a method based on a criterion with a residual distance [123].

Both methods 1 and 2 use the neighborhood algorithm to eliminate most of the points in the planar shape by means of the inclusion test procedure [118]. In order to determine the true maximal disks, the exhaustive algorithm is used to screen out the rest of the non-CMDs. The testing procedure is very computationally intensive because all the possible sets of points are used in the inclusion test procedure. By applying the bounding box-based algorithm before the exhaustive algorithm, the computational time for method 2 can be reduced and a smaller number of false CMDs

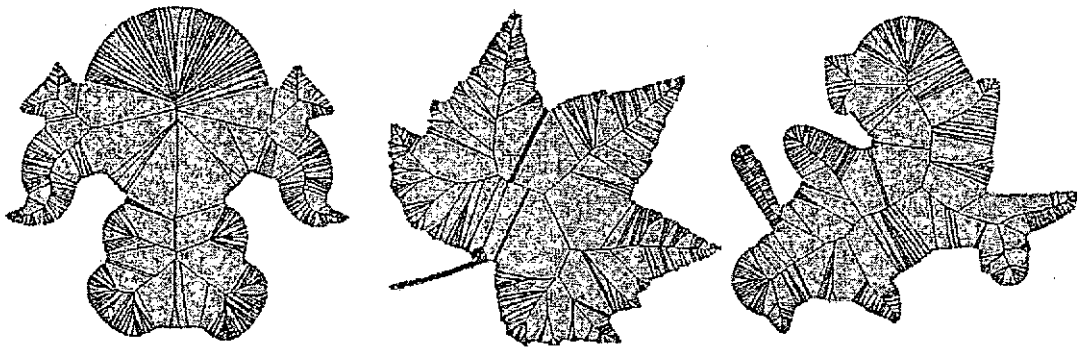
will be generated. Both methods 1 and 2 can be used to reconstruct the original contour exactly, and so be used as a means for compression, but they require a linking algorithm [118] to obtain a connected skeleton. Method 3 can extract a pruned connected skeleton directly without using a linking algorithm.

Input images		8SSED (sec)	Method 1: Neighborhood + Exhaustive Method (sec)	Method 2: Neighborhood + Bounding Box-based + Exhaustive Method (sec)	Method 3: Criteria of a skeleton (sec)	Proposed Method (sec)
#1	320×240	0.0700	5.6680	5.5380	4.3770	0.1000
	640×480	0.2810	65.9650	58.3340	34.2800	0.1610
	1280×960	1.1520	-	-	-	0.6510
#2	320×240	0.0700	5.1570	5.0170	3.2450	0.1000
	640×480	0.2810	61.1280	55.3400	28.6110	0.1500
	1280×960	1.1520	-	-	-	0.5910
#3	320×240	0.0700	5.0570	5.5680	3.7650	0.1200
	640×480	0.2810	28.6110	28.6110	29.3020	0.1810
	1280×960	1.1520	-	-	-	0.57000

Table 4.1: A comparison of different methods with different images.

As shown in Table 4.1, the runtimes for methods 1, 2 and 3 are longer than that for the proposed method. Theoretically, the complexity of an exhaustive algorithm is in the order of $O_1(n^2)$, where n is the number of points in the planar shape. With the neighborhood method, the complexity in terms of testing and finding the maximal disks in the neighborhood of each point in the planar shape is in the order of $O_2(kn)$, where k is the size of the neighborhood. The computation time is reduced significantly compared to the exhaustive algorithm due to the use of the bounding box-based algorithm to eliminate the irrelevant disks. For method 3, the complexity of the algorithm is in the order of $O_3(mn)$, where m is the total number of contour points. This method computes the distances to the contour instead of checking the inclusion and overlapping of the maximal disks. For the proposed method, its computational complexity is in the order of $O(8n)$ only, because it only considers the pixel under consideration and its 8 neighbors by means of the connectivity criterion. In addition,

the square root operation is not needed in this algorithm. In Table 4.1, we can observe that the runtimes required by the three methods increase tremendously when the image size increases. For our proposed method, the runtime is linearly proportional to the number of pixels in the planar shape. Figure 4.8 illustrates some of the extracted skeletons based on our algorithm.



(a) Example #1 (b) Example #2 (c) Example #3

Figure 4.8: The skeletons based on our proposed algorithm with an image of size 640×480.

4.6 Summary

With the use of the connectivity criterion proposed in this chapter, an accurate, simple and efficient algorithm for the extraction of a well-connected Euclidean skeleton is devised with the use of the signed sequential Euclidean distance map. The nearest contour points of the pixel under consideration and its 8 neighbors are generated to form a set of 8 point pairs, which are then used to determine whether the pixel is a skeleton point. This method can generate a connected Euclidean skeleton without requiring square root operation, a linking algorithm or any iteration. The complexity of this algorithm is linearly proportional to the number of the pixels in an image. The computational complexity is $O(8n)$ where n is the total number of pixels in the image.

Chapter 5

Maximal disk-based Histogram for Shape Retrieval

After extracting the skeleton of an object, a set of maximal disks can be obtained using our proposed skeletonization technique with the simple pruning algorithm as described in the previous chapter. In this chapter, we propose a robust and efficient representation scheme for shape retrieval, which is based on the normalized maximal disks used to represent the shape of an object. The logarithm of the radii of the normalized maximal disks is used to construct a histogram to represent the shape. The retrieval performance of this maximal disk-based histogram approach is compared to other methods, including moment invariants, Zernike moments, and curvature scale-space. Experimental results show that our proposed representation scheme outperforms the other methods under affine transformation, different distortions and noise levels.

5.1 Introduction

In designing a content-based image retrieval system [4,128], retrieval efficiency and accuracy are the two most important issues to be considered, especially for large multimedia databases. Research on content-based image retrieval has attracted great

interest and attention over the past decade. Feature selection and extraction is the most important step for image retrieval. To describe different types of images, different visual features, such as shape, texture, color, etc., should be considered.

Shape retrieval is one of the most challenging areas covered by the literature that deals with retrieval based on shape similarity. There are different traditional approaches for representing the shape characteristics, such as Fourier descriptors [73,74], moment invariants [60,64,65], Zernike Moments [4,68,69], etc. However, these methods are sensitive to noise and distortion. The curvature scale-space (CSS) method [22,27,28] is one of the most successful contour-based features used as a shape descriptor, and has been a shape descriptor in the MPEG-7 standard [14]. The representation of the curvature scale-space has been shown to be robust under the similarity transformations such as scaling, orientation changes, translation, and even shearing. However, the method is easily affected by distortion or occlusion. A shape descriptor which is compact, robust to noise and distortion, and provides an accurate representation of an object's shape is highly desirable.

Histogram analysis has been a widely used approach for content-based image/video retrieval. Even though histograms are commonly used for color description, they are quite limited in descripting shapes. The advantages of the use of histograms are due to their robustness, effectiveness, and their implementation and computational simplicity. In this chapter, a robust and efficient representation scheme based on the distribution of the maximal disks used to represent a shape is proposed. The maximal disks of a shape are extracted by a fast skeletonization technique with a pruning algorithm [123], which has been presented in Chapter 4. The robust feature is formed by the logarithm of the radii of the normalized maximal disks of an object. A

histogram based on the distribution of the radii is constructed and used for shape matching.

This chapter is organized as follows. In Section 5.2, we will give a detailed description of our proposed shape descriptor. Experimental results are illustrated in Section 5.3, where the performance of our proposed scheme is compared to other shape representation schemes, namely moment invariants, Zernike moments, and curvature scale-space. The robustness and accuracy of our proposed shape descriptor are also evaluated. Finally, a summary is given in Section 5.4.

5.2 A new Shape Descriptor

The shape of an object can be obtained by contour extraction or image segmentation methods, such as the adaptive snake method, which has been presented in Chapter 3, or any edge follower method. The extracted contour or shape is then encircled completely by a minimum disk, as shown in Fig. 5.1(a). The radius of this minimum disk is denoted as r_{max}. In order to obtain a descriptor independent of scale, the shape is normalized such that the radius r_{max} is equal to a fixed value $N/2$. In other words, the object forms an image of size $N \times N$. The boundary of the object is then re-sampled and represented by 512 evenly spaced points. In order to reduce the effect of noise along the contour of the shape, the contour is smoothed by converting the contour points to the frequency domain. Then, 75% of the highest frequency components are discarded. Based on the smoothed contour, the skeleton of the shape is then extracted using a fast skeletonization method with a pruning algorithm. The skeletonization method uses the loci of those maximal disks inside an object to represent its skeleton. In our scheme, the distribution of these radii is used to represent

the object. Figure 5.1(b) shows the result after skeletonization, which contains many branches. This is due to the discrete representation of the shape. In order to remove the branches, a pruning algorithm is applied, and the pruned result is shown in Fig. 5.1(c). The radii of those maximal disks associated with the pruned skeleton are then used to form a histogram for shape representation. The corresponding histogram for the skeleton in Fig. 5.1(c) is shown in Fig. 5.1(d).

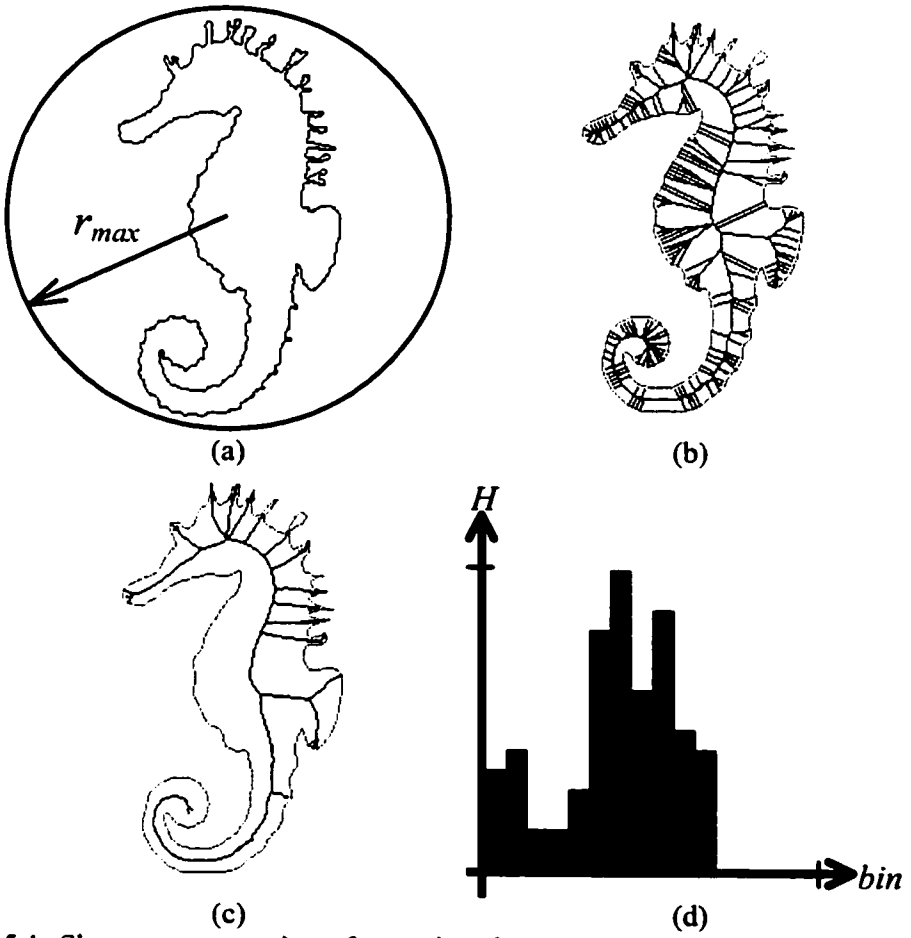


Figure 5.1: Shape representation of a marine shape: (a) the contour, (b) the skeleton, (c) the pruned skeleton, and (d) the maximal disk-based histogram.

5.2.1 Maximal disk-based histogram

Similar shapes will result in similar histograms for the radii of the maximal disks. The maximal disks are extracted by our skeletonization algorithm. Each

representative radius of the maximal disks and its corresponding percentage form a pair of attributes that can be used to describe the shape characteristics in an image. Basically, the histogram is formed by the maximal disks along the skeleton of a shape, so it is independent of the translation and orientation of the object. The radii of those maximal disks used to represent the object are normalized by the radius r_{max}, so histogram formed on the basis of these normalized radii is also invariant to the object's scale. Maximal disks with a larger radius capture the global shape, while those with a smaller radius represent the fine details. As the number of maximal disks with small radii is much more than that of large radii, the allocation of the normalized radii to the histogram bins should be performed in a non-linearly manner. Therefore, the radius r of a maximal disk is normalized by r_{max} and then quantized to a bin value, $bin(r)$, of the histogram as follows:

$$bin(r) = \frac{1}{2}\log_{10}\left(\frac{r}{r_{max}} \times 100\right) \times N_{bin} \tag{5.1}$$

where N_{bin} is the maximum number of bins and $bin(r)$ is in the range of $[1, \ldots, N_{bin}]$. The histogram of a shape with maximal disks of radii in the range of $[1, R]$, where $R \leq r_{max}$, is defined as follows:

$$H(bin(r)) = \frac{count(bin(r))}{n}, \qquad r = 1, 2, \ldots, R \tag{5.2}$$

where $count(bin(r))$ represents the number of maximal disks assigned to $bin(r)$ of the histogram, n is the total number of maximal disks for the object, and $\sum H(bin(r)) = 1$.

In order to have a compact representation of the shape descriptor, the histogram is set to have 16 bins, and each bin is represented by 4 bits. Therefore, 64 bits are used to represent a histogram or a shape. The 4-bit value is used to represent the number of

maximal disks with a particular range of radii in the corresponding bin. The maximal disk-based histogram of a marine shape is shown in Fig. 5.1(d).

5.2.2 Histogram Comparison

The difference between two histograms is computed by using the quadratic-form distance measure [13,127], which takes the cross-bin similarity into account when measuring the distance. Suppose that the maximal disk-based histograms of two images, P and Q, are represented as $\mathbf{H}_p = (H_p(1),\ldots H_p(N_{bin}))^T$, and $\mathbf{H}_q = (H_q(1),\ldots H_q(N_{bin}))^T$, respectively. The quadratic-form distance measure D is then defined as follows:

$$D(P,Q) = \sqrt{\left(\mathbf{H}_p - \mathbf{H}_q\right)^T \mathbf{A}\left(\mathbf{H}_p - \mathbf{H}_q\right)} \tag{5.3}$$

where $\mathbf{A}=[a_{ij}]$ is a similarity matrix that incorporates the cross-bin similarity, and a_{ij} denotes the similarity between bins i and j.

Each bin of the histogram corresponds to a particular range of logarithm radii of the normalized maximal disks, and the similarity matrix represents the correlation between the bin under consideration and its neighbors. The similarity matrix **A** used is defined as follows:

$$a_{ij} = \exp\left(-\left(\frac{i-j}{\sigma}\right)^2\right) \tag{5.4}$$

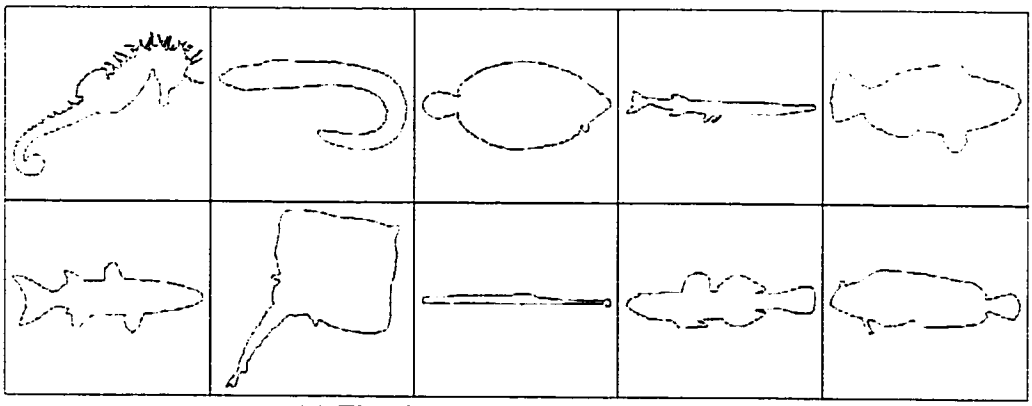
where $i, j = [1,\ldots,N_{bin}]$ and σ is the variance of the similarity parameter. The value of this distance measure D can then be used to determine the similarity between the two shapes.

5.3 Experimental Results

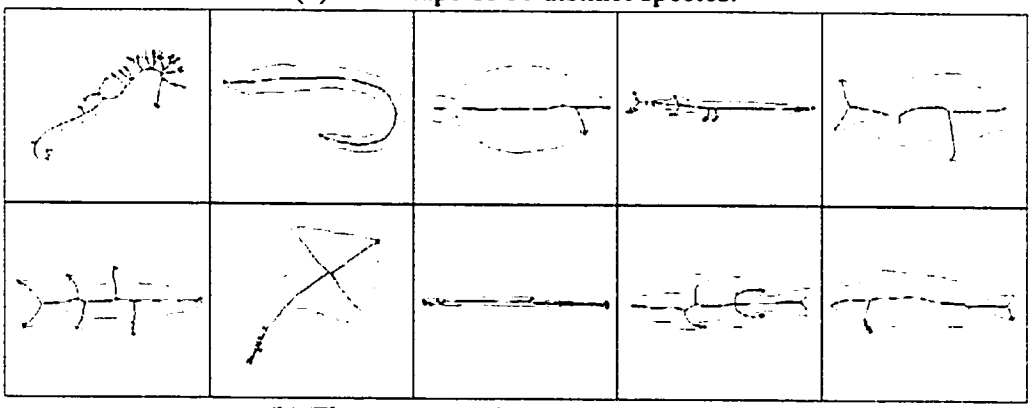
The retrieval performance of our proposed algorithm is compared to the traditional algorithms including Zernike moments [4], moment invariants [60] and curvature scale-space [14,27]. Part of the SQUID database (Shape Queries Using Image Databases) [27] was used in the experiments. The effects of affine transformation, noise, and distortion on the performances of the different algorithms were investigated and evaluated. The experiments were conducted on a Pentium 4 2.4GHz PC.

5.3.1 Generation of image databases

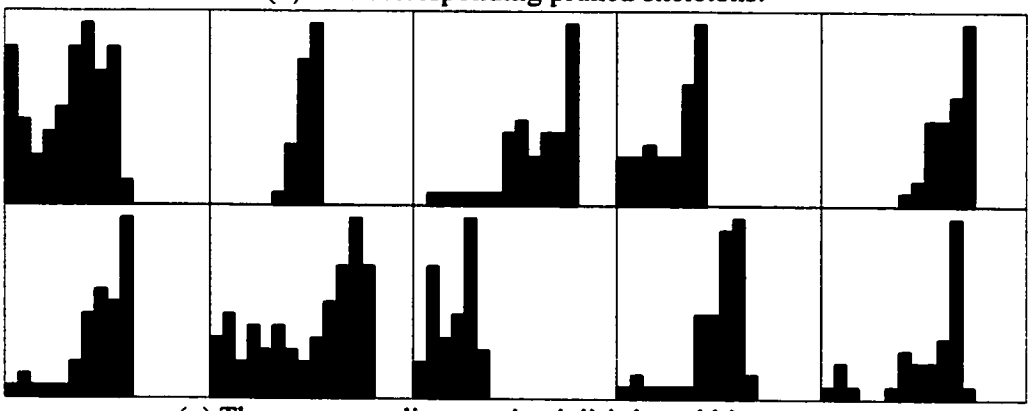
In our experiments, 10 distinct species of marine creatures were selected from the SQUID database. Each of the marine images is represented as a set of closed-contour points, then converted to binarized images, and normalized to a size of 256×256. The 10 different species form 10 classes, and each class contains at most 8 similar species. In our system, this makes up a database consisting of 76 images in all, which is similar to the databases constructed in [27]. One shape from each of the classes, its corresponding skeleton, and the maximal disk-based histogram are shown in Fig. 5.2. We can observe that the histograms of the distinct species are different.



(a) The shape of 10 distinct species.



(b) The corresponding pruned skeletons.



(c) The corresponding maximal disk-based histograms.

Figure 5.2: The 10 representative shapes used in the experiments.

Four types of databases were generated by applying different affine transformations, different levels of noise added to the contour points, different distortions added to the contour points in the original database, and all of the above variations. The first database was generated by using scaling factors between 0.8 and

0.9, and rotated by an arbitrary angle. The second database was generated by using 10 levels of noise between 0.0 and 18.0. The third database was generated by randomly adding different distortions to the contour points of the marine images. The fourth database was generated by using different scaling factors, rotations, and noise levels, and adding different distortions to the contour points. Hence, the number of classes in each of these databases is the same. The query set used in the experiments was formed by all the images within the database. The runtime required for extracting the proposed shape descriptors for 760 images in each of the databases is approximately 24 seconds, and the average runtime for retrieval for each query is about 168 ms.

5.3.2 Comparisons of different shape descriptors

In order to visualize the robustness of the different shape descriptors, the effects of affine transformation, noise, and distortion are investigated in the following sections.

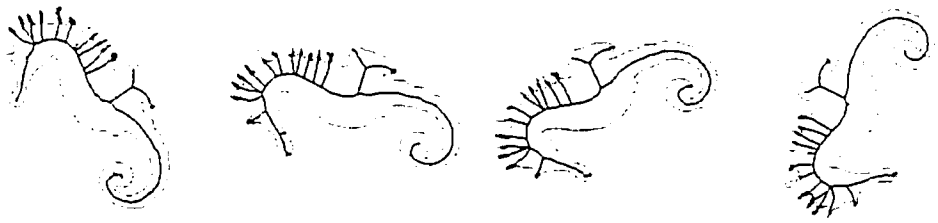
5.3.2.1 Effect of affine transformations

The first database generated by using scaling factors between 0.8 and 0.9, and rotated by an arbitrary angle was used. Suppose that $C=\{x, y\}$ represents the original contour and C' represents the contour under affine transformation. Then

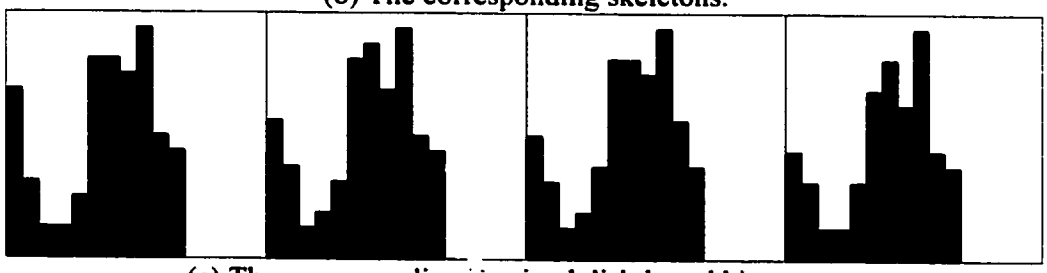
$$C' = \left\{ x' = s\left(x\cos\theta + y\sin\theta\right),\quad y' = s\left(x\sin\theta - y\cos\theta\right) \right\} \tag{5.5}$$

where s is the scaling factor and θ is the arbitrary angle. Figure 5.3 shows the maximal disk-based histograms with the object under different orientations and scales. It can be observed that the skeletons extracted are robust under the affine transformations, so the histograms of the images are very similar to each other.

(a) The simulated images with different orientations and scales.



(b) The corresponding skeletons.



(c) The corresponding maximal disk-based histograms.

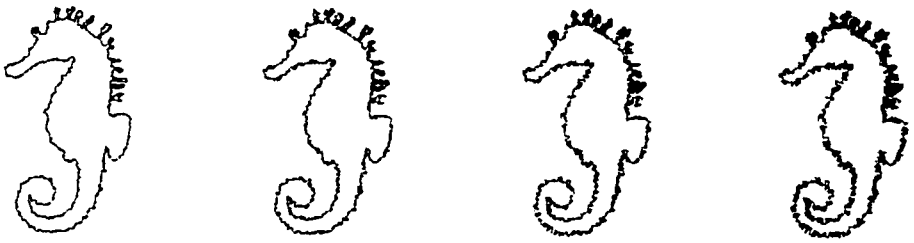
Figure 5.3: The maximal disk-based histograms of a marine shape with different orientations and scales.

5.3.2.2 Effect of noise levels

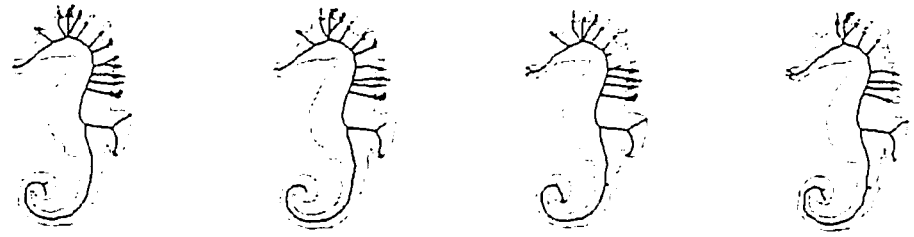
The second database was generated by using 10 different noise levels between 0.0 and 18.0. Suppose that $C=\{x, y\}$ represents the original contour and C' represents the contour with noise added along the boundary. Then

$$C' = \{x' = x + noise \times random(\ ),\quad y' = y + noise \times random(\ )\} \qquad (5.6)$$

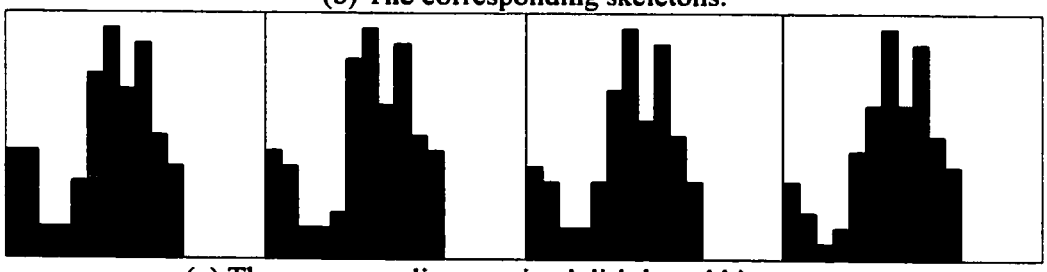
where *random*() is a random number in the range of {-1,1} and *noise* is the level of noise variance. Figure 5.4 shows the histograms using our proposed representation scheme for a shape under different levels of noise. It can be observed that the skeletons extracted are insensitive to noise. With the use of a pruning algorithm, the histograms with different noise levels are similar to each other.



(a) The simulated images with different noise levels (4, 6, 8, 10).



(b) The corresponding skeletons.

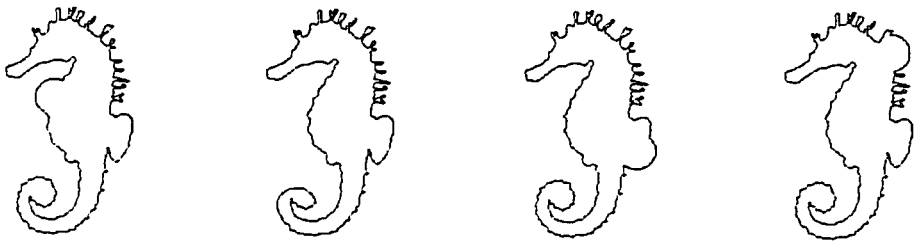


(c) The corresponding maximal disk-based histograms.

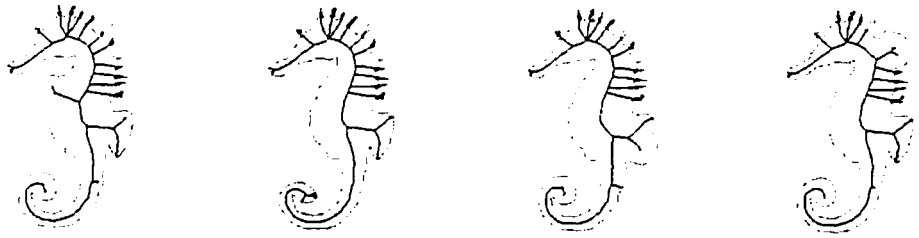
Figure 5.4: The maximal disk-based histograms of a marine shape under different noise levels.

5.3.2.3 Effect of distortions

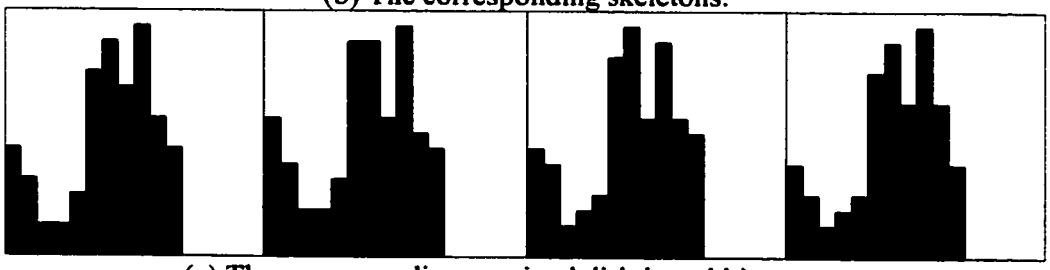
The third database was used to investigate the effect of distortions on the different shape description schemes. The distortion is defined as a disk with a fixed radius r arbitrarily added along the contour of the shape, and we set r=25 used in the experiments. Figure 5.5 shows the maximal disk-based histograms under different distortions. It can be observed that the skeletons extracted are the same except for those distorted parts close to the distorted regions of the shape. Since the effect of distortion is very localized to the skeleton, the respective histograms with different distortions are only changed slightly, and the corresponding histograms are similar to each other.



(a) The simulated images with different distortions.



(b) The corresponding skeletons.



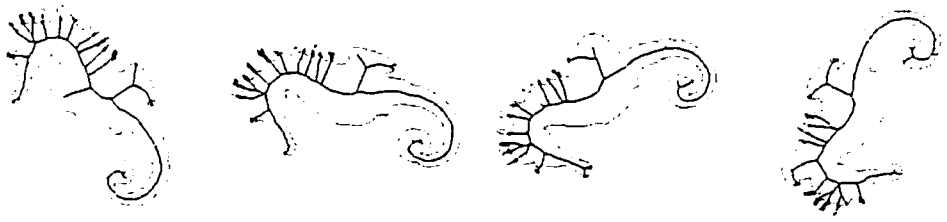
(c) The corresponding maximal disk-based histograms.

Figure 5.5: The maximal disk-based histograms of a marine shape under different distortions.

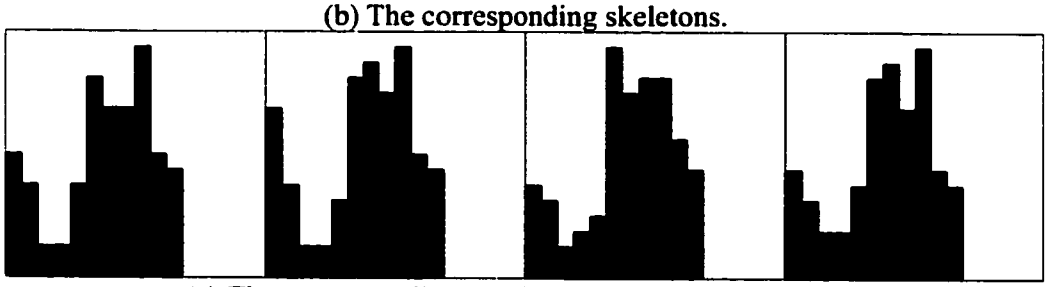
5.3.2.4 Effect of combing all the variations

The fourth database, which was generated by combining all the previous variations in the object shapes, was used to evaluate the retrieval performances of the different retrieval schemes. Figure 5.6 shows the maximal disk-based histograms of an object under different combinations of variations. It can be observed that the effects of the variations on the extracted skeleton are localized and limited, so the histograms are similar to each other.

(a) The simulated images with different types of variations.



(b) The corresponding skeletons.



(c) The corresponding maximal disk-based histograms.

Figure 5.6: The maximal disk-based histograms of a marine shape under different variations.

5.3.3 The retrieval performances

To measure the retrieval performances of the different shape representation schemes, the precision and recall rates for the different retrieval approaches are measured. The precision rate and recall rate are defined as follows:

$$\text{Precision Rate} = \frac{\text{number of relevant images selected}}{\text{total number of retrieved images}} \tag{5.7}$$

$$\text{Recall Rate} = \frac{\text{number of relevant images selected}}{\text{total number of similar images in the database}} \tag{5.8}$$

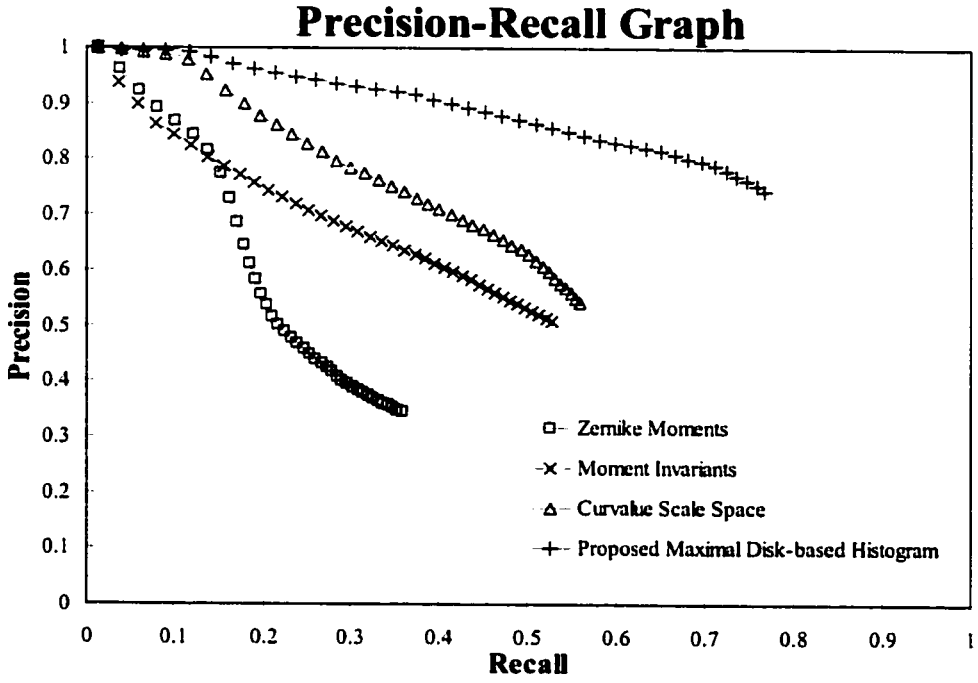
In our experiments, the 80 best-matched marine images were retrieved from the different databases for each query image. The average values of the precision rate and recall rate were computed and plotted as precision-recall graphs. The horizontal and

vertical axes represent the recall rate and the precision rate, respectively. The top-left point of the graph corresponds to the precision/recall values when only one retrieved image is considered, while the bottom right point corresponds to the precision/recall values when considering the entire answer set with 80 retrieved images. The Zernike moments, the moment invariants, and the curvature scale-space, which were the features used in other content-based retrieval systems, were compared to our proposed feature based on the precision-recall graphs. Also, the effects of different numbers of bins and different quantization levels were investigated by means of the precision-recall graphs.

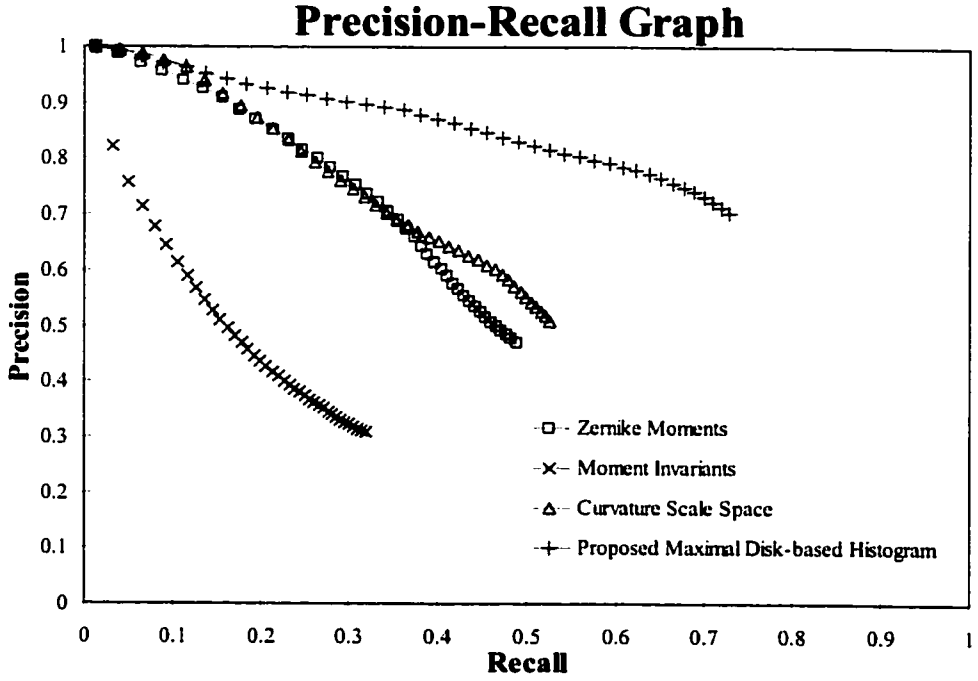
5.3.3.1 Comparisons based on different algorithms

In order to evaluate the accuracy of our proposed algorithm, the precision-recall graphs based on the Zernike moment, the moment invariants, the curvature scale-space, and the proposed features, were measured. Figure 5.7 shows the precision-recall graphs of the different algorithms based on the four databases. Figure 5.7(a) shows that the performance of the Zernike moments falls dramatically when the shapes are rotated and scaled, as compared with that of the other approaches. Figure 5.7 (b) shows that the performance of the moment invariants under noise is dropped when compared with others, due to its sensitivity to noises. Figure 5.7 (c) shows that the performance of the curvature scale-space when the shapes are distorted also falls dramatically compared to the others. Our proposed scheme has the best retrieval performances under affine transformation and different noise levels. Under distortion, our proposed algorithm still outperforms the moment invariants and the curvature scale-space, but is only very slightly inferior to that of the Zernike moments for a small answer set. Figure 5.7(d) shows the precision-recall graphs of the different

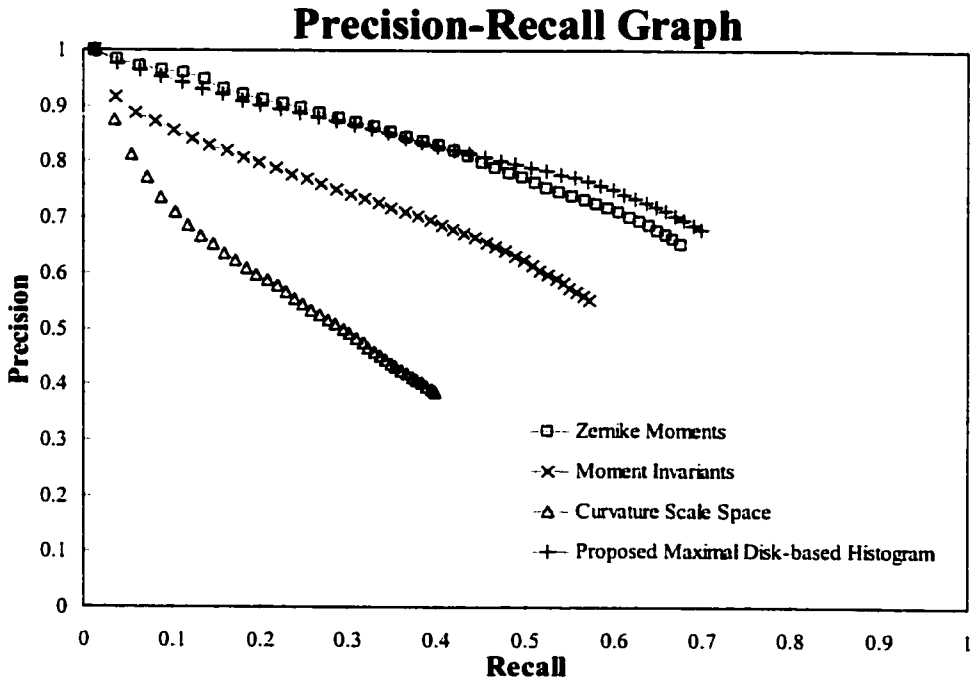
algorithms based on the fourth database. Our proposed scheme has the best retrieval performance while the performances of the other algorithms are dropped as they are sensitive to either noise, distortion or affine transformation.



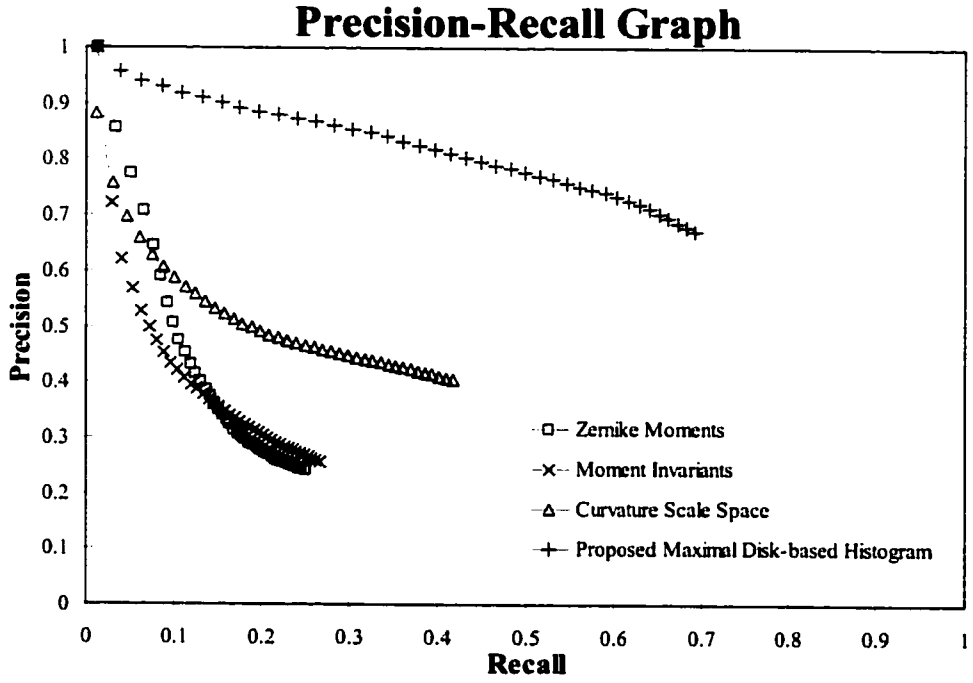
(a) Shapes are rotated and scaled.



(b) Noises are added to the shapes.



(c) Shapes are distorted.

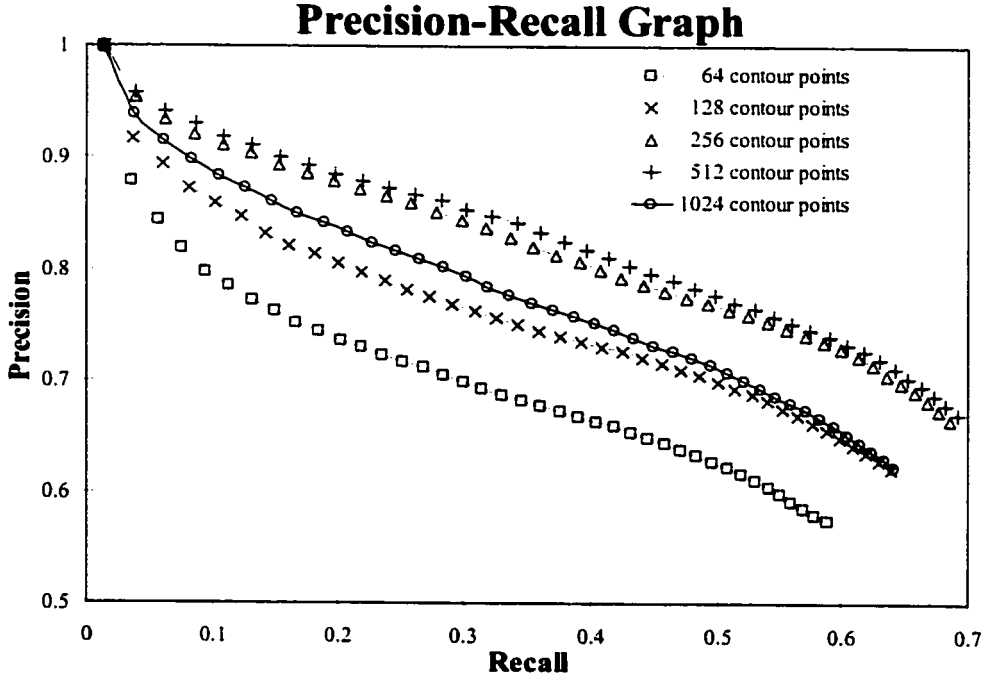


(d) Shapes are rotated, scaled, and distorted with noises.

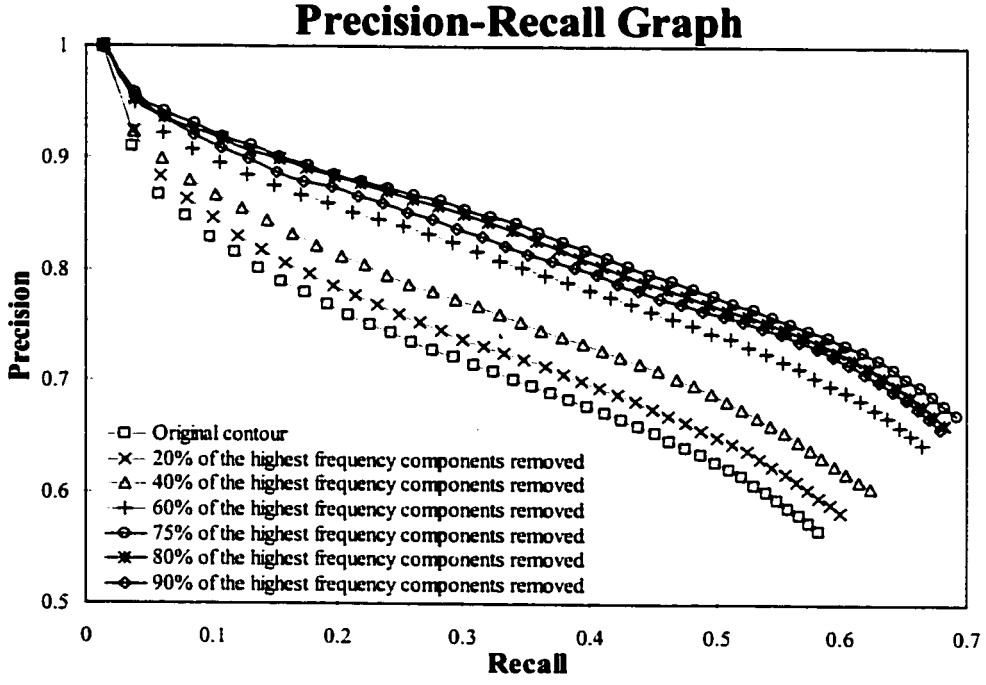
Figure 5.7: Comparison of the precision-recall graphs with 80 retrieval images using the four different shape databases: (a) affine transformed, (b) different noise levels, (c) different distortions, and (d) all the above variations.

5.3.3.2 Performances with different numbers of contour points and portions of high frequency components removed

To optimize the performance of our proposed scheme, removing different numbers of contour points and different portions of high frequency components was considered. To investigate their effects on retrieval performances, the fourth database was used for comparison. Figure 5.8(a) shows the retrieval performances based on different numbers of contour points. The use of 512 contour points achieves the best retrieval performance. Figure 5.8(b) shows the retrieval performances based on different portions of the highest frequency components discarded. The removal of 75% of the highest frequency components achieves the best performance.



(a) Different numbers of contour points.



(b) Different percentages of highest frequency components discarded.

Figure 5.8: Comparison of the precision-recall graphs with 80 retrieval images using the fourth shape database based on (a) different number of contour points, and (b) different percentage of the highest frequency components discarded.

Figure 5.9 illustrates a contour and the corresponding smoothed contours obtained by removing 40%, 60%, and 80% of the highest frequency components in the shape as well as their corresponding pruned skeletons. The results show that the noise along a contour is removed according to the percentage of the highest frequency components removed, so the extracted skeletons are affected accordingly.

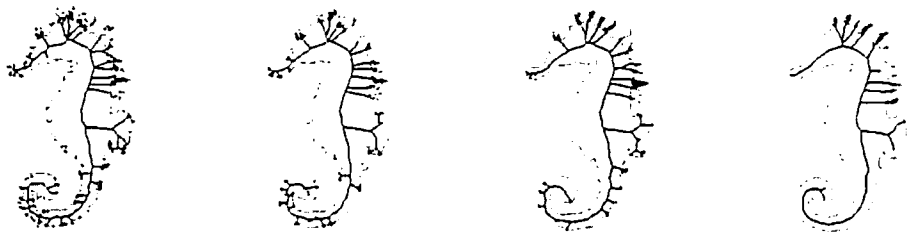
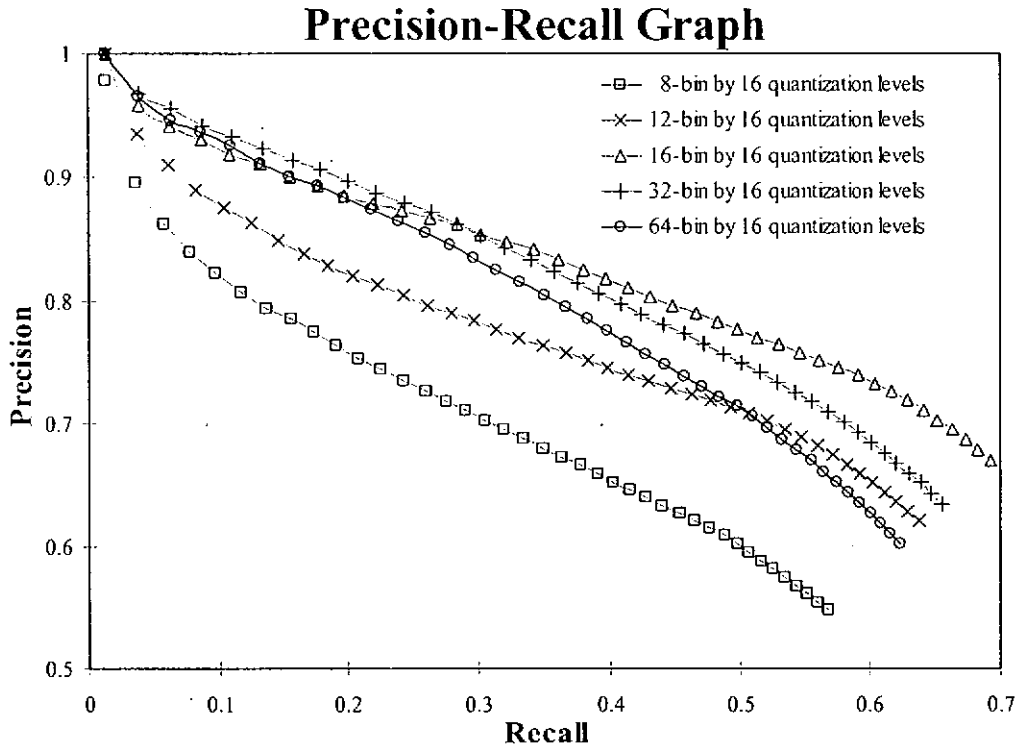


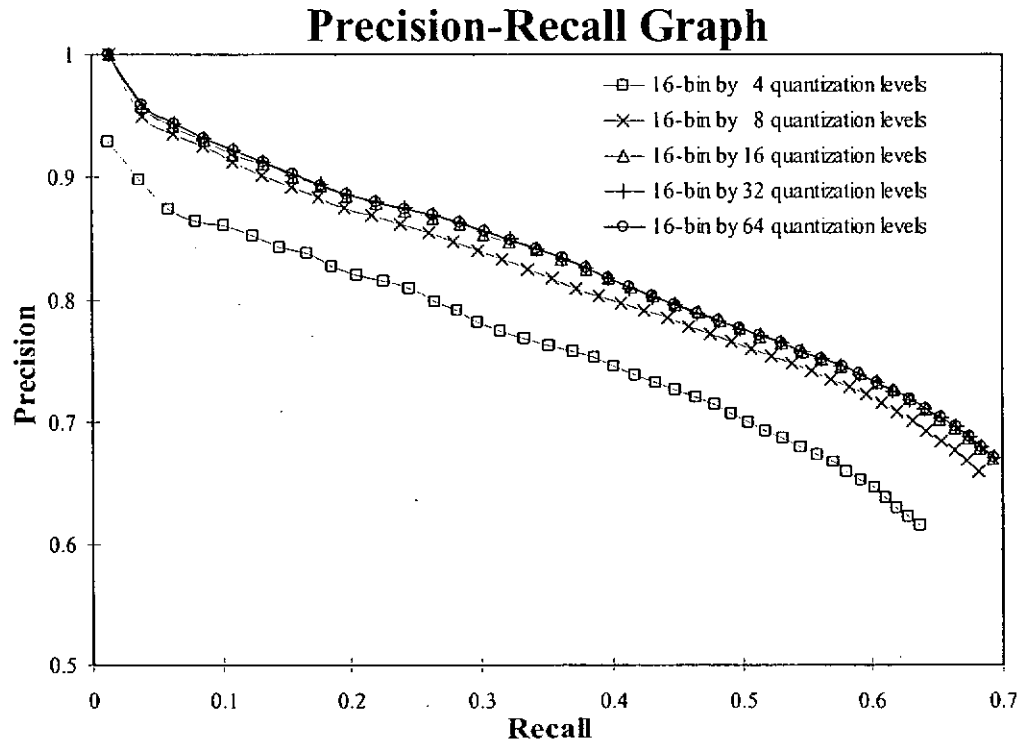
Figure 5.9: The original contour and the corresponding smoothed contours when 40%, 60%, and 80% of the highest frequency components are discarded.

5.3.3.3 Performances with different numbers of bins and quantization levels

To further evaluate the performance of our proposed scheme, different numbers of bins and quantization levels were considered. In this part, the fourth database, which was generated by combining all types of variations, was used for comparison. Figure 5.10(a) shows the retrieval performances based on different numbers of bins with 16 quantization levels. The 16-bin histogram with 16 quantization levels outperforms others for a large answer set. Figure 5.10(b) shows the retrieval performances based on a 16-bin histogram with different quantization levels. A better retrieval performance level can be obtained by using more quantization levels. Nevertheless, the retrieval performances of using 16, 32 and 64 quantization levels are nearly the same. Considering both the representation efficiency and retrieval accuracy of the shape descriptor, we choose the 16-quantization level histogram in our shape description scheme.



(a) Different numbers of bins used in the shape descriptor.



(b) Different quantization levels per bin used in the shape descriptor.

Figure 5.10: Comparison of the precision-recall graphs with 80 retrieval images using the fourth shape database based on (a) different number of bins, and (b) different quantization levels per bin used in the shape descriptor.

5.4 Summary

This chapter proposed a robust and efficient representation scheme for representing and retrieving shapes. Our algorithm is based on the histogram of the logarithmic radii of the normalized maximal disks of a shape. The centers of these maximal disks are located along the skeleton of the shape. This representation scheme is compact and can achieve a good retrieval performance level. The performance of our proposed scheme is superior to the moment invariants, Zernike moments, and curvature scale-space methods when the shapes are under affine transformation, at different noise levels, and under different distortions.

Chapter 6

Object Matching based on Line-Feature Hausdorff distance

The Hausdorff distance can be used to measure the similarity of two point sets. In matching the two point sets, one of them is translated, rotated and scaled in order to obtain an optimal matching, which is a computationally intensive process. In this chapter, a robust line-feature-based approach based on the Hausdorff distance for model-based recognition is proposed, which can achieve a good performance level in matching, even in a noisy environment or with the existence of occlusion. The robust features are extracted based on the possible longest segments in a point set. These features are insensitive to noise and can find the rotation and scale of the image point set accurately and reliably, so 2D-2D matching algorithm can be adopted. The 2D-2D matching can greatly reduce the required memory and computation when compared to a 4D-matching. Both the performance and the sensitivity to noise of our algorithm are evaluated using simulated data and a database of trademarks. Experiments show that our 2D-2D algorithm can give a high performance level when determining the relative scale and orientation of two point sets and locating an object, and the first 2D-matching is sufficient for similarity measure.

6.1 Introduction

Object matching is an important task in computer vision, model-based recognition [128, 129] and content-based retrieval [4, 6, 126, 130]. Humans can recognize or match the objects solely from their outlines or contents even under different translations, orientations and scales. Much research has been devoted to the properties of shape descriptors and the algorithms for recognition, retrieval, and indexing [19-21]. The shape descriptors selected in MPEG-7 [7, 10, 12, 14] are the ones that best fulfil the requirements as defined by the experts. The shape descriptors can be simply divided into four types: 3-D shape descriptor [24], 2-D/3-D shape descriptor [25], contour-based shape descriptor [22], and region-based shape descriptor [23]. Most of the real-world objects are 3-D, and one of the descriptors for 3-D shapes is based on the shape spectrum [29, 30], which can be used to represent the local convexity of a 3-D surface. However, the image and video world usually deals with 2-D projections of real-world objects. Therefore, 2-D shape descriptors such as contour shape, region shape, and 2-D/3-D descriptors [25, 26] can be used to represent the visual feature of a 3-D object from different viewing angles.

A contour-based descriptor expresses the shape properties of an object's outline while the region-based descriptor describes the pixel distribution within the 2-D object's region. For contour-based algorithms, Chang *et al.* [50] has proposed a shape recognition scheme based on relative distances between the contour points and their centroid. Other feature extraction methods include the corner point detection [51] and the dominant point detection [52] for contour representation. These important contour points can be used as features in the matching process. However, if part of a shape is missed, or an additional part is added to the boundary of the shape, there may be

significant effects on the features and this will cause false matching results. The curvature scale-space (CSS) [22, 27, 28] approach has been proposed as a contour-based shape descriptor used for search and retrieval in MPEG-7. The algorithms based on these shape descriptors usually analyze the contour points in computing the similarity of the shapes, forming an essential part of the retrieval and recognition systems. However, the applications of contour-based descriptors are more specific and limited when compared to those of region-based descriptors. This is because the latter does not need to extract the contours. For example, a trademark may consist of text, images, and occasionally other media such as scent or sound. Therefore, images of a trademark retrieval system [62, 63] are usually represented by region-based descriptors. Many well-known region-based descriptors, such as moment invariants [64-67], Zernike moments [4, 68, 69], Fourier coefficients [73, 74], angular radial transformation (ART) [23], etc., have been proposed and can provide invariant features with respect to the affine transformation. Both the high-order moments and the Fourier descriptors for two-dimensional images for object matching are sensitive to the uneven distribution of noise, and will produce a significant degree of error when the object is occluded or distorted.

One of the critical problems is how to match two point sets efficiently and accurately with the existence of noise, partial occlusion or spurious parts, and under different translations, orientations and scales. Hausdorff distance [81] has been used for matching two point sets because of its simplicity and relatively insensitivity to noise, and because it requires no explicit correspondence between the two point sets. Different Hausdorff distance measures [131] for object matching have been investigated, and a modified Hausdorff distance [82, 132] has also been applied in human face recognition. Most of the Hausdorff distance algorithms can find the best

match with respect to translation only. Although efficient matching algorithms incorporated with Chamfer distance matching [133] and combing translation and/or rotation transformation have been proposed by Goodrich *et al.* [134], the computational complexity is still intensive. The algorithm only considers the translation and/or rotation transformations for approximate point set pattern matching, which cannot tolerate the errors caused by different orientations and scales. A huge amount of computation is also required for matching objects of different orientations and scales. In [135], 2D-2D matching based on line features with Hausdorff distance was proposed. A line segment formed by two consecutive points along a contour is represented by its mid-point (x, y), the logarithm of its segment length, and its orientation. The first 2D-matching that computes the rotation and scaling factor is sensitive to noise. A 4D-matching was therefore proposed in [136]. However, this approach requires a large memory and is computationally intensive. More importantly, the use of line-segments along the contour will have a segmentation problem [136]. In this chapter, we propose a robust line-feature-based approach, which can be used to recognize an object based on its contour or its region. For shape matching, the contour of the object under consideration is extracted, the contour points form the required point set. For region-based matching, the edge map of the object is computed and is then used as the point set for matching. Hence, our approach can be used for both contour-based matching and region-based matching. In our algorithm, a segment is formed by joining a point and its corresponding farthest point in a point set, and then the features of this segment are extracted. This arrangement will solve the segmentation problem in [136]. We will also prove that this arrangement can make the extracted features robust to noise and occlusion, so a 2D-2D matching algorithm instead of a 4-D matching algorithm can be used. The first

2D-matching is to determine the relative scale, s, and orientation, ϕ, while a modified M-estimation Hausdorff distance is used for the second 2D-matching.

This chapter is organized as follows. In Section 6.2, an introduction to the Hausdorff distances is given. In particular, the M-estimation Hausdorff distance can determine the translation between two shapes efficiently even if noise and occlusion exist along the two object boundaries. In Section 6.3, our new matching algorithm is presented, which can accurately find the scale and orientation of a point set relative to another one. A robust line-segment Hausdorff distance is also described for matching two point sets. In Section 6.4, experimental results based on simulated data and a database of trademarks are presented. The accuracy of our 2D algorithm to determine the relative scale and orientation of two point sets and the effect of noise on the contour points are also evaluated. Finally, summary is given in Section 6.5.

6.2 Hausdorff Distance for Shape Matching

The Hausdorff distance is a kind of metric measurement used to measure the degree of mismatch between all possible relative positions of two point sets. Given two finite point sets $A = \{a_1, a_2, \ldots, a_m\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, the Hausdorff distance $H(A,B)$ for these two point sets is defined as follows:

$$H(A,B) = \max(h(A,B), h(B,A)) \tag{6.1}$$

$$h(A,B) = \max_{a_i \in A} \min_{b_j \in B} d(a_i, b_j), \tag{6.2}$$

where $h(A,B)$ is the directed Hausdorff distance and $d(a,b)$ is the Euclidean distance between two points a and b.

The relative position between the two point sets can be calculated by searching for a minimum value of the Hausdorff distances on the (x, y)-plane. If a translation

transformation, t, applied to the point set B is best matched with the point set A, a minimum value of the Hausdorff distance $H(A,t(B))$ for point set A and its transformed point set $t(B)$ can be obtained. The searching function for matching can be written as follows:

$$H_{\min} = \min_{t} H(A,t(B)) \tag{6.3}$$

where $t(.)$ represents the translation transformation function. For a specific transformation t, the minimum value of the Hausdorff distance, H_{min}, represents the similarity between the two point sets. If this value is great, it implies that the degree of mismatch between the two point sets A and B is high. Conversely, if this value is small, it implies that the two point sets are similar to each other. The translation parameters (x, y) obtained from the minimizer of the transformation t is used to transform the point set B before computing the similarity between A and B.

Different Hausdorff distance measures have been proposed for shape matching. Huttenlocher *et al.* [81] proposed the generalized Hausdorff distance measures, which include the traditional Hausdorff distance (HD), the modified Hausdorff distance (MHD), and the ranked Hausdorff distance. The MHD can achieve the best performance level when matching two point sets. Sim *et al.* [137] proposed the least trimmed square Hausdorff distance (LTS-HD) and the M-estimation Hausdorff distance (ME-HD), which are more robust to outliers and occlusions. The ME-HD measure only requires the comparison and summation operations, whereas the LTS-HD measure requires the sorting and summation operations, so the ME-HD is used in our algorithm. Given two finite point sets $A = \{a_1, a_2,..., a_m\}$ and $B = \{b_1, b_2,..., b_n\}$, the M-estimation Hausdorff distance, $H_M(A,B)$, for the point sets A and B is defined as follows:

$$H_M(A,B) = \max(h_M(A,B), h_M(B,A)) \tag{6.4}$$

and the directed M-estimation Hausdorff distance, $h_M(A,B)$, is

$$h_M(A,B) = \frac{1}{N_A}\sum_{a\in A}\rho(d_B(a)) \tag{6.5}$$

where $\rho(x) = \begin{cases} |x|, & |x| < \tau \\ \tau, & |x| \geq \tau, \end{cases}$

$d_B(a)$ represents the minimum distance value from point a to the point set B, and N_A is the number of points in the point set A. The threshold τ is used to eliminate outliers. This distance measure can therefore eliminate those outliers yielding large errors. The matching performance depends on the parameter τ. If τ is set to infinity, the ME-HD is equivalent to the conventional modified HD. The parameter τ cannot be accurately determined [137], and depends on the amount of noise and occlusion. For example, if the value of τ is set at 5.0, the range of the measured distance using ME-HD is between 0.0 and 5.0. The zero value means that the two point sets are exactly the same, while a value of 5.0 means that the two point sets are totally unmatched with each other.

Two objects can be compared either by their contours or by their overall patterns. The shape of an object can be represented as a set of contour points along its boundary. The contour points can be obtained by using an adaptive active contour model, which can extract the contour of a highly irregular object even under noisy environment. For region-based representation, the edge points of the objects are considered. Matching based on contour or region also requires the computation of the relative orientation, scale and position so that the line segments formed by the corresponding point sets can be matched with each other as closely as possible. The Hausdorff distance can be used to determine the translation parameters (x, y)

efficiently. The minimum value of the Hausdorff distance can also be used to represent their similarity to each other. If an object is rotated and scaled, we have also to determine the scale and orientation (ϕ, s). There are two approaches for computing the relative position, scale, and orientation between two objects. The first one is 2D-2D matching, which determines the relative scale and orientation in the first 2D-matching, and their relative position and similarity in the second 2D-matching. This approach is simple, but the scale and orientation must be determined accurately in the first 2D-matching. Otherwise, error results in the first matching will be passed onto the second 2D-matching. The second approach is 4-D matching, which determines the position, scale, and orientation in a single process. This approach is accurate and robust to noise, but it has a very large memory requirement and is very computationally intensive.

Yi *et al.* [136] proposed a line-feature-based approach for model-based recognition using a 4-D matching based on Hausdorff distance. Line segments are formed by joining the points along the object's contour. A line segment is represented by its midpoint, the logarithm of its length, and its orientation, which form a feature point (x, y, θ, $log\ l$) for its representation. The relative orientation, scale and translation between two different shapes can be obtained by using the 4-D Hausdorff distance measure. For the object-based matching, the direction of its orientation corresponding to each line segment has a range of 0° to 360° in the θ_m axis because the contour segments are connected sequentially. False alarms [136] will occur in the first matching if the rotation angle is limited to the range from 0° to 180°. This algorithm can be used for shape matching, and has the segmentation problem [136]. If a line segment is split into two or more segments, there will have little or no change

visually. However, the feature points will change significantly when the line segments are split or merged. Furthermore, as the shape of an object is represented by a set of contour points, different contour points will be extracted due to the presence of noise or distortion along its boundary. In other words, different line segments, and therefore feature points, will be formed if the line-segment approach is used. In addition, if more points are used to represent the contour, the corresponding line segments will become shorter. This segmentation problem makes this algorithm dependent on the contour points selected to represent the contour. Although 4-D matching was used to improve its accuracy, the algorithm is still not robust to the presence of noise and occlusions. Therefore, in this chapter, an approach based on robust line segment feature which can be used for both contour-based and region-based matching is proposed and evaluated.

6.3 A Robust Feature for Object Matching

Feature selection is an important issue for object matching. The selected features should be robust to noise and invariant to position, scale and orientation. We propose a robust feature for object matching using Hausdorff distance which allows 2D-2D matching to be adopted instead of 4D matching. The feature is based on the line segments formed between each point and their corresponding farthest point in a point set. The orientation and length of each of the line segments are then used to form a feature point in the (θ, log l)-plane. The features extracted are robust to noise and partial occlusion, so 2D-matching can be used to determine the scale and orientation between the point sets accurately. Having rotated and scaled the query point set, the

relative position and the similarity between the point sets are computed in the second 2D-matching in the (x, y)-plane.

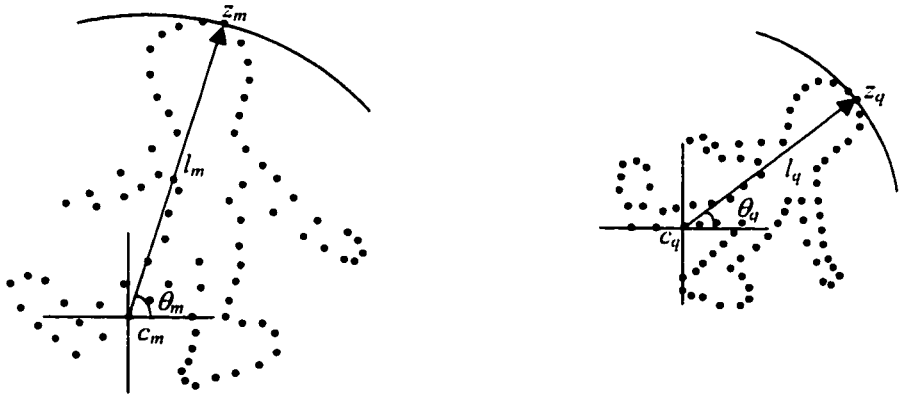
6.3.1 Robust features for matching under different orientations and scales

Suppose that we have a model object, S_m. A query object, S_q, is obtained by rotating and scaling the object S_m by an angle ϕ and a scaling factor s, respectively, as shown in Fig. 6.1(a) and (b). Depending on the context to be considered, the point sets for the model object and the query object are formed by their contour points for shape matching and by their edge maps for region-based matching. Robust line-segment features are then extracted by considering line segments formed by joining each point and its corresponding farthest point in the point set. To simplify the explanation, we consider shape matching only in the following discussions, which can be extended to region-based matching. Consider two points, z_m and z_q, which are the corresponding farthest points from the points, c_m and c_q, of the two shapes. Two line segments, $\overline{z_m c_m}$ and $\overline{z_q c_q}$ for the two objects are therefore formed. These two line segments are converted to feature points, $(\theta_m, \log l_m)$ and $(\theta_q, \log l_q)$, in the $(\theta, \log l)$-feature plane, which can then be used to measure the relative scale and orientation of the objects. Since c_m and c_q are the corresponding points in the two objects, the lengths and the orientations of the two line segments have the following relationship:

$$\log(l_q) = \log(l_m) + \log(s) \tag{6.6}$$

$$\theta_q = \theta_m + \phi \tag{6.7}$$

where the parameters ϕ and s represent the orientation and the scaling factor, respectively.

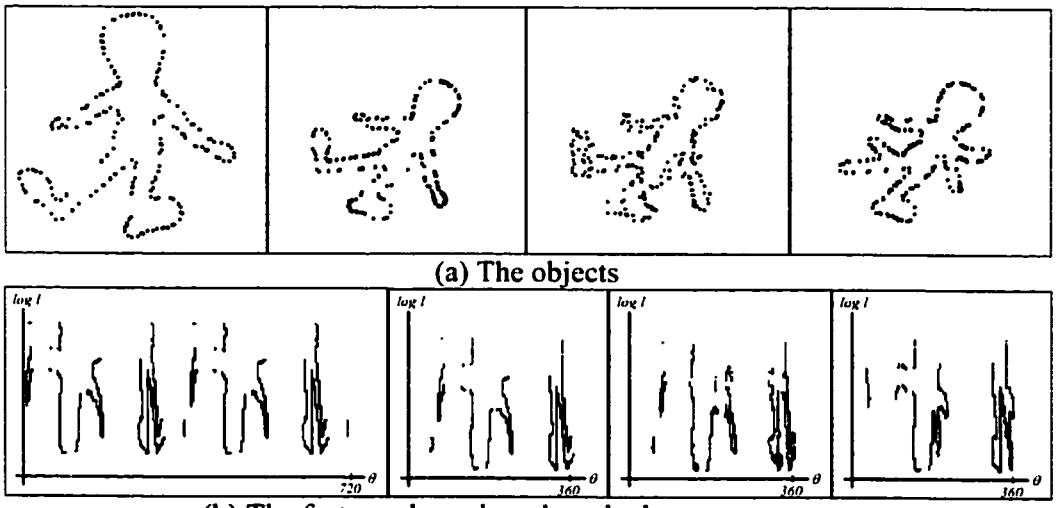


(a) The model object S_m (b) The query object S_q

Figure 6.1: A model and a query object, which are represented by the point sets, under different translations, orientations and scales.

A set of feature points can be extracted for each object to form a feature pattern in the (θ, log l)-feature plane. Figure 6.2(a) shows a model shape, a query shape, and two more shapes which are the rotated and scaled versions of the model shape with additional noise or distortion. By projecting the logarithmic lengths and the orientations of the longest line segments of these four objects into their corresponding feature planes, as shown in Fig. 6.2(b), the minimum value of the ME-HD can be obtained at a specific relative position between the model feature pattern and an image feature pattern. The methods used for forming line-segments have a high tolerance to noise and distortion. The scale s and the orientation ϕ can then be computed by matching the model and image feature patterns based on equations (6.6) and (6.7). In order to find their relative orientation within the range of 0° to 360°, the feature pattern of the model object is duplicated to the range of 360° to 720°, as shown in the first feature plane in Fig. 6.2(b). The feature patterns of these shapes in the (θ, log l)-feature plane are similar, but are translated relative to each other. Therefore, the minimum value of the ME-HD can be used for similarity measure in the first 2D-matching. After the relative orientation and scale between two point sets

are obtained, the query object can be resized and rotated for matching. The relative position between the transformed query object and the model object can then be obtained by using the ME-HD in the second 2D-matching.



(a) The objects

(b) The feature planes based on the longest segments

Figure 6.2: The objects and their corresponding (θ, log l)-feature planes.

6.3.2 Robustness with respect to noise and occlusions

Considering the feature pattern of an object, the effect of noise and occlusion can be minimized since the feature pattern is extracted based on the line segments formed by each point to its corresponding farthest point in a point set. Suppose that $S=\{x, y\}$ represents the point set of an object and S_{noise} represents the object with Gaussian noise added to the point set. Then

$$S_{noise} = \{x + \nabla x, y + \nabla y\} \tag{6.8}$$

where ∇x and ∇y represent the noise level of the variances with respect to x- and y-coordinates.

Since the line segments considered are the longest segments, the extracted features, the orientation and the logarithm of their lengths, are affected to a lesser extent than other possible line segments formed by the points in the point set. Suppose

that noise is added to a contour point and its corresponding farthest contour point, the change in length of the line segment formed by these two points is as follows:

$$\begin{aligned} \nabla l &= \left\| l_x + \nabla x, l_y + \nabla y \right\| - \left\| l_x, l_y \right\| \\ &\approx \frac{l_x \nabla x + l_y \nabla y}{\sqrt{l_x^2 + l_y^2}} \end{aligned} \tag{6.9}$$

The change in orientation can be computed as follows:

$$\begin{aligned} \nabla \theta &= \arctan\left(\frac{l_y + \nabla y}{l_x + \nabla x}\right) - \arctan\left(\frac{l_y}{l_x}\right) \\ &\approx \arctan\left(\frac{l_y \nabla x - l_x \nabla y}{l_x^2 + l_y^2}\right) \end{aligned} \tag{6.10}$$

where (l_x, l_y) are the relative (x, y)-coordinates between the contour point and its farthest contour point along the boundary and $(\nabla x, \nabla y)$ are the maximum resultant shift of the points due to noise. Since the line segment is the longest segment that can be formed in the point set and the displacement, ∇x and ∇y, are small compared to l_x and l_y, the values of ∇l and $\nabla \theta$, i.e. the deviations of the length and orientation of the line segment, are the smallest compared to other possible line segments in the point set.

On the other hand, if an object is distorted or occluded in some parts, only the corresponding parts of the feature pattern are affected, as shown in the fourth column in Fig. 6.2(a) and (b). The feature pattern is still similar to the original one. By using the ME-HD, the effect of changes to the feature pattern due to noise, distortion or occlusion can be minimized. If more feature points are considered, the orientation and scaling factor between two objects can be obtained more accurately. Consequently, our algorithm using the ME-HD can achieve good robustness when comparing two objects, even in the presence of noise, distortion or occlusion.

6.3.3 M-estimation segment Hausdorff distance for translation matching

After the relative orientation and scale between two point sets are obtained, the query object can be resized and rotated for matching. For region-based matching, the relative position between the transformed query object and the model object can then be obtained by using the ME-HD in the second 2D-matching. The conventional Hausdorff distance considers only the point sets, but without considering the points along the line segments formed by the consecutive contour points. For contour-based matching, the relative position between the transformed query object and the model object can be obtained by using an M-estimation Segment Hausdorff distance (MES-HD) in the second 2D-matching. Given two sets $A = \{a_1, a_2, \ldots, a_m\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, the M-estimation Segment Hausdorff distance, $H_{seg}(A,B)$, between A and B is defined as follows:

$$H_{seg}(A,B) = \max\left(h_{seg}(A,B), h_{seg}(B,A)\right) \tag{6.11}$$

and the directed M-estimation Segment Hausdorff distance, $h_{seg}(A,B)$, is

$$h_{seg}(A,B) = h_M(A, seg(B)) \tag{6.12}$$

where $h_M(A,seg(B))$ is the directed ME-HD from A to the point set formed by $seg(B)$. The point set $seg(B)$ includes all the points along the line segments formed by joining the consecutive points along the contour of object B, which is defined as follows:

$$seg(B) = \left\{ c \,\middle|\, c \in \overline{c_s c_e}; c_s, c_e \in B \right\} \tag{6.13}$$

where c_s and c_e are the starting and the end points of any line segment along the boundary of an object, and $\overline{c_s c_e}$ is the line formed by joining c_s and c_e. The Euclidean distance map of $seg(B)$ can also be obtained by the same distance transform [120].

Distance transform is a means of making the computation of the Hausdorff distance efficient. For the MES-HD, besides considering the contour points, all the points along the line segments formed by the consecutive contour points are also taken into account and used to generate the distance map. As the computation required depends on the size of the distance map [120], the computation required for this Hausdorff distance is similar to that of using the point set only. However, the computed distance will be more accurate, as all the points along an object's boundary are considered. Nevertheless, we must have a high-level knowledge about the representation of an object's boundary in order to use the MES-HD. If the point set of an object is obtained by edge detection techniques, which can provide low-level information about the points only, the MES-HD can no longer be used.

6.4 Experimental Results

The matching performance of our proposed algorithm is compared to the line-segment feature proposed by Yi *et. al.* [136]. The shapes used in the first part of the experiments are closed-contour, which are distorted by different levels of noise variance. The effect of occlusion and distortion on the matching performance will also be evaluated by removing part of a shape and/or including an additional part to the shape. In the second part of the experiment, the trademarks used are region-based object and a trademark retrieval system was implemented based on our algorithm. The retrieval rate of this trademark retrieval system based on our algorithm was also evaluated. The experiments were conducted on a Pentium II 400MHz PC.

6.4.1 Generation of the model and query sets

In this experiment, 10 different shapes extracted from images of size 640×480 by an adaptive snake method, which has been presented in Chapter 3, and represented by a set of 40 contour points are used to form the model set, as shown in Fig. 6.3. These 10 model shapes are first rotated by an angle of 37° and scaled by a factor of 0.7. A query set containing 100 shapes is then generated by applying different noise levels to the transformed model shapes. Another query set containing 10 shapes is also generated by arbitrarily removing and/or adding parts of contours to the transformed model shapes.

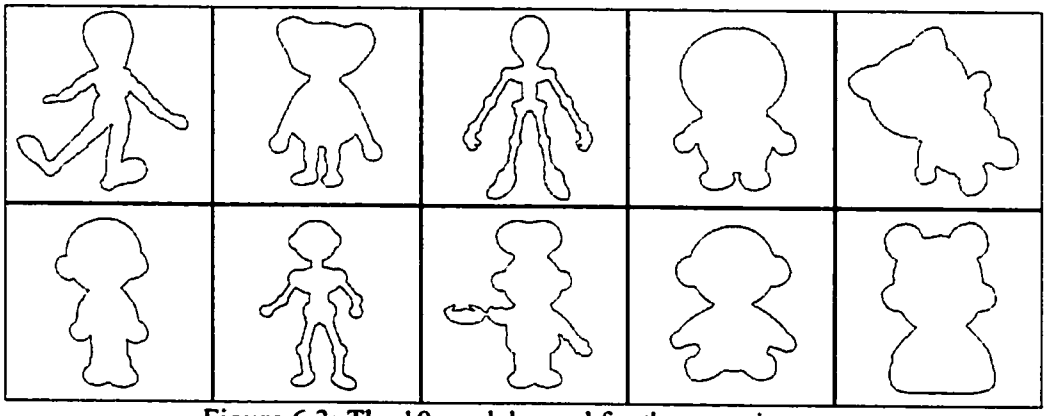


Figure 6.3: The 10 models used for the experiments.

6.4.2 The effects of noise levels and distortion

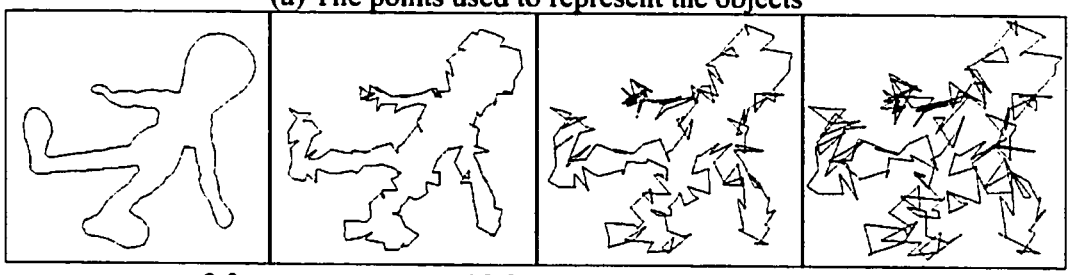
Figures 6.4 and 6.5 illustrate the effect of noise and of distortion/occlusion added to an object on the matching performance. The rotated and scaled point sets representing the objects with or without modifications are shown in the first row. In order to visualize the effect of noise or distortion on the object's shape, the successive points along the contour are connected and shown in the second row. Their corresponding feature patterns based on the line-segment approach [136] and our approach, robust line-segment, are used for matching. The scale and orientation of a

transformed shape relative to the original object can be computed by using the ME-HD in the respective feature planes. Having transformed the shapes accordingly, the MES-HD is applied to match the objects in the (x, y)-plane. The matching results based on the line-segment approach and our proposed approach are illustrated in the third and fourth rows, respectively.

Three different factors will affect the line segments of a shape and its feature pattern. These factors include the number of contour points used to represent the shape, the noise level of the shape, and the distortion or occlusion of the shape. Figure 6.6(a) illustrates an object represented by different number of contour points. The corresponding feature patterns generated by the line-segment approach and our proposed approach are shown in Fig. 6.6(b) and (c), respectively. The x-axis represents the orientation of the longest segments while the y-axis represents the logarithmic length of the longest segments. The feature patterns generated based on the line-segment approach change significantly with the number of contour points used to represent the object. On the contrary, the feature patterns generated by our approach are similar to each other and will be more accurate if more contour points are used. Our proposed approach does not have the segmentation problem, while the line-segment approach requires the careful selection of the contour points. In order to observe the effect of different noise levels added to a shape, and distortion or occlusion on the feature patterns, more contour points are used to represent an object. The corresponding feature patterns are shown in Fig. 6.6(d) and (e), respectively. However, in order to reduce the required computation, the number of contour points should be as small as possible, while it is sufficient to represent the object's shape. Algorithms such as the Discrete Curve Evolution (DCE) [138] and the Local Maximal Curvature (LMC) Point Decomposition [139] can be used to reduce the number of

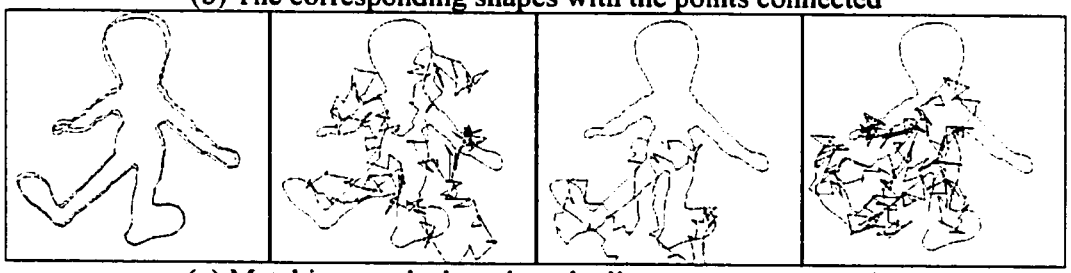
contour points. The visual appearance of a shape can be preserved and the required computational time for the matching process can be reduced. As the noise level increases, our approach will result in minimal change to the feature pattern, as shown in Fig. 6.6(d). Similarly, as parts are added or removed, as shown in Fig. 6.5(a), the feature patterns generated using our approach, as illustrated in Fig. 6.6(e), are affected to a lesser extent than the line-segment approach. Consequently, the matching performance based on our approach outperforms the line-segment approach when the shapes are distorted by noise and occlusion, as shown in Fig. 6.4 and Fig. 6.5. This is mainly due to the fact that the extracted features are obtained based on the longest possible line segments formed in a point set. Noise or disturbance on the contour points will have a relatively lesser effect when compared to the features extracted from other possible line segments of the point set. Furthermore, in measuring the Hausdorff distance in the second 2D-matching, line segments are considered instead of the point set only.

(a) The points used to represent the objects

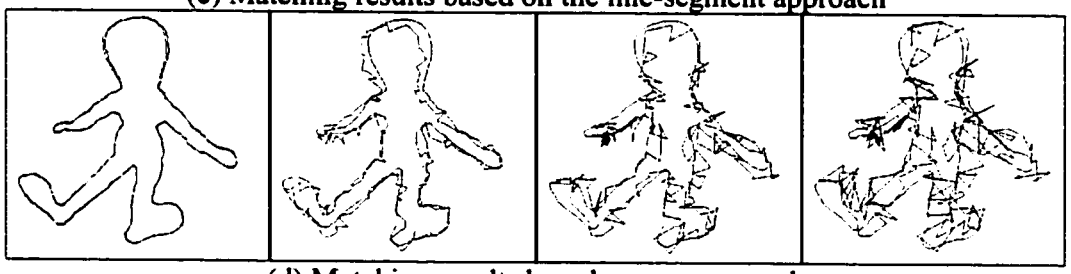


σ = 0.0 σ = 10.0 σ = 20.0 σ = 30.0

(b) The corresponding shapes with the points connected

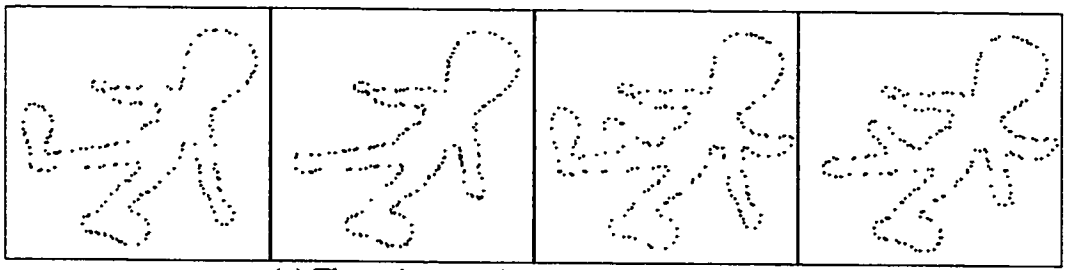


(c) Matching results based on the line-segment approach

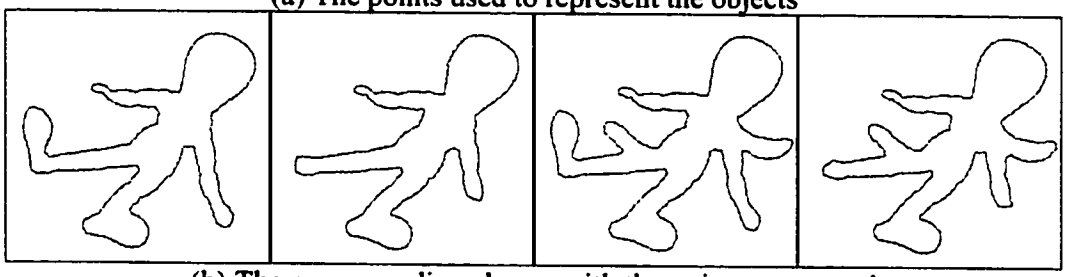


(d) Matching results based on our approach

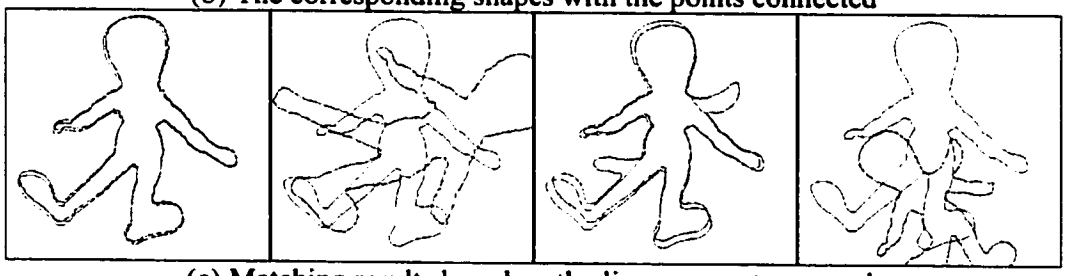
Figure 6.4: The comparison of matching performance using the line-segment and the robust line-segment features with different levels of noise.



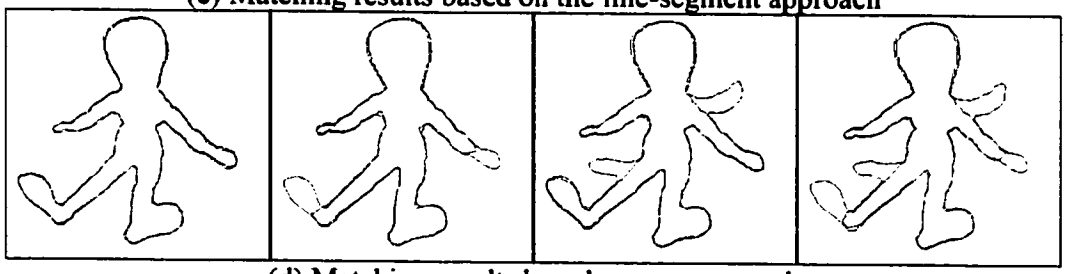
(a) The points used to represent the objects



(b) The corresponding shapes with the points connected

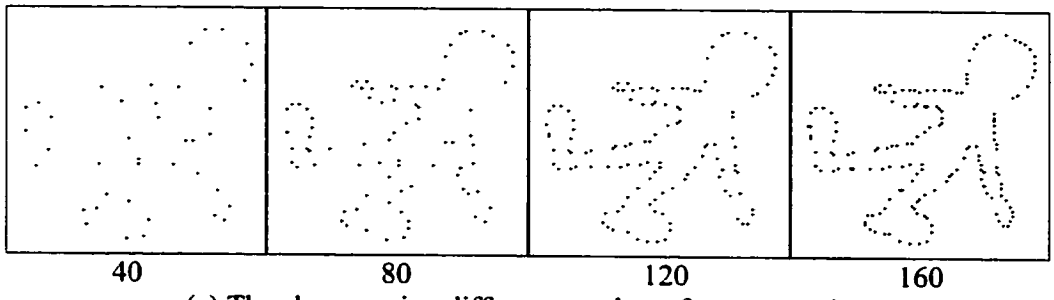


(c) Matching results based on the line-segment approach

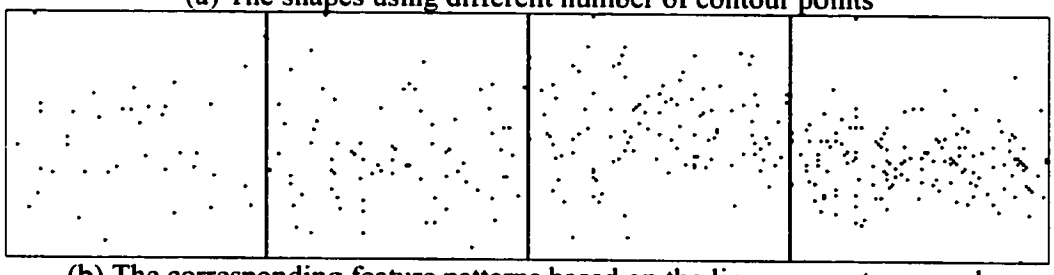


(d) Matching results based on our approach

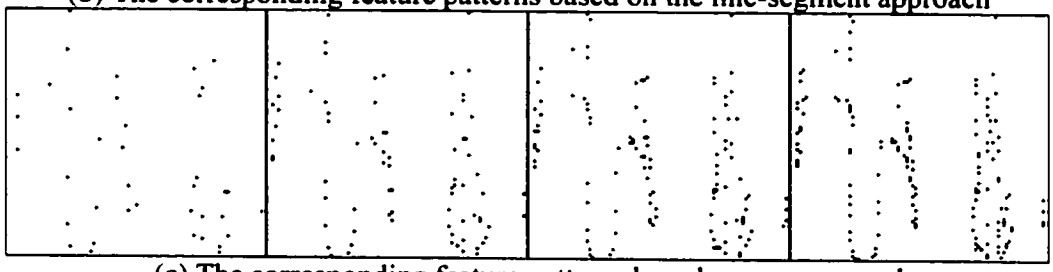
Figure 6.5: The comparison of matching performance using the line-segment and the robust line-segment features with occlusion and additional parts to the shapes.



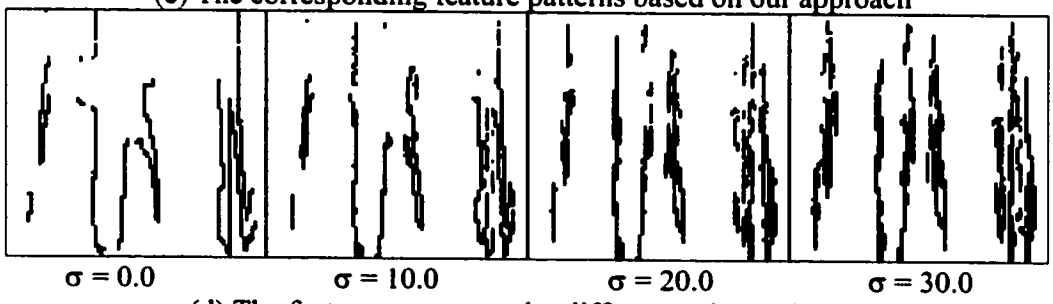
(a) The shapes using different number of contour points



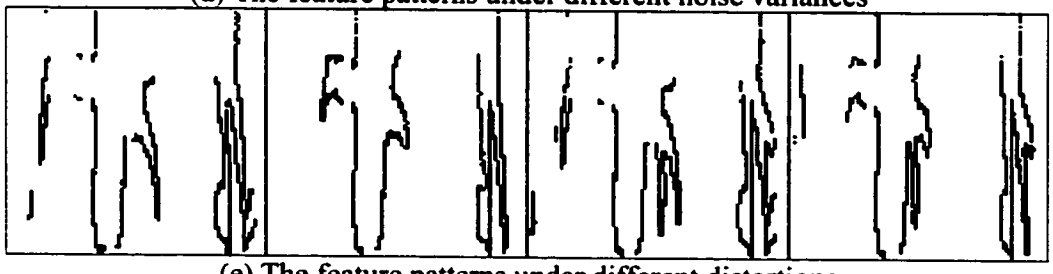
(b) The corresponding feature patterns based on the line-segment approach



(c) The corresponding feature patterns based on our approach



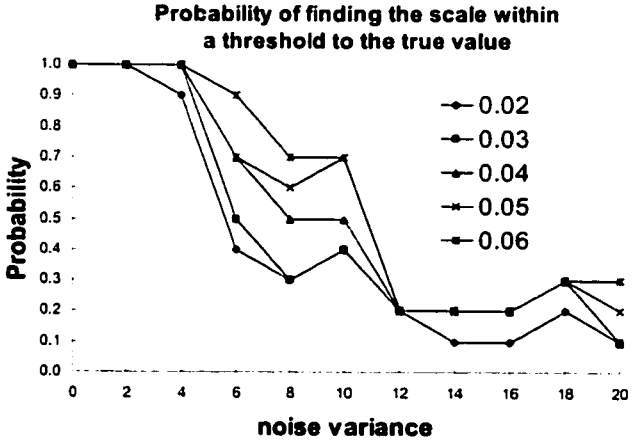
(d) The feature patterns under different noise variances



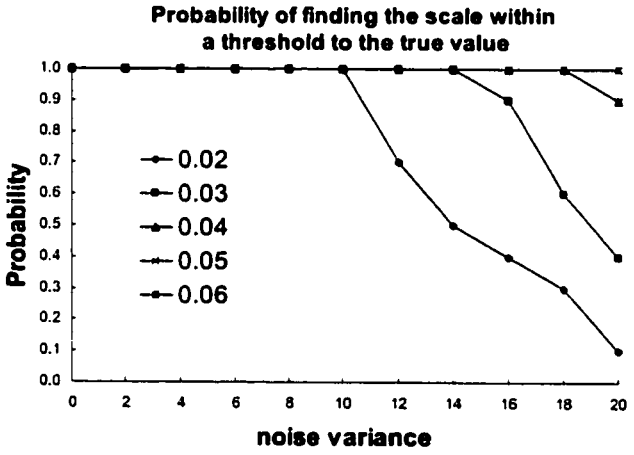
(e) The feature patterns under different distortions

Figure 6.6: The comparisons of (a) the shapes using different number of contour point, and their corresponding feature patterns in the (θ, log l)-feature plane by using (b) the line-segment approach and (c-e) our proposed approach. The horizontal and vertical coordinates of the feature planes represent the parameters θ and log l, respectively.

6.4.3 Computation of scale and orientation



(a)

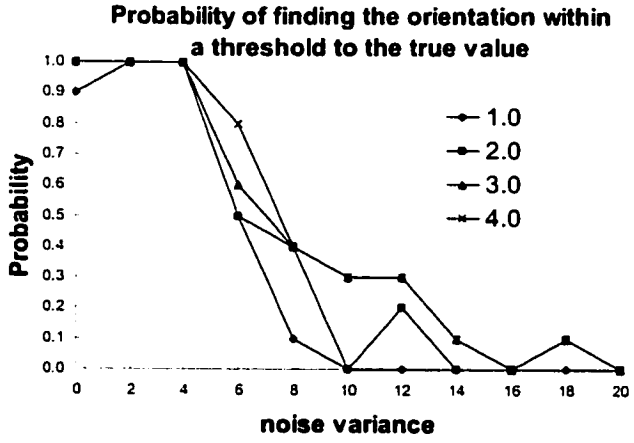


(b)

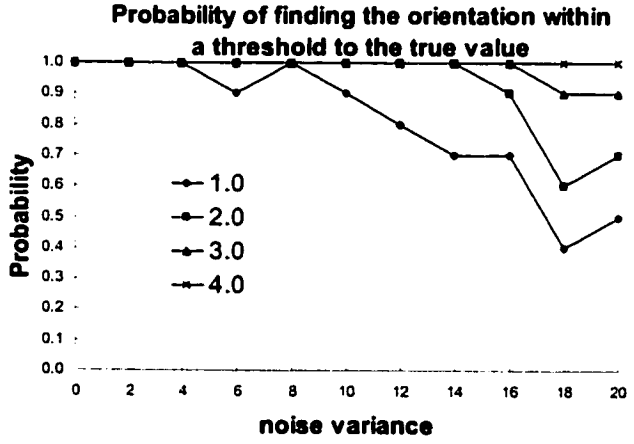
Figure 6.7: The probabilities of finding the scaling factor by (a) the line-segment approach and (b) the robust line-segment approach.

To evaluate the performance of computing the relative scale and orientation between two shapes, the probabilities of finding the scale and orientation are measured based on the shapes in the model set and the query set. The probabilities of finding the scale and orientation correctly within a threshold to the true value against different levels of Gaussian noise using the line-segment approach and our approach are plotted in Fig. 6.7 and Fig. 6.8, respectively. According to the results, we can observe that the probabilities drop rapidly when the noise level is higher than 4.0 for

the line-segment approach. Our proposed feature can be used to find these two factors accurately even if the noise is increased to 20.0.



(a)



(b)

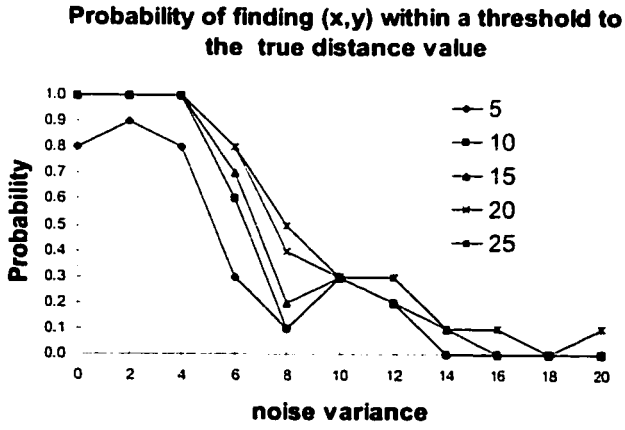
Figure 6.8: The probabilities of finding the orientation by (a) the line-segment approach and (b) the robust line-segment approach.

6.4.4 Computation of relative position

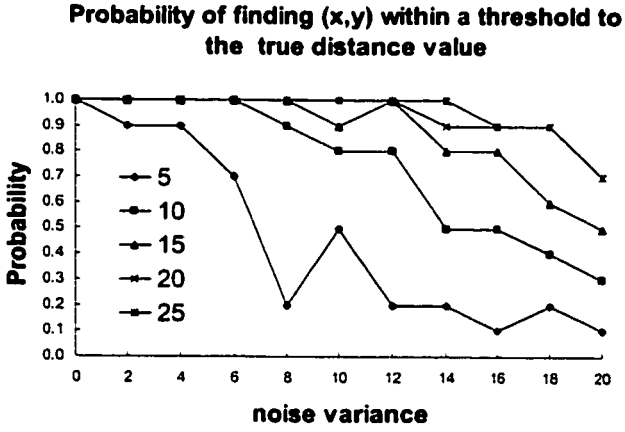
After the orientation and scaling factor between the objects are obtained by using the line-segment method and the robust line-segment method, the query object is transformed accordingly and then compared to the model object. The errors on the orientation and scaling factor obtained in the first 2D-matching will propagate to the second 2D-matching. This will affect the accuracy of the computed relative position

and the similarity between the two objects in the second 2D-matching. To evaluate the performance of computing the relative position between the two shapes, the probabilities of computing the relative position within a threshold with the ME-HD are shown in Fig. 6.9(a) for the line-segment approach and in Fig. 6.9(b) for the robust line-segment approach. We can observe that the matching results are degraded significantly when the noise variance is increased to 4.0 with the line-segment approach. Although the same second 2D-matching is applied, the robust line-segment approach has a much better matching performance than the line-segment approach. The matching performance of the MES-HD used in the second 2D-matching is also measured. The probability of finding the best translation parameters within a threshold to the true value for the robust line-segment approach with the MES-HD is plotted in Fig. 6.9(c). Experimental results show that the matching performance for the MES-HD is better than that for the ME-HD with different thresholds.

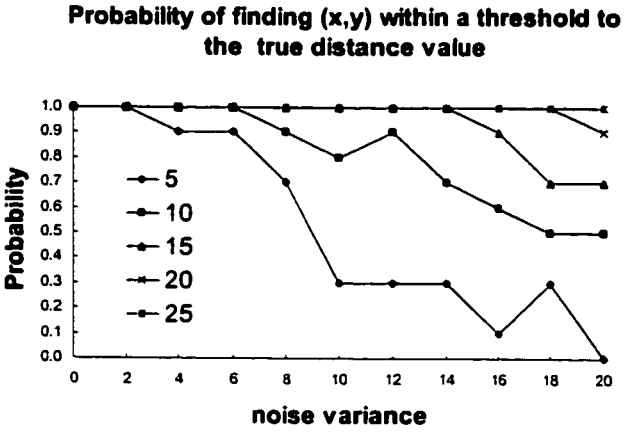
Similarly, the probabilities of the computed Hausdorff distance within a threshold with the ME-HD are shown in Fig. 6.10(a) for the line-segment approach and in Fig. 6.10(b) for the robust line-segment approach. The probabilities for the robust line-segment approach are much higher than those for the line-segment approach. Figure 6.10(c) shows the probability for the robust line-segment approach with the MES-HD, which can give a much better performance than the ME-HD in object-based matching.



(a)

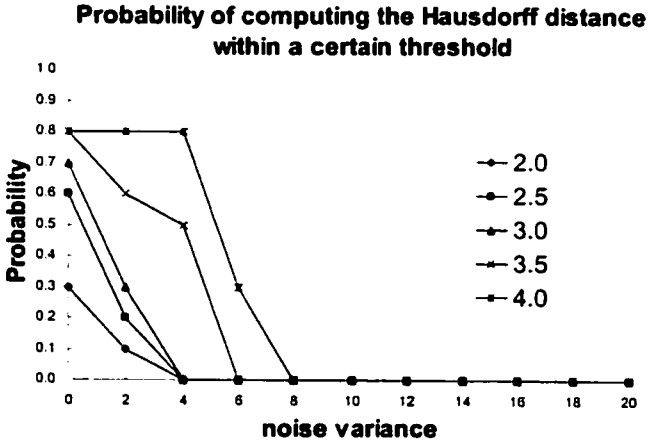


(b)

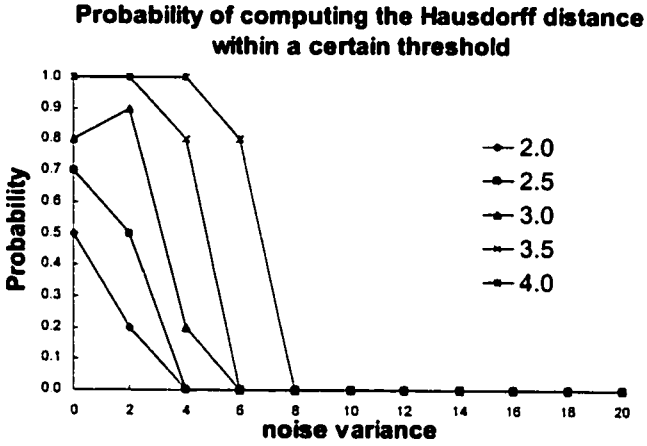


(c)

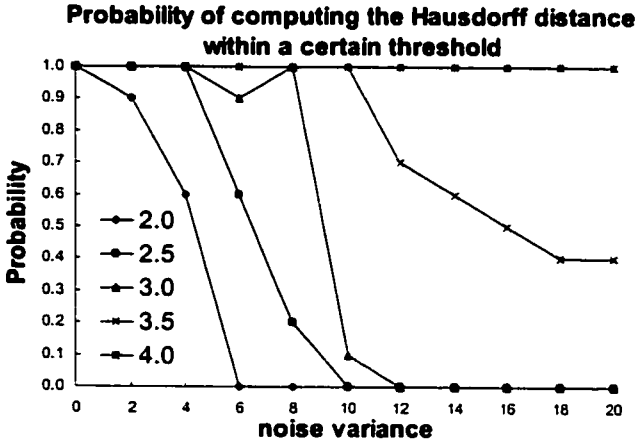
Figure 6.9: The probabilities of finding the relative position between two shapes using (a) the line-segment approach, (b) the robust line-segment approach, and (c) the robust line-segment approach with the MES-HD.



(a)



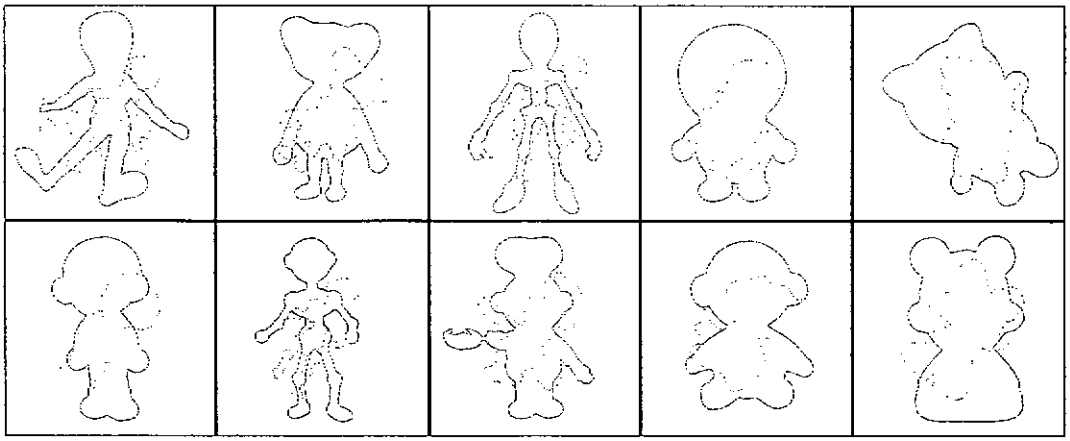
(b)



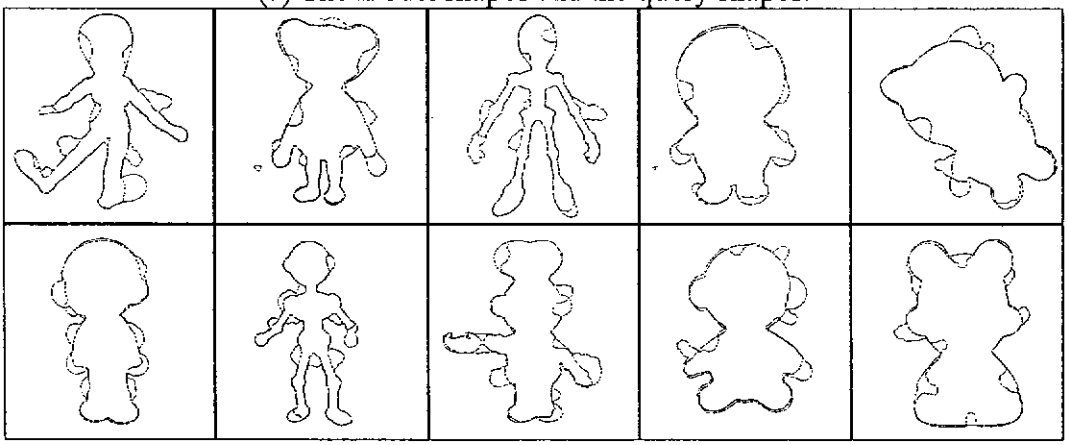
(c)

Figure 6.10: The probabilities of finding the Hausdorff distance within a certain threshold using (a) the line-segment approach, (b) the robust line-segment approach, and (c) the robust line-segment approach with the MES-HD.

Figure 6.11 illustrates more matching results, where the solid lines represent the model shapes, while the dotted lines represent the query shapes which are generated by arbitrarily removing and/or adding parts of contours to the transformed model shapes. The results show that our proposed algorithm can be used for matching even when the objects are distorted, rotated and scaled.



(a) The model shapes and the query shapes.



(b) The matching results

Figure 6.11: The matching of the model shapes and the query shapes.

6.4.5 Precision-Recall Graphs for trademark databases

To further evaluate the performance of our proposed algorithm, two trademark databases consisting of 500 images were used. All trademark images were scanned and binarized to black and white, and normalized to a size of 100×100. As the internal details of a trademark must also be considered, the edge image was therefore used to form its point set for its representation. Consequently, the ME-HD was used instead

of the MES-HD in the second 2D-matching for this application. Based on the edges of the trademark images, a database was constructed and integrated to form a retrieval system. As the classification of a trademark may be a subjective task in accordance with human perception, we divided the trademarks into 50 classes with 10 similar trademarks for each class to construct a database consisting of 500 images. The second database was generated by randomly rotating and scaling each of the trademark images in the first database. The scaling factors used are random numbers between 0.8 and 0.9. Hence, the number of classes and images in the second database are the same as the first one. The query set used in the experiments was formed by selecting an image from each class of the first database. The 50 query images are illustrated in Fig. 6.12.

To measure the performance of the trademark retrieval system, we computed the precision and recall rates for different matching approaches. The precision rate and recall rate are defined as follows:

$$\text{Precision Rate} = \frac{\text{number of relevant images selected}}{\text{total number of retrieved images}} \tag{6.14}$$

$$\text{Recall Rate} = \frac{\text{number of relevant images selected}}{\text{total number of similar images in the database}} \tag{6.15}$$

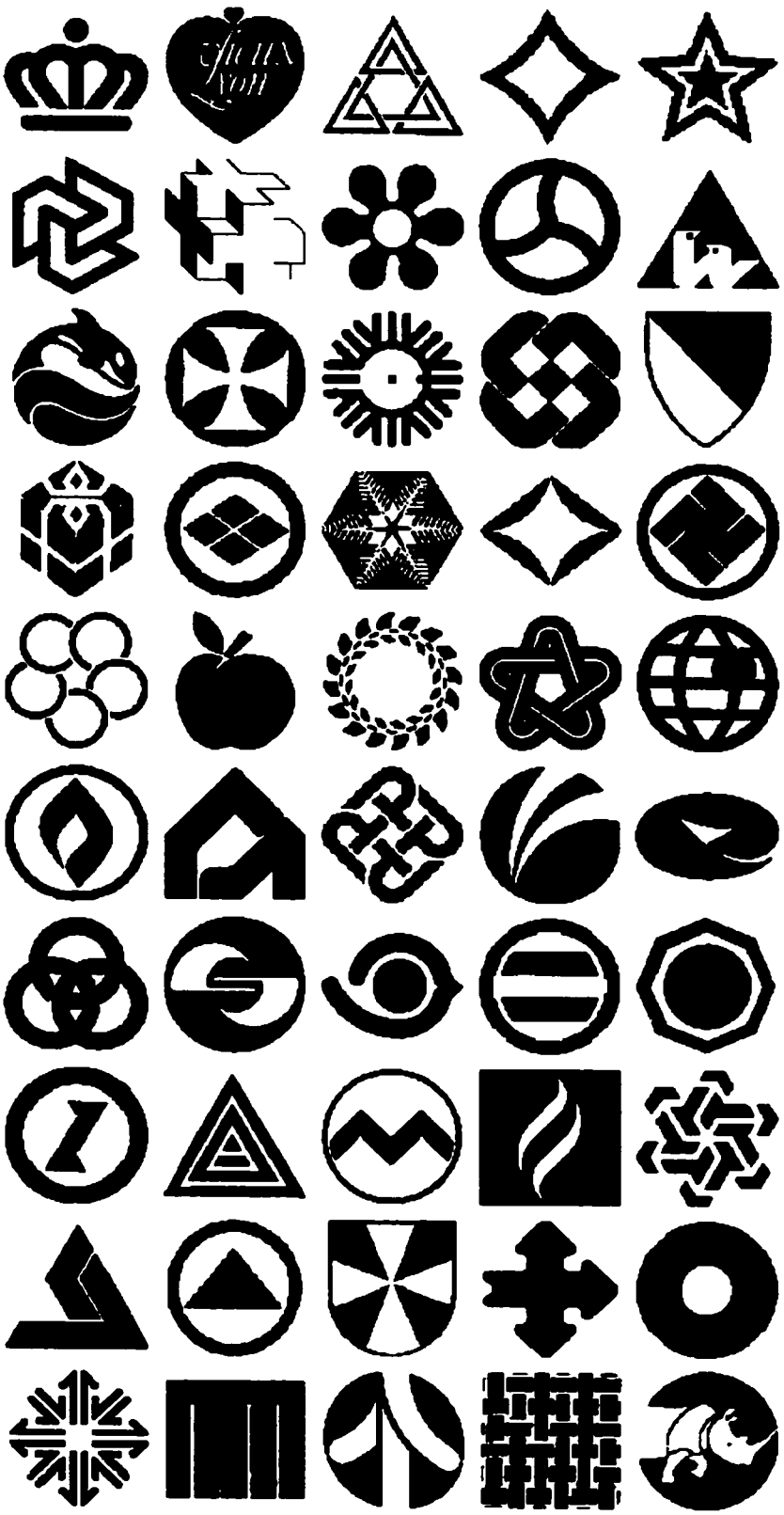


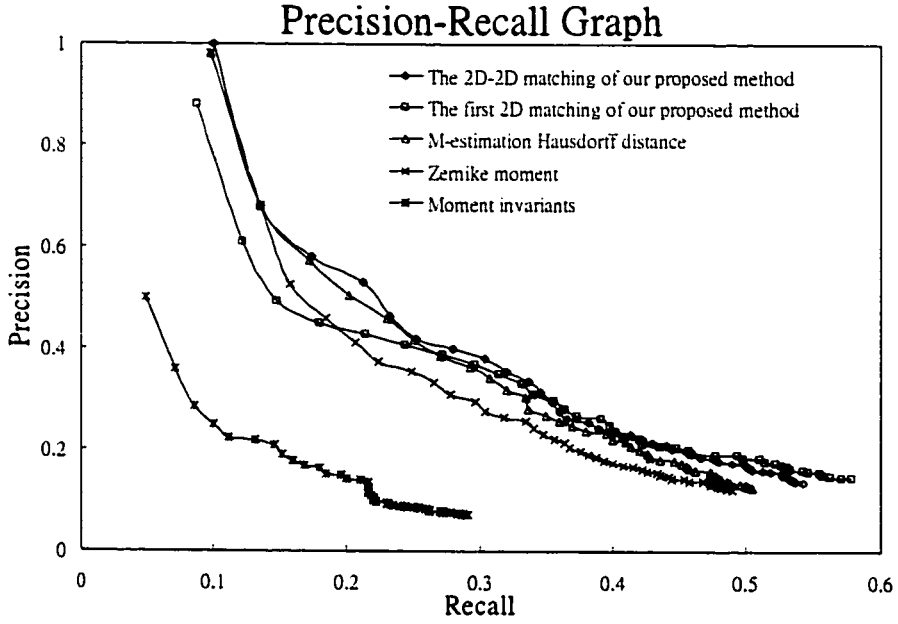
Figure 6.12: The 50 representative images in the trademark database.

The best matched 40 trademark images were retrieved from the two databases for each query image. The average values of the precision and recall rate were computed and plotted as the precision-recall graphs. The horizontal axis represents the recall rate while the vertical axis corresponds to the precision rate. The performance of a matching algorithm is represented by a curve which has 40 points corresponding to considering from 1 to 40 retrieved images in measuring the precision rate and the recall rate. The top-left point of the graph corresponds to the precision/recall values for considering 1 retrieved image while the bottom right point corresponds to the precision/recall values for the entire answer set with 40 retrieved images. The Zernike moments [4] and the moment invariants [66], which were the features used in other content-based trademark retrieval systems, were compared to our algorithm. The precision-recall graphs based on the Zernike moment features, the moment invariants, the M-estimation Hausdorff distance, the robust line-segment approach with first 2D-matching only, and the robust line-segment approach with 2D-2D matching, were measured.

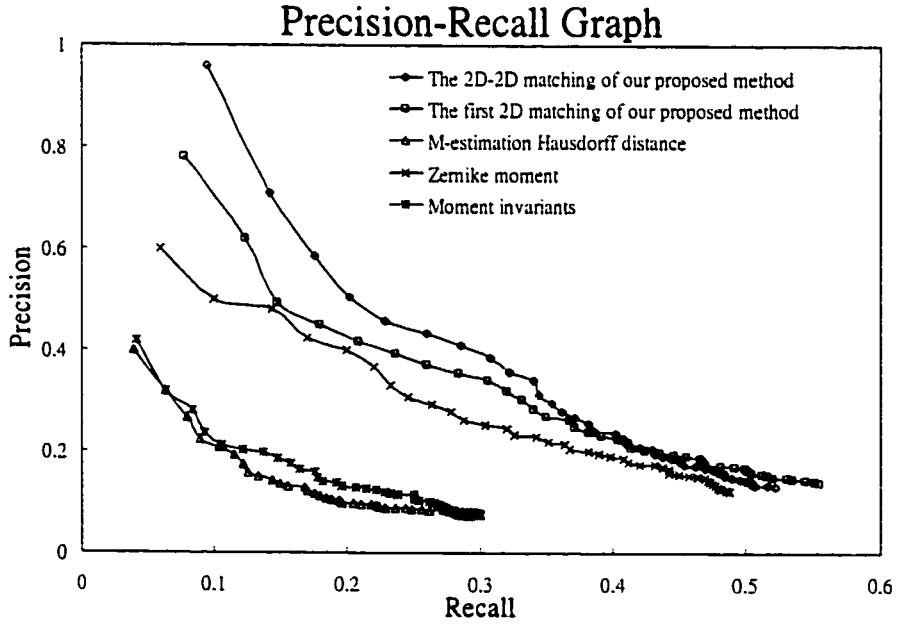
Figure 6.13 shows the precision-recall graphs of the different matching methods based on the two trademark databases: the original images, and the rotated and scaled images. Figure 6.13(a) shows the performance of the different approaches when the database of the original trademark images was used. For small answer sets, our robust line-segment approach with the 2D-2D matching performs slightly better than the Zernike moments and the M-estimation Hausdorff distance. Although our robust line-segment approach with the first 2D matching only has 12% lower precision rate than the 2D-2D matching method for small answer sets, it performs better than other methods for larger answer sets, achieving up to 9% better recall rate and 2% better precision rate compared to the Zernike moments. Figure 6.13(b) shows the

performance of the different approaches using the database of transformed trademark images. For small answer sets, the performances of most of the approaches are dropped compared to when the first database was used. Our robust line-segment approach with the 2D-2D matching shows nearly no change and yields the best performance. The performance of the M-estimation Hausdorff distance method drops significantly as it does not consider the differences in rotation and scale between two images. Since the trademark images are scaled between 0.8 and 0.9 with different orientations, the number of pixels used to represent a transformed image will be different from that of the original one. Therefore, the precision rate of the Zernike moments for small answer sets drops nearly 38% when compared to using the original trademark images. Although our proposed approach with the first 2D matching has a 10% lower precision rate than that using the original trademark images for small answer sets, it performs the best for larger answer sets.

Experimental results show that our robust line-segment approach with first 2D matching and with 2D-2D matching outperform the other three approaches in terms of the precision and recall rates. For a larger answer set, the recall rate of the first 2D matching algorithm was slightly greater than that of the 2D-2D matching. This means that the Hausdorff distance obtained in the first 2D matching based on our proposed approach can be used directly in similarity measure. The second 2D matching is used to locate the required object in an image, which is not necessary for retrieval purposes. Instead of using the 2D-2D matching, the computational time for the retrieval system can be reduced if only the first 2D-matching is used. Experimental results also show that our algorithms are invariant with respect to the orientation and scale of an image.



(a)



(b)

Figure 6.13: Comparison of the precision-recall graphs with 40 retrieved images for the trademark databases (a) based on the original images, and (b) based on the rotated and scaled images.

6.5 Summary

In this chapter, a robust line-feature-based approach for model-based recognition is proposed. This approach can be used for both contour-based matching and region-based matching. It can provide a good level of performance in a noisy environment or with the existence of occlusion. This new approach is insensitive to noise, and can find the rotation and scale of the image point set more accurately and reliably than other approaches. It solves the problems of determining the translation, rotation and scale between two objects. The feature points are generated based on the line segments formed by each contour point and its corresponding farthest point in a point set. The robustness and performance of our method have also been addressed. Experiments show that our approach can achieve accurate results and is robust to noise. Consequently, 2D-2D matching can be used instead of 4D-matching. This can greatly reduce the memory requirement and the required computation. In addition, a trademark retrieval system was implemented based on our proposed approach. Experimental results show that the first 2D-matching based on our proposed feature is sufficient for similarity measure.

Chapter 7

Conclusions and Further Work

7.1 Conclusions

In this thesis, we have given an overview of the content-based retrieval system and the recent development of the MPEG-7. The visual descriptors considered in the standard of MPEG-7 such as color, texture, shape and motion have been introduced. Also, various techniques of feature extraction and recognition for content-based image retrieval have been reviewed. Shape descriptors, which are high level description, have been emphasized in this research work.

In our research, we proposed an adaptive snake method, which can locate highly irregular boundaries of objects in an image. An adaptive force is applied when the image forces around a point are small, and points on the snake can be deleted or inserted such that the distance between adjacent points can be kept more constant. An additional terminating criterion based on area is also proposed such that unnecessary iterations can be prevented. By identifying contour and non-contour segments in a snake, the algorithm can extract multiple objects in an image by the processes of splitting and connecting. Experimental results show that the new algorithm can

achieve a better contour representation with a required runtime similar to that of the fast greedy algorithm.

After locating the boundaries, the skeleton can be extracted according to the extracted contour points. With the use of the connectivity criterion proposed in this thesis, an accurate, simple and efficient algorithm for the extraction of a well-connected Euclidean skeleton is devised with the use of the signed sequential Euclidean distance map. The nearest contour points of the pixel under consideration and its 8 neighbors are generated to form a set of 8 point pairs, which are then used to determine whether the pixel is a skeleton point. This method can generate a connected Euclidean skeleton without requiring a linking algorithm or any iteration. The complexity of this algorithm is linearly proportional to the number of the pixels in an image.

After extracting the skeletons, a set of maximal disks can be obtained by our proposed skeletonization technique with the simple pruning algorithm. We proposed a robust and efficient representation scheme for shape retrieval, which is based on the normalized maximal disks used to represent the shape of an object. The logarithm of the radii of the normalized maximal disks is used to construct a histogram to represent the shape. Our proposed representation scheme outperforms the other methods, including moment invariants, Zernike moments, and curvature scale-space, under affine transformation, different distortions and noise levels.

In addition, we also studied the problem of matching between two objects. In order to handle the variations due to different scales, orientations and locations of the objects, we proposed a robust line-feature-based approach for model-based recognition. This approach can be used for both contour-based matching and region-based matching. It can provide a good level of performance in a noisy environment or

with the existence of occlusion. This new approach is insensitive to noise, and can find the rotation and scale of the image point set more accurately and reliably than other approaches. It solves the problems of determining the translation, rotation and scale between two objects. The feature points are generated based on the line segments formed by each contour point and its corresponding farthest point in a point set. The robustness and performance of our method have also been addressed. Experiments show that our approach can achieve accurate results and is robust to noise. Consequently, 2D-2D matching can be used instead of 4D-matching. This can greatly reduce the memory requirement and the required computation. A trademark retrieval system was implemented based on our proposed approach. Experimental results show that the first 2D-matching based on our proposed feature is sufficient for similarity measure.

Finally, we believe that the techniques for extracting the contour and its skeleton, and matching algorithm are important for content based retrieval. In this thesis, we have demonstrated the extraction of object boundaries, the skeleton and the maximal disk-based histograms, as well as the matching of two objects based on contours/region of the objects for shape retrieval. In addition, two retrieval systems based on a set of SQUID images and a set of trademark images have been developed to demonstrate the overall performance of the different techniques.

7.2 Further Work

Current research into high-level features for image retrieval is at an early stage, and it will be a long time before any generally useful systems emerge. As described in previous sections, we have proposed an adaptive snake method, which can locate the boundaries of objects in an image, and an efficient and accurate algorithm for extracting a skeleton based on maximal disks. Also, we have proposed the matching algorithm for contour/region–based object. The results obtained in this research work are useful contributions to the development of content based retrieval from image database. It is far from certain that any current approach will lead to effective image retrieval based on the feature extraction and recognition techniques. Although our work provides solutions to shape feature extraction and matching/retrieval algorithm, there are rich areas for further research.

In our further work, we will focus on developing a better shape descriptor such as the skeleton representation, and a better similarity measure capable of dealing with differences between similar feature vectors. After locating the boundaries and extracting the skeletons, the extracted information can be used as a useful query feature for retrieving similar images from a database. The skeleton feature consists of the spatial and hierarchical structure information which can be used to represent the shape more efficiently. However, a drawback of the use of the maximal disk-based histogram as presented in Chapter 5 is due to the fact that different shapes may have similar or identical shape histograms. A set of maximal disks can form other different spatial structures which have the same distribution of the maximal disks. Our future work will concentrate on using the spatial information of the skeleton to form a spatial domain histogram such that its retrieval performance can be further improved. Hence,

different techniques for skeleton matching are being studied and investigated to further improve its efficiency.

In order to further improve the performance of the trademark retrieval system we have developed in our work, there are a number of research issues which need to be addressed. Our system can be extended to match on the basis of a set of salient features rather than a single feature representing the entire image. This would require a robust and automatic segmentation algorithm, which can extract not only multiple closed-contour objects but also the mesh contour objects in more general trademark images. A future extension is to allow partial matching of objects based on some specific features which are useful in eliminating many false matches and retrieving more similar objects. Also, a future extension may be in the direction of allowing the system to learn through feedback from each user query. Another area of research is in automatic grouping or clustering of the trademark images to improve retrieval performance. However, we believe that there may be other approaches or techniques used in specific applications that are suitable to retrieval systems.

Bibliography

[1] V. N. Gudivada and V. V. Raghavan, "Content-Based Image Retrieval Systems", *Computer*, pp. 18-22, September 1995.

[2] R. K. Srihari, "Automatic Indexing and Content-Based Retrieval of Captioned Images", *Computer*, pp. 49-56, September 1995.

[3] C. L. Huang and D. H. Huang, "A content-based image retrieval system", *Image and Vision Computing*, vol. 16, pp. 149-163, 1998.

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

[5] R. Mehrotra and J. E. Gary, "Similar-Shape Retrieval In Shape Data Management", *Computer*, pp. 57-62, September 1995.

[6] 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*, vol. 22, no. 12, pp. 1349-1380, December 2000.

[7] S. F. Chang; T. Sikora, and A. Purl, "Overview of the MPEG-7 standard", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 688–695, June 2001.

[8] IBM Almaden Research Center, "Query by Image and Video Content: The QBIC System", *Computer*, pp. 23-31, September 1995.

[9] J. M. Martinez, "Overview of the MPEG-7 Standard Version 6.0", *ISO/IEC JTC1/SC29/WG11/N4509*, Pattaya, December 2001.

[10] O. Avaro and P. Salembier, "MPEG-7 Systems: overview", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 760–764, June 2001.

[11] J. Hunter, "An overview of the MPEG-7 description definition language (DDL)", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 765-772, June 2001.

[12] T. Sikora, "The MPEG-7 visual standard for content description-an overview", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 696-702, June 2001.

[13] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 703-715, June 2001.

[14] M. Bober, "MPEG-7 visual shape descriptors", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 716-719, June 2001.

[15] S. Jeannin and A. Divakaran, "MPEG-7 visual motion descriptors", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 720-724, June 2001.

[16] S. Quackenbush and A. Lindsay, "Overview of MPEG-7 audio", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 725-729, June 2001.

[17] M. Casey, "MPEG-7 sound-recognition tools", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 737-747, June 2001.

[18] P. Salembier and J. R. Smith, "MPEG-7 multimedia description schemes", *IEEE Transactions on Circuits and System for Video Technology, vol.* 11, no. 6, pp. 748-759, June 2001.

[19] S. Berretti, A. Del Bimbo, and P. Pala, "Retrieval by shape similarity with perceptual distance and effective indexing", *IEEE Transactions on Multimedia*, vol. 2, no. 4, pp. 225-239, Dec. 2000.

[20] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H. J. Zhang, "Image classification for content-based indexing", *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 117-130, Jan. 2001.

[21] R. B. Johnson, "Multimedia databases and MPEG-7", *Electronics & Communication Engineering Journal*, vol. 13, no. 3, pp. 98-99, June 2001.

[22] M. Bober, "Shape descriptor based on curvature scale space", Lancaster, U.K., *MPEG-7 proposal P320*, February 1998.

[23] W. Y. Kim and Y. S. Kim, "A new Region-Based Shape Descriptor", *ISO/IEC MPEG99/M5472*, Maui, Hawaii, December 1999.

[24] T. Zaharia, F. Preteux, and M. Preda, "The 3D shape spectrum descriptor", *ISO/IEC MPEG/M5242*, Melbourne, Australia, October 1999.

[25] K. Müller, J. R. Ohm, J. Cooper, and M. Bober, "Results of 2D/3D shape core experiments MS-4", *ISO/IEC MPEG99/M6190*, Beijing, China, July 2000.

[26] S. Abbasi and F. Mokhtarian, "Affine-similar shape retrieval: application to multiview 3-D object recognition", *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 131-139, January 2001.

[27] F. Makharian and S. Abbasi, "Shape similarity retrieval under affine transforms", *Pattern Recognition*, vol. 35, pp. 31-41, 2002.

[28] F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 789-805, August 1992.

[29] C. Dorai and A. K. Jain, "Shape spectrum based view grouping and matching of 3D free-form objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1139-1145, October 1997.

[30] C. Dorai and A. K. Jain, "COSMOS-A representation scheme for 3D free-form objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1115-1130, October 1997.

[31] A. Akutsu, T. Tonomura, H. Hashimoto, and Y. Ohba, "Video indexing using motion vectors", *Proc. Visual Communications and Image Processing*, vol. 1818, pp. 1522-1530, 1992.

[32] E. Ardizzone, M. La Cascia, A. Avanzato, and A. Bruna, "Video indexing using MPEG motion compensation vectors", *IEEE International Conference on Multimedia Computing and Systems*, vol. 2, pp. 725-729, 1999.

[33] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 602-615, September 1998.

[34] O. Hori and T. Kaneko, "Results of Spatio-Temporal Region DS Core/Validation Experiment", *ISO/IEC MPEG99/m5414*, Maui, HI, December 1999.

[35] S. Jeannin and B. Mory, "Video motion representation for improved content access", *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 645-655, August 2000.

[36] W. Y. Ma and B. S. Manjunath, "EdgeFlow: a technique for boundary detection and image segmentation", *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1375-1388, August 2000.

[37] Y. Deng, C. Kenney, M. S. Moore, and B. S. Manjunath, "Peer group filtering and perceptual color image quantization", *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 21-24, 1999.

[38] L. L. Winger, "Linearly constrained generalized Lloyd algorithm for reduced codebook vector quantization", *IEEE Transactions on Signal Processing*, vol. 49, no. 7, pp. 1501-1509, July 2001.

[39] Y. Hamamoto, S. Uchimura, M. Watanabe, T. Yasuda, Y. Mitani and S. Tomita, "A Gabor Filter-based Method for Recognizing Handwritten Numerals", *Pattern Recognition*, vol. 31, no. 4, pp. 395-400, 1998.

[40] D. M. Weber and D. P. Casasent, "Quadratic Gabor filters for object detection", *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 218-230, February 2001.

[41] F. Murtagh and J. Starck, "Pattern Clustering Based on Noise Modeling in Wavelet Space", *Pattern Recognition*, vol. 31, no. 7, pp. 847-855, 1998.

[42] I. R. Greenshields and J. A. Rosiene, "A Fast Wavelet-based Karhunen-Loeve Transform", *Pattern Recognition*, vol. 31, no. 7, pp. 839-845, 1998.

[43] T. S. Lee, "Image Representation Using 2D Gabor Wavelets", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959-971, October 1996.

[44] A. Kumar and G. K. H. Pang, "Defect detection in textured materials using Gabor filters", *IEEE Transactions on Industry Applications*, vol. 38, no. 2, pp. 425-440, March-April 2002.

[45] B. S. Manjunath and W. Y. Ma, "Texture Feature for Browsing and Retrieval of Image Data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837-842, August 1996.

[46] M. Park, J. S. Jin, and L. S. Wilson, "Fast content-based image retrieval using quasi-gabor filter and reduction of image feature dimension", *Proceedings. Fifth*

IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 178-182, 2002.

[47] R. Porter and N. Canagarajah, "Robust rotation-invariant texture classification: wavelet, Gabor filter and GMRF based schemes", *IEE Proceedings- Vision, Image and Signal Processing*, vol. 144, no. 3, pp. 180-188, June 1997.

[48] T. N. Tan, "Rotation Invariant Texture Features and Their Use in Automatic Script Identification", ", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no.7, pp. 751-756, July 1998.

[49] R. Porter and N. Canagarajah, "Gabor Filters for Rotation Invariant Texture Classification", *IEEE International Symposium on Circuit and Systems*, pp. 1193-1196, June 1997.

[50] C. C. Chang, S. M. Hwang, and D. J. Buehrer, "A Shape Recognition scheme based on relative distances of feature points from the centroid", *Pattern Recognition*, vol. 24, no. 11, pp. 1053-1063, 1991.

[51] S. C. Pei and J. H. Horng, "Corner point Detection using nest moving average", *Pattern Recognition*, vol. 27, no. 11, pp. 1533-1537, 1994.

[52] A. Held, K. Abe, and C. Arelli, "Towards a Hierarchical Contour Description via Dominant Point Detection", *IEEE Transactions on Systems*, Man, and Cybernetics, vol. 24, no. 6, pp. 942-949, June 1994.

[53] J. W. Chung, J. H. Moon and J. K. Kim, "Conditional differential chain coding for lossless representation of object contour", *Electronics Letters*, vol. 34, no. 1, pp. 55-56, January 1998.

[54] E. Bribiesca, "A new chain code", *Pattern Recognition*, vol. 32, pp. 235-251, 1999.

[55] D. Cheng and H. Yan, "Recognition of Handwritten digits based on contour information", *Pattern Recognition*, vol. 31, no. 3, pp. 235-255, 1998.

[56] D. Zhang and G. Lu, "Enhanced generic Fourier descriptors for object-based image retrieval", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 3668-3671, 2002.

[57] M. F. Wu and H. T. Sheu, "Contour-based correspondence using Fourier descriptors", *IEE Proceedings- Vision, Image and Signal Processing*, vol. 144, no. 3, pp. 150-160, June 1997.

[58] R. Tello, "Fourier descriptors for computer graphics", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 5, pp. 861-865, May 1995.

[59] G. M. T. Man, J. C. H. Poon, and W.C. Siu, "Viewpoint-invariant Fourier descriptor for 3-D planar object recognition", *Electronics Letters*, vol. 30, no. 20, pp. 1664-1665, September 1994.

[60] J. H. Chuang, "A potential-based approach for shape matching and recognition", *Pattern Recognition*, vol. 29, no. 3, pp. 463-470, 1996.

[61] A. K. Jain, Y. Zhong and S. Lakshmanan, "Object Matching Using Deformable Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 267-277, March 1996.

[62] I. S. Hsieh and K. C. Fan, "Multiple Classifiers for Color Flag and Trademark Image Retrieval", *IEEE Transactions on Image Processing*, vol. 10, no. 6, pp. 938-950, June 2001.

[63] J. P. Eakins, J. M. Boardman, and M. E. Graham, "Similarity retrieval of trademark images", *IEEE Multimedia*, vol. 5, no. 2, pp. 53–63, April-June 1998.

[64] M. K. Hu, "Visual Pattern Recognition by Moment Invariants", *IRE Trans. Inform. Theory*, vol. IT-8, pp. 179-187, February 1962.

[65] T. H. Reiss, "The revised fundamental theorem of moment invariants", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, Aug 1991.

[66] C. C. Chen, "Improved Moment Invariants for shape discrimination", *Pattern Recognition*, vol. 26, no. 5, pp. 683-686, 1993.

[67] C. H. Teh and R. T. Chin, "On image analysis by the methods of moments", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 496-513, July 1988.

[68] A. Wallin and O. Kubler, "Complete sets of complex Zernike moment invariants and the role of the pseudoinvariants", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, pp. 1106-1110, November 1995.

[69] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489-497, May 1990.

[70] S. X. Liao and M. Pawlak, "On the accuracy of Zernike moments for image analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1358-1364, December 1998.

[71] L. Wang and G. Healey, "Using Zernike moments for the illumination and geometry invariant classification of multispectral texture", *IEEE Transactions on Image Processing*, vol. 7, no. 2, pp. 196-203, February 1998.

[72] P. Raveendran, S. Omatu, and S. H. Ong, "Neuro-pattern classification using Zernike moments and its reduced set of features", *Intelligent Systems Engineering*, vol. 3, no. 4, pp. 230-235, Winter 1994.

[73] A. Blumenkrans, "Two-Dimensional Object Recognition using a Two-Dimensional Polar Transform", *Pattern Recognition*, vol. 24, no. 9, pp. 879-890, 1991.

[74] B.G. Quinn, "Estimating frequency by interpolation using Fourier coefficients", *IEEE Transactions on Signal Processing*, vol. 42, no. 5, pp. 1264-1268, May 1994.

[75] F. H. Cheng, W. H. Hsu, and M. Y. Chen, "Recognition of handwritten Chinese characters by modified Hough transform techniques", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 4, pp. 429-439, April 1989.

[76] M. Atiquzzaman, "Multiresolution Hough transform-an efficient method of detecting patterns in images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1090-1095, November 1992.

[77] R. Cucchiara and F. Filicori, "The vector-gradient Hough transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 746-750, July 1998.

[78] N. Bennett, R. Burridge, and N. Saito, "A method to detect and characterize ellipses using the Hough transform", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 7, pp. 652-657, July 1999.

[79] O. Chutatape and L. Guo, "A modified Hough transform for line detection and its performance", *Pattern Recognition*, vol. 32, no. 2 pp. 181-192, 1999.

[80] G. Borgefors, "Distance transformations in digital images", *Computer Vision, Graphics, Image Processing*, vol. 34, pp. 344-371.

[81] H. C. Liu and M. D. Srinath, "Partial Shape Classification Using Contour Matching in Distance Transformation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 11, pp. 1072-1079, November 1990.

[82] D. P. Huttenlocher, G. A. Klanderman and W. J. Rucklidge, "Comparing Image Using the Hausdorff Distance", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850-863, September 1993.

[83] B. Takacs, "Comparing face images using the modified Hausdorff Distance", *Pattern Recognition*, vol.31, no. 12, pp. 1873-1881, 1998.

[84] D. G. Sim and R. H. Park, "Two-dimensional object alignment based on the robust oriented Hausdorff similarity measure", *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 475-483, March 2001.

[85] J. A. Sirat, "A Fast Neural Algorithm for Principal Component Analysis and Singular Value Decomposition", *International Journal of Neural Systems*, vol. 2, nos. 1&2, pp. 147-155, 1991.

[86] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, January 1990.

[87] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[88] H. Peng and D. Zhang, "Dual eigenspace method for human face recognition", *Electronic Letters*, vol. 33, no. 4, pp. 283-284, February 1997.

[89] W. L. Poston and D. J. Marchette, "Recursive Dimensionality Reduction using Fisher's Linear Discriminant", *Pattern Recognition*, vol.31, no. 7, pp. 881-888, 1998.

[90] D. L. Swets and J. Weng, "Using Discriminant EigenFeatures for Image Retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831-836, August 1996.

[91] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transactions on Pattern Analysis and Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.

[92] F. Goudail, E. Lange, T. Iwamoto, K. Kyuma, and N. Otsu, "Face Recognition System Using Local Autocorrelations and Multiscale Integration", *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, pp. 1024-1028, October 1996.

[93] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models", *International Journal Computer Vision*, pp. 321-331, 1988.

[94] K. M. Lam and H. Yan, "Fast greedy algorithm for active contours", *Electronic Letters*, vol. 30, pp. 21-22, January 1994.

[95] K. M. Lam and H. Yan, "Fast algorithm for locating head boundaries", *Journal of Electronic Imaging*, vol. 3, no. 4, pp. 351-359, October 1994.

[96] K. M. Lam and H. Yan, "Locating and extracting the eye in human face images", *Pattern Recognition*, vol. 29, no. 5, pp. 771-779, 1996.

[97] L. D. Cohen, "On Active Contour Models and Balloons", CVGIP: *Image Understanding*, vol. 53, no. 2, March, pp. 211-218, 1991.

[98] Y. Y. Wong, P.C. Yuen, and C. S. Tong, "Segmented snake for contour detection", *Pattern Recognition*, vol. 31, no. 11, pp. 1669-1679, 1998.

[99] Y. Y. Wong, P.C. Yuen, and C. S. Tong, "Contour length terminating criterion for snake model", *Pattern Recognition*, vol. 31, no. 5, pp. 597-606, 1998.

[100] H. Blum and R. N. Nagel, "Shape Description Using Weighted Symmetric Axis Features", *Pattern Recognition*, vol. 10, pp. 167-180, 1978.

[101] B. K. Jang and R. T. Chin, "One-Pass Parallel Thinning: Analysis, Properties, and Quantitative Evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1129-1140, 1992.

[102] P. K. Saha and B. B. Chaudhuri, "Detection of 3D Simple Points for Topology Preserving Transformations with application to Thinning", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 10, pp. 1028-1032, 1994.

[103] L. Lam and C. Y. Suen, "An Evaluation of Parallel Thinning Algorithms for Character Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp. 914-919, 1995.

[104] S. S. O. Choy, C. S. T. Choy, and W. C. Siu, "Note: New Single-Pass Algorithm for Parallel Thinning", *Computer Vision and Image Understanding*, vol. 62, no. 1, pp. 69-77, 1995.

[105] Y. Xia, "Skeletonization Via the Realization of the Fire Front's Propagation and Extinction in Digital Binary Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 10, pp. 1076-1086, 1989.

[106] F. Leymarie and M. D. Levine, "Simulating the Grassfire Transform Using an Active Contour Model", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, pp. 56-75, 1992.

[107] U. Montanari, "Continuous Skeletons from Digitized Images", *J. ACM*, vol. 16, pp. 534-590, 1969.

[108] F. L. Bookstein, "The Line Skeleton", *CGIP*, vol. 11, pp. 123-137, 1979.

[109] D. T. Lee, "Medial Axis Transform of a Planar Shape", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, pp. 363-369, 1982.

[110] R. L. Ogniewicz and O. Kübler, "Hierarchic Voronoi skeletons", *Pattern Recognition*, vol. 28, no. 3, pp. 343-359, 1995.

[111] J. W. Brandt and V. R. Algazi, "Continuous Skeleton Computation by Voronoi Diagram", *CVGIP: Image Understanding*, vol. 55, pp. 329-338, 1992.

[112] L. Ji and J. Piper, "Fast Homotopy-Preserving Skeletons Using Mathematical Morphology", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 6, pp. 653-664, 1992.

[113] P. E. Trahanias, "Binary shape recognition using the morphological skeleton transform", *Pattern Recognition*, vol. 25, pp. 1277-1288, 1992.

[114] R. Kresch and D. Malah, "Skeleton-based Morphological Coding of Binary Images", *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1387-1399, 1998.

[115] C. Arcelli and G. Sanniti di Baja, "Euclidean skeleton via centre-of-maximal-disc extraction", *Image and Vision Computing*, vol. 11, no. 3, pp. 163-173, 1993.

[116] R. Kimmel, D. Shaked, and N. Kiryati, "Skeletonization via Distance Maps and Level Sets", *Computer Vision and Image Understanding*, vol. 62, no. 3, pp. 382-391, 1993.

[117] F. Nilsson and P. E. Danielsson, "Note: Finding the Minimal Set of Maximum Disks for Binary Objects", *Graphical Models and Image Processing*, vol. 59, no. 1, pp. 55-60, 1997.

[118] Y. Ge and J. M. Fitzpatrick, "On the Generation of Skeletons from Discrete Euclidean Distance Maps", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 11, pp. 1055-1066, 1996.

[119] M. A. Butt and P. Maragos, "Optimum Design of Chamfer Distance Transforms", *IEEE Transactions on Image Processing*, vol. 7, no. 10, pp. 1477-1484, 1998.

[120] P. E. Danielsson, "Euclidean Distance Mapping", *Computer Graphics and Image Processing*, vol. 14, pp. 227-248, 1980.

[121] Q. Z. Ye, "The signed Euclidean distance transform and its applications", *in Proceedings, 9th International Conference on Pattern Recognition*, Rome, pp. 495-499, 1988.

[122] H. I. Choi, S. W. Choi, and H. P. Moon, "Mathematical Theory of Medial Axis Transform", *Pacific Journal of Mathematics*, vol. 181, no. 1, pp. 57-87, 1997.

[123] W. P. Choi, K. M. Lam, and W. C. Siu, "An Efficient and Accurate Algorithm for Extracting a Skeleton", *in Proceedings, 15th International Conference on Pattern Recognition*, Barcelona, vol. 3, pp. 742-745, 2000.

[124] W. P. Choi, K. M. Lam, and W. C. Siu, "An Adaptive Contour Model for Highly Irregular Boundaries", *Pattern Recognition*, vol. 34, no. 2, pp. 323-331, 2001.

[125] D. Shaked and A. M. Bruckstein, "Pruning Medial Axes", *CVIU*, vol. 69, no. 2, pp. 156-169, 1998.

[126] B. Günsel and A. M. Tekalp, "Shape similarity matching for query-by-example", *Pattern Recognition*, vol. 31, no. 7, pp. 931-944, 1998.

[127] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 729 -736, July 1995.

[128] L. K. Huang and M. J. J. Wang, "Efficient shape matching through model-based shape recognition", *Pattern Recognition*, vol. 29, no. 2, pp. 207-32, 1995.

[129] H. Nishida, "Model-based shape matching with structural feature grouping", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, 315-320, March 1995.

[130] R. Brunelli and O. Mich, "Image retrieval by examples", *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 164-171, Sept. 2000.

[131] M. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object Matching", *Proc. 12th Int. Conf. on Pattern Recognition (ICPR)*, Jerusalem, Israel, pp. 566-568, 1994.

[132] K. H. Lin, B. F. Guo, K. M. Lam and W. C. Siu, "Human face recognition using a spatially weighted modified Hausdorff distance", *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 477-480, 2001.

[133] G. Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 849 -865, Nov. 1988.

[134] M. T. Goodrich, S. B. Mitchell and M. W. Orletsky, "Approximate Geometric Pattern Matching Under Rigid Motions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 371-379, April 1999.

[135] X. Yi and O. I. Camps, "Line Feature-Based Recognition Using Hausdorff Distance", *Proc. Int'l Symp. Computer Vision*, pp. 79-84, November 1995.

[136] X. Yi and O. I. Camps, "Line-Based Recognition Using A Multidimensional Hausdorff Distance", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 901-916, September 1999.

[137] D. G. Sim, O. K. Kwon, and R. H. Park, "Object Matching Algorithms Using Robust Hausdorff distance Measures", *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 425-429, March 1999.

[138] L. Jia and L. Kitchen, "Object-Based Image Similarity Computation Using Inductive Learning of Contour-Segment Relations", *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 80-87, January 2000.

[139] L. J. Latecki and R. Lakämper, "Shape Similarity Measure Based on Correspondence of Visual Parts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1185-1190, October 2000.