MUVIS: A System for Content-Based Image Retrieval

Faouzi Alaya Cheikh
Tampere University of Technology

Abstract

The thesis considers different aspects of the development of a system called **MUVIS** [1] developed in the **MuVi** [2] project for content-based indexing and retrieval in large image databases. The issues discussed are system design, graphical user interface design, shape features extraction and their corresponding similarity measures, and the use of relevance feedback to enhance the retrieval performance.

Query by content, or content-based retrieval has recently been proposed as an alternative to text-based retrieval for media such as images, video and audio. Text-based retrieval is no longer appropriate for indexing such media, for several reasons. Firstly, keyword annotation is labor intensive, and it is not even possible when large sets of images are to be indexed. Secondly, these annotations are drawn from a predefined set of keywords which cannot cover all possible concepts images may represent. Finally, keywords assignment is subjective to the person making it. Therefore, content-based image retrieval (CBIR) systems propose to index the media documents based on features extracted from their content rather than by textual annotations. For still images, these features can be color, shape, texture, objects layout, edge direction, etc.

Query by content has been a very active research field, with many systems proposed by industrial and academic teams. Building such systems requires expertise in different fields of information technology: databases and indexing structures, system design and integration, graphical user interfaces (GUI), signal processing and analysis, man-machine interaction, user psychology, etc.

The development of the MUVIS system for content-based image retrieval was a challenging work that required solutions to different problems. Therefore, this thesis contributed to different phases of the system development; including system design, graphical user interface development, feature extraction, similarity measures, and relevance feedback. The system is designed in a client-server architecture. The graphic user interface proposes a new way of visualizing the color features used and permits intuitive interactions with it. Moreover, it allows several ways of query formulation: by example image, by histogram, or by single object

[1] MUVIS is a content-based image retrieval system

[2] MuVi stands for Multimedia Data Processing Applied to Image and Video Databases Manipulation

or composed image.

The second contribution of this thesis is the study of shape features and related similarity measures. These can be classified into two categories: wavelet-based approaches and ordinal correlation based ones. The former is a set of contour-based descriptors, where the features are extracted at significant topological points of the objects contours, which are the high curvature points (HCP); detected based on the local maxima of the wavelet transform of the 1D contour representation. The retrieval results are compared to those of the curvature scale space (CSS) and to a ranked list of images by a group of human users. The results show that our proposed schemes have comparable or better results than the CSS. Moreover, the proposed algorithms are computationally more efficient than the CSS, and are not sensitive to noise corruption or partial occlusion of the contour. The latter approach is a region-based technique, which applies the ordinal correlation framework on the 2D distance transform of binary contour image to extract the shape features. The distance transformation disseminates the information highly concentrated in the binary contour into the surrounding pixels, making the approach less sensitive to small variation of the contour caused by noise, imperfect segmentation or small object alignment errors. Being region-based, this approach can be applied to simple contours as well as shapes composed of several independent regions which may even contain internal holes.

The third contribution of this thesis consists of the implementation of a relevance feedback (RF) scheme in our CBIR system. RF techniques allow the CBIR systems to learn from the user interaction and enhance their performance in subsequent retrieval iterations. The proposed technique combines both re-weighting of the feature vector entries and adaptive resolution of the feature vectors to achieve a higher performance in discriminating relevant and irrelevant objects for a given query.

Preface

The work performed in this thesis has been carried out at the Signal Processing Institute of Tampere University of Technology, Finland. The work is part of the MuVi project, whose emphasis is on content-based image and video retrieval.

In the first place my gratitude goes to Prof. Moncef Gabbouj, the supervisor of my thesis, for his valuable guidance and expertise as well as for his kind advice, encouragement, and constant support. I'm grateful to Prof. Jaakko Astola, who always gave me help and support. I would like to thank the thesis reviewers Dr. Noel O'Connor and Prof. Erkki Oja for their feedback and constructive comments on the manuscript.

I would like to thank all the colleagues at the Institute of Signal Processing for the pleasant work atmosphere. Special thanks go to all those who made the institute scientifically as well as culturally very rich. I'm indebted to the members of the Image and Video Analysis group for the interesting and fruitful discussions, especially, Azhar Quddus, Bogdan Cramariuc, Mejdi Trimeche and Iftikhar Ahmed, with whom I worked very closely. Prof. F. Mokhtarian and S. Abbasi are acknowledged for providing us the fish image database, used in parts of this work.

The research work published in this thesis was financially supported by the Tampere Graduate School in Information Science and Engineering (TISE) and Academy of Finland Center of Excellence (SPAG) and EU network of excellence (SCHEMA), whom I gratefully acknowledge.

Special thanks go to all my friends especially to the small Tunisian community in Tampere.

Finally, I wish to thank my parents, my sister and my brothers for their endless love and support.

Faouzi Alaya Cheikh
Tampere, March 2004

Contents

List of Abbreviations

1D	-	One-Dimensional
2D	-	Two-Dimensional
3D	-	Three-Dimensional
AR	-	Auto-Regressive
ART	-	Angular Radial Transform
AV	-	Audio-visual
BiM	-	Binary format for MPEG-7
CAR	-	Circular Autoregressive
CBIR	-	Content-Based Image Retrieval
COST 211	-	Coopération européenne dans la recherche Scientifique et Technique 211
COST 211 AM	-	COST 211 Analysis Model
DBMS	-	Database Management System
D	-	Descriptor
DFT	-	Discrete Fourier Transform
DS	-	Description Scheme
DT	-	Distance Transform
DDL	-	Description Definition Language
EM	-	Expectation Maximization
FD	-	Fourier Descriptor
GLCM	-	Gray Level Co-occurrence Matrix
GoF	-	Group of Frames
GoP	-	Group of Pictures
HCP	-	High Curvature Point
HMM	-	Hidden Markov Model
HMMD	-	Hue-Max-Min-Diff color space
HVS	-	Human Visual System
HSV	-	Hue, Saturation and Value
IR	-	Information Retrieval
ISO	-	International Organization for Standardization
IVA	-	Image and Video Analysis
JNI	-	Java Native Interface

LSI	-	Latent Semantic Indexing
MIDP	-	Mobile Information Device Profile
ML	-	Maximum Likelihood
MPEG	-	Motion Picture Experts Group
MPEG-1	-	ISO standard
MPEG-2	-	ISO standard
MPEG-4	-	ISO standard
MPEG-7	-	ISO standard
MuVi	-	Multimedia Data Processing Applied to Image and Video Databases Manipulation
MUVIS	-	Multimedia Video Indexing and Retrieval System
OCF	-	Ordinal Correlation Framework
PDA	-	Personal Digital Assistant
QBIC	-	Query By Image Content
RF	-	Relevance Feedback
RGB	-	Red, Green, Blue
SD	-	Shape Desriptor
SOM	-	Self-Organizing Map
SVM	-	Support Vector Machine
TEKES	-	The National Technology Agency of Finland
TS-SOM	-	Tree Structured Self-Organizing Map
UA	-	Universal Axes
VQ	-	Vector Quantization
VSM	-	Vector Space Model
WT	-	Wavelet Transform
WTMM	-	Wavelet Transform Modulus Maxima
W3C	-	WWW Consortium
WWW	-	World Wide Web
XM	-	eXperimentation Model
XML	-	Extensible Markup Language

List of Tables

List of Figures

Chapter 1

Introduction

The world we live in is becoming inherently inter-connected and digital. The amount of information available nowadays, in digital format, to ordinary people is breath taking. The fast growth of the amount of digital material available can be explained by the new models of digital media production, distribution and consumption. Earlier, only information consumption was accessible to the common public. Information creation and distribution were restricted to experts and media companies. This changed radically with the appearance of the Internet and the first web browser; which revolutionized the distribution of information. The ease of creating new Web documents and linking them to existing ones caused exponential growth of the publicly available digital material. The ease of information exchange incited millions of people to create their own web pages and to enrich it with images, video, audio and music. In 1993 the first web browser, **Mosaic** [8], was released, and the growth rate of the Internet was an incredible 341% and it still growing at around 100% per year [25]. The web browsers were, mainly, graphical user interfaces that access the World Wide Web and retrieved the data stored therein. As a graphical user interface, these web browsers also allowed easy access to multi-media data such as pictures and sound files.

The wealth of information available freely on the web induced a major problem for the end users: how to find the information needed? This problem was felt in the very early years of the web. As early as 1993, solutions to the online information retrieval problem were proposed: keyword-based “Web Search Engines” such as Lycos and Yahoo. At the end of 1993, Lycos indexed a total of 800,000 web pages [8]. Many other search engines followed soon after that, such as the very popular Altavista and Google. A search engine is mainly a database and the tools to generate that database and search it. The popularity of web search engines, which never decreased, proves the need for efficient tools to search, filter and browse through the wealth of digital media available.

Traditional search engines were built with the assumption that the media in general and the one available on the web in particular, was mainly composed of text or its content can be described by a set of keywords associated to it. Therefore early web search tools were based on the classic techniques of text-based information retrieval. With the availability of cheap digital media acquisition tools such as web cameras, scanners, etc. the digital material on the web became inherently rich in images, audio and video sequences. Therefore, tools to browse and search such audio-visual (AV) media were needed. A straightforward way of using the existing tools for information retrieval to index and search these collections of audiovisual material, is to first annotate the records by keywords and then use the text-based Database-Management Systems (DBMS) to retrieve it. Several approaches were proposed to use keyword annotations to index and retrieve images [13, 15, 133]. Comprehensive surveys in information retrieval can be found in e.g. [14, 142]. These approaches are no longer valid however, since annotating images by textual keywords is neither desirable nor possible in most of the cases. Therefore, new ways of indexing, browsing and retrieval of still images are needed.

The rest of this thesis will focus on the problem of indexing and retrieval of still images, as a particular case of the general problem of content-based indexing and retrieval of multi-media documents. However, parts of the discussion in this thesis are still valid for other types of media, especially video; most of the algorithms and approaches depend heavily on prior knowledge of the type of media at hand and application context.

1.1 Query by Content

"A Picture is Worth One Thousand Words"

– A proverb ascribed to the Chinese philosopher Confucius [1]

This proverb is a variation of the original sentence: "One Picture is Worth Ten Thousand Words" actually fabricated by an advertising executive in California in 1927 [5], which suggests that pictures can replace words. The executive assumed that consumers would be compelled to buy a product that had the weight of Chinese philosophy behind it.

Recently, this proverb has been used extensively in the information technology research community to say that image content cannot be well described by text annotations and therefore traditional information retrieval systems are not appropriate for image retrieval. As pointed out by Picard [106], if there were a unique set of words which best characterized an image, the query by content problem would be effectively solved.

The framework of keyword-based image retrieval is no longer appropriate to handle such rich media because it suffers from two major drawbacks:

[1] Confucius, a Chinese philosopher who lived from 551 to 479 B.C. From Liki (Record of Rites), Chapter 42, "The Wisdom of Confucius".

- there is no fixed set of words that describe the content of images,
- the keywords annotation is subjective to the person who does it.

Therefore, even if we assume that in a specific context we can have a fixed set of keywords to annotate the images, using one word or another relies on the subjective judgment of the person performing the task. Due to the rich content of images, different persons may perceive them differently and annotate them with different keywords. These two problems became acute with the increase of the size of the image collections. The amount of manual labor required makes this approach prohibitive in certain applications.

To overcome these difficulties, query by content or content-based retrieval has recently been proposed as an alternative to text-based retrieval for retrieving media such as images, audio and video. It indexes the media documents based on features extracted from their content rather than by textual annotations. For still images these features can be color, shape, texture, object layout, edge direction, etc. This relatively new research field, visual information management systems, should not be considered as an application of the existing state of the art (in computer vision and databases) to manage and process images, since the development in these two fields have traditionally been (and still are) rather different.

Content-based image retrieval (CBIR) has been an active area of research, promising to provide powerful tools for database management in the near future. Building an efficient image retrieval system requires expertise from different disciplines to cover several major aspects; such as system design, feature extraction, high dimensional indexing structures, similarity measures, perception analysis, semantic analysis, and visual content abstraction from the automatically extracted low-level features e.g. using mechanisms such as relevance feedback, user interfaces and user studies.

CBIR captured the interest of both industrial as well as the academic research communities. The interest is large enough that the Motion Picture Experts Group (MPEG) dedicated a standard called MPEG-7: the “Multimedia Content Description Interface”, to allow the interoperability between the devices and applications attempting to solve parts of this problem. Since the early 90’s many techniques in this research topic have been developed, many special issues of leading journals [6, 30, 45, 85, 142] and entire books [9, 12, 31, 34, 35, 44, 76, 87, 127] have been dedicated to this topic, and many image retrieval systems are already in use. A good survey, even though incomplete, of CBIR systems can be found in [151].

Research in CBIR tries to solve some key problems, which have to be addressed when building such modern image database systems [92, 106]:

- How to abstract images?
- What computer measures can mimic human “perceptual similarity”?

- How do we weave context into such measures?
- What image model facilitates access to content while representing the image as efficiently as possible?
- What database structure will be the best for this new type of data?

This thesis does not attempt to cover all aspects of the content-based image retrieval problem nor does it represent a detailed literature review of the sheer amount of published work in this field.

1.2 Organization of the Thesis

This thesis consists of two parts: the first part gives an introduction to query by content, presents an overview of the MPEG-7 standard and reviews some of the major systems of query by content. The second part discusses the design and implementation issues related to the MUVIS system.

This chapter gives an introduction to the problem of content-based indexing and retrieval of images. Chapter 2 further presents an overview of the MPEG-7 standard and reviews the major CBIR systems found in the literature. Its first part focuses on visual information description: including low-level visual descriptors and segment description schemes; while its second part gives a brief overview of the major existing systems for CBIR.

Chapters 3–6 discuss the issues related to building the MUVIS system. They are largely based on the author's original publications [17, 18, 19, 20, 21, 22, 23, 109, 147]. These chapters focus on the fundamental bases for content-based image retrieval, i.e. multi-dimensional indexing, system design, visual feature extraction, similarity measures and finally learning from user interaction through relevance feedback. Chapter 3 presents the building blocks of the MUVIS system and their functionalities, and places each of these functionalities in the retrieval scenario [21]. In CBIR systems the user is the final judge of the performance of a retrieval system. Therefore, the users' interaction with the system is very important since his assistance in the query formulation can improve greatly the quality of the features extracted. Both in MPEG-4 [66] and MPEG-7 [87, 88] human assistance in tasks like segmentation and analysis are valid procedures since they can improve the quality of the resulting representation of the content of the audiovisual data. Such interactions may influence greatly the performance of retrieval systems. Therefore, the interfaces of such systems ought be user friendly and allow a high level of interactivity. In this chapter the GUI of MUVIS is particularly emphasized, since it plays a crucial role in the query formulation and retrieval process. Special attention is given to the histogram editing tool and the image editing tool. Also the different ways of formulating queries using these tools are described. The most straightforward query is by example image, with the image being internal or external to the database. The image editing tool can be used as a composition tool allowing the user to create a new image or to edit an existing one and query with it. Furthermore, it allows objects based queries where the user outlines an area or an object and use it as the query. The histogram editing tool allows the user to extract, edit and query with a histogram. It further allows the association of weights to the histogram entries to emphasize, reject or ignore one or more colors. This tool was developed to test and compare different histogram features and similarity measures. It is intended to be used by advanced users to manipulate the feature vectors and query with them. Later a brief description is given of the features and the similarity measures used to compare them. Finally, two extensions of MUVIS, to mobile devices and to image sequences, are mentioned at the end of this chapter.

In Chapters 4 and 5, two categories of shape features are proposed and corresponding similarity measures are derived. The first category, presented in Chapter 4, is contour-based approaches, where the salient points on the contour are detected based on the modulus maxima of the wavelet transform coefficients. Two types of features are extracted at these high curvature points, the first are shape visual features e.g. topological measurements. The second type of features is directly extracted from the wavelet coefficients and used as entries of the feature vectors to describe the object. For each type of features we derived a similarity measure to compare the objects based on their corresponding representations. Our work on these contour-based approaches was published in [22, 23, 109, 147].

The second category is a region-based shape similarity estimation framework, presented in Chapter 5. Being region-based this approach allows the estimation of similarity between objects made of a single or multiple regions, which may even contain internal holes. In this approach the binary image containing the contour is transformed using a distance transform (DT) to obtain a distance map. The distance map representations are compared using the ordinal correlation framework. This allows to compare the contour activity in a region of the image in a scale relative to the block size of the ordinal correlation mesh. Moreover, the distance transform reduces the sensitivity of the similarity measure to small alignment errors. The sensitivity of the system to the contour details depends on the block size. Therefore, shapes can be compared at different scales or resolutions. Furthermore, the DT is derived using two approaches, depending on whether the user wants to look for similar objects based on their boundary details or their inner structure (skeleton). Work on this region-based framework was published in [18, 19, 20].

As the user is the consumer of the retrieval results, he is only judge of the retrieval results relevance to his query. The users' feedback (used in several CBIR systems) allows the system to learn more about the interest of the user, bridges the gap between the low-level features extracted automatically and the high-level concepts the user has and may also be used to account for the users' subjectivity. Relevance feedback techniques are reviewed in Chapter 6 and an approach to integrate a hybrid relevance feedback mechanism into the ordinal-correlation framework is proposed. The proposed approach operates in two ways: first by similarity measure refinement using weight estimated based on statistics extracted from the positive and negative feedback examples; second by query expansion, where important regions of the image are analyzed using finer resolution. This work was published in [17], additional results are being submitted to future conferences.

1.3 Author's Contribution

The author's contribution to the field of query by content is presented in Chapters 3 to Chapter 6. The original contribution is in the MUVIS system design and implementation. Thus the structure of the thesis follows the logical development phases of MUVIS; covering the major bases of CBIR i.e. system and graphical user interface design, automatic visual feature extraction and comparison, and relevance feedback.

The main contributions of this thesis can be summarized in the following points:

- System design, in [16] and [21]
- GUI design especially query formulation tools design and implementation,
- Contour-based shape feature extraction:
  - high curvature points detection based on wavelet transform modulus maxima ,
  - definition of a topological feature extracted at the detected HCPs [109],
  - definition of two wavelet coefficient-based features for shape characterization, extracted at the HCPs [22, 23, 147],
  - similarity measures and similarity estimation algorithms for object based retrieval, based on the above mentioned features [22, 23, 109, 147],
- Region-based Shape similarity estimation framework: [18, 19, 20]
  - extension of the ordinal-correlation framework to compare objects based on distance transform of the binary contour images,
  - algorithm for feature vectors extraction and their comparison,
- Hybrid relevance feedback algorithm to tune the ordinal-correlation framework similarity measure and expand the query [17].

Chapter 2

Query by Content – An Overview

The large interest in CBIR resulted in a very fast increase of the amount of published research work and the number of systems developed, see Section 1.1. Furthermore, the different backgrounds of the researchers interested in CBIR made cooperation and results comparison a difficult task. These two factors emphasized the need for a standardization effort to allow the interoprability between the developed systems and to have a common terminology, and test material. Without such standardization effort, the evaluation of the performance of an specific approach or a given system would not be possible.

In this context the Motion Picture Experts Group (MPEG) dedicated a standard called MPEG-7: the “Multimedia Content Description Interface”, to allow this interoperability between the devices and applications attempting to solve parts of the query by content problem. MPEG-7, is the standard for rich descriptions of multimedia content, enabling sophisticated management, browse, search, and filtering of that content. MPEG-7 enables data exchange with standards such as Dublin Core (www.dublincore.org) and TV-Anytime (www.tv-anytime.org).

The next Section presents an overview of the MPEG-7 focusing on the visual aspects; while, Section 2.2 reviews the major CBIR systems. The last section briefly reviews some of the commonly used similarity measures in CBIR systems.

2.1 MPEG-7 Overview

MPEG-7 [88, 87], formally called “Multimedia Content Description Interface" is a standard for describing the multimedia data content. Its relation to previous ISO standards MPEG-1, MPEG-2 and MPEG-4 is summarized in Figure 2.1. MPEG-7 tries to develop forms of audiovisual information representations that go beyond the simple waveform or sample-based, compression-based (such as MPEG-1 and MPEG-2) or even object-based (such as MPEG-4) representations. These forms support some degree of interpretation of the information meaning, which can be passed onto, or accessed by, a device or a computer code. MPEG-7 is not aimed at a specific application in particular. It provides a rich set of standardized tools to describe multimedia content, allowing the development of a wide range of applications. Human users as well as automatic systems that process audiovisual information are within the scope of MPEG-7.

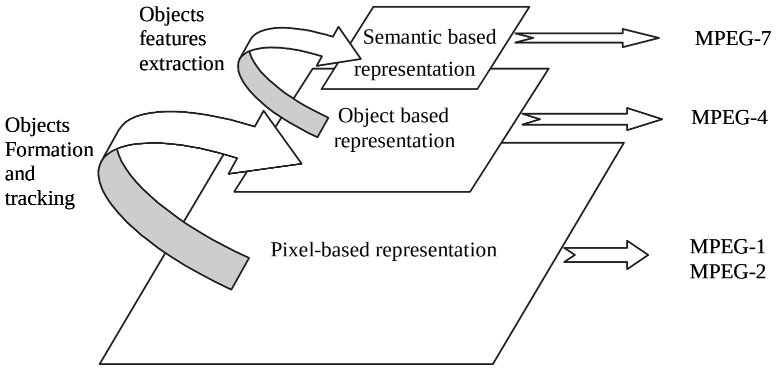


Figure 2.1: Relation between the MPEG standards

MPEG-7 offers a comprehensive set of audiovisual description tools (the metadata elements and their structure and relationships that are defined by the standard as descriptors (D) and Description Schemes (DS)). It specifies a Description Definition Language (DDL) to efficiently index and search material with associated MPEG-7 data. These searches will permit inquiries based on scenes, motion and visual content as well as text-based queries. AV material that has MPEG-7 data associated with it, can be indexed and searched for. This material may include: still pictures, graphics, 3D models, audio, speech, video, and information about how these elements are combined.

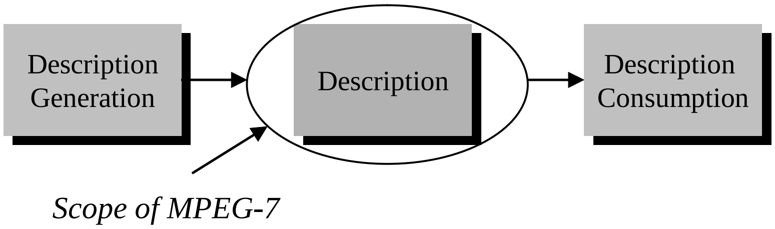


Figure 2.2: Scope of MPEG-7

The goal of the MPEG-7 standard is to allow the interoperability among devices and applications for searching, indexing, filtering, and accessing audiovisual (AV) content. MPEG-7 specifies the description of features related to the AV content as well as information related to the management of AV content. As illustrated in Figure 2.2, the scope of MPEG-7 is to define the representation of the description, i.e. the syntax and the semantics of the structures used to create

MPEG-7 descriptions. It does not provide normative tools for the generation nor for the consumption of the description. Such tools are not necessary to guarantee interoperability of MPEG-7 compliant systems. Furthermore, it allows future improvements to be included in MPEG-7 based applications and systems. To guarantee interoperability MPEG-7 specifies part of the extraction process of some of the low-level features.

As illustrated in Figure 2.3, MPEG-7 specifies four types of normative elements: Descriptors ($D_1, D_2, ...$), Description Schemes ($DS_1, DS_2, ...$), a Description Definition Language (DDL), and coding schemes. The MPEG-7 descriptions can be either in a textual form, based on the Extensible Markup Language (XML), suitable for editing, searching, filtering, and browsing or in a binary form suitable for storage, transmission, and streaming.

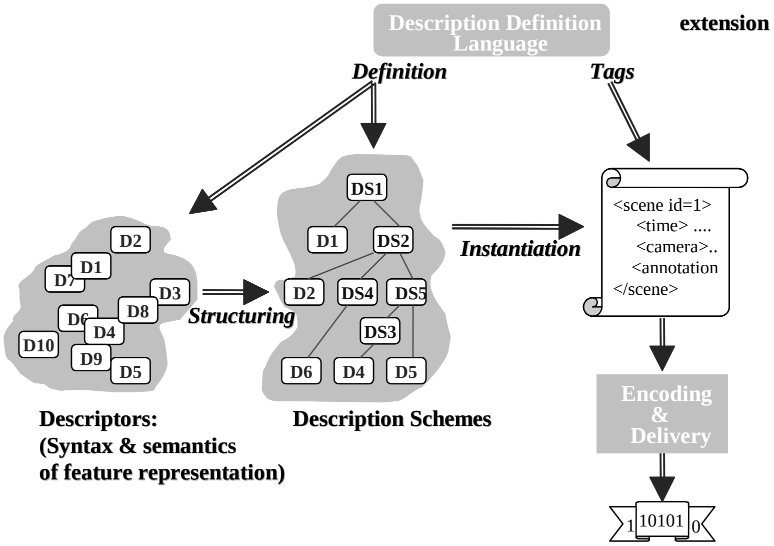


Figure 2.3: MPEG-7 normative elements

In order to describe AV content, a set of Descriptors is used. A Descriptor defines the syntax and the semantics of a given elementary feature. It can be used to describe low-level features as well as high-level features. The syntax of MPEG-7 Descriptors is defined by the Description Definition Language (DDL) which is an extension of the XML schema language [33]. Moreover, DDL can be used to create the syntax of new Descriptors.

Descriptors of an AV item are structured and related within a common framework based on Description Schemes (DSs). The model of the AV content description is defined by the DSs using the Descriptors as building blocks; see Figure 2.4, and the DDL is used to define the syntax of these DSs.

An instance of DSs and Descriptors is used as the description of given AV content. This description can be stored in the form of an XML document. This format

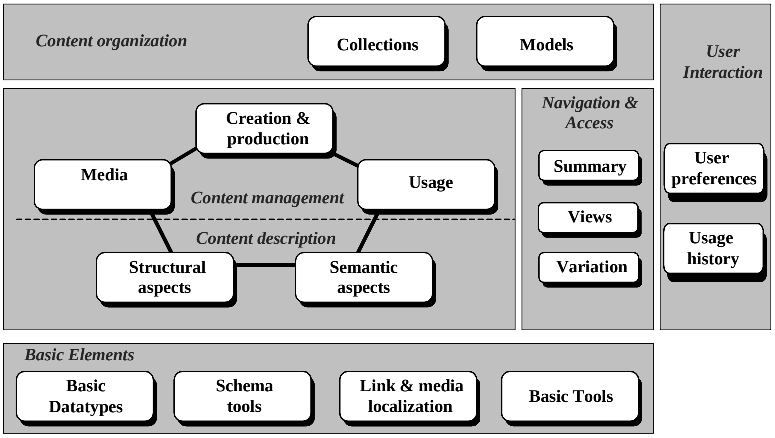


Figure 2.4: Overview of the MPEG-7 Multimedia DSs

allows efficient editing, searching, filtering, and processing. Since many available tools are XML-aware, XML was adopted as the first normative format in MPEG-7. On the other hand, this format is not suitable for streaming and is sensitive to transmission errors. To overcome these shortcomings, MPEG-7 defines a second format: Binary format for MPEG-7 (BiM). This format is suitable to streaming and compression. Moreover, MPEG-7 defines coding and decoding tools for this format. For a given piece of AV content, both of its representations in XML and BiM format are equivalent and can be encoded and decoded losslessly [87].

2.1.1 Multimedia Description Schemes

The multimedia DSs are organized into different functional areas, see Figure 2.4: basic elements, content management, content description, navigation and access, content organization, and user interaction. The MPEG-7 DSs represent a set of description tools. For a particular application, these DSs can be used to describe multimedia content. In the following we give a brief overview of each of these functional areas, further details can be found in [87].

The first set of DSs are called Basic elements because they represent the building blocks for descriptions or DSs. They provide the basic description functions including a number of schema tools, basic data types that provide a set of extended data and mathematical structures such as vectors and matrices.

MPEG-7 DSs for AV content management, describe information that is not present in the content itself but is important to many applications, such as title, creator, dates, genre, subject, language and parental guidance. These tools describe the following information:

1. creation and production,

2. media coding, storage and file formats,

3. content usage.

In MPEG-7, content description refers to information that can be perceived in the content. It can be subdivided into two types of descriptions: the first emphasizes the structural aspect of the AV signal whereas the second focuses on the conceptual aspects of the content.

The description of the structure of AV content relies on the notion of segments. The segment DS describes the result of a spatial, temporal, or spatio-temporal partitioning of the AV content. It can describe a hierarchical decomposition resulting in a segment tree. Moreover, the segment relation DS describes possible additional relationships among segments and allows the creation of graphs.

For some applications the structure of the AV content is not relevant, whereas the semantics of the content is, thus MPEG-7 proposes the semantic DS. The focus is put on events, objects in narrative worlds, concepts and abstractions rather than segments.

Similar to the segment DS, the conceptual aspects of description can be organized in a graph. The graph structure is defined by a set of nodes, representing the semantic notions, and a set of edges specifying the relationship between the nodes. Edges are described by the semantic relation DSs.

To allow easy navigation of AV content, MPEG-7 proposes DSs for describing summaries, views, and variations. The Summary DS describes semantically meaningful summaries and abstracts of the AV content. It allows navigation of the content in a hierarchical or sequential way.

The View DS describes structural views of the AV signals in the spatial or frequency domain to enable multiresolution access and progressive retrieval; while, the Variation DS describes the relationships between variations of the AV content. These variations can be compressed or low-resolution versions, summaries, different languages, and different modalities, such as audio, video, image, text, etc.

The content organization is built around two main DSs: the Collections DS and the Models DS. The former includes tools for describing collections of AV material, AV content descriptions, semantic concepts, mixed collections and collection structures in terms of relationships between collections. The latter describes parameterized models of AV content, descriptors, or collections. It involves two important DSs: the probability model and the analytic model DSs. The probability model DS describes different statistical functions and probabilistic structures, which can be used to describe samples of AV content and classes of Descriptors using statistical approximation. The analytic model DS describes a collection of examples of AV content or clusters of descriptors that are used to provide a model for a particular semantic class.

The user interaction DS describes preferences of users pertaining to the consumption of the AV content, as well as usage history.

2.1.2 Visual Features

MPEG-7 describes several low-level visual features to characterize the AV content. In this section, we present briefly the color, texture, shape and localization features. Motion and face features will not be reviewed.

Color

Seven color descriptors are standardized by MPEG-7: color space, color quantization, dominant colors, scalable color histogram, color structure, color layout, and Group of Frames/Group of Pictures (GoF/GoP) color.

The first two descriptors, color space and quantization, are intended to be used in conjunction with other color descriptors. Possible color spaces include:

- R, G, B,
- Y, Cr, Cb,
- H, S, V,
- monochrome,
- linear transformation matrix of R, G, B.

Both linear and nonlinear color quantization and lookup-tables are supported.

In many applications, it is enough to describe the color of an object, region or an entire image with a few colors, the dominant color descriptor is suitable for such cases. Color quantization is used to extract a small number of representative colors in each region/image. The percentage of each quantized color in the region is calculated correspondingly. A spatial coherence on the entire descriptor is also defined and is used in similarity retrieval.

The scalable color histogram descriptor is a color histogram in the HSV color space, encoded with a Haar transform. Its binary representation is scalable in terms of bin numbers and bit representation accuracy. The Scalable Color Descriptor is useful for image-to-image matching and retrieval based on color.

The Color Structure Descriptor captures both color content (similarly to a color histogram) and its structure. Its main functionality is image-to-image matching and its intended use is for still-image retrieval. The extraction method embeds color structure information into the descriptor by computing the relative frequency of all colors in a structuring element of 8×8 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. Color values are represented in the double-coned Hue-Max-Min-Diff (HMMD) color space, which is quantized non-uniformly into 32, 64, 128 or 256 bins.

The color layout descriptor effectively represents the spatial distribution of color of visual signals in a very compact form allowing high-speed retrieval and browsing. It targets not only image-to-image matching and high-speed sequence-to-sequence matching, but also hand-drawn sketch queries which is not supported by other color descriptors. The descriptor represents the DCT values of an image or a region that has been previously partitioned into 8×8 blocks and where each block is represented by its dominant color. The default number of coefficients is 12 for video frames while 18 coefficients are recommended for still pictures to achieve higher accuracy.

The last color descriptor is the Group of Frames/ Group of Pictures (GoF/GoP) color descriptor extends the Scalable Color Histogram Descriptor defined for still images to a video segment or a collection of still images. Two additional bits define how the color histograms of each frame are combined by average, median or intersection prior to the Haar transformation.

Texture

Three texture descriptors are standardized: homogeneous texture, edge histogram, and texture browsing.

Homogeneous texture is 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 the texture features associated with the regions can be used to index the image data. It provides a quantitative representation using 62 numbers (quantified to 8 bits each). The computation of this descriptor is based on filtering using scale and orientation selective kernels. Thirty filters (using Gabor functions) are used: 5 “scales" and 6 “directions" used in the multi-resolution decomposition. The first and second moments of the energy in the frequency bands represent the components of the texture descriptor. It is used for accurate search and retrieval.

The Texture Browsing Descriptor is useful for representing homogeneous texture for browsing type applications, and requires only 12 bits (maximum). It provides a perceptual characterization of texture, similar to a human characterization, in terms of regularity, coarseness and directionality. This is followed by analyzing the filtered image projections along the dominant orientations to determine the regularity (four possible levels) and coarseness (four possible values). The second dominant orientation and second scale feature are optional. Combined with the Homogeneous Texture Descriptor, they provide a scalable solution to representing homogeneous texture regions in images.

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. Therefore, it primarily targets image-to-image matching (query by example or by sketch), especially for natural images with nonuniform edge distribution.

Shape

Two dimensional shape representation techniques can be classified in two categories: region-based and contour-based. Therefore, MPEG-7 has standardized two shape descriptors, namely Region Shape and Contour Shape, to describe 2D shapes.

The shape of an object can be composed of one or more regions, and may also have some holes. Region Shape descriptor makes use of all pixels constituting the shape within a frame. Thus the Region Shape descriptor can describe complex shapes efficiently in a single descriptor. It captures the distribution of all the pixels of the shape. It is also robust to minor deformation along the boundary of the object.

The region-based descriptor is based on a 2D complex transform defined with polar coordinates on the unit disk, called Angular Radial Transform (ART). The ART has separable basis functions along the angular and radial dimensions. Twelve angular and three radial basis functions are used. The descriptor represents the coefficients obtained by projection of the binary region onto the 36 ART basis functions. It is characterized by its small size, fast extraction time and matching. Thus this descriptor is suitable for tracking shapes in video data processing applications.

Both region-based and contour-based shape descriptors are intended for shape matching. They do not provide enough information to reconstruct the shape nor to define its position in the image.

The Shape Contour descriptor captures characteristics of a shape based on its contour. It relies on the curvature scale-space (CSS) [1] representation, which tries to captures perceptually meaningful features of the shape. This descriptor represents the high curvature points by their position and value of the curvature.

This representation has two important properties. First, it captures characteristic features of the shape, enabling efficient similarity-based retrieval. Second, it is robust to non-rigid deformation and partial occlusion. Further discussion of this descriptor is presented in Chapter 4.

It is common to represent 3D information as polygonal meshes. The MPEG-7 shape 3D descriptor provides an intrinsic shape description of 3D mesh models. It exploits some local attributes of 3D surfaces. The descriptor represents the 3D mesh shape spectrum, which is the histogram of the shape indexes [65] calculated over the entire mesh. The main MPEG-7 applications targeted here are search, retrieval and browsing of 3D model databases.

Two additional descriptors are used for shape representations: region locator and spatio-temporal locator. They combine shape and localization information and allow partial reconstruction of the shape and its position in the image. The region locator descriptor represents the region with a brief and scalable representation of a box or polygon. The spatio-temporal locator has the same functionality but describes moving regions in a video sequence. The descriptor specifies the shape of a region within one frame together with its temporal trajectory based on motion.

2.2 Major CBIR Systems

Content-based image retrieval systems have several differences to classic information retrieval systems. The major differences are that in CBIR systems images are indexed using features extracted from the content itself and the objective of CBIR systems is to retrieve similar images to the query rather than exact matches. Therefore, retrieval results are not perfect matches of the query image, but rather somehow similar to it. The similarity in most CBIR systems is quantified and the database entries are ranked based on their similarity to the query image. Similar images are retrieved as result of a query. It is possible however that there is not a single entry that satisfies the user's specifications. Moreover, different users may be interested in different parts of the same image. Therefore, similarity-based retrieval is a more flexible framework than exact matching, and offers better performance in queries such as: “Find me images similar to this one”. The rest of this section presents a brief overview of the major content based image retrieval systems. It will cover QBIC, Photobook, Netra, ImageRover, Virage, Webseek and VisualSeek, Mars, Blobworld, Istorama and PicSOM. This list tries to give a brief overview of the major systems in CBIR, but is not an exhaustive list. The next chapter is dedicated to the description of our system called MUVIS.

Query By Image Content (QBIC) [38, 154] is an image retrieval system developed by IBM. It extracts simple features from objects or images: color, texture and shape. Color features computed are: the 3D average color vector of an object or the whole image in RGB, YIQ, Lab, and Munsell color space and a 256-dimensional RGB color histogram. The texture features used in QBIC are modified versions of the coarseness, contrast, and directionality features. The shape features consist of shape area, circularity, eccentricity, major axis orientation and a set of algebraic moment invariants. QBIC also implemented a method of retrieving images based on a rough user sketch. For this purpose, images in the database are represented by a reduced binary map of edge points. QBIC allows combined type searches where text-based keywords and visual features are used in a single query.

MARS [116, 118, 119, 120, 55] developed at UIUC allows combined features queries. Moreover, it allows combinations of global or local image features with textual keywords associated with the images. Color is represented using a 2D histogram over the HS coordinates of the HSV space. Texture is represented by two histograms, one measuring the coarseness and the other one the directionality of the image, and one scalar defining the contrast. In order to extract the color/texture layout, the image is divided into 5×5 subimages. The shape of the boundary of the extracted object is represented by means of Fourier Descriptors (FD). Mars used relevance feedback techniques from the information retrieval (IR) domain in content-based image retrieval, to permit interactive CBIR.

MIT's Photobook [104] proposes a different solution to the problem of image retrieval from many of the other systems described in this review. Photobook centers its attention not on purely discriminant indices but instead on “semantic-preserving image compression". This means that the images are not accompanied by purely discriminatory based index coefficients (e.g. colour histograms) but by compressed representation of the image content. Photobook implements three different approaches to construct image representations for query purposes, each for a specific type of image content: faces, 2D shapes and texture images. The first two representations are similar in the way that they offer a description relative to an average of a few prototypes by using the eigenvectors of a covariance matrix as an orthogonal coordinate system of the image space. For texture description, an image is viewed as a homogeneous 2D discrete random field, which is expressed as the sum of three orthogonal components corresponding to periodicity, directionality and randomness.

Virage is a commercial organization that has produced products for content based image and video retrieval [4]. A basic concept in this framework is that of a primitive, which denotes a feature's type, computation and matching distance. Five abstract data types are defined: global values and histograms, local values and histograms, and graphs. The VIR Image Engine provides a set of general primitives, such as global color, local color, texture and shapes. Apart from these, various domain specific primitives can be created when developing an application. When defining such a primitive, the developer supplies a function for computing the primitive's feature data from the raw image. Its GUI allows queries-by-sketch; it consists of a bitmap editor where the user can sketch a picture with drawing tools and color it using the colors from a palette. Also, the user can bring onto the canvas an image from an existing collection and modify it using the same drawing tools. Queries can be performed on various user-defined combinations of primitives.

WebSeek [136] is a video and image cataloging and retrieval system for the world-wide web developed at Columbia University. It automatically collects online visual material from the web and populates a database using an extendible subject taxonomy. Webseek uses text as well as color features to index the visual material. Color is represented by means of a normalized 166-bin histogram in the HSV color space. Users have the possibility of manually modifying an image color histogram before reiterating the search. Its extension VisualSeek [137] segments images to allow local and spatial queries using the back-projection of binary color sets.

Blobworld [11] is an image retrieval system developed at UC Berkeley. It uses the Expectation Maximization (EM) algorithm to segment the images into regions of uniform color and texture (blobs). The color is described by a histogram of 218 bins of the color coordinates in Lab-space. Texture is represented by mean contrast and anisotropy over the region. Shape is represented by approximate area, eccentricity, and orientation. Query-by-example is performed based on a region from one of the images presented to the user. Blobworld allows the user to view the internal representation of the submitted image and the query results; facilitating the understanding of the retrieval results.

The Istorama [67] system developed by ITI performs queries based on image region obtained using an unsupervised segmentation algorithm. Istorama allows the user to view the segmentation mask of the query image and to select a region and search similar image regions based on color, region size and its spatial location in the image. The user may change the weights associated with each feature to emphasize a specific feature he might be interested in.

ImageRover [129], a system developed in Boston University, combines textual and visual statistics in a single index for content-based search of a web image database. Textual statistics are captured using latent semantic indexing (LSI) based on text in the containing HTML document. Visual statistics are captured using color and texture orientation histograms. To initiate a search of the ImageRover index, the user specifies few keywords describing the desired images. Later the user can refine his query through relevance feedback. Both visual and textual cues are used in the relevance feedback loop. Based on relevance feedback from the user, the system selects the appropriate normalized Minkowski metric each time a query is made.

WebSeer [141] from the University of Chicago tries to classify an image as photograph or graphic. A number of color tests are applied to the images in order to separate photographs from drawings. Images determined to be photographs are subjected to a face detector based on a neural network. Keywords are extracted from the image file name, captions, hyperlinks, alternate text and HTML titles.

Netra [81, 82] is a system based on regions of homogeneous colors developed at the University of California Santa Barbara. It uses color, texture, shape and spatial location information for image indexing and retrieval. Images are segmented off-line using an edge flow segmentation technique, and each segment is characterized by its local features.

PicSOM [68, 69, 72, 71] is an image browsing system based on the Self-Organizing Map (SOM), developed at the Laboratory of Computer and Information Science at Helsinki University of Technology, Finland. The SOM is used to organize images into map units in a two-dimensional grid so that similar images are located near each other. PicSOM uses the average rgb color feature, texture feature, Fourier-based shape features and MPEG-7 features. PicSOM applies a tree-structured version of the SOM algorithm (Tree Structured Self-Organizing Map, (TS-SOM)) to create a hierarchical representation of the image database. During the queries, the TS-SOMs are used to retrieve images similar to a given set of reference images. As a basis for retrieving images, the PicSOM system uses a combination of several types of statistical features, which are computed from the image content. Separate feature vectors have been formed for describing colors, textures, and shapes found in the images. A distinct TS-SOM is constructed for each feature vector set and these maps are used in parallel to select the retrieved images. The image queries are performed through the Web interface and the queries are iteratively refined as the system exposes more images to the user. Image retrieval with PicSOM is an iterative process utilizing the relevance feedback approach. The image query begins with an initial set of images uniformly selected from the database. On subsequent rounds, the query focuses more on the user's need. This is achieved as the system learns the user's preferences from the selections made during previous rounds. It is worth noting that the Euclidean distance, inherently used in the PicSOM system, may be inappropriate for certain types of features.

2.3 Similarity Measures

In CBIR systems, image features are in general organized into n-dimensional feature vectors. Thus the query image and the database images can be compared by evaluating the distance between their corresponding feature vectors. Both metric and non-metric measures have been used in CBIR systems. Statistical distances such as the Mahalanobis distance have also been used [130].

Often different image features are indexed separately, thus similarity scores can be computed independently for each feature. Then the overall similarity score is obtained as a linear combination of these scores [4, 138]. The weights of this linear combination may be specified by the user or automatically adjusted by the system based on the feedback of the user [17].

Specific distances have been defined for specific features: e.g. for histograms commonly used measures are the histogram difference, histogram intersection [63, 140] or the quadratic distance [63, 134, 135]. The latter tries to account for the perceptual difference between any pair of bins in the histogram. The Hausdorff distance has been used to compare histograms as well as shapes in [123].

Many similarity measures are based on the L_p distance between two points in the n-dimensional feature space. For two points x, y the L_p distance is defined as $L_p(x, y) = (\sum_{i=0}^{n} |x_i - y_i|^p)^{1/p}$, called Minkowski distance. For $p = 2$ we get the Euclidean distance and for $p = 1$ we get the Manhattan, city block, or taxicab distance.

The L_1 and L_2 norms are analyzed in [139] and [138], and their performances are compared.

The retrieval performance of a system depends on the agreement between the similarity measure used and human judgments of similarity, since the end consumer of CBIR results is a human. Therefore, several measures in accordance with the human perception have been developed [128, 132] and in [18, 19, 20, 22, 23, 109, 147]. A good review of similarity measures for shape matching is presented in [152].

Amos Tversky (1977) proposed a feature-based "contrast model" of similarity [149], in which common features tend to increase the perceived similarity of two concepts, and where feature differences tend to diminish perceived similarity. For instance, Tomato and Cherry are similar by virtue of their common features Round, Fruit, Red and Succulent. Likewise, they are dissimilar by virtue of their differences, namely Size (Large versus Small) and Seed (Stone versus NoStone). This is an unsurprising and intuitive claim; however, Tversky's model claims that feature commonalities tend to increase perceived similarity more than feature differences can diminish it. That is, when assessing similarity we give more credence to those features that concepts have in common than to those that distinguish them.

Tversky's contrast model is a non-metric set-theoretic account of perceived similarity that aims to address some of the shortcoming of the distance models. Tversky's model is based on evaluating sets of matching and mismatching features:

$$Sim(x, y) = f(X \cap Y) - \alpha f(X - Y) - \beta f(Y - X) \tag{2.1}$$

where $Sim(x, y)$ reads the similarity of x to y, X is the set of features that represents x, Y is the set of features that represents y, $X \cap Y$ is the set of features common to x and y, $X - Y$ is the set of features uniquely possessed by x and $Y - X$ is the set of features uniquely possessed by y, α and β are free parameters and f is a function over sets of features related to the saliency of the features.

Chapter 3

MUVIS: A CBIR System

The **MUVIS** system has been developed in the context of the TEKES [1] MuVi [2] project. This system has been developed to provide a platform for testing CBIR algorithms developed by the Image and Video Analysis (IVA) group at Tampere University of Technology. However, MUVIS allows indexing of objects and images based on low-level features only, such as color [46], texture [101] and shape features [114, 146]. MUVIS graphical user interface (GUI) supports more complex queries by additionally allowing to use key words and objects layout inside the images with combinations of low-level features. MUVIS extensions to video sequences and to mobile devices are already under development and initial results have been published in [40, 56, 57].

MUVIS addresses several of the problems, listed in Section 1.1, faced when building such CBIR systems. The main contributions to this field, other than building the system, are in image content abstraction using automatically extracted features, development of similarity measures that mimic human perception of similarity between images and objects and use of relevance feedback to improve system performance.

The block diagram of MUVIS is presented and the functionalities of each block are reviewed, in Section 3.1. In Section 3.2 the different types of queries allowed by MUVIS are presented. An overview of the low-level features used in MUVIS is given in Section 3.3. Section 3.4 gives a brief idea about the extensions of MUVIS to mobile devices [56, 57] and to retrieve more complex multimedia data types [40].

[1] TEKES is the National Technology Agency of Finland

[2] Multimedia Data Processing Applied to Image and Video Databases Manipulation

3.1 MUVIS System

MUVIS is built as a modular, heterogeneous software system incorporating modules developed in different programming environments: **Java**, **ANSI C**. This is motivated mainly by the need of portability of the system to the variety of platforms used within the group. This combination of **Java**, **ANSI C** allowed us to harness the benefits of both languages and made possible and easy the extensions of the system to a distributed client-server system [21] and to mobile devices [56, 57]. MUVIS is designed in a client-server architecture, where the GUI resides on the client side and is implemented in Java, while the core of the retrieval system resides on the server side and is implemented in C. This type of architecture maybe inefficient when both sides are running on the same machine and direct communication by function calls result in some inefficient data manipulation procedures (resources, such as files, are loaded and discarded several times during the manipulation of the same database). This choice is well justified considering the end goal of the MuVi project of having a distributed retrieval system.

In a practical application of CBIR systems, it is reasonable to assume that graphical user interface, database management system (DBMS) and the image processing engine are running on different machines. For example: GUI runs on a remote (e.g. home personal computer) with a medium network connection while the DBMS and the image processing engine are running on powerful servers with a high bandwidth network connection between them.

Considering the above mentioned facts, a client-server architecture system seems to be the approach that will offer flexibility and efficiency in implementing a robust, portable and extendable CBIR system. Therefore, the MUVIS system is built in the following way:

- The Image Processing Library in C,
- the GUI interface in Java,
- the DBMS in C integrated with the Image Processing Library,
- the communication mechanism between the Java-GUI and the C-Library is done either via the Java Native Interface (JNI), when both sides reside on the same machine or using a client server communication sessions otherwise, see Figure 3.11.

3.1.1 Block Diagram of the System

As mentioned in the beginning of this chapter, MUVIS has been implemented as a platform for testing and developing new CBIR algorithms within the IVA group. The system is made of several building blocks, see Figure 3.1, each has specific functionalities. MUVIS modularity allows the design and testing of each module separately. This in turn, allows fast progress of the research work on each module independently of the other modules. The rest of this section describes briefly the functionality of each module.

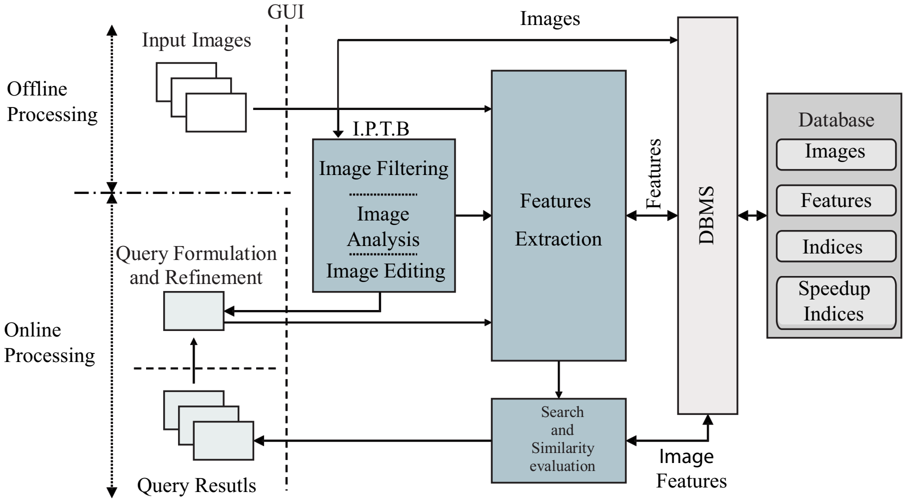


Figure 3.1: Block diagram of MUVIS

3.1.2 Image Processing Toolbox

The image processing tool-box module is an important module of the MUVIS system, and is composed of a collection of image analysis, filtering and editing functions. The image processing functions are used during all operations of database population, feature extraction of each of the images in the database, image coding and decoding.

The image analysis contains functions for image segmentation, edge detection, object contour extraction, etc. The results of such operations are in general very sensitive to noise present in the processed images. Therefore, image filtering capabilities are included in the toolbox, these are linear and nonlinear filters that we designed or are proposed in the literature, such as smoothing, edge and contrast enhancing, interpolation, etc.

The image editing functions allow image manipulations such as image composition using simple primitives such as lines, simple shapes as well as images or regions from images. Simple editing functions such as editing, copying, cutting, pasting as well as histogram editing functions, are very important for the query formulation, processing and refinement tasks.

Some of these functionalities are used on the client side (for query formulation and refinement) as well as on the server side (during the database population, and retrieval process). Therefore, these functionalities are implemented on both sides with Java and C. For example, the histogram extraction functionalities are used on the:

- client side to allow histogram manipulation and histogram based queries,
- server side to allow feature extraction from all images during database population.

Having this redundancy in implementation can be very useful in reducing the unnecessary client-server communication and data transfers.

3.1.3 Feature Extraction

The feature extraction block is at the heart of MUVIS, since it represents the focus of interest of our group in the query by content research field. The main goal of the group is to develop new algorithms for automatic image feature extraction, since most of the group members have signal processing backgrounds. Therefore, the development of MUVIS is a means to an end and not the main goal of the research project.

MUVIS is developed with the idea of indexing images and objects (image segments) based on their content. Therefore, features at both levels, image level and object level, are extracted from the given images and used for annotation, further discussed in Section 3.3 and Chapters 4– 5. The features extracted from the whole image are called global features, and they include:

- color attributes: color histogram (e.g. 166 bins in HSV color space), dominant color,
- spatial attributes: segmentation mask, number of regions, center of gravity of each region,
- texture attributes: Gabor features, Moment features, etc.

The features characterizing an object are called object features, and may include:

- color attributes: color histogram (e.g. 8 bins in HSV color space), dominant color,
- texture attributes: coarseness, directionality, moments, GLCM,
- shape attributes: contour, aspect ratio, main axes, CSS, WTMM features, ordinal correlation features,
- spatial attributes: coordinates of the center of mass.

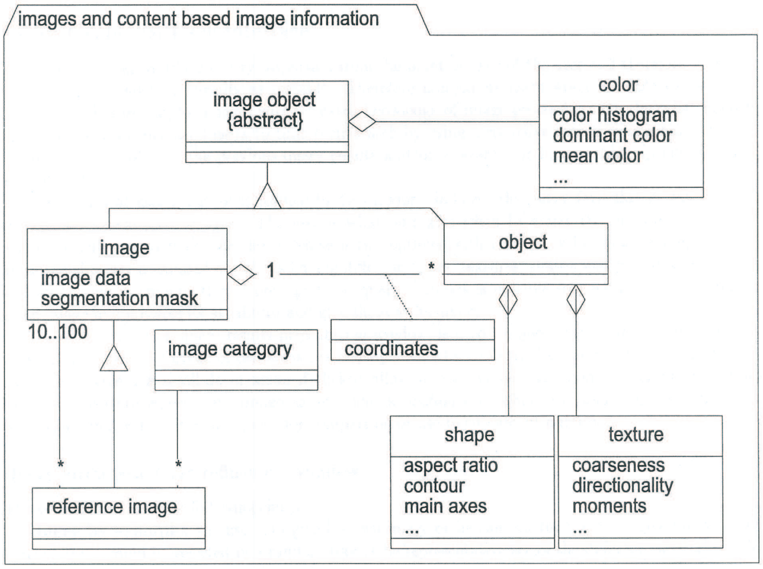


Figure 3.2: Conceptual model of the content-based features of images

Figure 3.2 shows the conceptual model of the content-based features of images. In this figure the image color feature is represented by: color histogram, dominant color and mean color, where the object features are shape and texture.

3.1.4 Internal Database

Indices

The database contains the indices, which identify each one of the images in the image database by a unique index and the set of all corresponding features.

This index database is arranged in a way so that the access time to a given image is minimized. For this purpose some techniques such as the R-trees , SS-trees and the SR-trees [44], which are good index structures for searching in high-dimensional feature spaces, have been investigated and an indexing structure based on the pyramid [7] indexing structure was implemented. The main idea behind the pyramid technique is to do a mapping of the k-dimensional points into one-dimensional points, and index them by a B^+-tree.

The mapping of the k-dimensional point into a one-dimensional point is a crucial step in the pyramid-technique. It is based on partitioning of the (k-dimensional) data space into $2k$ (k-dimensional) pyramids, having their center point as the center point of the data space. Each pyramid is further divided into several slices, parallel to its base as shown in Figure 3.3.

To each slice corresponds one data page of the B^+-tree. Therefore, each point q in the k-dimensional data space, is located in one of the $2k$ pyramids, say i,

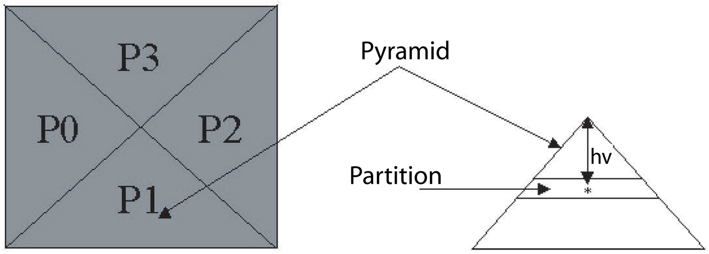


Figure 3.3: Partitioning the data space into pyramids (for 2D data space).

and has a height hv within that pyramid i. Hence, it can be indexed by the single value, $pv = i + hv$. In order to build the B^+-tree, pv can be used as the key and the k-dimensional points plus the pv values are stored in the leaf nodes of the B^+-tree, so that no inverse transformation is needed. The update and delete operations can be done in a similar way to the insert operation. However, query processing is a more complex operation than it is for other indexing structures such as the R-trees, the SS-trees or SR-tree [44].

Speedup Indices

To insure immediate response to a query by example image from the database, the system associates to each image in the database a record containing the indices of the top L most similar images. These L indices are called speedup indices, since there is no actual search done when the query submittedand the response time is reduced to the display time of the system. L is a fixed number and depends on the database size and number of similar images expected to be present.

The similarity computation may be done on a feature by feature basis for each image or on a default combination of features. For the case of single feature query for each feature, the top, say, 100 most similar images are found and their indices in the image database are stored. The speedup indices are very useful in retrieval by browsing, where the user does not change the parameters of the query but simply navigates through the database entries by just clicking on the image of interest in an iterative process until he is satisfied with the retrieval results displayed.

Images

This is the actual collection of images stored in the database. In general, this is a collection of images with different dimensions, different file formats, which are typically be in a compressed format.

3.1.5 Graphical User Interface

The database population is a time consuming task performed off-line, and launched from the command line console directly running on the server. Therefore, when we talk about the user interface in the following we mean the graphical user interface used in the online processing. The graphical user interface plays a crucial role in the browsing and retrieval processes as through it these processes are initiated and their results are presented.

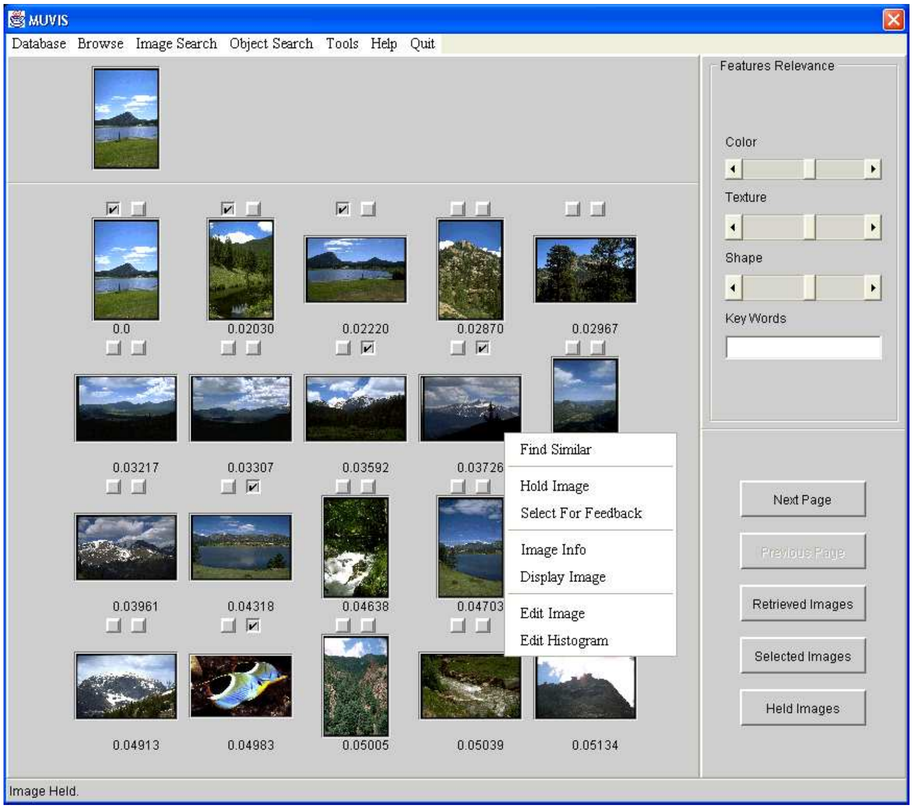
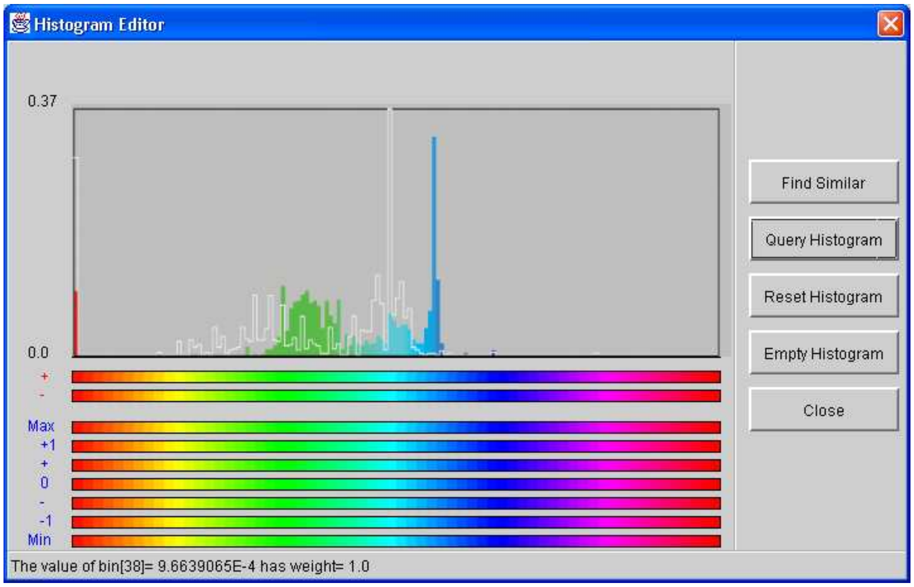


Image Held.

Figure 3.4: Main window of the graphical user interface of MUVIS.

The GUI of such systems should be as intuitive and user friendly as possible allowing non-expert users to browse and search images without extra knowledge of image processing or the mechanisms of features extraction or their similarity



The value of bin[38]= 9.6639065E-4 has weight= 1.0

Figure 3.5: Histogram editing tool.

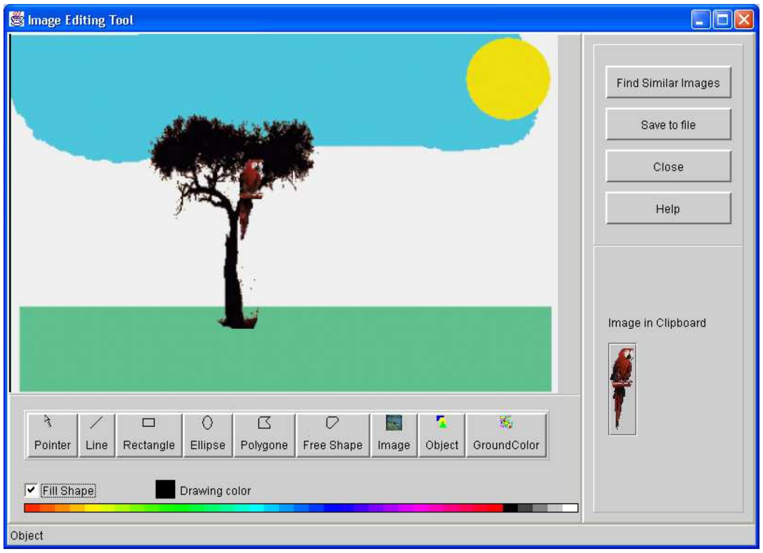
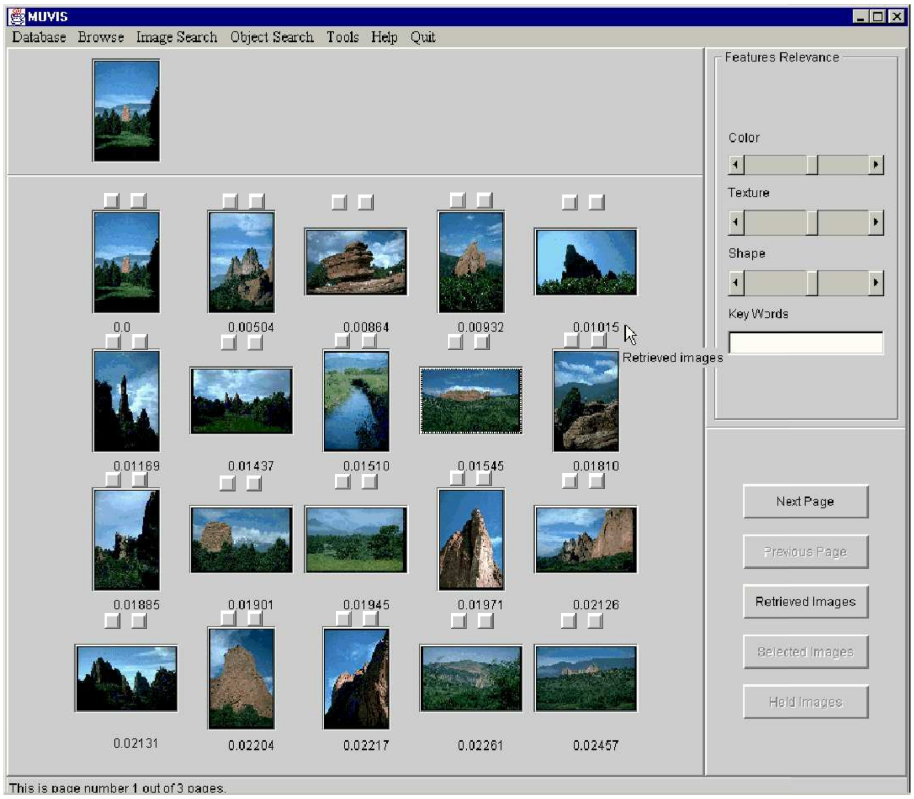


Figure 3.6: Image editing tool.

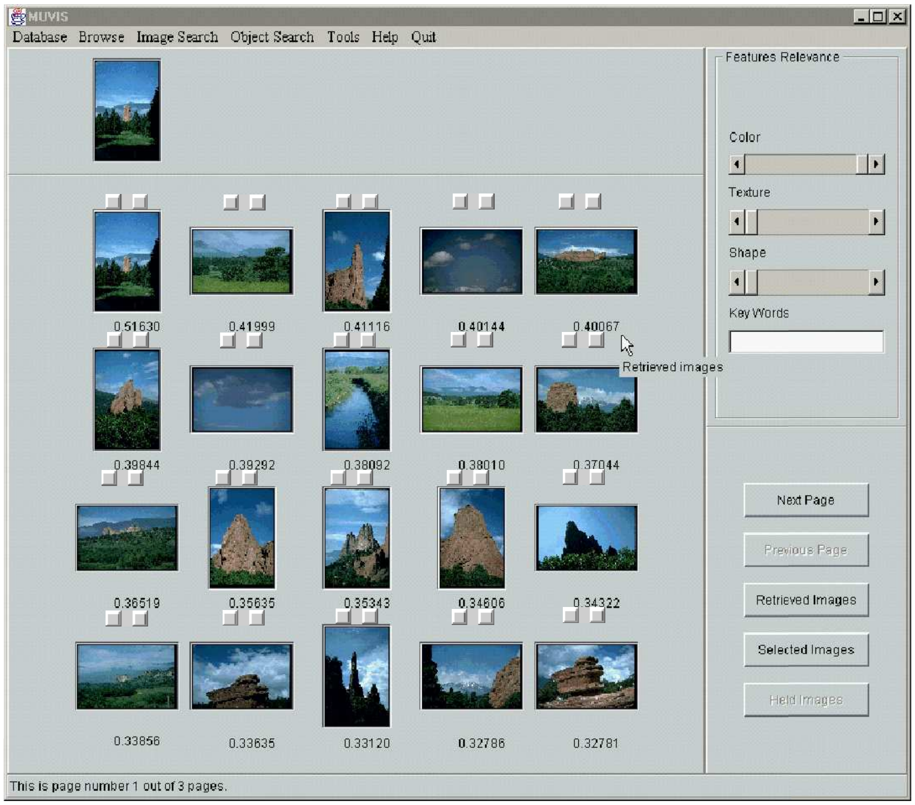


This is page number 1 out of 3 pages.

Figure 3.7: Retrieval results based on color using histogram difference.

measures used. While, on the other hand, more knowledgable users should be allowed to tune their search using advanced parameters settings, relevance feedback mechanisms for query refinement, or using a specific combination of features or algorithms. Since the consumer of the retrieval results is a human user, the results have to be presented in a simple yet meaningful way.

The MUVIS graphical user interface main window is composed of four panels, a status bar and a menu bar, see Figure 3.4. The central panel is the main one, where the retrieval results are presented. In it 20 images at most can be presented in a 4×5 matrix of thumbnail images. Above each of the retrieval result images presented in this panel two check boxes are used to select the image for feedback or to hold it in a repository of interesting results to the user. The right side of the window contains two panels aligned vertically. The top panel "Feature Relevance", contains three slide bars to adjust the weights associated with low-level features: color, texture and shape; and contains a text field where keywords can be entered. These weights and keywords maybe used to formulate multiple feature queries combined with textual annotations. The lower panel contains buttons used to navigate through the retrieved results (50 images are retrieved by default



This is page number 1 out of 3 pages.

Figure 3.8: Retrieval results based on color using histogram intersection.

displayed in three pages), and to switch between the retrieved results, the images selected for feedback and the images of interest to the user stored in the holding list. The fourth pane is positioned directly under the menu bar and on top of the main panel, and displays the thumbnail image representing the query. Under each retrieved image its similarity score is displayed.

The main window of the GUI contains a status bar where additional information is presented, and a menu bar at its top, from where the browsing and search actions are initiated. The menu entries are “Database”, “Browse”, “Image Search”, “Object Search”, “Tools”, “Help”, and “Quit”. From each of these entries a drop down menu with different entries are presented allowing a set of actions:

- Database: allows the selection of the database to be used in subsequent browsing or searching actions,

- Browse: initiates the browsing from a random set of images or images from each category in the database selected,

- Image search: allows to initiate queries using an example image from the

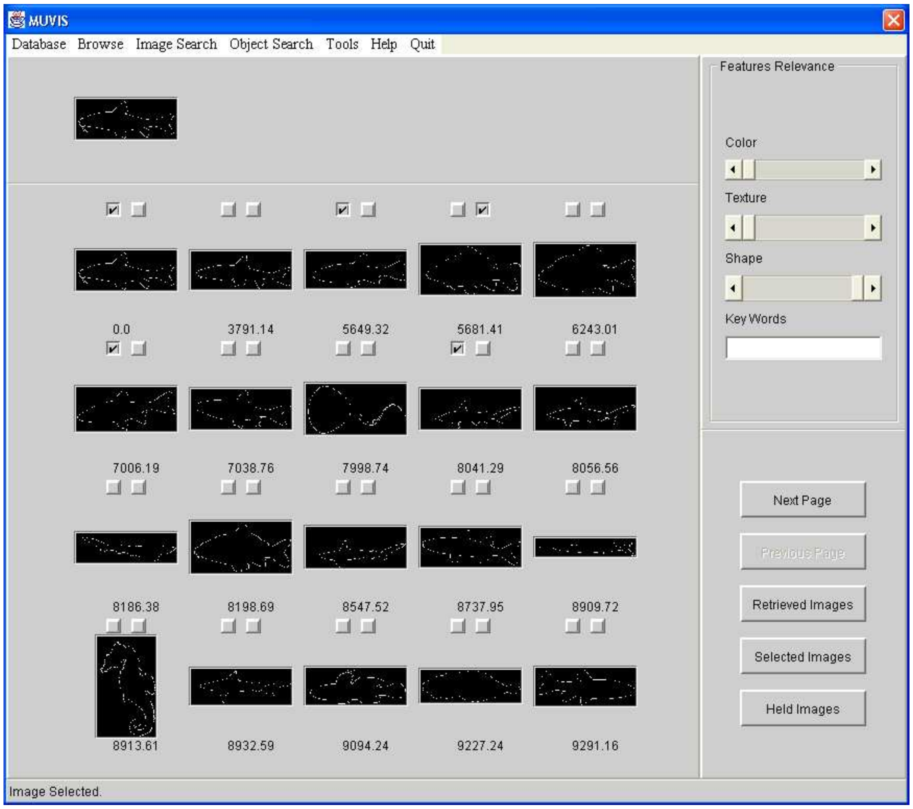


Figure 3.9: Retrieval results based on shape.

database or provided by the user, by a sketch or a histogram,

- Object Search: permits object-based queries with drawn sketches or provided in a file,

- Tools: gives access to a set of filters, image editing tool, histogram editing tool and a category tool,

- Help: help topics,

- Quit: allows to exit the current query or the system.

A single left mouse button click on any of the thumbnail images initiates a query with it using the default parameters. While a click with the right button of the mouse presents a drop-down menu with the following entries: “Find Similar”, “Image Info”, “Display Image”, “Edit Image”, “Edit Histogram”, “Hold Image”, and “Select for Feedback”, see Figure 3.4. These menu entries allow the following actions: query with the image, display the image file info, display the image in full

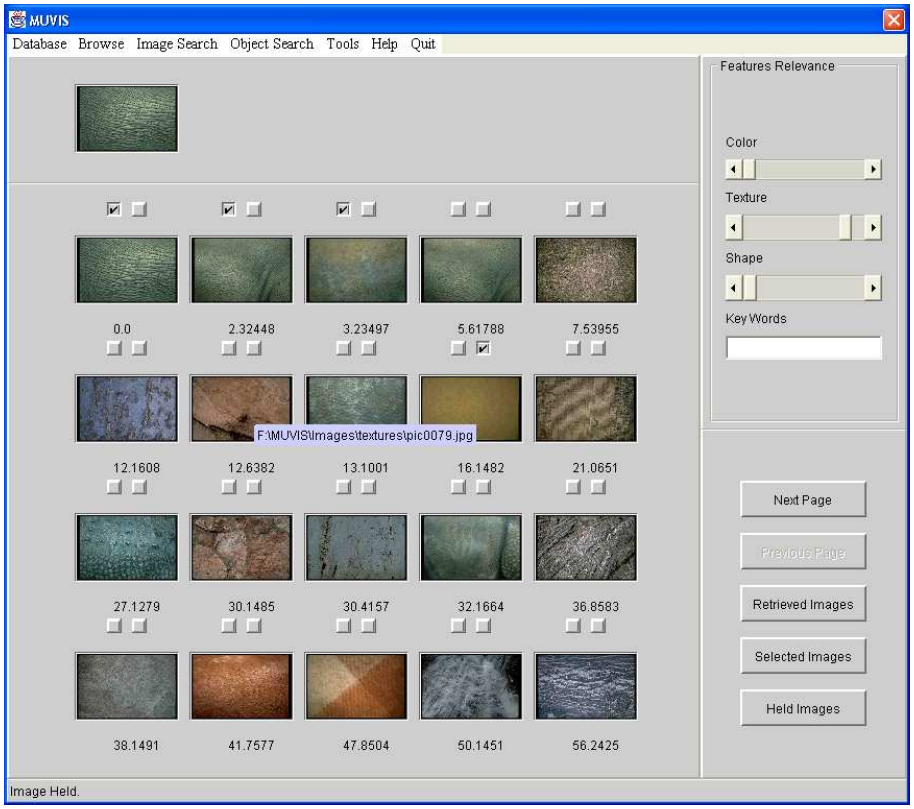


Image Held.

Figure 3.10: Retrieval results based on texture.

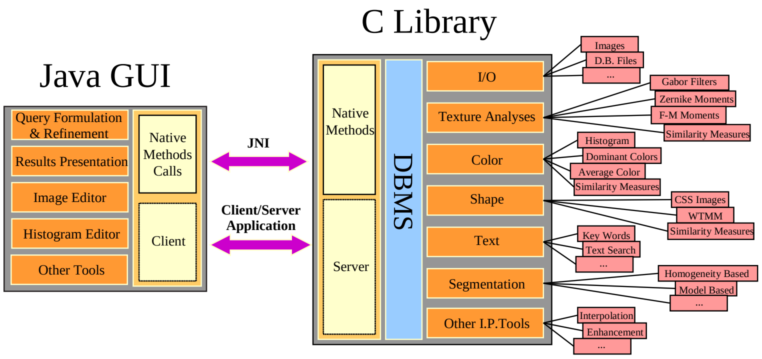


Figure 3.11: Software structure of MUVIS.

size, open the image in the image editing tool, extract its histogram and display it in the histogram editing tool, hold it in the repository for interesting images or set it as a feedback entry, respectively.

3.2 Query Formulation

Typically, a browsing or retrieval session starts by the selection of the database from the “Database” entry in the menu bar. In our work we have been using three different databases, one for each feature: color, texture and shape. Next, the user can either start to browse the database images or to initiate a search for an image or an object. Browsing starts from a set of 20 images randomly selected from the database images or from a set of images representing the categories in the database. Several types of queries can be formulated in MUVIS, which can be described as query by example image, query by sketch, or query by histogram.

Query by example: The user can query by an external image or an image from the database. The difference between these two options is that when an external image is used its features need to be extracted while if the image is already in the database, its features are already extracted and stored in the database along the image. Therefore, when a query submitted using an image from the database, only the index of the image in the database is transferred to the server. In both cases the features used are imposed by the database selected at the start.

Query by a sketch: Here the user is prompted with the image editing tool; which consists of a vector graphic editor where the user can draw a sketch composed of primitive colored shapes, objects and images. In a similar way the user can bring onto the canvas an image and modify it using the same drawing tools, see Figure 3.6. An example of sketch-based query is shown in Figure 3.6, where the simple shapes such as rectangles and circles are combined with more complex objects (tree and bird) from other images to create the query sketch.

Query by sketch allows to search for objects or regions in the images. The image editor is used to select an object or a region from an existing image and query with it. In this case any of the region features can be used to estimate the similarity between the object and objects in the database. This is only supported now in the queries based on the color feature or shape. Histogram intersection is used to estimate the similarity between the object and the images based on color, while all the shape features can be used to estimate the similarity between the boundary of the object to the boundaries of objects in the shape database. No segmentation algorithms are used here, since the similarity estimation is based on global features when the query is a composed of multiple objects, or the object features when the query is made of a single object outlined by the user.

Query by Histogram: Here the user is given the choice of using either an image histogram as is, to modify it or build one from scratch and query with it. All these options are done through the histogram editor, shown in Figure 3.5. The histogram editing tool allows the association of weights to the bins in order to emphasize, reject or ignore a set of colors.

Query by Text: The text is used as keywords and captions which could be attached to the whole image or to an image segment. This feature has not been used until now, even though some 4000 color images have been annotated using a fixed set of keywords. The keywords were associated with the image manually and not to regions of the image.

Even though the GUI allows for the combination of features using weights assigned to each feature as can be seen in Figure 3.4, combined feature queries have not been used in MUVIS but are being considered in the extensions of the system.

Theoretically a query can contain: text, colors, textures, shapes and spatial layout of objects:

$$Query = QT_1(op_1) * QT_2(op_2) * ..., \tag{3.1}$$

where QT_n =query term n, op_n =operator n. A QT contains a simple query about histogram, texture or shape. The $*$ operator can be any logical operator AND, OR, NOT .

3.3 Low-Level Features

Several features have been used in MUVIS, these features have been representation of the color, texture or shape information in the images. In the following these features are briefly reviewed.

3.3.1 Texture Features

Texture is one of the basic characteristics for the analysis of many types of images. It provides important information for segmentation of images in distinct objects or regions as well as for classification or recognition of surface materials. But due to the large diversity of natural and artificial textures, a universal definition of texture is still unavailable. A large range of descriptors have been used for texture analysis, such as: Fourier power spectrum, spatial texture energy, autocorrelation function, structural elements, Markov random field model, fractal dimension, spatial gray-level co-occurrence probability and multichanel representation [86, 107].

The information content of a texture obtained from the gray-level intensity of pixels is sometimes unsatisfactory as it strongly depends on the lighting conditions. More important are the local variations of the image intensity. Also the scale is an important aspect to take into account because it should be related with the size of a textural element which cannot be known “a priori”. Experiments proved that the human visual cortex has separate cells that respond to different frequencies and orientations, which is equivalent to a multiscale analysis system.

All these are arguments for multiresolution characterization of the textural images by means of localized filter-bank [87]. Each of the involved filters will emphasize a certain local property of the analyzed input. This emphasises the efficiency of techniques such as the Gabor filters, when the end user of the results is a human.

To support image retrieval applications such as aerial and satellite imagery analysis, MPEG-7 adopted three texture descriptors: homogeneous texture descriptor, texture browsing descriptor and edge histogram descriptor [87]. As MUVIS has been buit as a platform for testing the techniques developed in our research group, it did not adopt the MPEG-7 texture features. Several texture features have been used in MUVIS to characterize images and objects, e.g. Gabor Filters, Co-occurrence matrices [101] and moments [108]. A brief description of each of these techniques is presented below.

Gabor Filters Analysis

Gabor filter-banks were shown [97] to model very well the visual cortex and they are optimally localized in the sense that they reach the lower bound for simultaneous localization in the spatial and frequency domain according to Heisenberg's uncertainty principle. The Gabor filters are a special case of windowed Fourier filters. They are also analogous to wavelets because the whole family is generated by rotations and dilations of a single Gabor filter. The "fingerprints" associated to textures by this method can be of further use in segmentation problems especially if highly specific frequencies or orientations are involved.

Gabor functions are Gaussians modulated by complex sinusoids (with center frequencies U and V along x and y axes respectively). A 2D Gabor function can be written as follows:

$$h(x,y) = g(x',y')exp(2\pi j(Ux+Vy)), \tag{3.2}$$

where $(x',y') = ((xcos\Phi + (-y)sin\Phi), (xsin\Phi + ycos\Phi)$ are the rotated coordinates, and where

$$g(x,y) = (\frac{1}{2\pi\lambda\sigma^2})exp(-\frac{(x/\lambda)^2+y^2}{2\sigma^2}). \tag{3.3}$$

The parameters involved here are the following: λ is the aspect ratio of the Gaussian (the ratio between the axes of ellipses that are the level curves of $g(x,y)$), σ its width in the spatial domain also called scale parameter and Φ the orientation angle of its major axis with respect to the horizontal. The Fourier transform of the $h(x,y)$ function has the following expression:

$$H(u,v) = exp(-2\pi^2\sigma^2[(u'-U')^2\lambda^2 + (v'-V')^2]), \tag{3.4}$$

where $(u',v') = ((ucos\Phi + (-v)sin\Phi), (usin\Phi + vcos\Phi))$. This means that the frequency response has the shape of a Gaussian centered about the frequency (U,V) and with extensions inversely proportional to σ.

Filters designed following these functions are highly selective in both position and frequency. This happens because the 2D Gabor functions are mapping $R^2 \longrightarrow C$ simultaneously achieving the lower bounds of the uncertainty inequalities $\Delta x \Delta u \geq 1/(4\pi)$ and $\Delta y \Delta v \geq 1/(4\pi)$. Attaining the equality, maximal possible resolutions in the 2D visual and 2D Fourier domains are ensured.

The output of filtering is obtained by the convolution of the input image with the filter response: $h(x, y) * i(x, y)$. Computing the magnitude $|h(x, y) * i(x, y)|$ for each of the filters in the bank gives a features space where each point in the initial image is described by as many coordinates as the number of filters. The feature vectors may be useful in classification or segmentation problems also. The results of using the Gabor filters features to retrieve rock images has been investigated in [101, 102] and their retrieval performance compared to those of Gray Level Co-occurrence Matrix (GLCM) features. Recent work in [80] shows how texture and color features affect natural image retrieval.

Moments Features

In this method the texture features are obtained directly from the gray-level image by computing the moments of the image in local regions. The intensity image is regarded as a function of two parameters $f(x, y)$. The $(p + q)$th order moment of $f(x, y)$ with respect to the origin $(0, 0)$ are defined as:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) x^p y^q dx dy, \tag{3.5}$$

The lower order moments have well defined geometric interpretations. For example, m_{00} is the area of the region, $\frac{m_{10}}{m_{00}}$ and $\frac{m_{01}}{m_{00}}$ give the x and y coordinates of the center of the region, respectively. The moments m_{20}, m_{11} and m_{02} can be used to estimate the elongation of the region, and the orientation of its major axis.

In MUVIS [108] a fixed number of the lower-order moments for each pixel in the image is used. The moments are computed within small local windows around each pixel. Given a window size, the coordinates are normalized to the range $[-1, 1]$, the pixel being at the origin. The moments are then computed with respect to this normalized coordinate system.

The set of values for each moment over the entire image can be regarded as a new feature image. The moments alone are not sufficient to obtain good texture features in certain images. Hence a hyperbolic tangent function is used as a nonlinear transformation function to obtain texture feature images corresponding to the moments images M_k with mean $\overline{M}$ as in [108]. The following is the transformation function:

$$F_k(i, j) = \frac{1}{N} \sum_{(a,b) \in W(i,j)} |tanh(\alpha(M_k(a, b) - \overline{M}))| \tag{3.6}$$

where N is the number of pixels in the processing window $W_{i,j}$, and (i, j) is the center of this window and α controls the shape of the function.

Gray Level Co-Occurrence Matrix

Gray-level co-occurrence matrix (GLCM) [49] is the two dimensional matrix of joint probabilities $P_{d,r}(i, j)$ between pairs of pixels, separated by a distance, d, in a given direction, r. It is popular in texture description and is based on the repeated occurrence of some gray level configuration in the texture; this configuration varies rapidly with distance in fine textures and slowly in coarse textures. Haralick [49] defined 14 statistical features from gray-level co-occurrence matrix for texture classification. Only a few of these features have been adopted in MUVIS [101, 102]:

$$\text{Energy} \quad \sum_i \sum_j P^2_{d,r}(i,j) \tag{3.7}$$

$$\text{Entropy} \quad -\sum_i \sum_j P_{d,r}(i,j) log P_{d,r}(i,j) \tag{3.8}$$

$$\text{Contrast} \quad \sum_i \sum_j (i-j)^2 P^2_{d,r}(i,j) \tag{3.9}$$

$$\text{Inverse Difference Moment} \quad \sum_i \sum_j \frac{P_{d,r}(i,j)}{|i-j|^2},\ i \neq j. \tag{3.10}$$

3.3.2 Color Features

Color is a very important visual attribute for human vision as well as computer vision systems. Therefore, it has been one of the most widely used features in CBIR systems, because simple color features are easily extracted from the images and objects. Moreover, color is invariant to orientation and scale and analysis of retrieval results based on color are intuitive. Furthermore, the color features can be computed from arbitrarily shaped regions.

The ability of the color features to characterize perceptual similarity colors is greatly influenced by the selection of the color space and color quantization scheme. The RGB color space is used in CBIR, even though it is perceptually not uniform. Other perceptually more uniform color spaces are HSV and L*a*b* [59], are obtained from RGB by using a nonlinear transform. Color quantization is essential in the extraction of any color feature due to the large numbers of colors that can be present in a single image. The color space is quantized to reduce the number of distinct colors in an image, and simplify its feature vector.

The most widely used color feature is the color histogram [140], which gives an idea about the colors present in an image and their percentages.

Image retrieval systems have implemented one or more color-based features, e.g. dominant colors [137], color moments [138], color sets [134], color coherence vector [103], color correlogram and autocorrelogram [53].

Eight color descriptors [87] were defined as part of the MPEG-7 standard: Color space, Dominant Colors, Color Quantization, GoF/GoP Color, Color Layout, Color-Structure and Scalable Color Histogram.

The dominant color as well as histograms in different color spaces have been investigated in MUVIS [63]. Both histogram intersection and histogram difference were used as similarity measures, see Figure 3.8 and Figure 3.7. Usually texture and color are closely related to the objects in natural images. Therefore queries combining both texture and color features have been recently been investigated in MUVIS [80] and were shown to provide better retrieval results than single feature queries.

3.3.3 Shape Features

Shape representations can be classified into contour-based and region-based techniques. Both categories of techniques have been investigated and several approaches have been developed. First we have implemented contour-based techniques: curvature scale space and polygonal approximation based on high curvature points detected based on wavelet transform modulus maxima (WTMM). Later we developed the ordinal-framework for region-based shape similarity estimation and used it to compare objects boundaries as well as segmentation masks. An overview of shape characterization techniques as well as features used in MPEG-7 and in MUVIS are described in Chapters 4 and 5.

3.4 MUVIS Extensions

Two extensions of MUVIS have been under investigation since 1999. The first extension is to index and retrieve audiovisual data; while the second is to allow queries initiated from mobile devices. These two extensions are briefly mentioned in this section, further details can be found in the mentioned references.

3.4.1 MUVIS Extensions to Image Sequences

Figure 3.12 shows the block diagram of the extended MuVi system, which is able to retrieve image sequences in addition to still images. The new blocks needed for image are marked with bold borders. The most important part of the extension is the temporal segmentation block, which detects the beginning and the end of each shot. Furthermore, several attributes of each shot are extracted. After the extraction of key frames for each shot, they are passed to the feature extraction

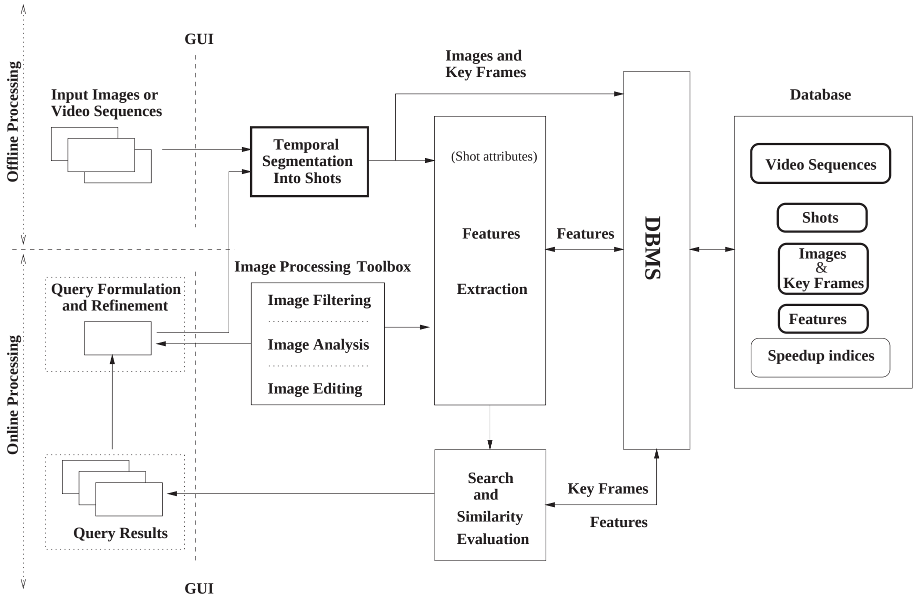


Figure 3.12: Extended block diagram of the system.

block. The feature extraction block is the same as for still images possibly with some modifications taking into account the temporal information.

When formulating the query, the user can retrieve image sequences by a sample image, a sketch, or a sample image sequence. The two first methods imply a query which is similar to the retrieval of still images. Where key frames are manipulated in similar way to still images in CBIR. The latter case starts with a sample sequence, which is segmented into shots, and whose key frames are extracted. The key frames can then be compared against those in the database.

The work on extending MUVIS to handle different types of media is on its way and initial results have been published in [40, 46].

3.4.2 MUVIS Extensions to Mobile Devices

The way people are communicating is changing very fast. A few years ago, mobile phones were lucrative items restricted to a very small community of businessman and government agents. Moreover, they were used exclusively for voice calls. Today the mobile terminal penetration is growing steadily and continuously. And their use is no longer restricted to voice communication only. In Finland, it is widely accepted among youngsters to use a mobile phone for exchanging short messages, to chat with friends or to play games. Adults may be more interested in checking their stocks or paying a bill using their wireless terminal. Third generation phones will create new opportunities for content providers, by providing a way of transmitting text, voice, images, and streamed video. Moreover, their ability to always be connected to the Internet will provide users with new ways of interaction with the available information. Users will then face the problem of how to retrieve the information of interest to them in an efficient manner. The goal is to allow for searching and navigation in this wealth of data without the need to make text-based queries for three obvious reasons:

- The user may be unable to type in commands,
- keyboards of portable devices are not very comfortable for text-based commands,
- text-based queries may not be very appropriate in the case of multimedia data.

Therefore, a content-based indexing and retrieval engine coupled to a speech recognition engine could be the ultimate interface to such a system. We investigated the possibilities of image retrieval over mobile devices in [56, 57].

Although the newly introduced pervasive devices have faster processors, larger memories and wider communication bandwidth, their capabilities remain far behind those of personal computers. Therefore, designing CBIR systems for such devices requires the understanding of their characteristics, their hardware and software limitations. Furthermore, one has to understand the ways users can interact with such devices and their implications on applications design. In [56] we investigated the extension of MUVIS to the NOKIA 9210 communicator based on Personal Java application and a client server architecture, see Figure 3.13. In [57] another extension of MUVIS has been proposed and tested on the Mobile Information Device Profile (MIDP), using a Java-based client server paradigm. The Demo shows that such an implementation is feasible. However, due to the limiting factors in both the hardware and software of the wireless terminal as well as the communication channel, limited results have been obtained. Image retrieval is the bottleneck which is more severe with the size of the images. The good news is that with the advances of third generation networks, offering higher data rates and more processing power in wireless devices and more memory, such an application will be common. Other mobile devices such as Personal Digital Assistants (PDAs) have been used to browse and search large repositories of audio-visual material [74, 47]. In this case the limitations imposed by the device are not as severe as those imposed by the mobile phones: more processing power, larger screen size, higher bandwidth available and broswer-based interface.

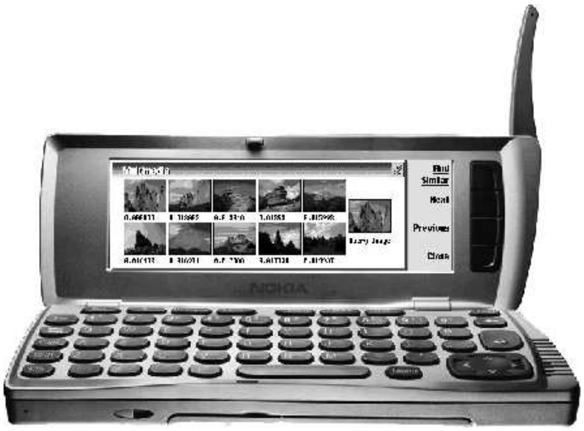


Figure 3.13: MUVIS over the 9210 NOKIA communicator.

Chapter 4

Contour-Based Shape Features

Our daily interaction with the environment around us is performed by visual means, where the information captured by the eyes is processed by the human visual system (HVS) and then transmitted to the brain for storage or analysis. Due to the inherent redundancy present in all the visual material, the human visual system pays great attention to abrupt transitions and changes in this information, e.g. change of color or texture between two smooth regions. Therefore, the human visual system focuses on edges and tends to ignore uniform regions [51, 100]. This capability is hard-wired into the retina, see Figure 4.1. Connected directly to the rods and cones of the retina are two layers of neurons that perform an operation similar to the Laplacian. This operation is called *Lateral Inhibition* and helps us to extract boundaries and edges.

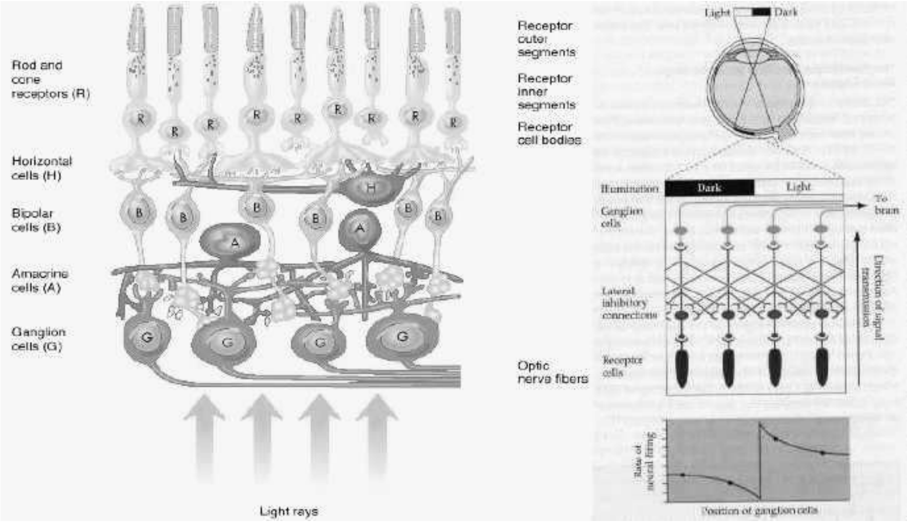


Figure 4.1: Lateral inhibition mechanism in the HVS

The illustration of the response of our retina to a transition, right hand in Figure 4.1, shows the signal transmitted via the optical nerve to the brain. It can be clearly seen that the lateral inhibition emphasizes the transition by an under-shoot and an over-shoot centered around the transition step. Knowing that there is a particular change of light intensity at a particular place and no change in light intensity until another particular place, the brain can “fill-in” the intensity between the two. Therefore, the brain compensates for the information thrown away by the eyes.

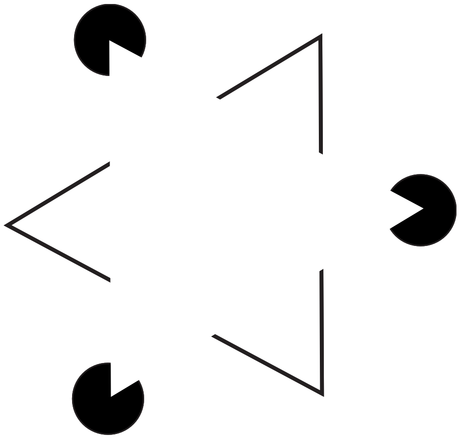


Figure 4.2: Optical illusion: “Kanisza’s Triangle”

Figure 4.2 illustrates the side effects of lateral inhibition and the brain compensation of the missing information on the perceived world. It represents the “Kanisza’s Triangle”, which physically consist of 3 circles with pie shaped wedges removed and 3 angles formed from straight lines. When you look at the images, you will likely see one triangle on top of another. The top triangle typically appears brighter, although the background is physically uniform. The triangular forms and the apparent brightness of the top triangle in comparison with the bottom one are subjective.

The *shape* of objects plays an essential role among the different aspects of visual information. Therefore, it is a very powerful feature when used in similarity search and retrieval. Unlike color and texture features, the shape of an object is strongly tied to the object functionality or identity. The importance of shape information to humans goes beyond the visual sense and appears in our daily vocabulary and expressions, e.g. *take shape, be in shape, best shape*. Unfortunately, semantically meaningful shapes are not easy to obtain from images, due to the poor performance of todays’ automatic segmentation algorithms. Moreover, the existence of noise, occlusion and distortions introduced during the image formation process, make object extraction and similarity estimation a more difficult task. Image formation is basically a projection of the 3D world onto the 2D pictorial space. A 2D silhouette often conveys enough information to allow the correct recognition of the original 3D object, as seen in Figure 4.3. Therefore, techniques developed for 2D shape analysis can be applied to the analysis of 3D objects, hence the importance of 2D shape indexing and retrieval techniques. Shape-based retrieval is one of the most challenging aspects in content-based image retrieval.

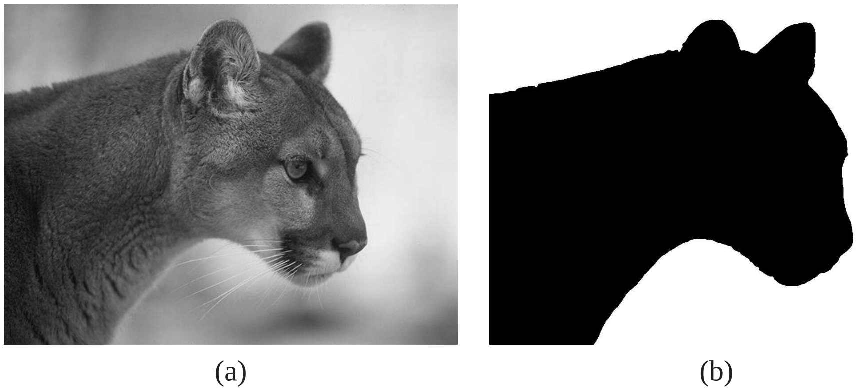


Figure 4.3: (a) Image containing a 2D projection of a 3D object (Tiger), (b) The 2D silhouette of the object.

The rest of this chapter is split into two sections, Section 4.1 gives an overview of shape representations; while Section 4.2 proposes our multiscale feature extraction techniques and their similarity measures. Region-based techniques will be covered in the next chapter.

4.1 Overview of Shape Representations

Shape representation techniques are generally characterized as being boundary-based or region-based. The former (also known as contour-based) represents the shape by its outline, while the latter considers the shape as being composed of a set of two-dimensional regions. Techniques in both categories can be further subdivided into scalar (or space domain) and transform domain. Figure 4.4 shows a subdivision of shape representations [124]. Note that, transform domain representations can be further subdivided into: single scale and multiscale representations.

As it can be seen in Figure 4.4, the shape representations can be subdivided into contour-based and region-based categories. Each of the two categories can be further subdivided into sub-categories. The contour-based approaches are subdivided into two classes:

- spatial domain techniques

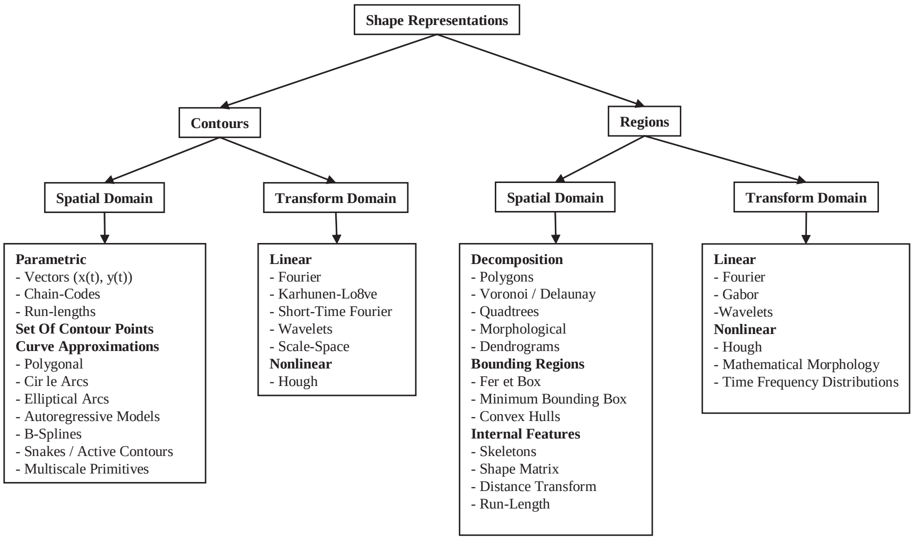


Figure 4.4: A taxonomy of shape representation techniques.

  - Parametric Contours: the boundary is represented as a parametric curve by an ordered sequence, e.g. Chain-codes,
  - Set of contour points,
  - Curve approximation: by a set of geometric primitives fitted to the contour, e.g. splines,

- Transform domain techniques: e.g. Fourier transform of the parametric representation.

Similarly, the region-based approaches are also subdivided into two classes:

- Spatial domain techniques
  - Region Decomposition: the shape region is partitioned into simpler forms and represented by the set of primitives, e.g. polygons,
  - Bounding Regions: e.g. enclosing rectangle,
  - Internal Features: e.g. Skeleton,

- Transform Domain Techniques: such as Gabor filters.

Selecting a set of features from the shape representation to characterize an object for a certain application is not easy, since one must take into consideration the variability of the shapes and the specific characteristics of the application domain. Feature comparison can be understood as a way of quantifying the similarity/dissimilarity between corresponding objects. This is a very difficult problem since it tries to mimic the human perception of similarity between objects [100].

Several shape features have been proposed in the literature for shape characterization [29]. Many of these techniques however, cannot be used for content-based retrieval due to their complexity or because they lack a counterpart in the human visual system. Therefore, techniques based on simple and visually meaningful shape features have been used in several content-based retrieval systems, (e.g. QBIC [38], MUVIS [16, 21, 148]) such as high curvature points [1, 110, 143], polygonal approximation [64], morphological and topological features and others [29, 122].

4.1.1 Simple Geometric Features

Contour-based features can be simple scalar values extracted from the boundary itself, e.g. aspect ratio illustrated in Figure 4.5, boundary length, eccentricity, circularity, circumference, complexity, directedness, relative area, right-anglessness, sharpness, straightness, transparency [32].

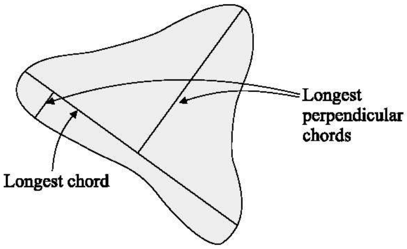


Figure 4.5: Aspect ratio: ratio of the sum of the lengths of chords which are oriented perpendicular to the longest chord and the longest chord length.

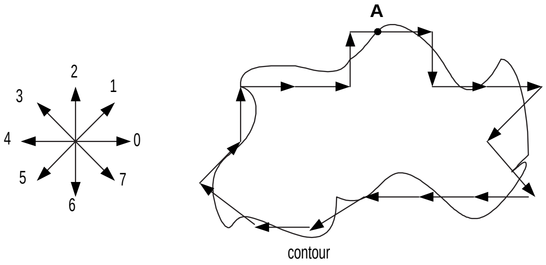
More complex contour-based shape feature extraction starts by creating a parametric contour representation; which is typically a one-dimensional function constructed from the two-dimensional shape boundary points. Later, the constructed one-dimensional function (e.g. curvature function) is used to extract a feature vector describing the shape of the object. In the following, we will describe some contour based features and for some of them the associated similarity algorithms that may be used in CBIR systems.

4.1.2 Spatial-Domain Features

Chain Codes

In chain coding (also known as Freeman code [39]), the direction vectors between successive boundary pixels are encoded. Figure 4.6 shows a commonly used chain code [59]. Here eight directions are used, coded by 3-bit code words. Typically, the chain code contains the start pixel address followed by a string of code words.

Such codes can be generalized by increasing the number of allowed direction vectors between successive boundary pixels.



Algorithm:

1. Start at any boundary pixel A,
2. Find the nearest edge pixel and code its orientation. in case of a tie chose the one with largest (or smallest) code value.
3. Continue until there are no more boundary pixels.

Boundary pixel orientations: (A), 060057444543120020

Chain code: A 000 110 000 000 101 111 100 100 100 101 100 011 001 010 000 000 010 000

Figure 4.6: Chain code representation.

The basic form of the chain code described above is sensitive to scaling and rotation, hence modified versions were derived [10]. Chain code representation is of no use in CBIR, due to its sensitivity to noise, and scaling. It is also hard to normalise to account for invariance with respect to rotations. A short review of chain-code techniques can be found in [91].

Stochastic Methods

The methods in this class are based on stochastic modeling of a 1D parameterized function of the contour itself or of some significant points over the contour. The 1D function is interpreted as the realisation of a stochastic process. The model parameters obtained by estimation are used as shape descriptors (feature vector).

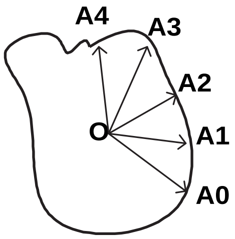


Figure 4.7: The centroid-to-boundary distance function r_t

Autoregressive Models An autoregressive (AR) model is a parametric equation that expresses each sample of an ordered set of data samples as a linear combination of a specified number of previous samples. The representation can be either one or two-dimensional depending on whether the Cartesian or polar coordinate system is used. An example of stochastic methods are the circular autoregressive models (CAR) proposed in [61]. The method models the parameterized distance function between the centroid and the boundary points. The CAR model is characterized by a set of unknown parameters and an independent noise sequence. Since the boundary is closed, the 1D time series distance function r_t is a periodic function, see Figure 4.7. The CAR model that was utilized is a stochastic process defined by the following m-th order difference equation:

$$r_t = \alpha + \sum_{j=1}^{m} \theta_j r_{t-j} + \sqrt{\beta} w_t \tag{4.1}$$

where w_t are independent random noise sources. The parameters $(\alpha, \theta_1, \ldots, \theta_m, \beta)$ are unknown and need to be estimated. Maximum likelihood (ML) parameter estimation can be used to determine these parameters. In this method, the ML estimated parameters θ_j are invariant to translations, rotations and scale. Parameters α and β are not scale invariant, but the quotient $(\alpha/\sqrt{\beta})$ is. Thus the vector $[\theta_1, \ldots, \theta_m, \alpha/\sqrt{\beta}]^T$ can be used as a shape descriptor which is robust to geometric transformations. The similarity can then be measured by computing the weighted Euclidean distance between shape feature vectors. The weighting of the descriptor vector can be achieved so that the components that are common within a training class are emphasized while the components that differed are de-emphasized. The experimental results showed that the CAR method is quite effective in measuring similarity between shapes [61]. The disadvantage with AR boundary modeling is that in the case of complex boundaries, a small number of AR parameters is not sufficient for effective description.

Hidden Markov Models Another example of stochastic methods is the use of Hidden Markov model (HMM) of shape boundary to extract the shape feature. Hidden Markov Models [112] are finite stochastic automata which have been successfully applied to continuous speech and online handwriting recognition but recently also to new applications such as gesture recognition and video indexing.

Figure 4.8 shows a continuous three state HMM with transition probabilities $a_{ij} = Pr(q_t = s_j | q_{t-1} = s_i)$, $(i, j) \in \{1, 2, 3\}^2$ and output probability density functions $b_j(\vec{o})$, where q_t denotes the actual state at time t, s_j is a distinct state and $\vec{o}$ denotes an observation vector. The pdf $b_j(\vec{o})$ of state s_j is usually given by a finite Gaussian mixture of the form

$$b_j(\vec{o}) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\vec{o}, \vec{\mu}_{jm}, \vec{\Sigma}_{jm})$$

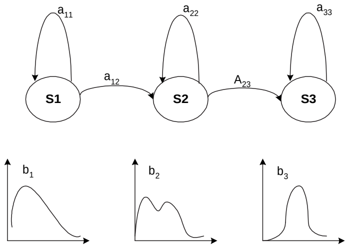


Figure 4.8: Continuous linear Hidden Markov Model

where c_{jm} is the mixture coefficient for the m-th component of the mixture and $\mathcal{N}(\vec{o}, \vec{\mu}_{jm}, \vec{\Sigma}_{jm})$ is a multivariate Gaussian probability density function with mean vector $\vec{\mu}_{jm}$ and covariance matrix $\vec{\Sigma}_{jm}$.
A HMM $\lambda(\vec{\pi}, \bar{a}, \vec{b})$ with N states is fully described by the $N \times N$-dimensional transition matrix $\bar{a}$, the N-dimensional output probability density function vector $\vec{b}$ and the initial state distribution vector $\vec{\pi}$. After the model λ has been trained using, for example, the Baum-Welch algorithm, the feature sequence $\vec{O} = \vec{o}_1, \ldots, \vec{o}_T$ can be scored according to:

$$Pr(\vec{O}|\lambda) = \sum_{q} \pi_{q_1} bq_1(\vec{o}_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(\vec{o}_t) \tag{4.2}$$

In practice, $Pr(\vec{O}|\lambda)$ is usually estimated using the Viterbi algorithm. The Hidden Markov Model described above can be used to model a function of the boundary of a contour [50].

Polygonal Approximations

Hoffman et al. [52] argued that when the human visual system decomposes objects it does so at points of high negative curvature, see Figure 4.9. Therefore, approximating curves by straight lines joining these high curvature points (HCP) retain the maximal amount of information necessary for successful shape recognition. This can be explained by the fact that our visual system focuses on singularities and ignores smooth curves due to lateral inhibition. Thus for a large class of objects, the polygonal approximation of contours based on high curvature points makes sense. Moreover, it compresses effectively the information conveyed by a given shape in a small set of vertices.

In this approach, the shape representation reduces to determining the vertices of the polygon. Several approaches to determine the best set of vertices can be found in the literature: a straight forward technique consists of fitting a polygon with vertices at equal intervals inside the contour with a pre-determined accuracy. Another technique, more consistent with the human visual properties, consists in

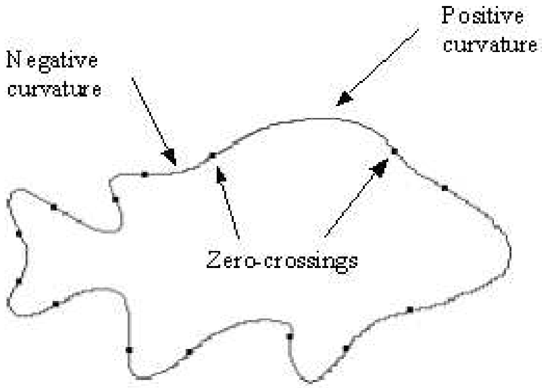


Figure 4.9: Positive curvature, negative curvature and zeros of curvature of a fish contour.

assigning the vertices to the existing high curvature points of the contour. We have used this latter method to perform the polygonal approximation of contours, and as shown in Figure 4.10 the visual properties of shapes are preserved even when few vertices are used. The high curvature points were extracted by thresholding the Wavelet Transform Modulus Maxima (WTMM) of the orientation profile. More details on this technique can be found in Section 4.2.

A standard method of representing a simple polygon A is to describe its boundary by giving a list of its vertices, each vertex represented by its coordinates pair. The similarity measures associated with polygonal representation of shapes depend on the method used to describe the polygon. If a fixed number of vertices is used to describe all the polygons, the Euclidean distance may be used to estimate their similarity. The Euclidean distance fails to make the difference between small errors on each of the vertices of the polygon and a large error on a single vertex, which is a key factor used by the HVS in distinguishing shapes. On the other hand if the number of vertices is not fixed for all the objects in the database, Minkowski type distances are not applicable. Therefore, we developed similarity measures which try to mimic the human way of comparing polygons. These methods are explained in detail in Section 4.2.

An alternative representation of the boundary of a simple polygon A is achieved through the *turning function* [3] $\Theta_A(s)$. The function $\Theta_A(s)$ measures the angle of the counterclockwise tangent as a function of the arc length s, measured from some reference point O, which can be any point on the polygon boundary. $\Theta_A(s)$, keeps track of the turning that takes place as a function of $s \in [0, 1]$, increasing with left-hand turns and decreasing with right-hand turns (Figure 4.11). Formally, if $\kappa(s)$ is the curvature function of a contour, then $\kappa(s) = \Theta'_A(s)$. The function $\Theta_A(s)$ is piecewise constant, making its computation easy and fast, it is also in-

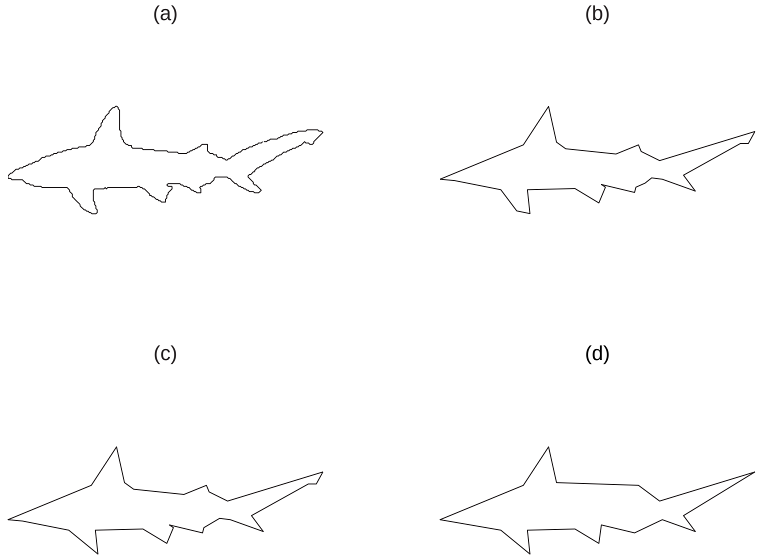


Figure 4.10: Polygonal approximation of a contour based on high curvature points, (a) original contour, (b) corners detected at level 1, (c) level 3, (d) level 5.

variant under translation and scaling. A rotation by an angle θ results in a simple shift, so that the corresponding turning function becomes $\Theta_A(s) + \theta$. Note also that changing the start point O by an amount $t \in [0, 1]$ along the perimeter of a polygon A corresponds to a horizontal shift in the argument of the function $\Theta_A(s)$ to become $\Theta_A(s + t)$.

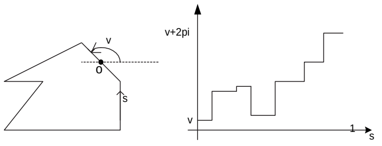


Figure 4.11: Turning function $\Theta_A(s)$

In order to evaluate similarity between polygons A and B, a distance function L_p is defined [3] to measure the distance between their turning functions $\Theta_A(s)$ and $\Theta_B(s)$ minimized with respect to vertical and horizontal shifts to account for rotations and change of starting point, respectively. For $p = 2$, the distance function can be computed [3] efficiently in $O(n^2 \log n)$, for a polygon with n vertices, which makes this measure accessible. Similarity is reasonably reflected in this measure. The measure is also robust to small occlusions. One major drawback of this distance function is that it is unstable with respect to nonuniform noise.

4.1.3 Transform-Domain Features

Fourier Descriptors

An effective representation of a curve boundary is obtained using the 1D Fourier descriptors (FD) [105]. These are complex coefficients of the Fourier series expansion of waveforms. The boundary is parameterized along the contour using a pair of one dimensional waveforms $x(t)$ and $y(t)$. N samples of the continuous boundary are taken such that:

$$u(n) = x(n) + jy(n), \quad n = 0, 1, \ldots, N-1. \tag{4.3}$$

For closed contours $u(n)$ will be periodic with period N. The Discrete Fourier Transform (DFT) [121] representation of $u(n)$ and its inverse is given by:

$$F(k) = \sum_{n=0}^{N-1} u(n) exp\left[\frac{-j2\pi kn}{N}\right] = M(k)e^{j\theta(k)}, \quad 0 \leq k \leq N-1 \tag{4.4}$$

$$u(n) = \frac{1}{N}\sum_{k=0}^{N-1} F(k) exp\left[\frac{j2\pi kn}{N}\right], \quad 0 \leq n \leq N-1 \tag{4.5}$$

The complex coefficients F(k) are called the *Fourier Descriptors* (FD) of the boundary. If $u'(n)$ is obtained from $u(n)$ by a rotation of angle ϕ, a scaling factor of α and a shift of l in the starting point :

$$u'(n) = \alpha u(n-l)e^{j\phi} \tag{4.6}$$

then the corresponding Modified Fourier Descriptors (MDF) [121] are given by

$$F'(k) = \alpha e^{-j(\frac{2\pi nlk}{N} - \phi)} F(k) = M'(k)e^{j\theta'(k)} \tag{4.7}$$

where $M'(k) = \alpha M(k)$ and $\theta'(k) = \phi + \theta(k) - \frac{2\pi lk}{N}$

From the above equation, we see that the magnitude of the Fourier Descriptors ($M(k)$) is invariant to changes of starting point, rotations and reflections. It is also easy to normalize $M(k)$ for scale invariance by forcing all contours in the database to have a fixed perimeter and taking an equal number of samples N for all contours.

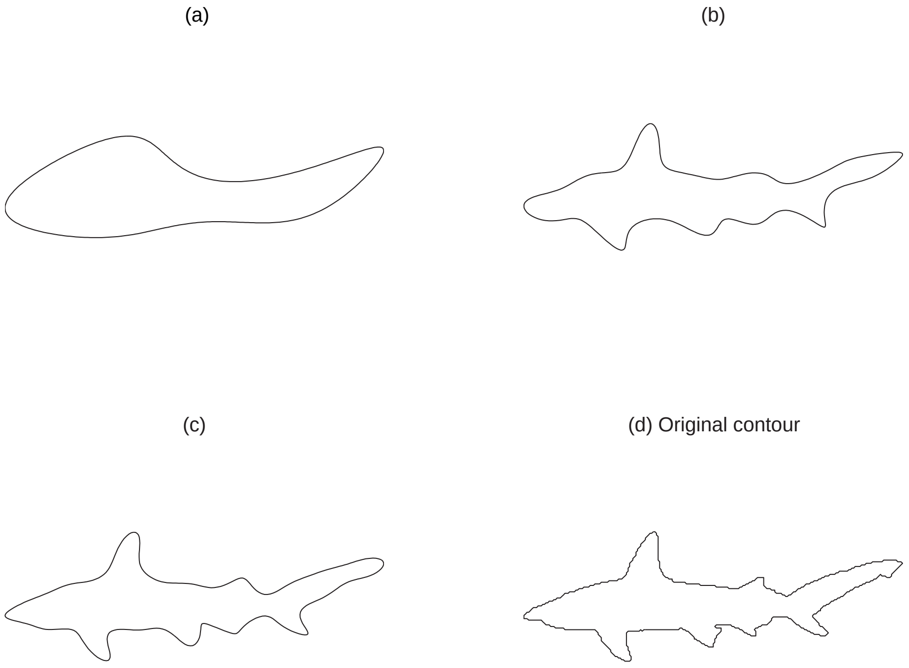
The Fourier Descriptors based shape feature vector can be defined as:

$$F = [M(0), M(1), \ldots, M(N-1)] \tag{4.8}$$

the length N of the feature vector, or the number of FD's used, depends on the precision level of the matching, as the higher coefficients contain high frequency details of the contour. As shown in Figure 4.12, a contour can be adequately reconstructed using only the first 25 coefficients, thus allowing for a relatively short feature vector.

A similarity measure commonly used with FD feature vector is the Euclidean distance: for an image I from the database and a query image Q their similarity distance can be calculated as:

$$d(F^I, F^Q) = \sqrt{\sum_{j=1}^{n} (M^I(j) - M^Q(j))^2} \tag{4.9}$$



(a)

(b)

(c)

(d) Original contour

Figure 4.12: The contour as reconstructed from few coefficients, (a) 5 coefficients are used (b) 15 (c) 25 (d) Original contour

Figure 4.13 shows an Example of search results based on the Fourier Descriptors method, the search is carried over the full database of 1130 marine animals. The 15 closest matches to the query image are displayed. The similarity distances between the query image and the retrieved images are shown on top of each image. Further experimental results and comparisons of FD to other shape descriptors can be found in [146].

From these results, we can say that the Fourier Descriptors method performed rather poorly on similarity retrieval, in fact, most similar images, as judged by humans, are absent from the 15 first images retrieved by the method.

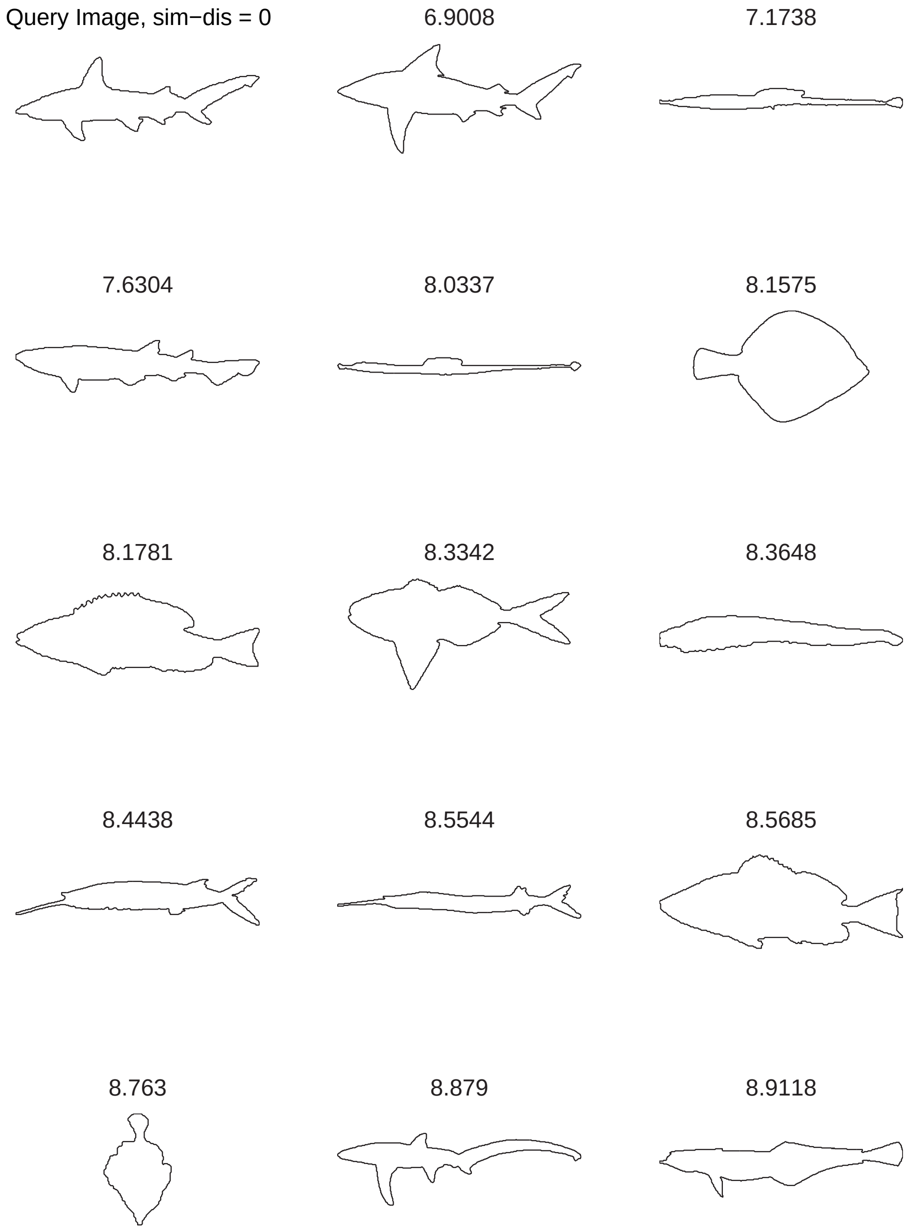


Figure 4.13: Example of search results based on Fourier Descriptors method, where the search is carried over the full database (1130 images). The similarity distance to the query image is shown above each retrieved image.

The FD method is robust to noise and geometric transformations. The major advantage of this method is that it is easy to implement (accessible). A major disadvantage, is that the frequency terms have no obvious counter-part in human vision. Therefore, retrieval results cannot be analysed in terms easily understandable by common users. Another difficulty with the Fourier Descriptors method arises from the fact that they use global sinusoids as basis functions, which fail to provide for a spatial localization of the coefficients, thus occlusions and local deformations of the shape will be reflected in all the coefficients and therefore change all the entries of the descriptor.

Scale-Space Techniques

Several research groups worked on multiscale representations for shape analysis: Rosenfeld [115] developed a multiscale edge detection scheme and recently Witkin [155] proposed the scale-space filtering approach that uses a set of parameterized Gaussian kernels to smooth the curvature function along the contour of an object and extract the descriptive primitives such as the locations of curvature extrema. Moktharian and Mackworth [94, 95, 96] used this filtering technique to detect and track inflection points across different scales and then used the resulting curvature scale space (CSS) representation to estimate similarity between different shapes.

The scale-space approach was first introduced by Witkin (in 1983) [155] and interest in this technique has been growing ever since. A key point in the work of Witkin is that important features of a signal are generally associated with extreme points, such as local extrema. This highlights the importance of detecting and locating these points based on the signal derivatives. In practical situations, the analytical signal derivatives are unknown thus, numerical differentiation methods are used. These are based on a neighborhood around each point, whose length defines the analyzing scale. As the differentiations enhance high frequency noise, Gaussian filtering was widely used to solve this problem.

The desired derivative is obtained by convolving $u^{(1)}(t)$, the derivative of $u(t)$, with a Gaussian $g(t)$, and applying the property of the convolution:

$$u^{(1)}(t) * g(t) = (u(t) * g(t))^{(1)} = u(t) * g^{(1)}(t) \tag{4.10}$$

This means that the signal $u(t)$ can be differentiated by convolving it with the first derivative of the Gaussian, and the extrema of $u(t)$ can be found at zero-crossings of the convolution output.

The standard deviation of the Gaussian determines the scale of analysis. For each scale, a set of extreme points are detected. The scale-space representation of the signal $u(t)$ is defined as the evolution of the set of these points as a function of scale. So, each zero crossing occupies one position in one scale and their positions change smoothly with increasing scale until they vanish pairwise.

Therefore, the number of extrema decreases from a maximum at the original scale (lowest scale contains all the details) to zero when the contour becomes a convex contour and thus contains no local extrema anymore.

The most important scale-space technique in the context of content-based image retrieval is called Curvature Scale Space (CSS) [1, 93, 94, 95, 96]. It was adopted as a contour-based shape descriptor by MPEG-7. In the rest of this section we will describe in more detail the CSS technique.

Shapes are represented first as chain codes, later the contour is smoothed iteratively at increasing scales to smooth the contour until it becomes convex. The curvature at each point is computed at each iteration, and the zero crossings of the curvature function are located and plotted. When the contour becomes convex, the curvature function of the contour has no more zero crossing. The CSS is obtained by plotting all the zero crossings obtained during the filtering. The resulting image, i.e. the graph representing the scale vs the indexes (of parameterization) of the zeros of curvature, is called the Curvature Scale Space Image (CSS image).

Figure 4.14 shows an example of a multi-scale representation of a contour [96] via its CSS, and the smoothed contour at three different scales.

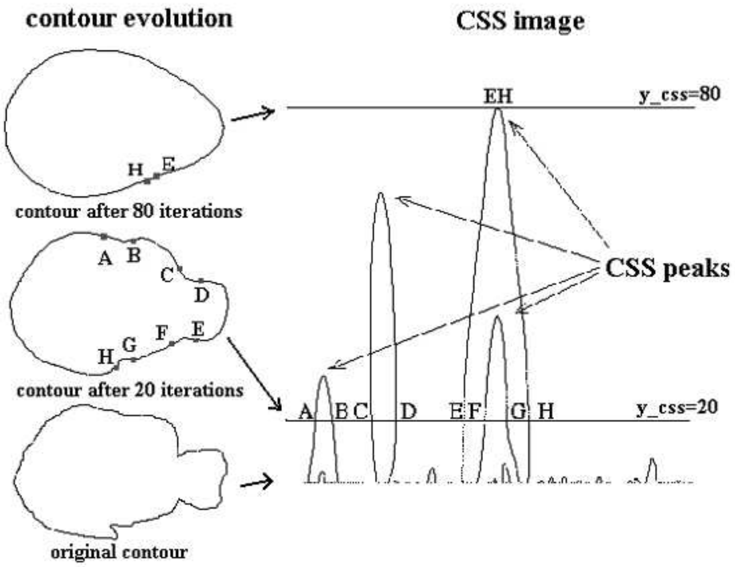


Figure 4.14: A fish contour represented at three different scales, with the evolution of the zeros of curvature points on the contour and in the CSS image.

This representation is invariant under translation. Besides, a rotation of the object usually causes a circular shift of its CSS image, which can easily be determined during the matching process. A change in the starting point of the contour has the same effect. Moreover, if the curves to be compared are normalized (by normalizing the indexes of the zeros), then, scaling does not change the representation, and if noise creates some small lobes in the CSS images, the main ones will remain unaffected. These properties make the CSS representation suitable for similarity-based retrieval techniques.

As stated previously, the largest lobes in the CSS image will not be affected by noise. Thus a compact and efficient way to represent the object contour is to only keep the location of the maxima of its CSS image. Besides, as small lobes will only give information on the existing noise in the contour, the smallest maxima should be discarded, and therefore those maxima which appear at scales lower than 10 $\%$ of the largest scale in the CSS image are eliminated.

The basic idea behind the similarity estimation using the CSS images is to obtain a coarse match using the structural features (the CSS image maxima) of the input curves. Such a match can be found quickly and reliably since at high scales of the CSS images, there are relatively few maxima to be matched. The reason for using these maxima is that they are the most significant points of the lobes of the CSS images. The CSS coordinates of a maximum convey information both on the location and the scale of the corresponding lobe, while a whole lobe is in general similar (in its shape) to the other lobes in the CSS image (see in Figure 4.14). Furthermore, maxima are isolated point features and therefore solving the matching problem between two sets of maxima is relatively simple.

The task of the similarity estimation algorithm is to find the best correspondence between the two sets of maxima corresponding to two contours. The allowed transformation from one set to the other is mere horizontal translation.

The MPEG-7 contour Shape Descriptor (SD) based on the CSS consists of: the eccentricity and circularity values of the original and filtered contour, the index indicating the number of peaks in the CSS image, the magnitude of the lagest peak and the (x, y) position of the remaining peaks.

Wavelet Transform

Multiresolution analysis techniques decompose the signal into components at different scales, so that the coarsest scale components carry the global approximation information while the finer scale components contain the details information [24]. Additionally some recent psychophysics models suggest that the human visual system processes and analyses image information at different resolutions [98].

Wavelet representation is especially useful for shape analysis in vision problems because its theory provides powerful properties, such as algorithms for dominant points detection, and local periodic pattern analysis. In fact wavelets constitute an efficient mathematical tool for the detection and characterization of signals' singularities. Two important points should always be considered by multiscale shape analysis techniques:

1. Important shape structures are generally associated with transient events in the object,

2. Different events may occur at different scales.

An event is usually a specific characteristic along the signal, such as a discontinuity, singularity or local frequency. One of the main features of the wavelet transform is its ability to separate transient events from the signal, whilst ensuring that local signal modifications do not affect the whole representation.

Such a representation is highly desirable for shape analysis problems where different situations, such as partial occlusion, may imply local interferences to the shape, therefore avoiding one of the main disadvantage of global shape descriptors, such as the Fourier descriptors and moments.

In recent years, wavelet transform became an active area of research for multiresolution signal and image analysis. Chuang and Kuo [24] introduced the wavelet descriptor of planar curves and showed its desirable properties such as invariance, unicity and stability. Hwang and Mallat [84] proved that there could not be a singularity without a local maximum of the wavelet transform at the finer scales, therefore, the wavelet transform modulus maxima (WTMM) seem to be very appropriate for the description of contours, enabling the detection of high curvature points at different resolutions [75, 110]. In the rest of this chapter we will present several shape descriptors based on high curvature points detected at different scales. The descriptors are based on local features of the polygonal approximation of the original contour or directly based on the characteristics of the wavelet transform modulus maxima themselves. The methods are compared against the curvature scale space technique and a group of human users.

4.2 Wavelet-Based Feature Extraction

In section 4.1.2, we have seen that high curvature points are very efficient in compressing the information present in a closed contour, in a way that conforms closely to human visual perception. The techniques used to extract those points of interest can be broadly classified into two categories [36, 37, 111]: single scale and multiscale techniques. The former techniques might suffer from finding lots of unimportant details, while at the same time missing large rounded corners. The latter techniques avoid these problems, and provide additional information about the “structural” importance of the high curvature points.

In [96], the curvature scale space representation was used to extract the contour HCP. However, since planar curves smoothed using a Gaussian kernel suffer from shrinkage [78, 79], the tracking and the correct localization of high curvature points becomes more difficult. In contrast with the scale-space filtering approach, which serves primarily as an *analysis tool*, the wavelet decomposition provides an effective *synthesis tool*, which additionally, does not have any shrinkage problems. This makes the Wavelet decomposition very useful for detecting local features of a curve due to the spatial and frequency localization property of the wavelet bases [24].

The Wavelet Transform (WT) decomposes a signal into a family of functions that are the translation and dilation of a unique function $\psi(x)$ whose average is zero. This function is called the mother wavelet. The WT of a signal $f(x)$ is given by

$$Wf(s,a) = f * \psi_s(a) \tag{4.11}$$

where $*$ denotes the convolution operator and $\psi_s(x) = \frac{1}{s}\psi(\frac{x}{s})$. The so called "dyadic scales" denote the scales that are powers of 2, that is $s = 2^j$, where $j = 1, 2, 3 \ldots$.

In [84], Mallat has defined a class of basic wavelets that are regular and have compact supports. Additionally, their corresponding WT can be implemented by a fast algorithm. Mallat also proved that all singular points of a signal correspond necessarily to local maxima in the wavelet transform of that signal. This idea was adopted in [110] to detect all the singularities of a contour by determining the local maxima of the wavelet transform of the contours' orientation profile. It was also confirmed in [110] that the quadratic spline mother wavelet $\psi(x)$ given by equation (4.12) is very efficient for detecting singularities on the contour. $\psi(x)$ is the first derivative of the cubic spline smoothing function $\theta_s(x)$:

$$\psi(x) = \begin{cases} 24|x|^3/x - 16|x|^2/x & \text{if } |x| \leq 0.5 \\ -8|x|^3/x + 16|x|^3/x - 8|x|/x & \text{if } 0.5 \leq |x| \leq 1 \\ 0 & \text{if } |x| \geq 1 \end{cases} \tag{4.12}$$

$$\theta(x) = \begin{cases} 8|x|^3 - 8|x|^2 & \text{if } |x| \leq 0.5 \\ -8/3|x|^3 + 8|x|^2 - 8|x| + 8/3 & \text{if } 0.5 \leq |x| \leq 1 \\ 0 & \text{if } |x| \geq 1 \end{cases} \tag{4.13}$$

We transform the orientation profile of the contour $\phi(u)$ using the mother wavelet defined above. In the second step of the algorithm we detect the local maximum of the wavelet transform. These maxima correspond to all the singularities appearing on the contour $\Gamma(u) = (x(u), y(u))$.

$$W\phi(s,u) \quad = \quad \phi(u) * \psi_s(u) \tag{4.14}$$

where $\phi(u)$ is defined as:

$$\phi(u) = \tan^{-1}((dy(u)/du)/(dx(u)/du)) \tag{4.15}$$

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{(\dot{x}^2(u) + \dot{y}^2(u))^{3/2}}. \tag{4.16}$$

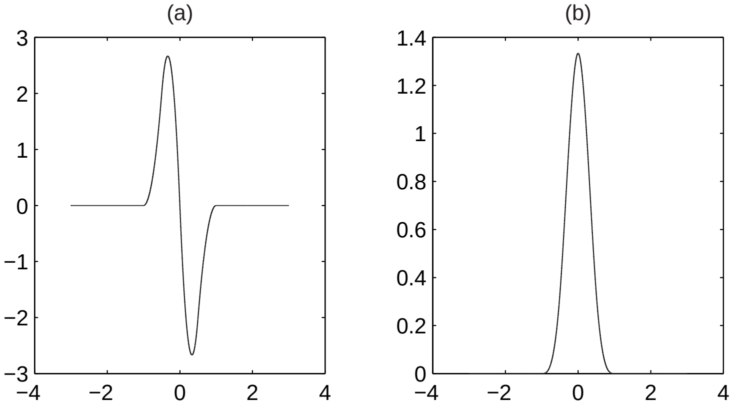


Figure 4.15: (a) The cubic spline function $\theta(x)$, (b) Mother wavelet $\psi(x)$ used for HCP detection (quadratic spline), obtained as the derivative of $\theta(x)$ in (a).

the curvature function $\kappa(u)$ defined in equation (4.16) is the derivative of the orientation $\phi(u)$.

Figure 4.16 shows the modulus maxima of the wavelet transform computed for dyadic scales 2^2 to 2^6. Note that as we increase the scale, the number of maxima decreases because of the smoothing process and only those maxima which correspond to important singularities survive the smoothing. For the purpose of extracting a few important high curvature points that are useful for polygonal approximation, we select the WTMM maxima above a certain threshold in T_j each level at scale s^j, ($T_j = 0.16$ was used), track them down to lower scales (to compensate for change of location across scales) and determine the exact location of these HCP on the contour. Figure 4.17 displays at different scales, those points which were extracted from the WTMM profile plotted in Figure 4.16. In Figure 4.18, the polygonal approximation using the high curvature points extracted at level $j = 4$, is plotted against the original contour. It can be safely assumed that this approximation captures the salient properties of the shape and gives us a clear idea about its structure. We used those vertices to represent the polygonal approximation of the contour of the original object.

Once the approximation of a contour by a polygon is achieved based on the HCP detected using the WTMM, different features are extracted at each vertex of the polygon. Three different algorithms were developed to characterize contours based on these features. These algorithms are presented below.

4.2.1 Wavelet Transform Modulus Maxima Features

In this section we propose a robust wavelet-based matching algorithm, which is suitable for shape matching based on the object's contour. The features extracted are insensitive to noise and are invariant to translation, rotation and scale change.

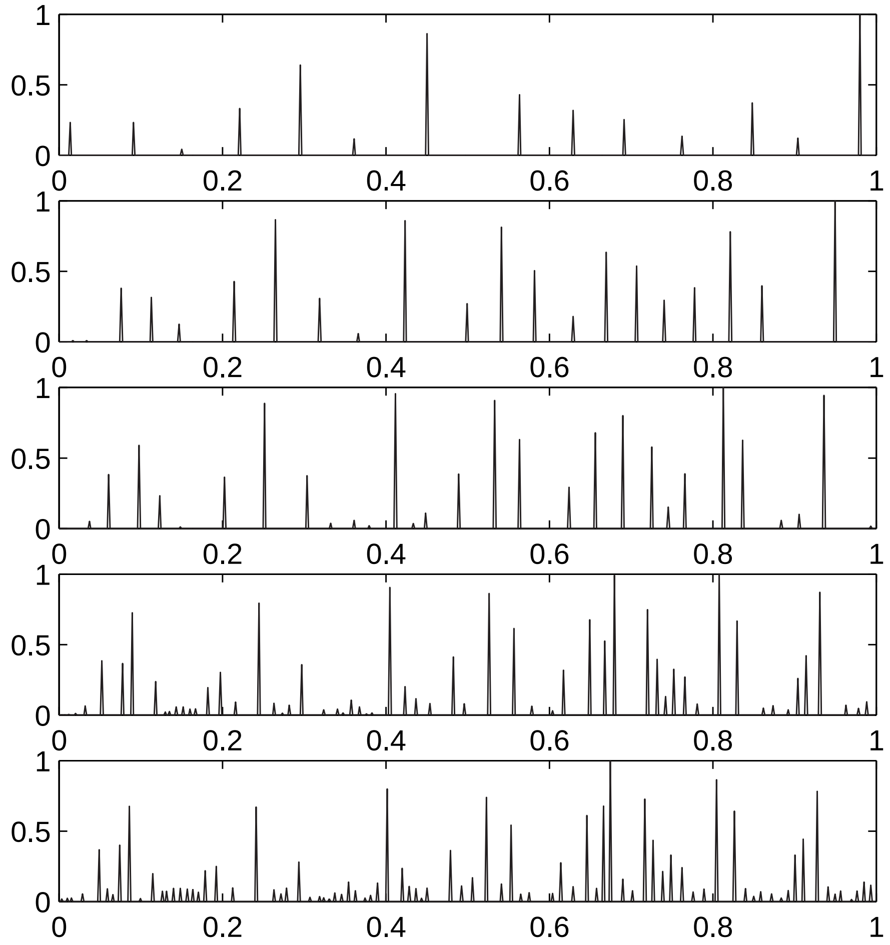


Figure 4.16: The wavelet transform modulus maxima (WTMM) for dyadic scales at levels $j = 2, ..., 6$, ($s = 2^j$), from bottom to top, respectively.

The algorithm uses WTMM to detect the location of high curvature points and to estimate the degree of similarity between two shapes at these points. Their performance is evaluated for estimating the similarity of natural objects. Retrieved images with the proposed approach are compared to those retrieved with the contour scale-space (CSS) technique [1, 93, 94, 95, 96] and those retrieved manually by a representative set of users.

Fast schemes for narrowing down the search space are essential in content-based retrieval systems where large sets of images are considered. Therefore, we used the aspect ratio γ to reduce the search space, by filtering out objects with error on γ larger than a fixed threshold are discarded. The remaining candidates go through the second step of the retrieval process, where low-level features are ex-

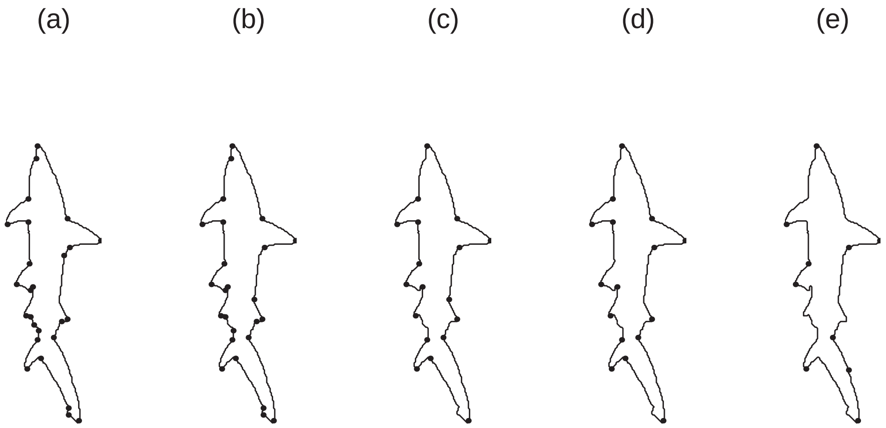


Figure 4.17: High curvature points detected for levels $j = 2, ..., 6$, from left to right, respectively.

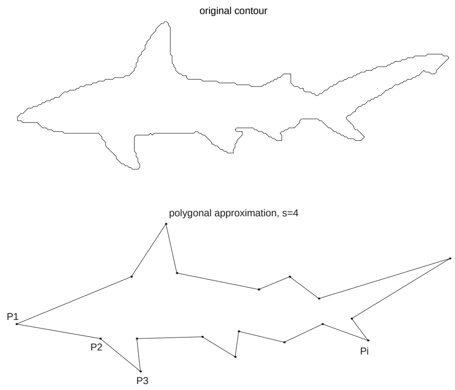


Figure 4.18: Polygonal approximation using high curvature points appearing at scales $s \geq 4$.

tracted at high curvature points and compared to those of the query image. Thus the boundary is tracked and its orientation profile is computed as in [21]. The orientation profile of each shape is up-sampled and interpolated in a way to have the same number of points for each contour. The Wavelet transform of the orientation profile is computed for dyadic scales from 2^1 to 2^6. WTMM are then computed, see Figure 4.19, and only those WTMMs larger than a certain threshold are considered important singularities. These singularities represent the HCPs of the contour, therefore the magnitudes and positions of the WTMM contain enough information to characterize the contour and thus are used as entries in the feature vector. Similarity between contours is estimated at each level of the decomposition independently, and the overall similarity measure is computed as the maximum value of the single level similarity scores.

Feature Extraction and Matching Algorithm

1. Select candidate objects with aspect ratios similar to that of the query object,
2. Compute the WTMM of the orientation profile of the contours,
3. Consider only the WTMM larger than a certain threshold T_{WTMM} and use their position and magnitude as the feature vector entries,
4. Compute a similarity score, at each level, between the query image and each candidate image,
5. The final similarity score is computed as the maximum of the scores at each level.

At each wavelet decomposition level l , we characterize the query image contour with two vectors, M_l^q and L_l^q, where, $M_l^q = [m_{l1}^q, m_{l2}^q, ..., m_{lm}^q]$ contains the m magnitudes of the WTMM and $L_l^q = [p_{l1}^q, p_{l2}^q, ..., p_{lm}^q]$ contains the m locations of the high curvature points on the normalized contour.

Similarly, at each decomposition level l, the candidate image contour is characterized with two vectors, M_l^c and L_l^c of length n.

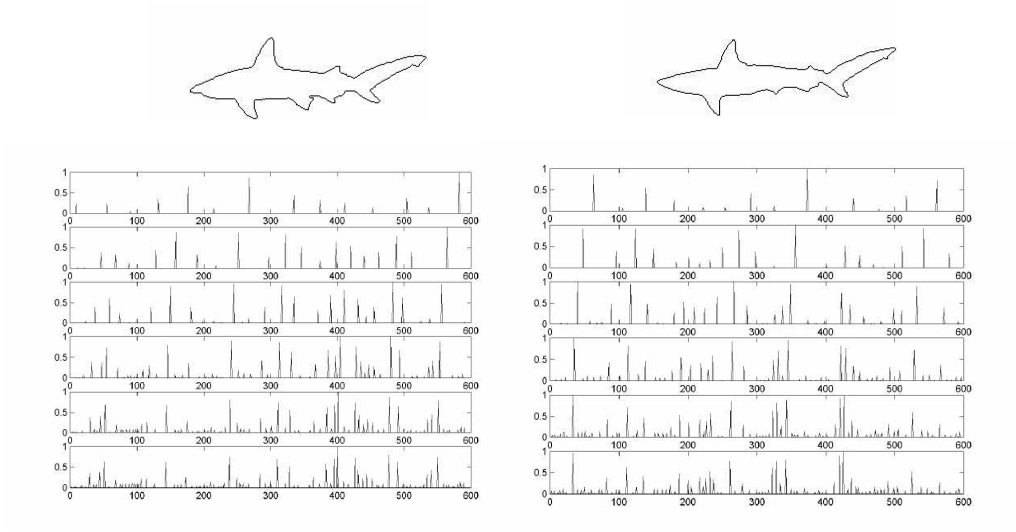


Figure 4.19: Two fish contours and their corresponding WTMM.

Before starting the matching, feature vectors from the query and candidate contours are shifted in a way to have their singularities corresponding to the largest WTMM aligned. Which means, that we start the matching process from the boundary points having the highest curvature at each decomposition level. This shift of the feature vector entries, makes the similarity estimation resilient to rotation and starting point variations. Considering the two contours in Figure 4.19 of the two contours Q and C (from left to right) of similar sharks. If we consider the top scale, it can be easily seen that if we shift the largest maximum of the WTMM of Q from its position ~ 600 to put it on the same coordinate of the largest maximum of the WTMM of C at ~ 400, other maxima from Q will be paired to maxima from C, e.g. the maximum at ~ 260 in Q and the one at ~ 60 in C. This pairing operation is the main operation in the similarity estimation algorithm. Since we are not interested in exact matching, the pairing operation is not expected to find exact correspondence between the maxima of the two contours, but rather a correspondence with an acceptable margin of error on the position and the magnitude of the maxima. Therefore, a valid match between two high curvature points is found if the differences between their corresponding WTMM locations and magnitudes are under the thresholds T_M and T_L respectively. In other words we are trying to match every maximum within a window, in the other contour representation. In Figure 4.20 the HCPs detected at levels $2, ..., 6$ of the two fishes in Figure 4.19 are highlighted and the polygonal approximation of both contours at each of these levels are shown in Figure 4.21.

Let K be the number of matched maxima. The similarity score at level l, for $l = j, ..., 6$, is computed as:

$$s_l = \frac{2 \times (K - \xi)}{m + n} \times 100, \tag{4.17}$$

where, $\xi = \sum_{i=1}^{K}(\frac{|\delta m_i|}{mean(m^q_{li}, m^c_{li})} + \frac{2\times|\delta p_i|}{L})$, where L is the length of the contour, δm_i and δp_i are the errors on the magnitude and position of the i^{th} matched maxima. ξ gives an indication on how good the match is between the two sets of high curvature points.

The lower levels are not considered in the matching process in order to make our measure unaffected by the presence of noise and small details on the contours. The overall similarity score is thus $S = max(s_l)$ for $l \in \{4, 5, 6\}$ only.

This approach is similar to the CSS technique [48], it is however more effective since the exact location of the HCP is determined with high precision by tracking the WTMM through the decomposition levels until the original contour. Moreover it is faster, since only a few decomposition levels are needed, unlike the CSS where the full decomposition is required. Moreover, by considering high curvature points only, just the visually important contour details are used to estimate the similarity of two shapes and redundant information is discarded by ignoring smooth curves. The proposed technique preserves most of the shape information, since the object contour can be accurately reconstructed from its WTMM [21].

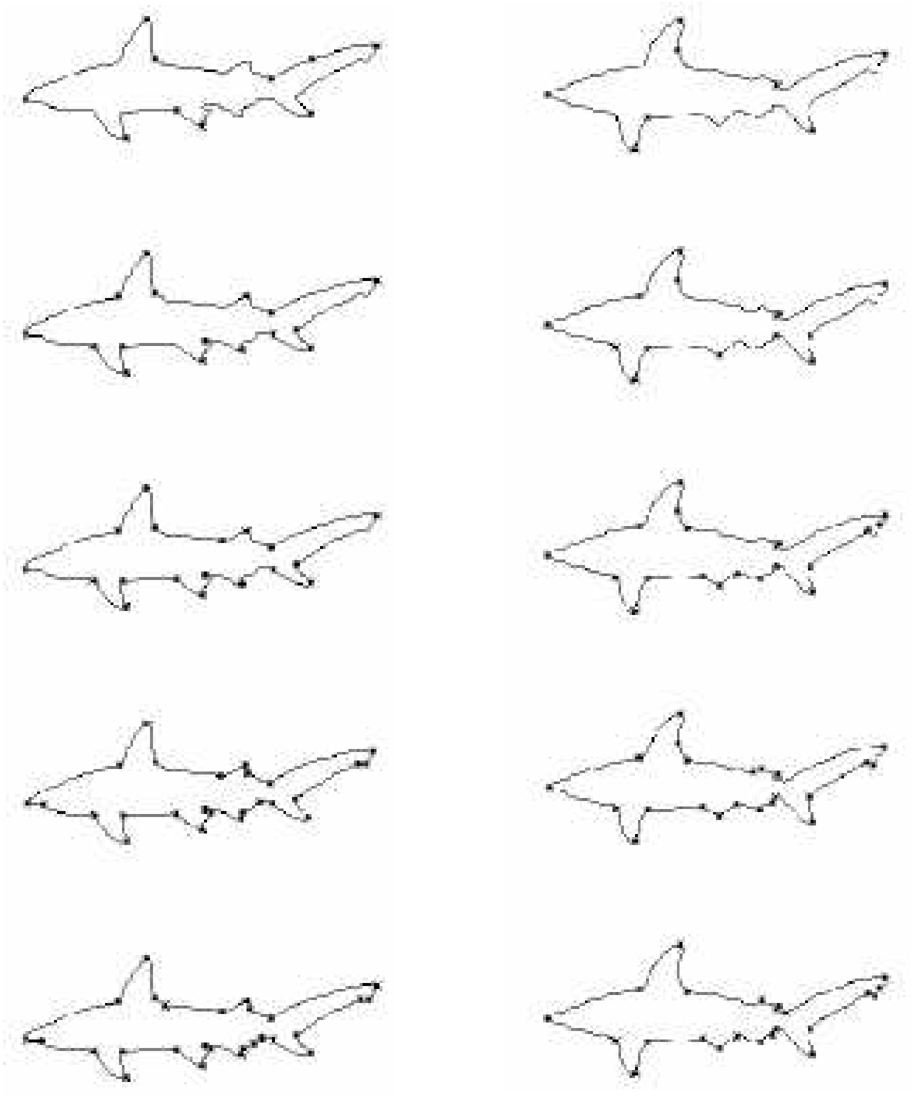


Figure 4.20: High curvature points detected for levels $j = 2, ..., 6$, for the two fish contours shown in Figure 4.19.

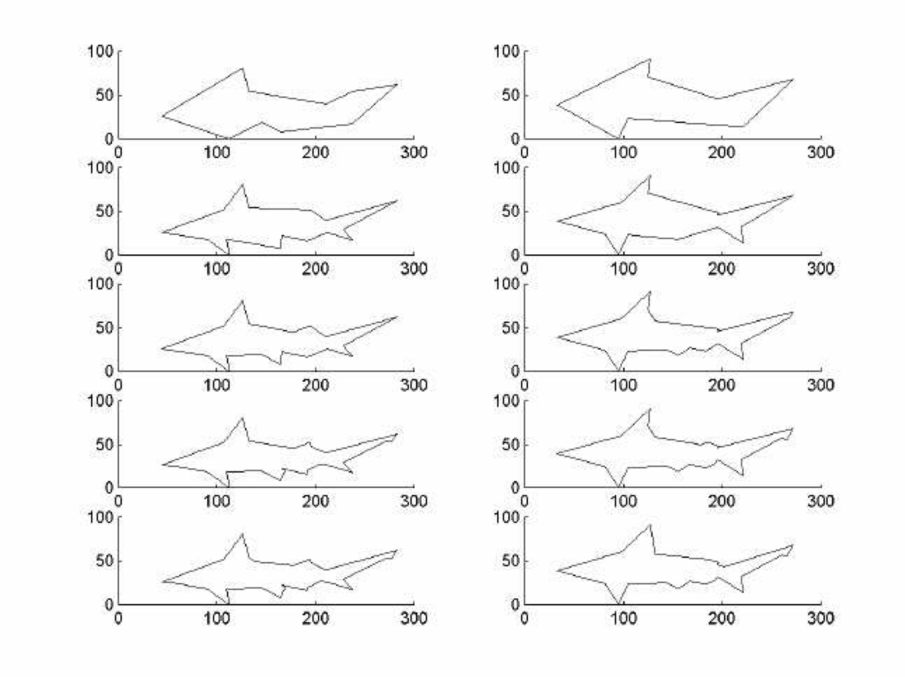


Figure 4.21: Polygonal approximations of the two contours at different scales, based on the high curvature points in Figure 4.20.

Experimental Results

In these experiments we used 1130 fish contour images. The boundary of each fish in the database is represented by a sequence of 1000 points. The results of querying the fish database with the image shown on the left in Figure 4.19 are presented in Figure 4.22. Here, the five most similar images retrieved using the CSS algorithm, the matching results of human users and the proposed algorithm are shown. Three thresholds were used in these experiments, T_{WTMM}, T_M and T_L. $T_{WTMM} = 0.6$, separates high curvature points from insignificant singularities which could be present in the fish boundaries due to noise introduced during the acquisition or the contour extraction processes. $T_M = 0.2$ and $T_L = 20$, are the tolerated errors on the magnitude and the location of the wavelet transform modulus maxima. We allowed an error of $\delta\gamma = 5\%$ on the bounding box.

In order to have reliable results from human evaluation, a query image and a subset of 50 images from the database (containing all the similar images to the query) were given to twelve persons. These persons were asked to rank the top 20 images according to their similarity to the query image. The set of images was randomly shuffled prior to the test. After the evaluation, each candidate image was assigned a final score, computed as the mean of the scores assigned by 10 of

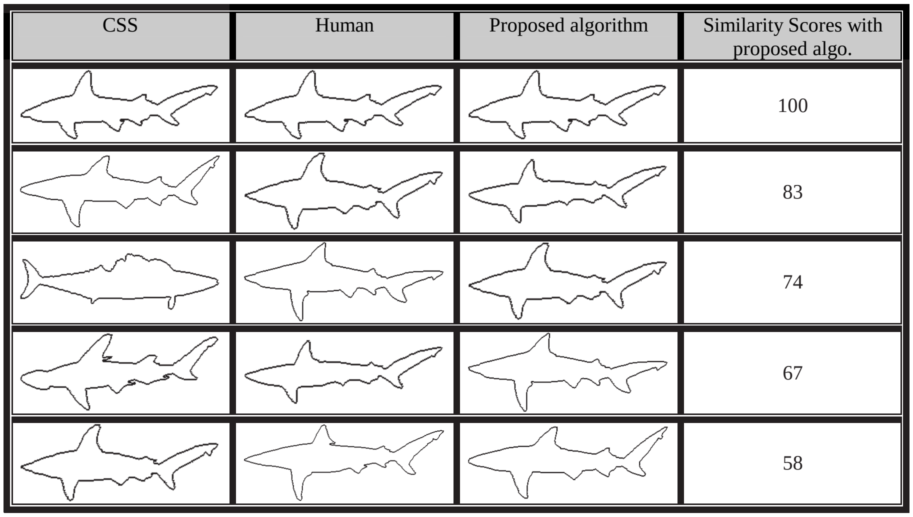


Figure 4.22: Query fish is located in the top of each column, the rest are the best candidates obtained by each method.

these persons to that image. The lowest and highest ranks are removed.

The shape of the neighborhood used is specified by the errors on the location and the magnitude and their relation. In our experiments we used a rectangular neighborhoods which puts no constraints on the values the magnitude and the location can take within the tolerated rectangle. One intuitive relation could be $T_M + T_L \leq Constant$, to allow a larger error on the location when the magnitude error is small and vise versa.

From Figure 4.22, one can easily see that the proposed algorithm produces results, which are closer to those given by the human users test.

Figure 4.23 shows the retrieval results for a query in the MPEG-7 test set B of CE-1 (1400 contours in 70 categories with 20 contours of the same object in each category) with the image “rat-01” shown in the first position of the figure, here 19 out of 20 images were retrieved as the most similar objects in the database.

Properties of this Technique

The proposed similarity estimation technique tries to mimic the human way of comparing two polygons. This approach is invariant to:

- Translation: since the orientation profile of the contour is independent of the location.

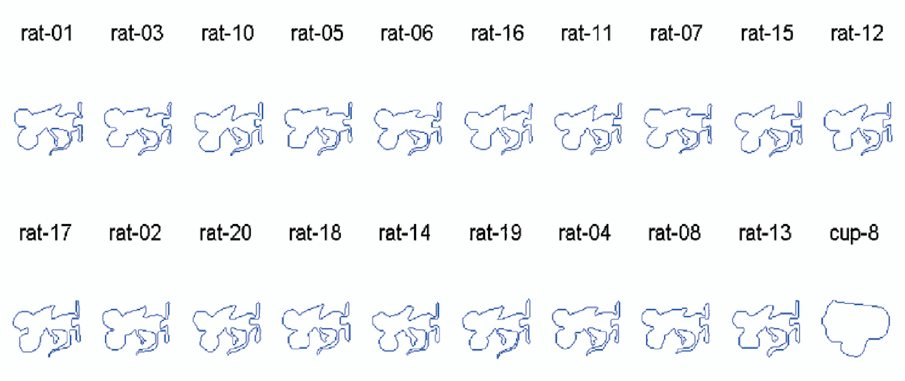


Figure 4.23: Results of a query with a sample image from the MPEG-7 shape test set B.

- Rotation and starting point change: the rotation of the shape or the change of the starting point induce circular shift of the maxima of the wavelet decomposition. Therefore, applying a circular shift to the feature vector until the largest maxima of the query and candidate shapes coincide solves these problems.
- Scale: achieved by using the same number of points to represent the orientation profile and the normalizing of the contour length to one.
- Noise corruption: achieved by considering only the WTMM larger then a fixed threshold.
- Flipping: the problem of mirrored images can be solved by replacing the maxima position with the relative position from the largest maxima, however this may create problems if we have partial symmetry around the largest maxima.
- Partial occlusion: the proposed similarity measure is not very sensitive to the appearance or elimination of few HCP; therefore it is robust to partial occlusions.

4.2.2 Topological Features

In this section, we propose another robust matching algorithm, which tracks the boundary and computes its orientation profile as in the previous algorithm. The wavelet transform of the orientation profile is computed for dyadic scales from 2^1 to 2^6. WTMM is then computed and scaled with the global maxima at the same scale. Only those WTMMs greater than a certain threshold are taken as valid candidates. These events are tracked in the lower successive levels to find their exact locations. This procedure is repeated from level six to two.

Feature extraction and Matching

As mentioned before the aspect ratio is used to filter out the majority of the objects. The remaining candidates go through the second step of the retrieval process, where a set of other low-level features are extracted and compared to those of the query image. These low-level features are the angles subtended at the corner points, distances from the centroid, distance ratios of the adjacent sides and their locations on the boundary.

Algorithm:

1. Select candidate objects with similar aspect ratios as the query object (narrowing the search space),
2. locate the important high curvature points based on WTMM using the approach in [110], for each level of the wavelet decomposition,
3. extract a set of low-level features from the polygonal approximation of the contour,
4. Compute a similarity score, at each level, between the query image and the image in the database,
5. The final similarity score is computed as the mean of the scores at each level.

Steps 3, 4 and 5 are carried out as follows: At each wavelet decomposition level l , we characterize the query image contour with four vectors, θ_l^q, D_l^q, R_l^q and L_l^q, where $\theta_l^q = [\theta_{l1}^q \theta_{l2}^q ... \theta_{lm}^q]$ is the vector containing the angles at the contour points corresponding to detected modulus maxima, $D_l^q = [D_{l1}^q D_{l2}^q ... D_{lm}^q]$ is the vector containing the distances of these points from the centroid of the contour, $R_l^q = [R_{l1}^q R_{l2}^q ... R_{lm}^q]$ is the vector of ratios of the vertices sides and $L_l^q = [L_{l1}^q L_{l2}^q ... L_{lm}^q]$ is a vector containing the locations of the vertices on the normalized contour. Similarly, we characterize the candidate image contour with four vectors, θ_l^c, D_l^c, R_l^c and L_l^c, of length n each.

Figure 4.24 (a) shows a typical database image contour and Figure 4.24 (b) shows the feature to be extracted in the framework of polygonal approximation. Here, C_g is the centroid of the contour, θ is the angle, d_{C_g} is the distance from the centroid and $d2/d1$ is the sides ratio.

Let K be the number of valid matched high curvature points at level l between the query and candidate object contours. A valid match is a pair of points with corresponding errors under certain thresholds $\theta_{err} \leq T_\theta, D_{err} \leq T_D, R_{err} \leq T_R, L_{err} \leq T_L$. The similarity score at level l, for $l = 2, ..., 6$, is defined as:

$$s_l = \frac{2K}{m+n} \times 100 - \xi, \tag{4.18}$$

where $\xi = \alpha \times mean(\theta_{err}, D_{err}, R_{err}) + \beta L_{err}$, and $\theta_{err}, D_{err}, R_{err}$ are the relative percentage errors on the angle, distance and distance ratio, while L_{err} is the absolute error on the location. The first term in the similarity score gives an indication of the number of points matched, while the second part provides information on how good the match is. Level one is not considered in the matching process in order to reduce the effect of noise. The overall similarity score is then $S = mean(s_l),$ for $l = 2, ..., 6$.

To initialize the algorithm, we consider the five most prominent points of the WTMM from each of the query and the target image. We then find the best matching pair of points using the same similarity score as in Equation 4.18. These are used as starting points for the matching process.

Experimental Results

In these experiments, we used 1130 fish contour images. The query objects are shown in Figures 4.25 and Figure 4.27 and sample retrieval results are presented in Figures 4.26 and Figure 4.28. The retrieval results are the five most similar images retrieved using the CSS algorithm [51], the proposed algorithm and a group of users. We found the threshold values $T_\theta = 35$, $T_D = 25$, $T_R = 20$ and $T_L = 0.065$ acceptable. For similarity score computation (according to Equation 4.18) we selected $\alpha = 0.2$ and $\beta = 100$. These values are selected experimentally so that the contribution of ξ to the similarity score does not exceed that of a pair of matching vertices.

In order to have reliable results from human evaluation, a query image and a subset of 50 images from the database (containing all the similar images to the query) were given to twelve persons. These persons were asked to rank the top 20 images according to their similarity to the query image. The set of images was randomly shuffled prior to the test. After the evaluation, each candidate image was assigned a final score, computed as the mean of the scores assigned by 10 of these persons to that image. The lowest and highest ranks are removed.

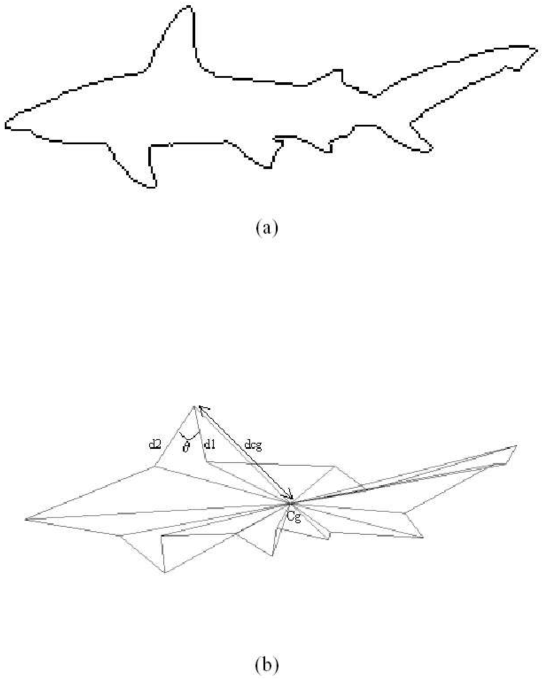
From the above figures, one can easily see that the proposed algorithm produces results which are often very close to those produced by human users. The produced results are clearly better than those obtained by CSS. Furthermore, the use of relatively few high curvature points and dyadic wavelet decomposition guarantees high computational efficiency.

4.2.3 Multi-scale Topological Features

In this approach, the shape feature vector is defined as:

$$F = \{(s_1, \theta_1, a_1, r_1) \ldots (s_i, \theta_i, a_i, r_i) \ldots (s_n, \theta_n, a_n, r_n)\} \tag{4.19}$$

where s_i is the largest scale at which the vertex P_i was detected, $(s_i \geq 2^4)$, θ_i is the angle at the vertex P_i, a_i is the logarithm of the ratio of consecutive segments



(a)

(b)

Figure 4.24: (a) Fish contour image, and (b) shows the features extracted for polygonal approximation at scale 2^4.

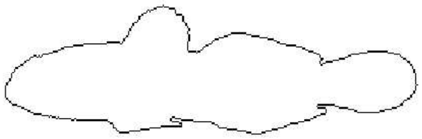


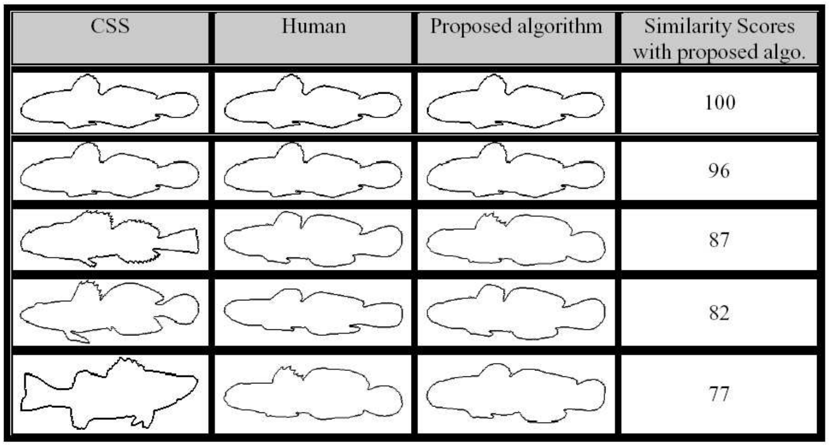
Figure 4.25: Query image.

and r_i is the logarithm of the ratio of the length of sides at each vertex to the overall perimeter length L:

$$a_i \overset{def}{=} \log\left(\frac{dist(P_i, P_{i+1})}{dist(P_{i-1}, P_i)}\right) \tag{4.20}$$

$$r_i \overset{def}{=} \log\left(\frac{dist(P_i, P_{i+1}) + dist(P_{i-1}, P_i)}{L}\right). \tag{4.21}$$

For closed contours, the feature vector F is stored in a linked list, thus when $i = 1$ in Equations (4.20) and (4.21), P_{i-1} becomes P_n and P_{i+1} becomes P_1 when $i = n$. Note that a reconstruction of an approximation of the shape is possible



CSS	Human	Proposed algorithm	Similarity Scores with proposed algo.
			100
			96
			87
			82
			77

Figure 4.26: Results with the query image of Figure 4.25.

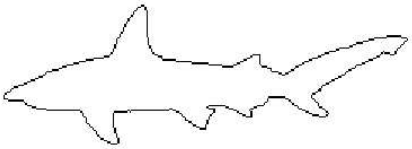
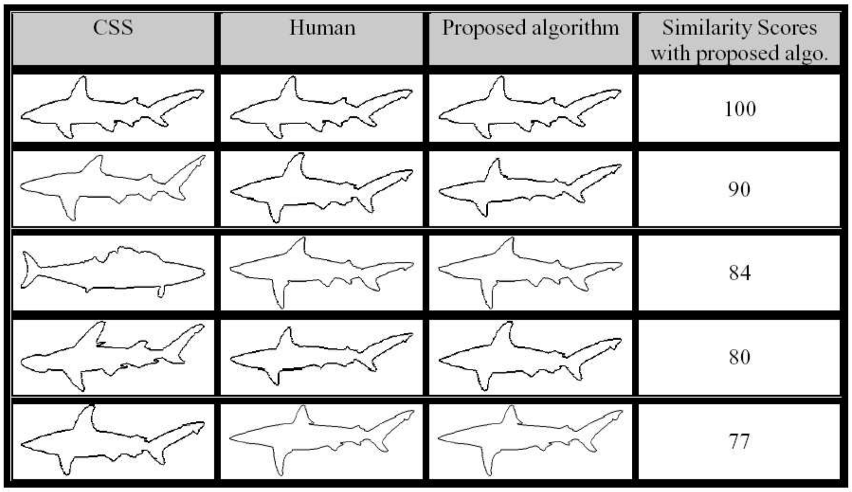


Figure 4.27: Query image.



CSS	Human	Proposed algorithm	Similarity Scores with proposed algo.
			100
			90
			84
			80
			77

Figure 4.28: Results with the query image of Figure 4.27.

using the shape feature vector defined above. For most applications, there is no need to use the very fine details of the contour. Therefore, we limit the vertices used in the representation to only those HCP appearing at scales $s \geq 2^4$, ignoring the fine details on the contour, which may be due to noise and could easily fool the matching algorithm discussed next. This also results in a significant reduction in the size of the feature vectors. Moreover, it ensures that the shape representation is stable and robust to noise. Since the curvature function is used to extract the feature vector, translation invariance is achieved automatically. The representation is also robust to scale as discussed above and rotation and change in starting point result in simple circular shift of the elements in the shape feature vector, which are efficiently dealt with in the matching algorithm discussed below.

Matching Algorithm

In this section, we present the matching algorithm used with the shape feature vector described above. The basic idea behind the algorithm is the following: if we want to compare two polygons, we match their vertices based on their scale, angle, ratio of consecutive segments and ratio to the overall length, $(s_i, \theta_i, a_i, r_i)$ and then exploit the information about the succession of the vertices in each polygon to align the matched vertices such that the correct correspondence between the two sets of vertices is maximized. The allowed transformation from one set to another is circular shift of the feature vectors.

We denote the feature vector of the query shape Q by

$$F^Q = \{(s_1^Q, \theta_1^Q, a_1^Q, r_1^Q) \ldots (s_i^Q, \theta_i^Q, a_i^Q, r_i^Q) \ldots (s_m^Q, eta_m^Q, a_m^Q, r_m^Q)\}, \quad (4.22)$$

and the feature vector of each candidate shape C by

$$F^C = \{(s_1^C, \theta_1^C, a_1^C, r_1^C) \ldots (s_i^C, \theta_i^Q, a_i^C, r_i^C) \ldots (s_n^C, \theta_n^C, a_n^C, r_n^C)\}. \quad (4.23)$$

1. Create the match matrix M of size (m, n) containing the list of all possible matches between the vertices in Q and in C. For convenience, always arrange M so that $m \leq n$, this can be done by simple transposition. Formally, every entry $M_{i,j}$ is calculated as follows:

$$\begin{aligned} M_{i,j} = min(s_i^Q, s_j^C), \text{ if } \quad & ((\theta_i^Q - \theta_j^C)_{mod(360)} \leq \Theta_{Min}) \quad (4.24) \\ & \text{and } (|a_i^Q - a_j^C| \leq A_{Min}) \\ & \text{and } (|r_i^Q - r_j^C| \leq R_{Min}) \\ M_{i,j} = 0, \text{ Otherwise.} & \end{aligned}$$

   This means that if the angles at vertices P_i^Q and P_j^C are within a reasonable range (Θ_{min}) as well as the ratios of consecutive segments and those of the lengths of the corners are close enough (A_{min} and R_{min} respectively), the match value is the the minimum common scale at which the two vertices first appear. The values of the "tolerated" deviations Θ_{min}, A_{min} and R_{min} are experimentally adjusted to account for the size and nature of the database. That is, if the database consists of highly "similar" shapes, then those values can be made small enough to allow only tight matches, otherwise, if the database is small or contains shapes of hybrid structures then those parameters are set to higher values to allow loose matches. To allow the matching of heavily occluded shapes, then R_{min} should be set to a large value. In the experiments, the values of Θ_{min}, A_{min} and R_{min} were set to $15°$, 0.45 and 0.3, respectively. With these values, good results are obtained when matching full shapes. For an enhanced performance over the matching of heavily occluded shapes, the value of R_{min} is set to 0.9.

2. for two *identical* shapes the match matrix M will contain a full diagonal of entries containing the scales s_i. The diagonal will be wrapped if the starting point is not the same:

$$M = \begin{bmatrix} \times & s_2 & \times & \times \\ \times & \times & s_3 & \times \\ \times & \times & \times & s_4 \\ s_1 & \times & \times & \times \end{bmatrix} \tag{4.25}$$

The example above shows the match matrix M for two identical shapes with four vertices, with a shift of one vertex at the starting points. The $\times$'s denote "wrongly" matched or more usually, unmatched vertices, so the value at $\times$ is usually 0.

To find the best matching score among the diagonal entries of M, a matrix M_D is constructed such that the "wrapped" diagonal entries of M are the columns of M_D. This is always possible since $m \leq n$. For the example matrix M in Equation (4.25), M_D is given as

$$M_D = \begin{bmatrix} \times & s_2 & \times & \times \\ \times & s_3 & \times & \times \\ \times & s_4 & \times & \times \\ \times & s_1 & \times & \times \end{bmatrix} \tag{4.26}$$

3. For every column j in M_D calculate the score of match C_j of that column as follows:

```
for every $i = 1 \ldots n$
    *if* $M_D(i,j) \neq 0$ (contains a matched vertex)
       $C_j(i) = M_D(i,j)$
   *else* (does not contain matched vertex)
       $C_j(i) = R_i(j_{closest})/(|(j+n) - j_{closest}| + 1);$
```

where $R_i(j_{closest})$ is defined as the closest nonzero element (in both directions) in the row $R_i = [M_D(i,:)M_D(i,:)M_D(i,:)]$ to the point $R_i(j+n)$. The idea here is when no match is found on a given vertex, the consecutive or previous vertex that matches is used. The number of skipped vertices penalizes the match. Figure 4.29 illustrates how $C_j(i)$ is calculated when $M_D(i,j) = 0$.

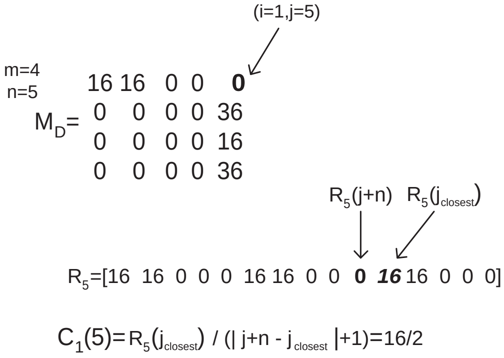

(i=1,j=5)
m=4
n=5
M_D=
16 16 0 0 0
0 0 0 0 36
0 0 0 0 16
0 0 0 0 36
R_5(j+n)
R_5(j_closest)
R_5=[16 16 0 0 0 16 16 0 0 0 16 16 0 0 0]
C_1(5)=R_5(j_closest) / (| j+n - j_closest |+1)=16/2

Figure 4.29: Calculation of $C_j(i)$

Finally, the matching score of each column j is calculated as:

$$C_j = \frac{\sum_{i=1}^{n} C_j(i)}{\sum_{i=1}^{m} s_i} \tag{4.27}$$

4. The column with the maximum score is selected as the best matched diagonal and the overall matching score will be given by:

$$Score = \max(C_j) \quad j = 1 \ldots n. \tag{4.28}$$

Experimental Results

The algorithm was tested on the database of 1130 contours of marine animals. Figures 4.30 through 4.35 show the retrieval results on different shape contours. The image shown on the top left corner of each figure is the query, thus its score is 1. The rest of the shapes are the most similar ones according to the similarity defined above. From these figures, it can be safely concluded that the retrieval results reflected similarity of shapes very sharply: most of the first matches are similar to the query image and their ranking according to their matching score are in concordance with our perception of similarity. This shows that a concise combination of the location and scale information in the matching algorithm can be very efficient in reflecting similarity of shapes. The system is insensitive to translations, rotations and scaling. Figure 4.31 shows the retrieval results for a query consisting of a rotated and scaled shape. The retrieved images were the same as the ones resulting from the search using the original query (Figure 4.30). This matching algorithm performed particularly well on heavily occluded shapes. In Figure 4.34, the query image was a fish tail ($\geq 50\%$ occlusion). The best retrieved image was the original fish to which the tail belonged, then the rest of the retrieved images contained tails similar to the query.

Figure 4.35 presents the normalized similarity scores for the same query image achieved by the Fourier Descriptors method, the CSS technique and by the proposed technique. The scores were normalized between 0 and 1. The best match (not the exact match) was given the score 1. The scores performed by our algorithm had the steepest descent, which means that they isolated efficiently the most similar shapes from the rest of the database. Knowing that the database used in these experiments did not contain more than 10 images of the same marine creature confirms the usefulness of our shape representation and matching algorithm in large databases.

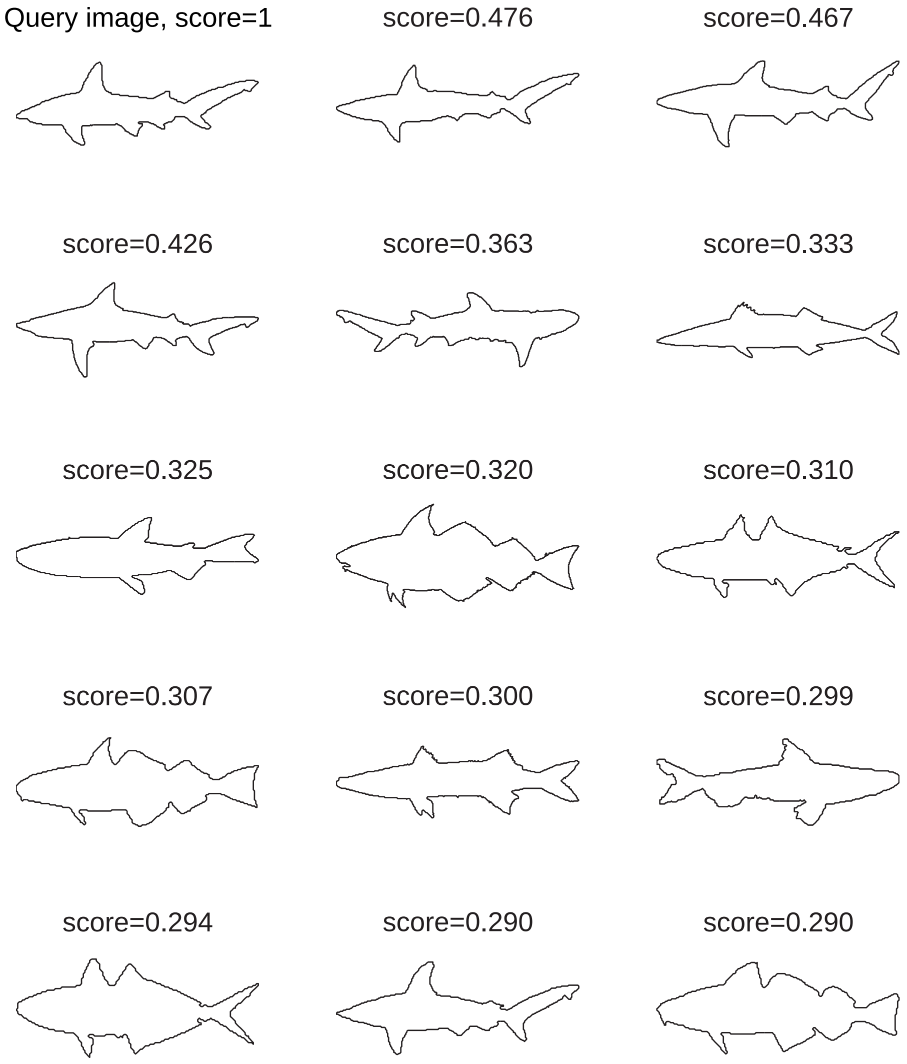


Figure 4.30: Example 1 of search results using WTMM-based technique.

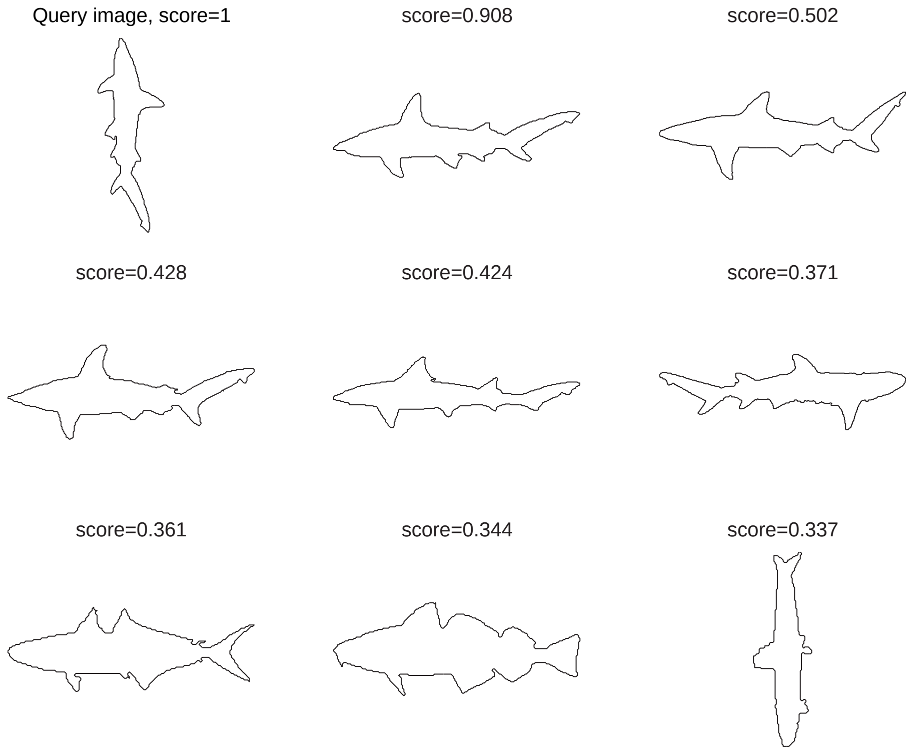
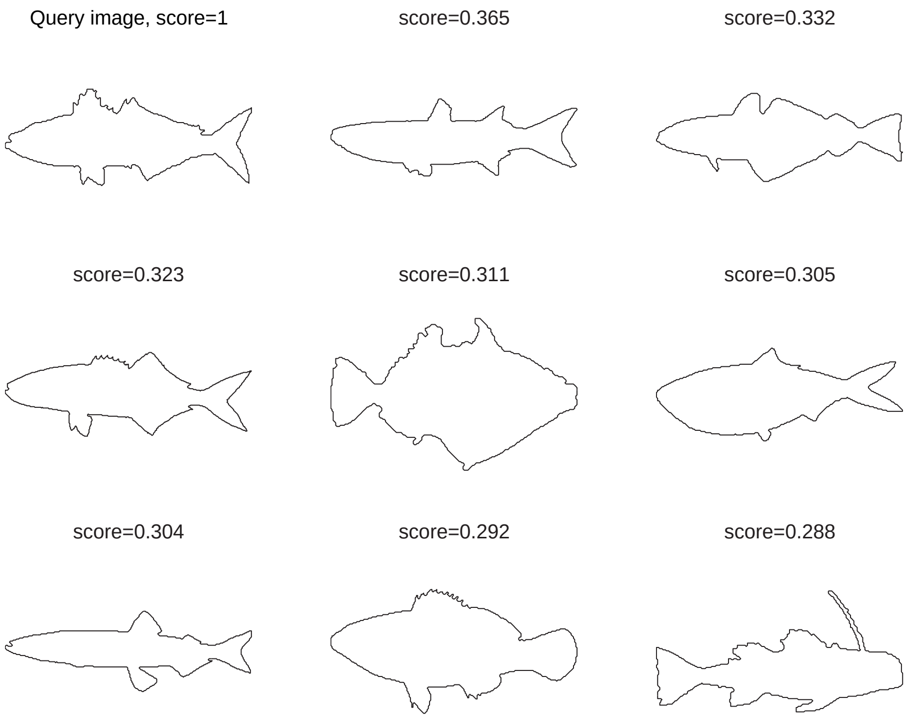


Figure 4.31: Example 2 of search results over a rotated shape using WTMM-based technique.



Query image, score=1

score=0.365

score=0.332

score=0.323

score=0.311

score=0.305

score=0.304

score=0.292

score=0.288

Figure 4.32: Example 3 of search results using WTMM-based technique.

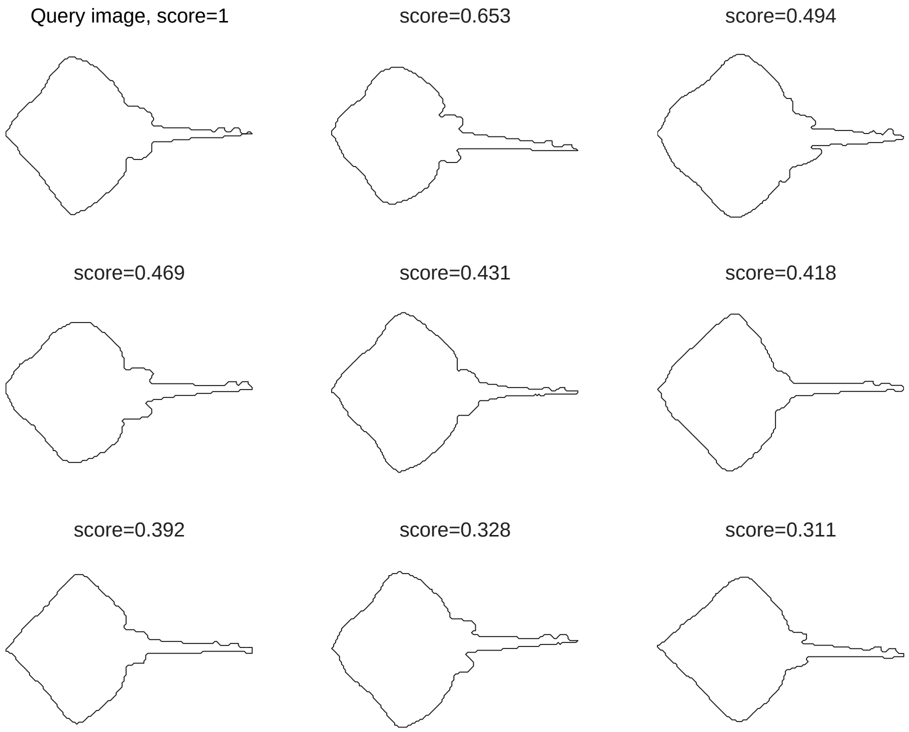


Figure 4.33: Example 4 of search results using WTMM-based technique.

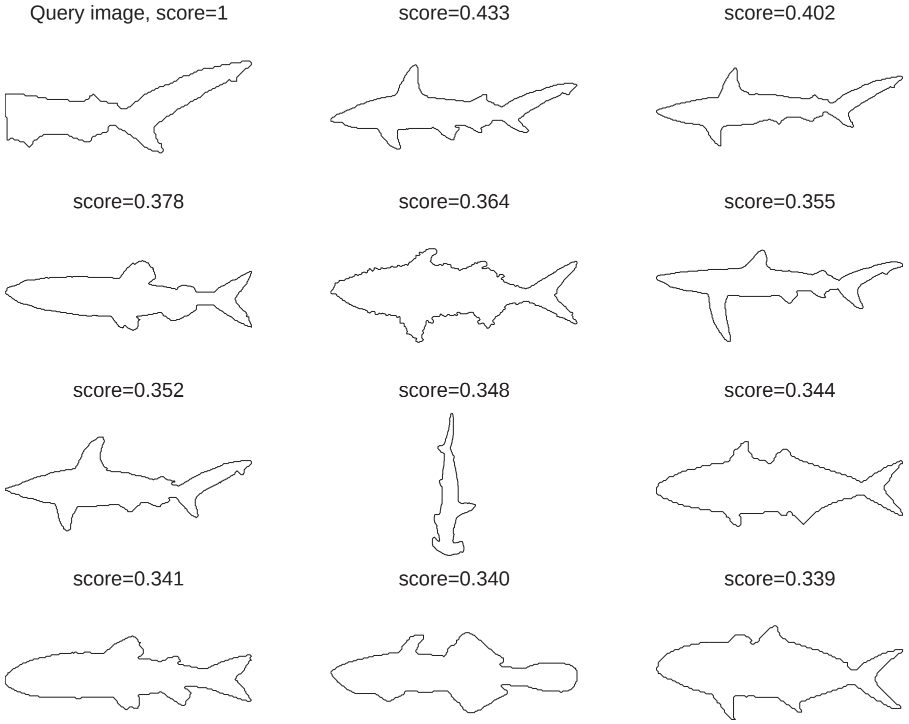


Figure 4.34: Retrieval result on a heavily occluded contour using the WTMM-based technique.

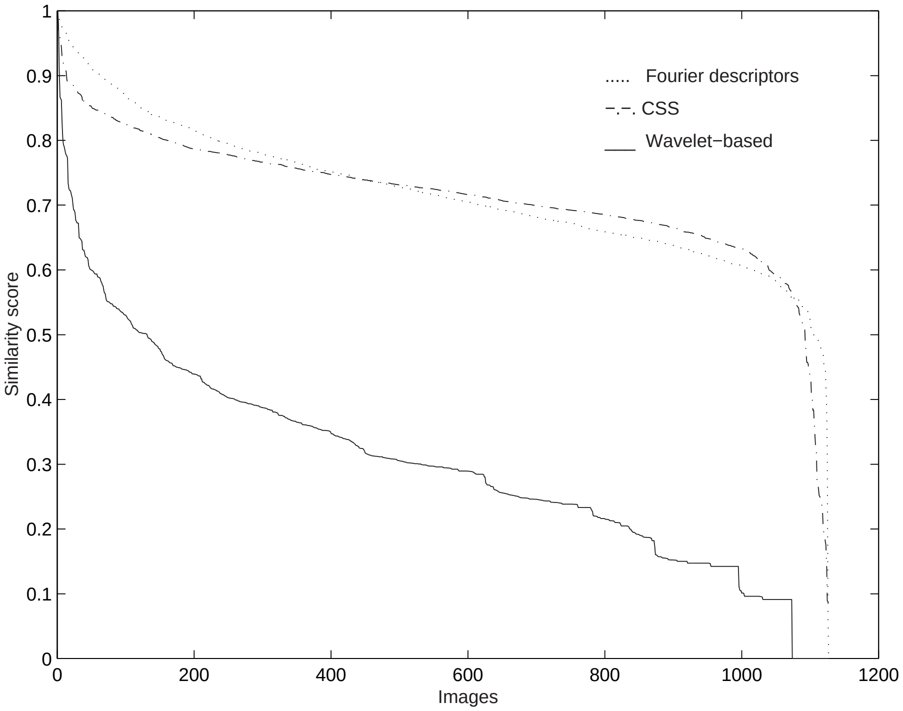


Figure 4.35: Plot of the similarity scores for the Fourier descriptors method, CSS technique and our technique respectively.

Chapter 5

Region-Based Shape Features

In this chapter, a region-based shape similarity estimation framework based on ordinal correlation is proposed. To illustrate the concordance of similarity scores obtained via this approach with human perception of object similarity, we apply this technique to the problem of performance evaluation of segmentation algorithms [90, 153]. We show in this way that the scores obtained for the segmentation mask are in line with our perception of the quality of the segmentation results when compared to a groundtruth mask. The problems of CBIR and segmentation performance evaluation may seem of different nature at first sight. They are very closely related however, if we consider the specific context of a foreground background segmentation scenario. Comparing the segmentation mask obtained automatically to a ground truth mask manually created is the same operation one needs to perform when comparing the query object contour to those of objects in the database. Moreover, if the segmentation results are going to be used in a scenario where the end user is a human, such as interactive TV or computer games, the similarity measure has to agree with the human visual perception of objects similarity. This is the same restriction imposed on the similarity measures used in CBIR.

The rest of the chapter is organized as follows: Section 5.1 gives an overview of the Angular Radial Transform (ART), Shape Descriptor (SD) adopted in MPEG-7. Section 5.2 presents an overview of the Ordinal Correlation Framework (OCF) followed by a detailed description of each of its processing steps. Experimental results for both segmentation performance evaluation and CBIR applications are presented in section 5.3, using segmentation masks obtained by the COST Analysis Model (COST AM) segmentation algorithms [2, 41] and a subset of the MPEG-7 shape test set.

5.1 Angular Radial Transform Descriptor

The Angular Radial Transform (ART) shape descriptor is adopted by MPEG-7, as a region-based shape descriptor, takes into account all the pixels of the objects and not only those representing the contour. Therefore, this category of descriptors can be used to represent complex object shapes possibly with multiple regions and holes, such as those in Figure 5.1 . The ART descriptor is computed by decomposing the shape image into orthogonal 2D basis functions.

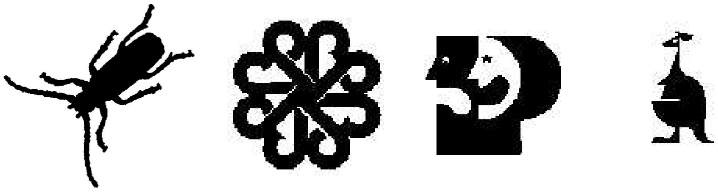


Figure 5.1: Example of complex shapes in which region-based descriptors are applicable.

The ART is the orthogonal unitary transform defined on a unit disk that consists of all the orthogonal sinusoidal basis functions in polar coordinates. The ART coefficients are computed as:

$$F_{nm} = (V_{nm}(\rho,\theta), f(\rho,\theta)) = \int_0^{2\pi}\int_0^1 V^*_{nm}(\rho,\theta) f(\rho,\theta)\rho d\rho d\theta, \tag{5.1}$$

where F_{nm} is an ART coefficient of order n and m, $f(\rho,\theta)$ is an image function in polar coordinates and $V_{nm}(\rho,\theta)$ is the ART basis function separable along the angular and radial directions, that is:

$$V_{nm}(\rho,\theta) = A_m(\theta)R_n(\rho), \tag{5.2}$$

To achieve rotation invariance, an exponential function is used for the angular basis function,

$$A_m\theta = \frac{1}{2\pi}exp(jm\theta), \tag{5.3}$$

The radial function is defined by cosine function,

$$R_n(\rho) = \begin{cases} 1 & n = 0, \\ 2cos(\pi n\rho) & \text{Otherwise} \end{cases} \tag{5.4}$$

The 2D ART basis functions, $V_{nm}(\rho,\theta)$, are complex functions. In Figure 5.2, the real parts of the first 36 basis functions ($n < 3$, $m < 12$) are shown, their imaginary parts being similar except for quadrature phase difference.

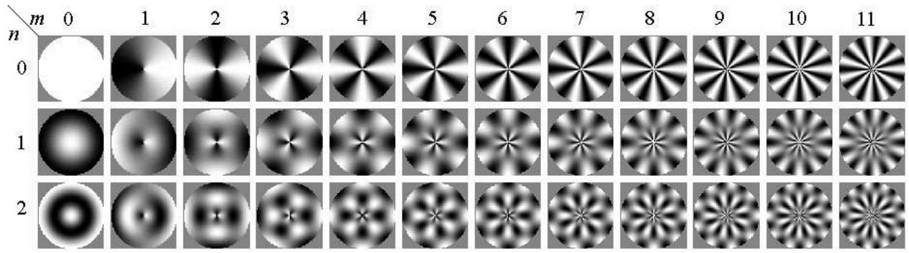


Figure 5.2: Real part of ART basis functions.

The ART descriptor is defined as a set of normalized magnitudes of complex ART coefficients, which guarantee rotation invariance. For scale normalization ART coefficients are divided by the magnitude of the ART coefficient of order $n = 0$, $m = 0$, which is not used as a descriptor element. Each one of the 35 coefficients is quantized to four bits. The similarity measure of the ART descriptor is the L_1 norm.

5.2 Ordinal Correlation Framework

Images in the target applications are representing either: a single object outline or a segmentation mask. Therefore, we will not discuss how to obtain the contours or the segmentation masks. Our goal is to compute a similarity score between any two shapes or two segmentation masks. The proposed method operates in three steps: alignment, binary to multilevel image transformation and similarity evaluation. The alignment step is not needed in the case of the segmentation performance evaluation, since we are comparing segmentation masks corresponding to the same image/frame.

Once the objects outer boundaries are aligned, the binary images containing the objects shapes are transformed into multilevel images through distance transformation (DT) [28, 29]. The obtained images are then compared using a version of the ordinal correlation measure introduced in [27, 132]. This ordinal measure estimates the similarity between gray-level images based on the ordinal correlation of their pixel values. An approach based on distance transform was used in [43] for the automatic recognition of road signs for an onboard vision system. The distance transform was applied to a single image and summed the distances of the pixels corresponding to the pixel positions of the patterns models they are looking for. Later they extended their work to pedestrian detection [42]. Applying the distance transform only on the candidate images makes the similarity estimation operation asymmetric, thus $d(C_1, C_2) \neq d(C_2, C_1)$. This is undesirable in the context of CBIR, therefore in our approach DT is applied to both query and candidate contours. In the rest of this section a detailed description of the approach is given.

5.2.1 Object Alignment Based on Universal Axes

The alignment is performed by first detecting three universal axes (UA) [77], with the largest magnitude, for each shape; then orienting the shape in such a way that these axes are aligned in a standard way for all the objects to be compared.

In this implementation of the universal axes determination algorithm we use the version number $\mu = 2l$ [77]. The steps of the alignment algorithm are detailed below.

Step 1: Translate the coordinate system so that the origin becomes the center of gravity of the shape S.

Step 2: Compute:

$$x_\mu^{(l)} + iy_\mu^{(l)} = \int\int_S (\sqrt{x^2+y^2})^\mu (\frac{x+iy}{\sqrt{x^2+y^2}})^l dxdy = \int\int_S r^\mu e^{il\theta} dxdy, \quad (5.5)$$

and using normalized counterpart (called universal axes)

$$\tilde{x}_\mu^{(l)} + i\tilde{y}_\mu^{(l)} = \frac{x_\mu^{(l)} + iy_\mu^{(l)}}{\int\int_S (\sqrt{x^2+y^2})^\mu dxdy}, \quad (5.6)$$

for $l = 1, 2, 3$.

Step 3: Compute the polar angle $\Theta_\mu \in [0, 2\pi]$ so that

$$R_\mu e^{i\Theta_\mu} = |x_\mu^{(l_1)} + iy_\mu^{(l_1)}| \quad (5.7)$$

with R_μ being the magnitude of $x_\mu^{(l_1)} + iy_\mu^{(l_1)}$. l_1 is the number of axes needed to align an object.

Step 4: The directional angles of the l_1 universal axes of the shape S are computed as follows:

$$\theta_j = \frac{\Theta_\mu}{l_1} + (j-1)\frac{2\pi}{l_1}, \quad \text{for} \quad j = 1, 2, ..., l_1. \quad (5.8)$$

In our implementation we used $l_1 = 3$, See Figure 5.3, since for $l_1 = 2$ the two universal axes orientation will verify $\theta_2 = \theta_1 + \pi$. Therefore, they can not be used alone to determine if an object is fliped around the direction they define or not.

Step 5: Once the three universal axes are determined we rotate the contour so that the most dominant UA (UA with the largest magnitude) will be aligned with the positive x-axis, see Figure 5.3.

Step 6: Then, if the y-component of the second most dominant UA is positive, we flip the contour around the x-axis. After this step the objects contours are

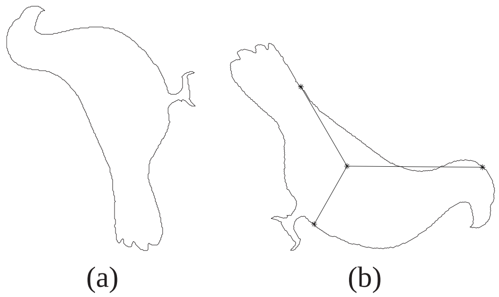


Figure 5.3: (a) The original bird contour before the alignment step, (b) the contour after alignment using three Universal Axes.

assumed to have similar orientation, therefore the next steps should not worry about rotation or flipping of the contour.

To illustrate the alignment performance we applied it to the set of contours in Figure 5.4. The results of the alignment are presented in Figure 5.5. It can be noticed that this alignment scheme solved both problems of rotation and mirroring. This operation is needed since the ordinal-correlation compares corresponding regions (blocks) of the two images, therefore it is not resilient to rotation and mirroring without the contour alignment operation.

5.2.2 Binary to Multilevel Image Transformation

Let S be a shape represented by its boundary C in a binary image. The binary image is transformed into a multilevel (graylevel) image G using a mapping function ϕ, such that the pixel values in G, $\{G_1, G_2, ..., G_n\}$, depend on their relative position to the boundary points $C_1, C_2, ..., C_p$:

$$G_i = \phi(C_k : k = 1, 2, ..., p), \quad \text{for} \quad i = 1, 2, ..., n, \tag{5.9}$$

where C_k is the position of the contour pixel k in the image G. It should be observed that several transformations satisfy this requirement, including any distance transform [29].

As a result of this mapping the information contained in the shape boundary will be spread throughout all the pixels of the image. Computing the similarity in the transform domain will benefit from the boundary information redundancy in the new image, making it less sensitive to errors in the alignment or contour extraction algorithms. There is no single optimal mapping; different mappings will emphasize different features of the contour.

$$G_i = \begin{cases} |V_0 \pm d(P_i, C)|, & \text{if} \quad d(P_i, C) \leq T_h \\ 0, & \text{Otherwise.} \end{cases} \tag{5.10}$$

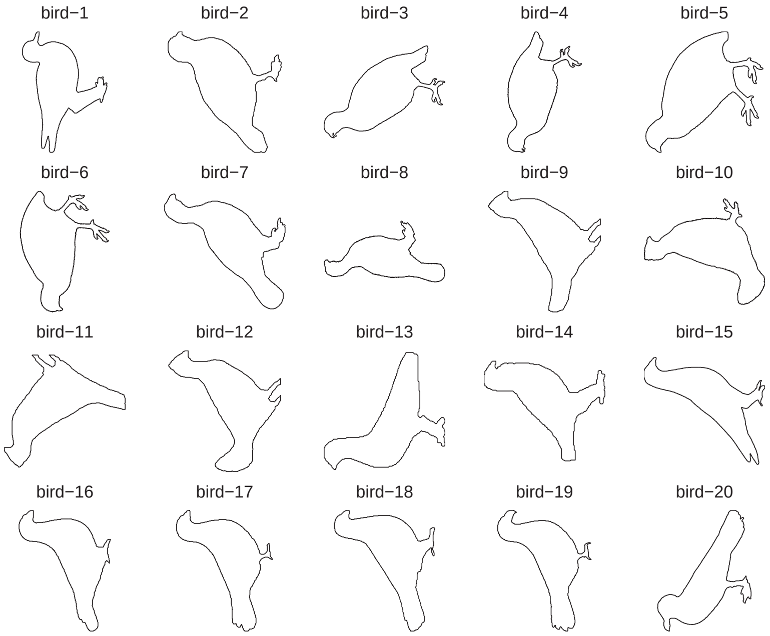


Figure 5.4: Bird contours from the MPEG-7 shape test set B.

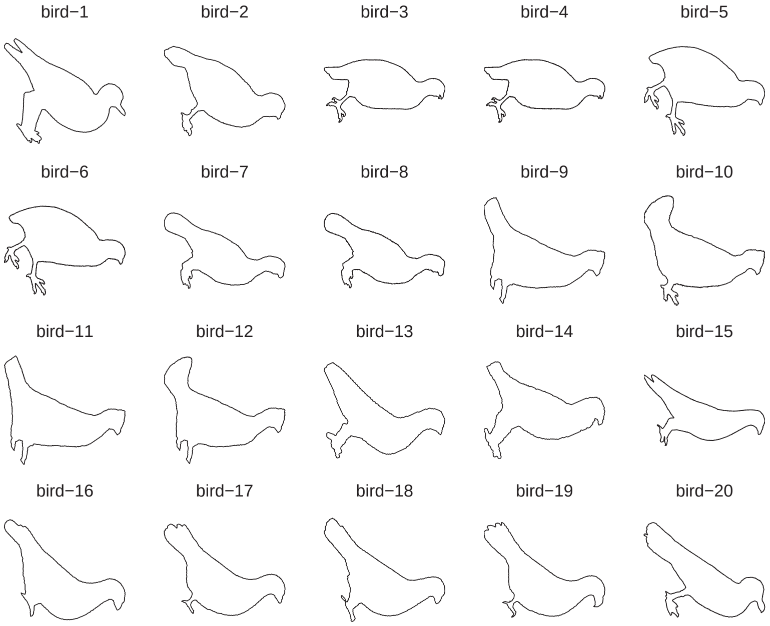


Figure 5.5: The contours in Figure 5.4 after alignment.

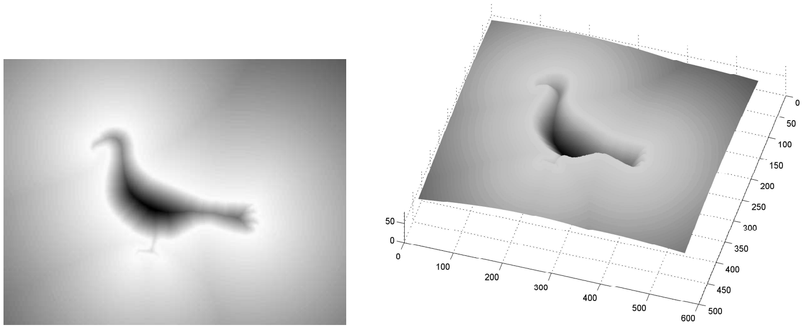
for $i = 1, 2, ..., n$ and V_0 is the pixel value on the contour points and $d(P_i, C)$ is the distance from any point P_i in the image to the object boundary C.
We distinguish however two special cases:

- in the first $V_0 > 0$, thus the larger the distance $d(P_i, C)$ from the contour points, the smaller the new pixel value will be. This mapping function emphasizes the detail variations on the object boundary, see Figure 5.6. Applying the distance mapping inside and outside the contour can lead to a better evaluation of segmentation results. One can even assign different weights inside and outside of the contours.

- in the second mapping $V_0 = 0$ thus $G_i = d(P_i, C)$. This mapping when applied inside the contour only it emphasizes the contour skeleton which is a very important feature of a shape, see Figure 5.7.

We implemented both mappings based on the geodesic distance [144]. The metric is integer and its application is done through an iterative wave propagation process [113]. The contour points are considered as seeds during the construction of the distance map.

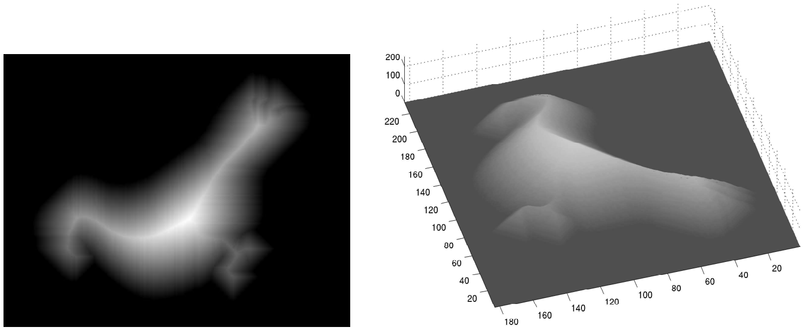
Figure 5.7 (a) represents an example of a distance map generated only inside the shape of a bird contour. Figure 5.7 (b) shows a 3D visualization of this distance map.



(a) (b)

Figure 5.6: (a) The distance map generated for the “bird-19” contour in Figure 5.3 inside and outside the contour, (b) a 3D view of this distance map.

After this transformation the image pixels contain information on how far each pixel is from the contour. Therefore, if we consider a small region in this image map, we can analyze the contour activity in the region. For instance, applying a simple Min operation on the pixel values inside the region tells us how close the region is to the contour. Thus, operating on the distance transform image and



(a) (b)

Figure 5.7: (a) The distance map generated for the "bird-19" contour in Figure 5.3 inside only, (b) a 3D view of this distance map.

using a region-based analysis of the contour activity is a good way of comparing shapes.

5.2.3 Block Based Similarity Evaluation

The evaluation of image similarity is based on the framework for ordinal-based image correspondence introduced in [132]. Figure 5.8 gives a general overview of this region-based approach.

Suppose we have two images, X and Y, of equal size. In a practical setting, images are resized to a common size. The pixel values in the transform images represent distances inside the images. Therefore, the resized transform images pixel values are computed by pixel sub-sampling and scaling of the initial transform images pixel values by the resizing factor.

Let $\{X_1, X_2, ..., X_n\}$ and $\{Y_1, Y_2, ..., Y_n\}$ be the pixels of images X and Y, respectively. We select a number of areas $\{R_1, R_2, ..., R_m\}$ and extract the pixels from both images that belong to these areas. Let R_j^X and R_j^Y be the pixels from image X and Y, respectively, which belong to areas R_j, with $j = 1, 2, ..., m$.

The goal is to compare the two images using a region-based approach. To this end, we will be comparing R_j^X and R_j^Y for each $j = 1, 2, ..., m$. Thus, each block in image X is compared to the corresponding block in image Y in an ordinal fashion. The ordinal comparison of the two regions means that only the ranks of the pixels are utilized. For every pixel X_k, we construct a so-called slice $S_k^X = \{S_{k,l} : l = 1, 2, ..., n\}$, where:

$$S_{k,l}^X = \begin{cases} 1, & \text{if} \quad X_k < X_l \\ 0, & \text{Otherwise} \end{cases} . \tag{5.11}$$

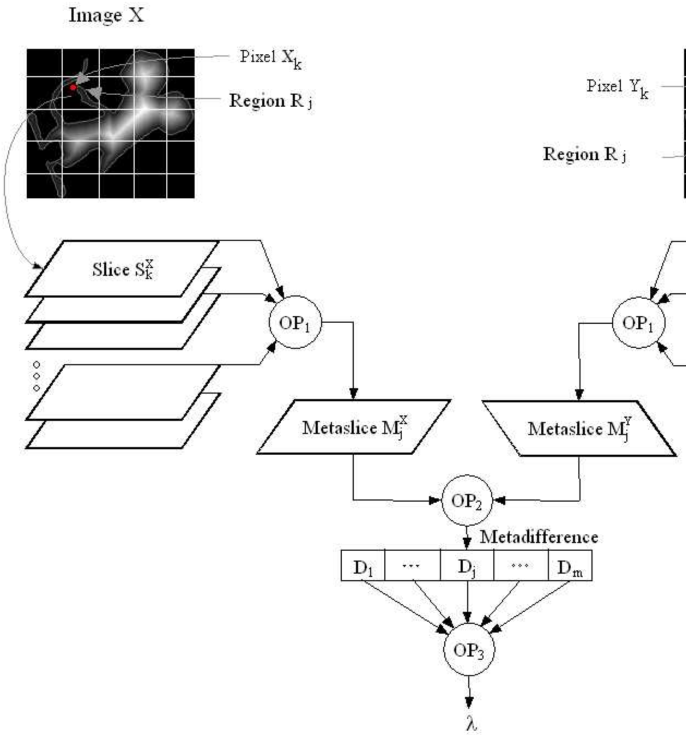


Figure 5.8: The general framework for ordinal correlation of images.

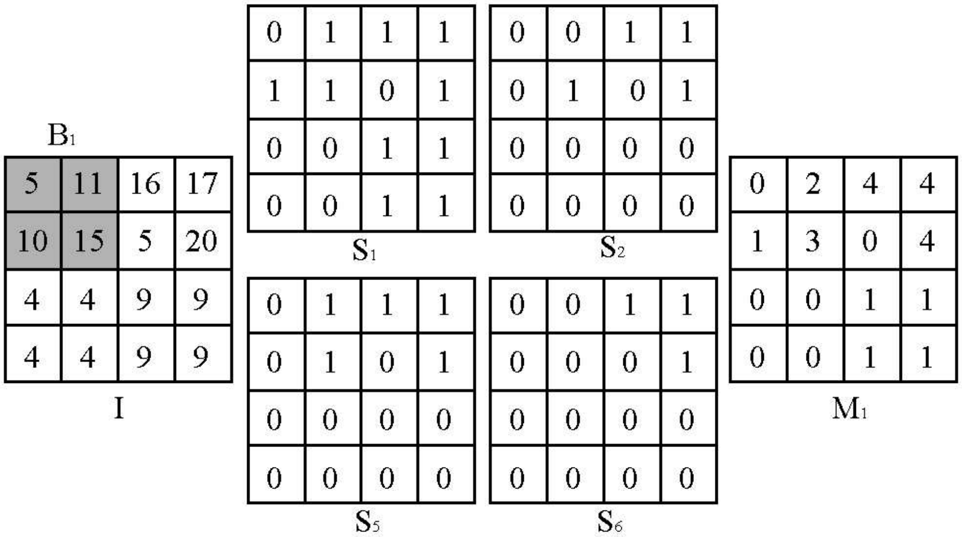
As can be seen, slice S_k^X corresponds to pixel X_k and is a binary image of size equal to image X. Slices are built in a similar manner for image Y as well.

To compare regions R_j^X and R_j^Y, we first combine the slices from image X, corresponding to all the pixels belonging to region R_j^X. The slices are combined using the operation $OP_1(.)$ into a metaslice M_j^X.

Figure 5.9 shows an illustration of the slices and metaslices creation for a 4×4 pixels image and regions of 2×2 pixels. The four slices S_1, S_2, S_5 and S_6 shown in this figure are computed for the four pixels in block B_1. The operation used in this illustration to create the metaslice M_1 is $OP_1(.) = \sum(.))$.

More formally, $M_j^X = OP_1(S_k^X : X_k \in R_j^X)$ for $j = 1, 2, ..., m$. Similarly, we combine the slices from image Y to form M_j^Y for $j = 1, 2, ..., m$. It should be noted that the metaslices are equal in size to the original images and could be multi-valued, depending on the operation $OP_1(.)$. Each metaslice represents the relation between the region it corresponds to and the entire image.

The next step is a comparison between all pairs of metaslices M_j^X and M_j^Y by using operation OP_2, resulting in the metadifference D_j. That is, $D_j = OP_2(M_j^X, M_j^Y)$, for $j = 1, 2, ..., m$. We thus construct a set of metadifferences



I

5	11	16	17
10	15	5	20
4	4	9	9
4	4	9	9

S1

0	1	1	1
1	1	0	1
0	0	1	1
0	0	1	1

S2

0	0	1	1
0	1	0	1
0	0	0	0
0	0	0	0

S5

0	1	1	1
0	1	0	1
0	0	0	0
0	0	0	0

S6

0	0	1	1
0	0	0	1
0	0	0	0
0	0	0	0

M1

0	2	4	4
1	3	0	4
0	0	1	1
0	0	1	1

Figure 5.9: Example of slices and metaslice for a 4×4 image using blocks of 2×2.

$D = \{D1, D2, ..., Dm\}$. The final step is to extract a scalar measure of correspondence from set D, using operation $OP_3(.)$. In other words, $\lambda = OP_3(D)$. It was shown in [132] that this structure could be used to model the well-known Kendall's τ and Spearman's ρ measures [62].

The following is a short description of the operations $OP_k(.), k = 1, 2, 3$ adopted for this measure. Operation $OP_1(.)$ is chosen to be the component-wise summation operation; that is, metaslice M_j is the summation of all slices corresponding to the pixels in block j or in other words, $M_j = \sum_{k:X_k \in R_j} S_k$.

Next, operation $OP_2(.)$ is chosen to be the squared Euclidean distance between corresponding metaslices. That is, $D_j = ||M_j^X - M_j^Y||_2^2$. Finally, operation $OP_3(.)$ sums together all metadifferences to produce $\lambda = \sum_j D_j, \quad \text{for} \quad j = 1, 2, ..., m$. Small values of λ mean similar objects.

Subdividing the image in non-overlapping regions (block-based is the simplest way of splitting the images) and using these blocks to compare the transform images is equivalent to comparing the contours at a scale proportional to the block size. If the transform image is used as a single block, than the distance of the block to the contour will be zero, meaning that we are on top of the contour in the image, which is equivalent to looking to the image at the largest scale. On the other hand if we consider the blocks to be equal to a single pixel each, than all the entries in the metaslice of a pixel will be either all zeros or all ones, meaning that the pixel is on or off the contour. Therefore, the differences D_i represent binary comparison of the pixels of the two images, $D_i = 0$ means both pixels are on the contour or off the contours while $D_i = 1$ means that there is a difference of two pixels status since one is on the contour and the other is not. Therefore in this case the summation of these error measures will give the number of non-corresponding pixels.

Therefore, the variation of the block size means the variation of the scale at which we are comparing the two contours. This is an advantage of this approach over classical ordinal correlation measures, since it is capable of taking into account differences between images at a scale related to the chosen block size.

5.3 Experimental Results

The proposed technique is applied to two important problems: performance evaluation of segmentation algorithms and content-based retrieval of shape images. The experiments performed are presented and their results analyzed in the rest of this section. The first set of experiments, segmentation algorithms performance evaluation, is intended to show the concordance of the similarity measure proposed with our visual perception of similarity between shapes.

5.3.1 Segmentation Quality Evaluation

The objective evaluation of the performance of segmentation algorithms is an important problem [89, 90, 99]. Even when a reference mask is available, comparing two segmentation masks is still a difficult problem. Several factors make such evaluation difficult, among the most important factors is the difficulty to discriminate between many small distributed error segments and a few large ones.

Our shape correspondence technique proposed in Section 5.2, discriminates easily between the two cases of segmentation errors, thanks to two important characteristics of this approach: first its ability to analyze the differences between the two images at a given block size (scale); second the maximum value assigned by the geodesic distance transformation, when applied inside the segments, is proportional to the size of the segment. Therefore, small regions yield small distances inside and therefore will generate pixels with low gray values. This allows discrimination between the two cases.

In this experiment, the segmentation masks resulting from the COST AM versions 5.0 and 5.1 [2, 41], are compared against a reference mask. The COST AM is the Analysis Model from COST 211 [1] a collaborative research forum facilitating the creation and maintenance in Europe of a high level of expertise in the field of video compression and related activities. The COST 211bis action was a project dealing with redundancy reduction techniques applied to video signals. COST 211ter was a follow-on project to this dealing with redundancy reduction techniques and content analysis for multimedia services. COST 211quat is the most recent COST 211 action which continues the work of COST 211ter.

[1] COST stands for “Coopération européenne dans la recherche scientifique et technique”.

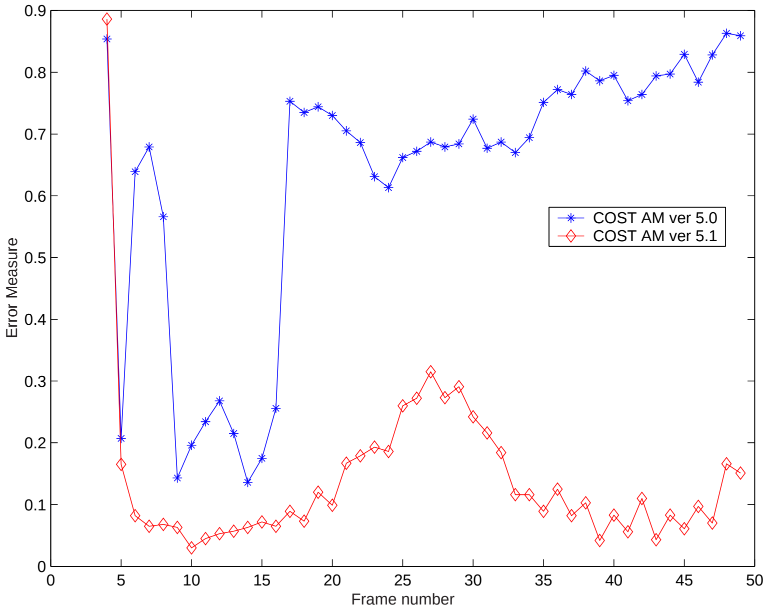


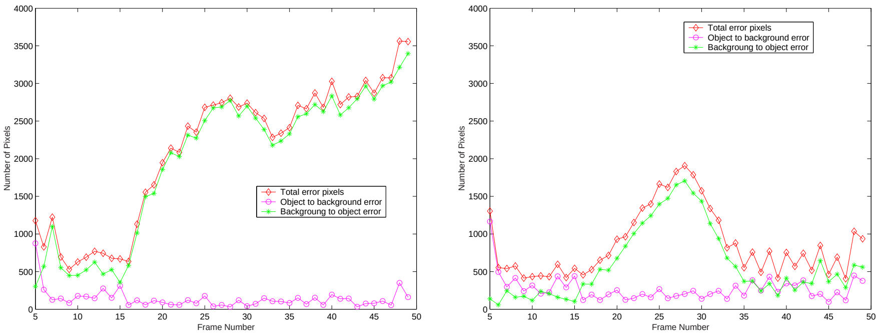
Figure 5.10: The segmentation performance scores for frames 4-49 of the sequence 'Erik', for COST AM ver5.0 and ver5.1.

The plot in Figure 5.10, shows the segmentation performance scores obtained by our technique, for the frames 4-49 of the "Erik" sequence. The plots in Figure 5.11 show quantitative measures of the errors in number of pixels from both COST AM ver. 5.0 and ver. 5.1, respectively. Three different numbers are computed for each frame:

- number of pixels of the background segmented as foreground pixels,
- number of pixels of the foreground segmented as background pixels,
- sum of the two previous numbers.

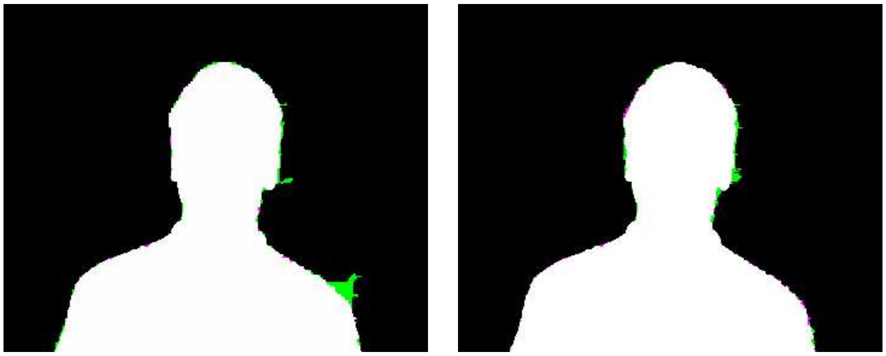
For illustration we present the colored segmentation masks for frames 15 and 20 from the "Erik" sequence in Figures 5.12 and 5.13. The frame pixels are colored as follows:

- black represents the background,
- white is the region where the reference and estimated masks overlap,
- green represents the areas of the background segmented as part of the object,
- purple represents the regions from the object merged with the background.



(a) (b)

Figure 5.11: Plots of the different segmentation errors in pixels for frames 5 to 49, for COST AM: (a) 5.0, (b) 5.1.



(a) (b)

Figure 5.12: The colored segmentation error of frame 15 from the “Erik” sequence, segmented using COST AM: (a) 5.0, (b) 5.1.

It can be easily seen that our segmentation performance scores in Figure 5.10, correlate very well with the variation of the total number of error pixels (size of the error segments). Moreover, it reflects the variation in both types of segmentation errors and inherently resolves the ambiguity between the case of many small error segments and the case of fewer but larger ones. As discussed above, the variation of the block size means the variation of the scale at which we are comparing the two contours. This is an advantage of this approach over classical ordinal correlation measures, since it is capable of taking into account differences between images at a scale related to the chosen block size. When the block is set to a single pixel, our measure reduces to simple quantitative segmentation evaluation based on the number of pixels difference between the groundtruth and the segmentation result.

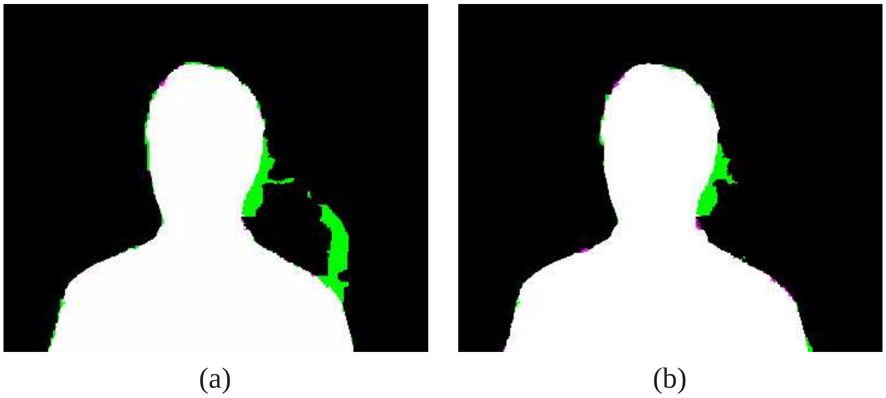


Figure 5.13: The colored segmentation result of frame 20 from the “Erik” sequence, segmented using COST AM: (a) 5.0, (b) 5.1.

5.3.2 Shape Similarity Estimation

The shape similarity estimation experiments were conducted on two sets of 20 images. The two sets are taken from the MPEG-7 CE Shape test set B, which contains 1400 images grouped in 70 categories. These test sets are chosen in such a way as to assess the performance of our technique in estimating the object similarity within a single category (intra-category) and between contours from different categories (inter-category). Therefore, the first test set contains all the samples in the bird category of the MPEG-7 Shape test set B, see Figure 5.4. The second set contains 20 objects taken from 4 different categories, see Figure 5.14. In both experiments the similarity score λ is computed for all the pairs of shapes in the set. The similarity scores obtained are presented in Table 5.1 and Table 5.2. All the scores are multiplied by 10^3 when they are presented in the tables and the figures. The distance maps were generated inside the objects only with $V_0 = 50$. This setting emphasizes the shape skeleton and gives less importance to contour details. The distance transformed images are resised to 32×32 pixels and blocks of size 4×4 were used. Larger images can be used if more precision is needed, this would imply the creation of more slices and therefore more computational power would be needed.

Figure 5.15 represents a surface plot of the similarity scores in Table 5.1. It shows that within this category the scores have small values, which means that the objects are very similar according to our measure. The MPEG-7 CE-1 test set contains several object categories with strong intra-category variability. The second test shows that our similarity measure easily discriminates between different objects.

It is worth noticing that the scores on the diagonals are zero which means that each object is identical to itself, so there is no bias in the similarity scores. It

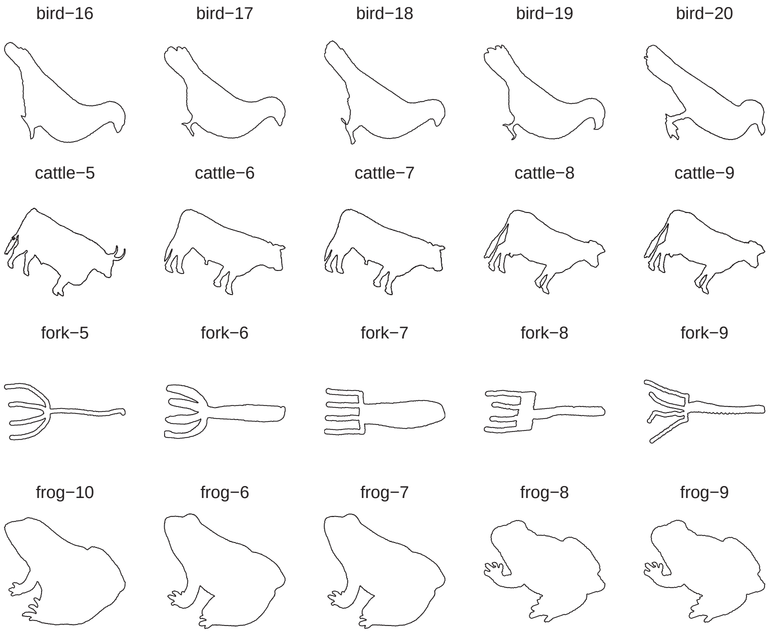


Figure 5.14: Contours of test set 2 after alignment.

is worth noticing that the scores obtained between the “bird-3”, “bird-4”, “bird-5”, “bird-6”, and the rest of the birds in this category are larger than the rest of the scores. This can be explained by the fact that these four birds have much shorter tails and have a more circular contours compared the rest of the birds. The similarity scores are low between the pairs (“bird-3”, “bird-4”), (“bird-5”, “bird-6”), (“bird-7”, “bird-8”) and (“bird-9”, “bird-11”). By visual inspection one can verify that each pair of contours represent the one bird contour rotated or rotated and scaled. Therefore, we can safely say that our measures have a 0.5% error, which can be explained by the small contour variation introduced by rotation and the size reduction of the distance maps. Lower error can be obtained by increasing the size of the distance map images and reducing the block size used for the metaslices creation.

Dark blue regions in Figure 5.15 represent very low scores (close to zero), which shows that objects in this category are very similar or even identical. To find out which are the most similar contours to a given contour in Figure 5.14 we sort the scores on the row corresponding to this contour in Table 5.2. Using Figure 5.16 one can easily estimate which are the most similar object within this category, based on the clustered dark blue cells. Figure 5.16 shows that similarity scores between subjects from the same category are low, while those from different categories are relatively high. Therefore, sorting the scores in ascending order will yield the most similar objects first.

The inter-category scores obtained by our similarity estimation technique are substantially larger than the intra-category ones. Therefore, this technique could be used as a shape classification technique. Moreover, it is sensitive to intra-category shape variations thus it can be used in similarity-based retrieval. When the $OP_1(.)$ is the summation, the computation of the meta-slices can be computed directly from the distance map without having to pass by the binary slices computation. Since each of their pixel values represent the rank of that pixel in the distance map when compared to the distance values of the pixels inside the block being considered.

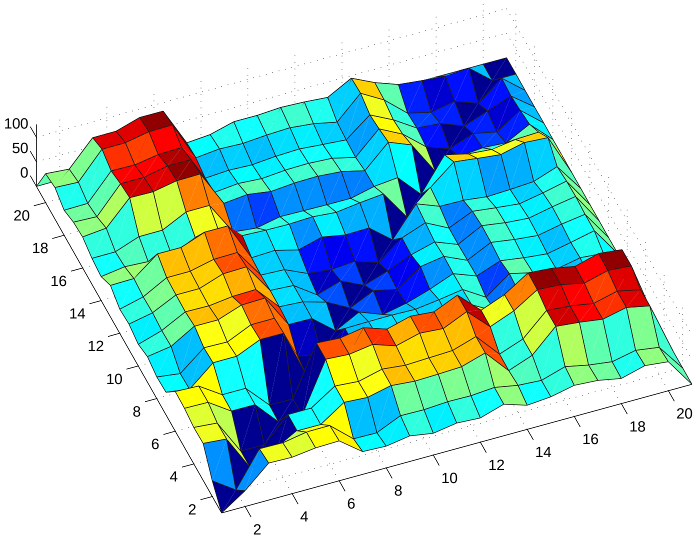


Figure 5.15: The similarity scores for the bird contours in Figure 5.5, dark blue cells mean most similar objects.

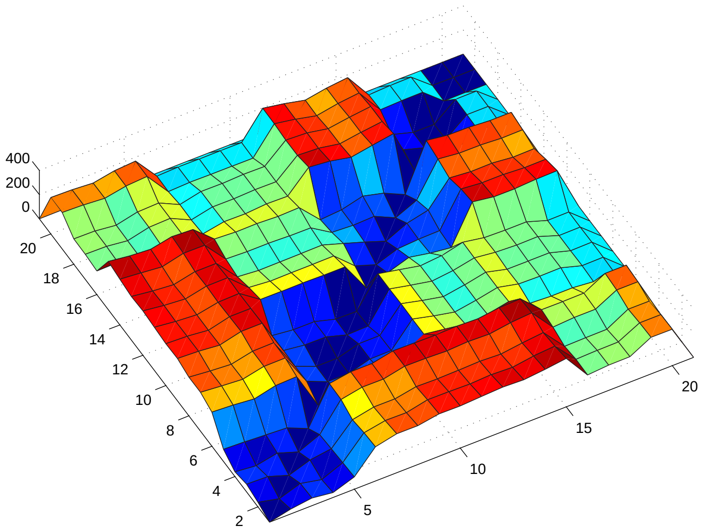


Figure 5.16: Similarity scores obtained for the contours in test set 2 presented in Figure 5.14.

Table 5.1: Similarity scores for the contours in Figure 5.5. These scores are multiplied by 10^3.

Object	*bird-1*	*bird-2*	*bird-3*	*bird-4*	*bird-5*	*bird-6*	*bird-7*	*bird-8*	*bird-9*	*bird-10*	*bird-11*	*bird-12*	*bird-13*	*bird-14*	*bird-15*	*bird-16*	*bird-17*	*bird-18*	*bird-19*	*bird-20*
bird-1	0	32	75	74	80	81	46	44	52	44	52	49	64	49	53	63	60	51	64	59
bird-2	32	0	71	70	79	79	39	38	60	59	62	61	65	56	53	68	58	52	62	55
bird-3	75	71	0	0	48	48	80	77	88	84	85	87	102	53	71	117	112	105	116	106
bird-4	74	70	0	0	48	48	80	76	88	83	85	87	102	53	71	116	112	104	116	106
bird-5	80	79	48	48	0	1	101	98	105	89	102	98	120	76	95	125	122	112	125	121
bird-6	81	79	48	48	1	0	101	98	106	90	103	99	120	76	95	126	122	113	126	121
bird-7	46	39	80	80	101	101	0	7	39	42	42	40	42	54	28	53	42	38	46	43
bird-8	44	38	77	76	98	98	7	0	38	39	39	38	44	52	23	58	47	41	51	47
bird-9	52	60	88	88	105	106	39	38	0	25	7	17	32	51	33	51	44	39	49	60
bird-10	44	59	84	83	89	90	42	39	25	0	22	12	45	53	32	55	48	43	53	60
bird-11	52	62	85	85	102	103	42	39	7	22	0	16	36	49	31	56	49	45	55	63
bird-12	49	61	87	87	98	99	40	38	17	12	16	0	37	55	31	52	46	41	50	61
bird-13	64	65	102	102	120	120	42	44	32	45	36	37	0	60	42	41	35	37	40	57
bird-14	49	56	53	53	76	76	54	52	51	53	49	55	60	0	47	86	80	75	86	83
bird-15	53	53	71	71	95	95	28	23	33	32	31	31	42	47	0	67	56	52	59	61
bird-16	63	68	117	116	125	126	53	58	51	55	56	52	41	86	67	0	17	22	19	44
bird-17	60	58	112	112	122	122	42	47	44	48	49	46	35	80	56	17	0	18	8	39
bird-18	51	52	105	104	112	113	38	41	39	43	45	41	37	75	52	22	18	0	17	35
bird-19	64	62	116	116	125	126	46	51	49	53	55	50	40	86	59	19	8	17	0	38
bird-20	59	55	106	106	121	121	43	47	60	60	63	61	57	83	61	44	39	35	38	0

Table 5.2: Similarity scores for the contours in Figure 5.14. These scores are multiplied by 10^3.

Objects	*bird-16*	*bird-17*	*bird-18*	*bird-19*	*bird-20*	*cattle-5*	*cattle-6*	*cattle-7*	*cattle-8*	*cattle-9*	*fork-5*	*fork-6*	*fork-7*	*fork-8*	*fork-9*	*frog-10*	*frog-6*	*frog-7*	*frog-8*	*frog-9*
bird-16	0	39	63	39	101	282	322	321	351	348	358	368	361	379	401	196	209	210	303	302
bird-17	39	0	58	22	94	273	306	304	342	341	337	345	339	363	377	199	206	208	297	297
bird-18	63	58	0	58	83	256	293	291	325	322	325	330	323	346	364	182	185	187	285	284
bird-19	39	22	58	0	74	298	332	331	369	366	360	368	363	388	402	215	227	228	318	318
bird-20	101	94	83	74	0	304	333	334	370	367	365	370	363	387	392	234	233	234	330	330
cattle-5	282	273	256	298	304	0	73	71	75	73	247	221	184	206	244	162	152	153	136	138
cattle-6	322	306	293	332	333	73	0	1	52	53	246	204	168	198	234	195	192	193	143	145
cattle-7	321	304	291	331	334	71	1	0	51	52	247	206	169	199	236	193	191	192	141	144
cattle-8	351	342	325	369	370	75	52	51	0	5	244	207	172	191	228	206	199	200	141	143
cattle-9	348	341	322	366	367	73	53	52	5	0	244	209	173	190	232	202	196	197	140	142
fork-5	358	337	325	360	365	247	246	247	244	244	0	59	118	84	63	373	347	346	356	359
fork-6	368	345	330	368	370	221	204	206	207	209	59	0	62	74	77	355	337	338	326	329
fork-7	361	339	323	363	363	184	168	169	172	173	118	62	0	81	123	320	304	305	287	290
fork-8	379	363	346	388	387	206	198	199	191	190	84	74	81	0	79	351	331	331	319	321
fork-9	401	377	364	402	392	244	234	236	228	232	63	77	123	79	0	379	354	352	339	340
frog-10	196	199	182	215	234	162	195	193	206	202	373	355	320	351	379	0	50	51	132	129
frog-6	209	206	185	227	233	152	192	191	199	196	347	337	304	331	354	50	0	1	138	138
frog-7	210	208	187	228	234	153	193	192	200	197	346	338	305	331	352	51	1	0	139	139
frog-8	303	297	285	318	330	136	143	141	141	140	356	326	287	319	339	132	138	139	0	2
frog-9	302	297	284	318	330	138	145	144	143	142	359	329	290	321	340	129	138	139	2	0

Chapter 6

Relevance Feedback

In this chapter, we discuss relevance feedback techniques in content-based image retrieval systems. We will first introduce the need and concept of relevance feedback in content-based image retrieval. Then, in Section 6.1 we review key issues in relevance feedback as a learning process and propose a brief overview of commonly used relevance feedback techniques. Our relevance feedback algorithm for the ordinal-correlation framework for shape retrieval is described in Section 6.2. In Section 6.3 a simple objective performance measure of the effect of the feedback loop is presented and simulations are shown in Section 6.4.

As mentioned throughout this thesis, content-based image retrieval (CBIR) systems attempt to automatically index and query image databases based on image content. They operate on features extracted automatically from the images, such as color, texture and shape. Their accuracy is too low however to make them popular tools, such as text-based information retrieval systems. This is mainly due to the gap between the extracted low-level image features and the high-level semantic concepts the images may contain.

Furthermore, the subjectivity of humans makes it even harder to understand what a user is looking for when he puts a query by example image. Human perception of similarity is subjective and task-dependent, that is, two pictures may seem similar to one person while being perceived as dissimilar by another. Moreover, the same person may perceive the same image differently depending on the context in which it is put.

Common users find it difficult to combine different features to form a coherent query representing the image they seek, because, each type of visual feature tends to capture only one aspect of the image content such as shape, texture or color. Moreover, the human mind manipulates the concepts in an a-modal fashion; which makes their formulation in terms of the independent low-level features used in CBIR systems a serious challenge for users as well as for GUI designers. CBIR systems usually represent each image as a high dimensional feature vector (vector model representation [126]) and use some similarity measure between each pair of feature vectors.These similarity measures are feature dependent and are not simple to tune manually.

To address these problems, relevance feedback (RF) techniques have been proposed to allow the system to learn from the users' interaction with the retrieval results. Relevance feedback was introduced by the information retrieval community in the late 1960's to increase the number of relevant document retrieved [60]. It was found to be very effective in text-based information retrieval systems [125]. In these systems, RF allows the user to interact with the retrieval results of a query by selecting terms from the documents he considers relevant to expand the original query. Terms from the documents considered irrelevant may also be used to modify the original query. Therefore, the key issue in relevance feedback is how to learn from the users feedback and use the positive and negative examples to refine the query or adjust the similarity measure.

In CBIR low-level features extracted from the image content are used instead of the precise and semantically meaningful terms found in IR. Moreover, humans have difficulty to precisely express their visual queries. Even with these handicaps, RF has been considered a major research direction in CBIR, promising better systems that can learn from the users' interaction to improve their performance.

In the next section of this chapter we will review the main ideas of RF in CBIR systems, then propose our relevance feedback technique adapted to the shape retrieval framework proposed in the previous chapter.

6.1 Relevance Feedback Overview

Several issues in relevance feedback techniques are relevant to CBIR, such as indexing structures, feature selection, learning schemes and scalability. We will focus our discussion on the consideration of relevance feedback in CBIR as a small sample machine learning problem, followed by a description of the learning and search natures of each algorithm. We begin the discussion with an overview of the classic relevance feedback approaches in CBIR.

6.1.1 Classical Algorithms

The early relevance feedback schemes for CBIR were adopted from feedback schemes developed for classical text documents retrieval. These schemes can be classified into two categories: query point movement (query refinement) and re-weighting (similarity measure refinement) [116]. Both of these categories of approaches were based on the vector model, the most popular model used in information retrieval [126]. The query point movement method tries to improve the estimate of the “ideal query point” by moving it towards positive example points and away from bad example points in the query space. The query estimate can be updated in various ways. A frequently used technique to iteratively improve this estimate is Rocchio's formula (see Equation 6.1.1). That is, for a set of relevant documents D'_R and non-relevant documents D'_N given by the user [117], the optimal query is defined as:

$$Q' = \alpha Q + \beta(\frac{1}{N_{R'}} \sum_{i \in D'_R} D_i) - \gamma(\frac{1}{N_{N'}} \sum_{i \in D'_N} D_i) \tag{6.1}$$

where α, β and γ are suitable constants; and N'_R and N'_N are the number of documents in D'_R and D'_N, respectively.

This technique was used in the MARS system to replace the document vector with visual feature vectors. Experiments show that retrieval performance can be improved by using these relevance feedback approaches.

The re-weighting method enhances the importance of a feature or parts of it that help retrieve relevant images and reduce that of the features or the parts of the features that hinder this process. This is achieved by assigning weights to the feature vectors or to its entries and using a weighted distance metric.

Let S be the weighted similarity metric defined as:

$$S = \sum_{i \in [N]} w_i |X_i^{(1)} - X_i^{(2)}| \tag{6.2}$$

Where the weights w_i are used to emphasize the dimension of the feature that contributes more to the similarity scores of the query results labeled as positive examples. Therefore, the weight for a feature component, w_i, is updated as follows:

$$w_i = w_i(1 + \overline{\delta} - \delta_i), \; \delta_i = |f(Q) - f(Rel_i)| \tag{6.3}$$

where $\overline{\delta}$ is the mean of δ.

On the other hand, the weights should depress the feature dimension that contribute more to the match of images labeled as negative examples. Therefore, the weights are updated as [54]:

$$w_i = w_i(1 - \overline{\delta} + \delta_i), \; \delta_i = |f(Q) - f(Rel_i)| \tag{6.4}$$

In MARS [118] well established text retrieval techniques were applied to image retrieval. Two factors, "component importance" and "inverse collection importance", were proposed for images in accordance to the factors "term frequency" and "inverse document frequency" in text retrieval. The vector space model was used for relevance feedback. They also used Gaussian normalization to put equal emphasis on each feature component, and then used the inverse of the standard deviation of each component for the images in the relevant feedback set as weights. They concluded that the approach adopted from text retrieval performed better than Gaussian normalization but the latter was more robust to unknown feature components. Another way of using the feedback to refine the similarity measure was proposed in [119], where a set of similarity measures is pre-defined, and the system selected the similarity measure which minimized the sum of the differences between the ranks of the retrieved images and the ranks of the relevant images selected by the user. They also proposed an alternative to hard switching between different similarity measures; using a weighted sum of all similarity measures. Similarly, in the ImageRover system [131], appropriate L_p Minkowski distance metrics are automatically selected to minimize the mean distance between the relevant images specified by the user.

The query space is updated by selecting feature models [92], assuming that each feature model represents a single aspect of the image content better than the other models. They used a learning scheme to dynamically determine which feature model or combination of models is best for the next retrieval iterations.

More computationally robust methods performing global feature optimization were proposed; such as the MindReader retrieval system [58] that formulates a minimization problem on the parameter estimating process. A further improvement over the MindReader approach is given in [117], where, optimal query estimation and weighting functions are derived by a unified framework, based on the minimization of the total distances of the positive examples from the updated query. The weighted average and a whitening transform in the feature space were found to be the optimal solutions.

6.1.2 Relevance Feedback as a Learning Problem

Relevance feedback can be approached as a learning problem where the system learns from the users feedback on the retrieval results and refines its retrieval process. Both the query-movement method and the re-weighting method are simple learning methods.

Machine learning is about constructing computer programs that automatically improve with experience. Therefore, machine-learning methods, such as decision tree learning [83], artificial neural networks [70], Bayesian learning [150], and kernel based learning [145] can be and have been applied to relevance feedback in CBIR.

The major difference in the case of CBIR is that users usually don't provide a large a large number of feedback examples. Therefore, the size of the training sets is small, typically less than 10 examples per query. Moreover, the feature dimensions in CBIR systems are usually high, limiting the number of machine learning techniques that can be used in CBIR. Techniques such as decision tree learning and artificial neural networks are therefore not very appropriate for this task.

Bayesian learning is advantageous in addressing the issue of learning from a small sample compared to other techniques. In [150] the feature distribution was considered as a Gaussian mixture and Bayesian inference used to learn from feedback iterations during a query session. This approach suffers from the complexity of the data model used, which has to be estimated from a small training set, and its computational efficiency.

Active learning methods have been used to select samples which maximize the information gain, or minimize entropy/uncertainty in decision-making. These methods increase the speed of the learning process and enable faster convergence of the retrieval result which in turn increases user satisfaction. The approach proposed in [118] used Monte Carlo sampling to search for the set of samples that will minimize the expected number of future iterations. The support vector machine (SVM) active learning algorithm proposed in [145] to select the sample which maximizes the reduction in the size of the version space in which the class boundary lies. Without knowing a priori the class of a candidate, the best strategy is to halve the search space each time. Selecting the points near the SVM boundary almost achieves this goal, and it was found to be more efficient than other more sophisticated schemes which require exhaustive trials on all the test items.

6.1.3 Relevance Feedback as Pattern Recognition

Relevance feedback in CBIR can be considered as a pattern recognition or classification problem. In which the images can be classified into two separate groups, relevant and irrelevant to the query. The users positive and negative examples are used as training samples of the classifier.

In this scenario many existing pattern recognition tools can be adopted such as linear classifier [157], nearest-neighbor classifier [156], Bayesian classifier [150], support vector machine [145], etc.

Typically in a pattern recognition problem, there are clear pattern classes. Therefore, each item can be put into one or more of the predefined classes. The algorithms' task is to separate the classes as clearly as possible. In the case of general purpose CBIR, there are no pre-defined classes. Moreover, the users subjectivity and his context dependent perception limits the performance of classification algorithms.

6.1.4 Semantics in Relevance Feedback

The approaches described until now perform relevance feedback based on low-level feature vectors by adopting the vector model developed for document retrieval. Even though, these approaches do improve the performance of CBIR, they have severe limitations. The major problem is that users often pay more attention to the semantic content (or a certain object/region) of an image than to the background; furthermore, the feedback images may be partially similar in semantic content, but vary greatly in low-level features. The opposite is true too since the same feature vector can be obtained from two completely different images, e.g. the histogram. Hence, using low-level features alone may not be effective in representing users' feedback and in describing their intentions. Therefore, there have been efforts to incorporate semantics in relevance feedback to image retrieval. FourEye [92] and PicHunter [26] systems, made use of hidden annotations through the learning process.

Some CBIR systems tried to propose ways to memorize the history of the query and feedback and to make use of it in future retrieval sessions. An attempt to explicitly memorize learned semantic information to improve CBIR performance, was proposed in [73]. The basic idea of this approach is to accumulate semantic relevance between image clusters learned from the user's feedback in correlation network. Mathematically, the correlation network is represented by a correlation matrix.

In the rest of this chapter, we introduce a relevance feedback technique and use it to perform both query refinement and similarity measure tuning of the ordinal correlation framework. The similarity measure tuning is done via a re-weighting technique while the query refinement is done by dynamically changing the query resolution. Adaptively changing the resolution at which the images are compared based on the edge information within a region (block), can be very beneficial in discriminating between similar objects based on the users RF. This is achieved by putting more emphasis on the regions with higher contour activity and use lower resolution elsewhere.

6.2 Hybrid Relevance Feedback Approach

In this section we focus on a scheme that makes use of the users' feedback on the retrieval results to tune the retrieval framework proposed in Chapter 5. The feedback is used to perform two tuning actions. The first action is similarity measure tuning via a re-weighting algorithm; while the second action is an adaptive query vector expansion approach to allocate more entries of the feature vector to the most important regions in the image. The important regions of the image are either the regions containing the contour pixels or the object structure, depending on what the users' interests are.

6.2.1 Shape Similarity Tuning

Here we propose a re-weighting technique as a direct way of integrating the relevance feedback information into the shape similarity estimation technique described in Chapter 5. This is easily done by replacing the $OP_3(.)$ of the ordinal correlation structure by a weighted sum of all the meta-differences to estimate dissimilarity score as $\lambda = \sum_j W_j \times D_j$, where W_j, are the weights of regions $R_j, j = 1, 2, ..., m$, estimated based on the relevance feedback information.

Therefore, a typical search session begins when the user draws a contour on the image editing tool or selects an image containing a contour. A first search iteration is done in the database using this query image and weights $W_j = 1$, for $j = 1, 2, ..., m$. Next, the results of this iteration are presented to the user. The user then labels few of these images as relevant and few more as irrelevant according to his information needs.

The aim is to find the weights that would make the dissimilarity scores smaller for the images in the relevant set, and larger for those of the rest of the images in the database, assumed not relevant to the query, thereby, making the discrimination between the relevant and irrelevant images easier.

6.2.2 Weights Estimation

The weights are automatically estimated based on the statistics of each feature vector entry separately, without need for user intervention or ad hoc thresholds. Features from both positive and negative feedback image sets are used in the weight estimation, and no assumption of optimal relevant set or irrelevant set are made.

For an entry in the feature vector to be useful in discriminating between images:

- its variation among all the images in the database should be large, assuming that the number of relevant images to a certain query is very small compared to the number of irrelevant ones,
- its variation among the relevant images should be small,
- its variation among the irrelevant images should be large.

Since the output of the operation $OP_2(.)$ is a vector containing differences between the meta-slices $D_j = ||M_j^X - M_j^Q||_2^2$, for $j = 1, 2, ..., m$, of the query image Q and candidate image X. The retrieval results of a given query are the images within a certain distance from the query image. Assuming that the feature vector space is dense enough (large set of images), the query image can be assumed to be the centroid of the retrieved set. Therefore, its features can be very well estimated as the mean of the features of the retrieved set entries. Thus, $D_j = ||M_j^X - Mean(M_j^I)||_2^2$, for $j = 1, 2, ..., m$ and I in the retrieved set. Thus the mean of D_j over the retrieved set is the squared standard deviation of the M_j^X, for X in that set. Therefore, for a given query image the mean of D_j over the retrieval set will be used as an estimate of the variation of the feature vector entries in that set.

Notation

We will adopt the following notation:

K**:** Number of iterative searches with user feedback.

R^0= {all images in the database}.

R^k= {retrieval set after the k'th search}, where

$$k = \begin{cases} 0 & \text{whole database,} \\ 1 & \text{search with the original query,} \\ 2, \ldots, K & \text{searches after user feedback.} \end{cases}$$

$R^k_{\textbf{rel}}$= {set of relevant images in R^k}. These images are the ones that are marked as relevant by the user at the end of the k'th search, $R^k_{\text{rel}} \subseteq R^k$.

$R^k_{\textbf{irrel}}$= {set of irrelevant images in R^k}. These images are the ones that are marked as irrelevant by the user at the end of the k'th search, $R^k_{\text{irrel}} \subseteq R^k$, $R^k_{\text{rel}} \cup R^k_{\text{irrel}} \subseteq R^k$.

N: Number of images in the database.

m: Number of features in a feature vector.

f_i= $[f_{i1} f_{i2} \cdots f_{im}]$. Feature vector of the i'th image where f_{ij} is the j'th component of the vector, $i = 1, \ldots, N$, $j = 1, \ldots, m$.

F^k= {feature vectors of the images in R^k} = $\{f_i | i \in R^k\}$.

$F^k_{\textbf{rel}}$= {feature vectors of the images in R^k_{rel}} = $\{f_i | i \in R^k_{\text{rel}}\}$.

$F^k_{\textbf{irrel}}$= {feature vectors of the images in R^k_{irrel}} = $\{f_i | i \in R^k_{\text{irrel}}\}$.

F^k_j= {set of values for the j'th components of the feature vectors of the images in R^k} = $\{f_{ij} | i \in R^k\}$.

$F^k_{\textbf{rel},j}$= {set of values for the j'th components of the feature vectors of the images in R^k_{rel}} = $\{f_{ij} | i \in R^k_{\text{rel}}\}$.

$F^k_{\textbf{irrel},j}$= {set of values for the j'th components of the feature vectors of the images in R^k_{irrel}} = $\{f_{ij} | i \in R^k_{\text{irrel}}\}$.

Let

$$\mu^0_j = \text{Mean}(F^0_j), \tag{6.5}$$

$$\mu^k_{\text{rel},j} = \text{Mean}(F^k_{\text{rel},j}), \tag{6.6}$$

$$\mu^k_{\text{irrel},j} = \text{Mean}(F^k_{\text{irrel},j}). \tag{6.7}$$

For the j'th feature in the $k+1$'st retrieval iteration the weight is estimated as

$$w^k_j = \begin{cases} 0, & \text{if } Max(F^k_{\text{rel},j}) \geq Min(F^k_{\text{irrel},j}) \\ \mu^k_{\text{irrel},j} / \mu^k_{\text{rel},j}, & \text{otherwise.} \end{cases} \tag{6.8}$$

Therefore the weights will be set to zero if the feature (difference between relevant image and the query images for a given region) of any of the relevant images is larger than that of any of the irrelevant images, $Max(F^k_{\text{rel},j}) \geq Min(F^k_{\text{irrel},j})$, to avoid increasing the dissimilarity score of the relevant images. This can simply mean that the corresponding region is not relevant to the comparison of the objects, therefore its contribution to the similarity score should be nil. The other regions weights are computed as the ratio $\frac{\mu^k_{\text{irrel},j}}{\mu^k_{\text{rel},j}}$ which will be larger than one. Therefore, increasing the dissimilarity scores proportionally to the value of the feature F^k_j. This should increase the dissimilarity scores of the irrelevant images much faster than those of the relevant images, making the discrimination easier.

For the case of iterative retrieval, the weights of iteration $k+1$ can either be estimated using all the information accumulated from the previous iterations or just a subset thereof. Using all the history information, $\Omega = \cup^k_{i=1} R^i_{\text{rel}}$, will give a better estimation of the variance of the features in the relevant and irrelevant image sets. On the other hand it hinders the convergence of the retrieval process. Using only the previous iteration R^k_{rel} feedback implies that the user has to mark several entries at each iteration to have enough samples to estimate the parameters needed.

We opted for using the feedback info from only two iteration at most for a given search. In this iterative tuning process object relevance decay is implemented, where relevant instances of the near past are considered more heavily than those of the early past. This is done practically in the weight estimation:

$$\mu^{k+2}_{\text{rel},j} = \frac{2 \times \mu^{k+1}_{\text{rel},j} + \mu^k_{\text{rel},j}}{3}, \tag{6.9}$$

$$\mu^{k+2}_{\text{irrel},j} = \frac{2 \times \mu^{k+1}_{\text{irrel},j} + \mu^k_{\text{irrel},j}}{3}. \tag{6.10}$$

6.2.3 Adaptive Resolution

The problem with the fixed resolution representation is that it uses the same resolution to represent the entire shape. There are certain regions of the shape where low resolution is sufficient thus increasing the resolution does not improve the quality of the representation and the discrimination power of the similarity framework. While, in other regions of the shape (regions with high edge presence) higher resolution would improve the discrimination capability of the OCF. Having the same high resolution for the entire shape is wasteful in terms of the number of dimensions used to represent the shape and hence the query cost. Therefore, additionally to using weights to tune the similarity measure we opted for adaptive resolution representation of shapes. Figure 6.1 presents the OCF with the adaptive resolution, where some of the blocks are further subdivided into smaller blocks.

This is to take into consideration the fact that some of the blocks into which we subdivided the distance map contain more information about the object contour (or its skeleton) than other blocks. Therefore, it makes perfect sense to allocate more entries in the feature vector to describe the content of these blocks. Keeping a fixed resolution implies that all blocks have the same importance. This approach expands the query vector by adding new entries corresponding to the finer resolution blocks of the more important regions.

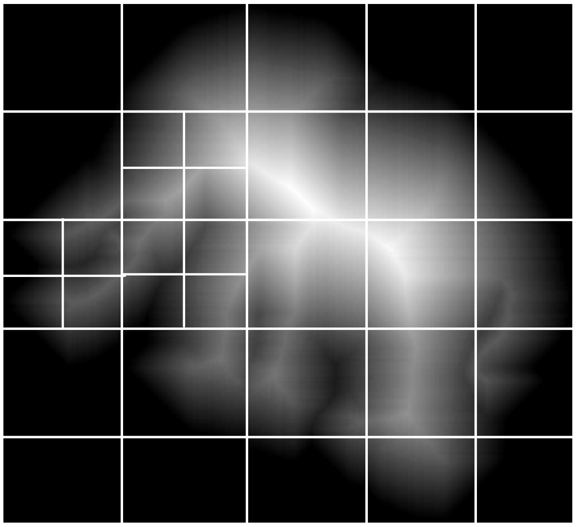


Figure 6.1: Adaptive resolution shape representation.

Depending on the users search criteria, the most important regions of the distance map may be those containing the image contour points or its skeleton. In both cases however, the distance map pixel values in these regions will be local maxima. Therefore, the contribution of these regions to the similarity computation should be emphasized by allocating more blocks to them and thus more entries in the feature vectors.

This subdivision of a region R_j^k (block) into smaller ones is performed only if two conditions are met:

1. $w_j^k = \mu^k_{\text{irrel},j}/\mu^k_{\text{rel},j} \simeq 1$, for relevant and irrelevant candidates,
2. block R_j^k contains object boundary info.

The first condition means that the weights are similar for relevant and irrelevant candidates, therefore weights are not very helpful here. The second condition, insures that the block is not a smooth area for which further subdivision does not bring any additional information. When the two conditions are met, subdividing the block into four smaller blocks may improve the discrimination between similar objects.

Comparison of the similarity scores obtained before and after the RF loop with adaptive resolution, shows that the range of the scores between subjects from the same category have increased (even though much less than the increase of the intra-category scores) thus showing that the similarity framework becames more sensitive to the small boundary variations, see Figure 6.5. It can be seen from this plot that the scores are more sensitive to the details variations of the contours (seen clearly in the scores of objects of the same category). Here, the query image is “bird-16” with “bird-17” and “bird-19” positive feedback, while “cattle-8” and “cattle-9” represented the negative feedback.

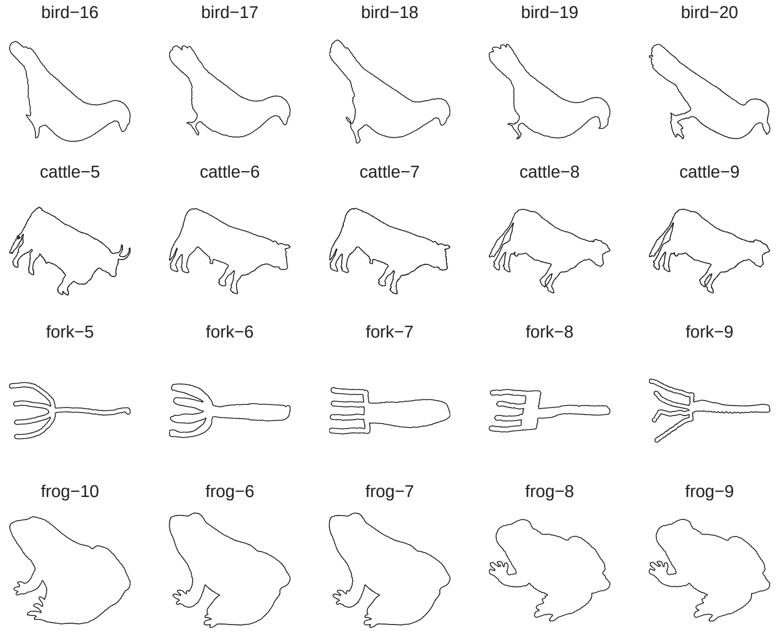


Figure 6.2: Contours of test set 3 after alignment.

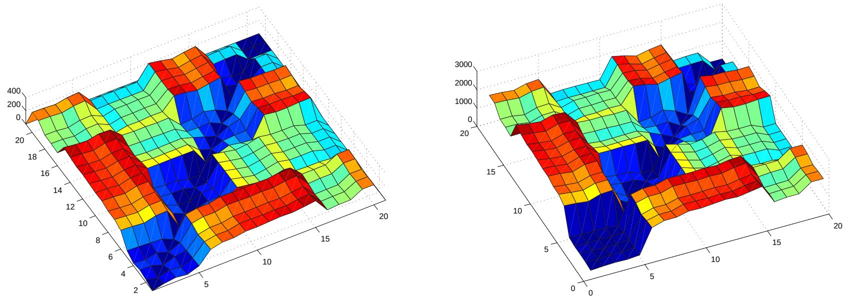
6.3 Objective Measure of the Discrimination Power

The goal of the weight adaptation is to increase the capability of the similarity measure to discriminate between the relevant and irrelevant images to a given query image. To measure how the weights influence the similarity scores of the retrieved images; we define a simple objective measure of the schemes’ discrimination power. Let $DP_k = \frac{E(\lambda^k_{irrel})}{E(\lambda^k_{rel})}$; where $E(\lambda^k_{rel})$ is the the mean of the similarity scores of the images in R^k_{rel} and $E(\lambda^k_{irrel})$ is the the mean of the similarity scores of the images in R^k_{irrel}.

6.4 Experimental Results

In this experiment four categories of objects were used, see Figure 6.2, and at each iteration three images were used as the relevant set and two as the irrelevant one. The dissimilarity scores of the search results are plotted in Figure 6.3 (a) without and (b) with the relevance feedback estimated weights, respectively. It can be noticed from the plot of Figure 6.3 (b), that the surfaces appear flatter on the diagonal of the plot and closer to zero, which means that the dissimilarity scores within a group of objects are not much larger. Noticing that the variance of the scores after RF is larger within the test set, therefore the scores are larger than those obtained without weights, making the discrimination between the objects easier. This can be seen by steeper transitions between the objects of different categories. The steep transitions between the plateaus of Figure 6.3, show that there is clear discrimination between between the objects from the same or different categories.

When the relevance feedback information is used to recompute the similarity scores, the DP increases, e.g. in the experiment mentioned above query with “bird-16”, the discrimination power measure passed from $DP = 4.5$ without feedback to $DP = 10.34$ with FB. This is explained by way the weights are estimated, to increase the scores of the irrelevant images much faster than those of the relevant ones.



(a) (b)

Figure 6.3: Similarity scores obtained for the contours in test set 3 presented in Figure 6.2: (a) before RF, (b) after RF weight estimation.

Figure 6.4 shows the evolution of the discrimination scores before and after the relevance feedback loop was used to tune the weights in the OCF.

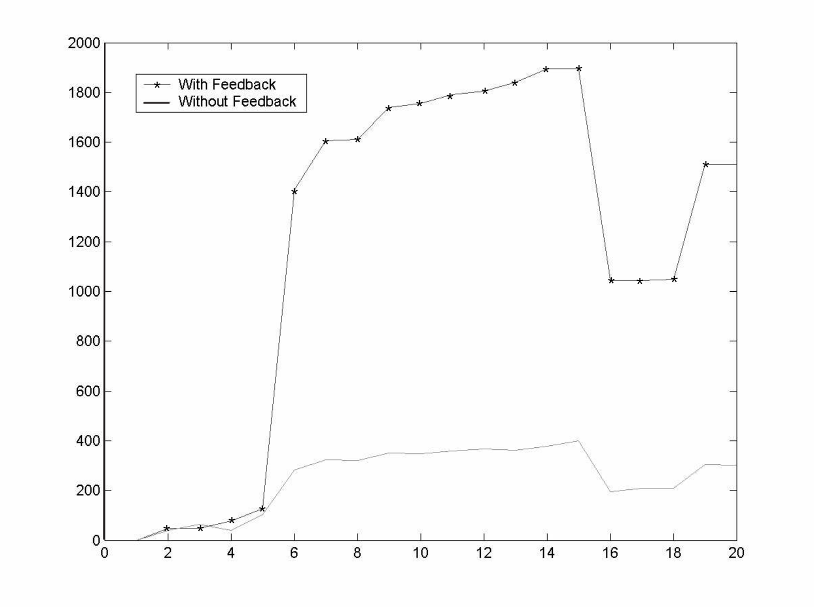


Figure 6.4: Evolution of dissimilarity scores before and after relevance feedback with fixed resolution.

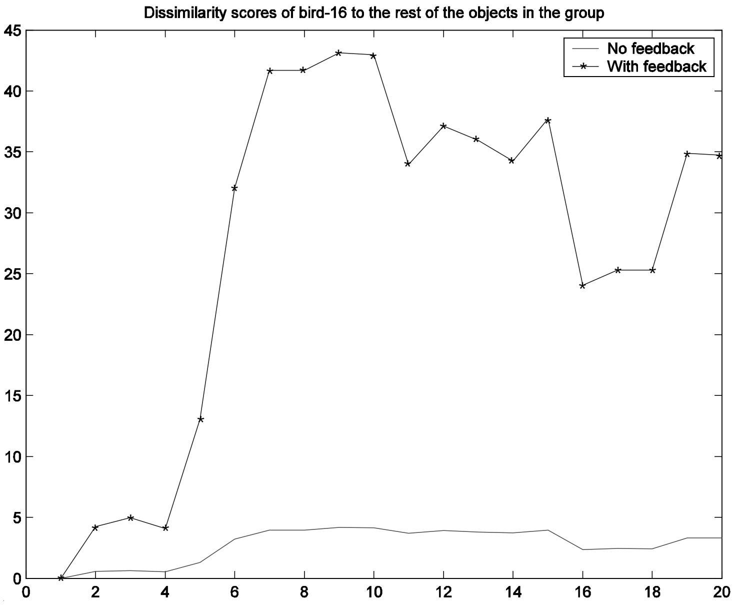


Figure 6.5: Scores before and after RF and adaptive resolution shape representation.

Chapter 7

Conclusions

Content-based multimedia document indexing and retrieval has been a very active research area. This has generated a significant amount of published research work, an ISO standard called MPEG-7, and a number of industrial and academic systems already in use. No one however can claim victory over the query by content problem yet, since the performance of most existing systems is rather poor especially when applied on a general database. Moreover, interoperability between systems is practically impossible now, since different systems may extract different features and use different similarity estimation schemes. Therefore, content-based multimedia retrieval will still be a challenging research topic for several more years.

Initially all the research groups were trying to find the best feature to characterize an image. Now, they realized that there is no single feature that can efficiently characterize the content of an image. Moreover, the semantic concepts of the content cannot be efficiently captured by the automatically extracted low-level features, because the interpretation of the content of an image is highly dependent on the context in which this action is performed, and on the person performing it. Furthermore, this interpretation requires the extraction of semantically meaningful entities from the image. This is naturally done by the human visual system without extra effort, and is used in most of our daily activities. Such automatic extraction is however not possible yet by the current image processing techniques, which constitutes the major obstacle to the development of efficient CBIR systems. This problem will be solved once the media stream acquisition devices become able to capture the scenes in more intelligent format, e.g. video camera which captures the scenes as 3D object models. One step on that direction can be to automatically annotate the actual images and video frames by simple contextual information and the range data to assist the automatic analysis algorithms to extract the semantics from the audio visual data. Even when we will have semantically meaningful entities extracted correctly from the AV material a challenging problem remains to be solved: how to mimic the human perception of similarity using numerical algorithms. This problem is still researched in text-based IR systems, where the semantic entities (words) have always been available, yet comparing the content of text documents is still a challenging problem.

Although a literature review of all the published work on CBIR is beyond the scope of this thesis, we gave a global picture of the problem and discussed it from all its aspects. This review part of the thesis tries to emphasize the major achievements and trends in the field and discusses the most prominent techniques in each aspect. After that a description of MUVIS is given. Our motivation for developing MUVIS has been to develop a general system structure that supports the development and testing of new algorithms for features extraction, similarity estimation and relevance feedback. Therefore the system has been developed as a framework for the implementation and testing of novel CBIR algorithms.

The contribution of this thesis can be split into three topics, system and GUI design and implementation, shape features extraction and similarity measures and relevance feedback techniques.

User interfaces play a very prominent role in CBIR systems, since the user-computer interactions are carried out through them. They are used for browsing, query formulation, results presentation and query tuning. The GUI of MUVIS is entirely designed and implemented by the author in JAVA. It allows simple and intuitive ways of browsing capabilities, where the user navigates through the database entries by simply clicking on the thumbnail images presented to him. These thumbnail images are retrieved based on their similarity to the image the user clicks on. More complex tools for query formulation are also available in the GUI for more advanced users. These tools are the image editing tool and the histogram editing tool. The former allows to submit queries using sketches and images composed of color-filled shapes, images, and regions extracted from other images. The composed image is in vector graphic format, therefore allows full image as well as object based queries. The latter allows editing and query by image histograms. It allows the visual inspection and comparison of the histograms of the query and any of the retrieved images, which gives the user a hint as to why the images are retrieved in certain order.

The major part of the the work is related to shape characterization and similarity estimation techniques, where we proposed three contour-based algorithms based on HCP extracted using the WTMM, and one region-based using the ordinal correlation framework. The huge amount of digital material available imposes a hard constraint on the complexity of the techniques to handle this massive amount of data. Therefore, simple and computationally efficient techniques are emphasized in our work. Moreover, the extension of our work to mobile devices with limited capabilities is another motivation for not considering computationally heavy approaches.

In the contour-based approaches, the salient points on the contour are detected based on the modulus maxima of the wavelet transform coefficients, and two types of features are extracted at these high curvature points. The first type of shape visual features are topological measurements. The second type of features is directly extracted from the wavelet coefficients and used as entries of the feature vectors to describe the object. For each type of features we derived a similarity measure to compare the objects based on their corresponding representations. These similarity measures try to mimic the human way of comparing shapes based on their polygonal approximations, and are thus more appropriate than Minkowski type metrics.

The second category is a region-based approach where the binary image containing the contour is transformed into a distance map by a distance transform. The distance map representations are compared using the ordinal correlation framework. This allows to compare the contour activity in a region of the image in a scale relative to the block size of the ordinal correlation mesh. Moreover, the distance transform reduces the sensitivity of the similarity measure to small alignment errors and to variations of the contours and even to partial occlusions. The sensitivity of the system to the contour details depends on the block size. Therefore, shapes can be compared at different scales or resolutions. Furthermore, the DT is derived using two approaches, depending on whether the user wants to look for similar objects based on their boundary details or their inner structure (skeleton).

The user is the consumer of the retrieval results, and is the unique judge of their relevance to the query. Therefore, users' feedback (used in several CBIR systems) allows the system to learn more about the user's needs. Moreover, it bridges the gap between the low-level features extracted automatically and the high-level concepts the user has about the target. The feedback helps also to rid the similarity measures from the subjectivity of the user. Relevance feedback techniques are reviewed and an approach to integrate a relevance feedback loop into the ordinal-correlation framework is proposed. The proposed approach operates in two ways: first it refines the similarity measure using weights estimated based on statistics extracted from the positive and negative feedback examples; second it uses adaptive resolution analysis, where smaller blocks are used for the more active/important regions of the image.

Future work on the extension of the system to audiovisual material indexing and retrieval has already started, with additional features extracted from the image sequences based on motion and audio track. The work done in MUVIS for CBIR can be used fully in the new version of the system based on the key-frames extracted from the video sequences. These algorithms can benefit very much from the combination of features extracted from different aspect of the AV material, such as the detection of events in the video sequence based on the audio and motion changes. Considerable improvements can thus still be expected from such multi modal features.

Bibliography

[1] S. Abbasi, F. Mokhtarian, and J. Kittler. Curvature Scale Space Image in Shape Similarity Retrieval. *Springer Journal of Multimedia Systems*, 7(6):467–476, 1999.

[2] A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora. Image Sequence Analysis for Emerging Interactive Multimedia Services - The European COST 211 Framework. *IEEE Transactions on Circuits and Systems for Video Tachnology*, 8(7):802–813, November 1998.

[3] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An Efficiently Computable Metric for Comparing Polygonal Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209–216, March 1991.

[4] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jam, and C. F. Shu. The Virage Search Engine: An Open Framework for Image Management. In *Proc. of the SPIE Storage and Retrieval for Still Image and Video Databases IV*, volume 2670, pages 77–87, February 1996.

[5] F. R. Barnard. One Picture is Worth Ten Thousand Words. Printers' Ink (Now Known as Marketing/Communications) Advertizing Trade Journal, March 1927. http://www.cs.uregina.ca/ hepting/proverbial/history.html.

[6] S. Basu, A. Del Bimbo, A. Tewfik, A. H., and H. Zhang. Special Issue on Multimedia Database. *IEEE Transactions on Multimedia*, 4(2), June 2002.

[7] S. Berchtold, C. Bohm, and H. Kriegel. The pyramid-tree: Breaking the curse of dimensionality. In *Proc. of the ACM SIGMOD Conference*, pages 142–153, Seattle, Washington, USA, June 1998.

[8] T. Berners-Lee and M. Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. HarperBusiness, November 2000.

[9] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, Inc., 1999.

[10] E. Bribiesca. A Geometric Structure For Two-Dimensional Shapes And Three-Dimensional Surfaces. *Pattern Recognition*, 25(5):483–496, 1992.

[11] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, August 2002.

[12] V. Castelli and L. D. Bergman, editors. *Image Databases: Search and Retrieval of Digital Imagery*. John Wiley & Sons, 2002.

[13] N. S. Chang and K. S. Fu. Query-by-Pictorial. *IEEE Trans. on Software Engineering*, 6(6), Nov. 1980.

[14] S.-K. Chang and A. Hsu. Image Information Systems: Where Do We Go From Here? *IEEETransactionsonKnowledgeDataEngineering*, 4(5):431–442, October 1992.

[15] S. K. Chang and T. Kunii. Pictorial Database Systems. *IEEE Computer*, pages 13–21, November 1980.

[16] F. Alaya Cheikh, B. Cramariuc, and M. Gabbouj. MUVIS: A System for Content-Based Indexing and Retrieval in Large Image Databases. In *Proc. of the VLBV98 workshop*, pages 41–44, Urbana, IL, USA, October 1998.

[17] F. Alaya Cheikh, B. Cramariuc, and M. Gabbouj. Relevance Feedback for Shape Query Refinement. In *Proc. of the IEEE International Conference on Image Processing (ICIP 2003)*, volume 1, pages 745–748, Barcelona, Spain, September 14–17 2003.

[18] F. Alaya Cheikh, B. Cramariuc, M. Partio, P. Reijonen, and M. Gabbouj. Shape Similarity Estimation using Ordinal Measures. In *Proc. of the International Workshop on Very Low Bitrate Video Coding (VLBV01)*, pages 44–49, Athens, Greece, October 2001.

[19] F. Alaya Cheikh, B. Cramariuc, M. Partio, P. Reijonen, and M. Gabbouj. Evaluation of Shape Correspondence Using Ordinal Measures. In M. M. Yeung, C.-S. Li, and R. W. Lienhart, editors, *Proc. of the IS&T/SPIE Electronic Imaging 2002 Symposium, Conference on Storage and Retrieval for Media Databases 2002*, volume 4676, pages 22–30, January 2002.

[20] F. Alaya Cheikh, B. Cramariuc, M. Partio, P. Reijonen, and M. Gabbouj. Ordinal-Measure Based Shape Correspondence. *Eurasip Journal of Applied Signal Processing, Special Issue on Image Analysis for Multimedia Interactive Services - Part I*, 2002(4):362–371, April 2002.

[21] F. Alaya Cheikh, B. Cramariuc, C. Reynaud, M. Quinghong, B. Dragos-Adrian, B. Hnich, M. Gabbouj, P. Kerminen, T. Mäkinen, and H. Jaakkola. MUVIS: a System for Content-Based Indexing and Retrieval in Large Image Databases. In *SPIE/EI'99 Conference on Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 98–106, San Jose, California, January 1999.

[22] F. Alaya Cheikh, A. Quddus, and M. Gabbouj. Contour Based Object Recognition Using Wavelet-Transform. In *Proc. of the X European Signal Processing Conference, EUSIPCO 2000*, volume 2, pages 2141–2144, Tampere, Finland, September 4–8 2000.

[23] F. Alaya Cheikh, A. Quddus, and M. Gabbouj. Shape Recognition based on Wavelet-Transform Modulus Maxima. In *Proc. of the 7th IEEE International Conference on Electronics, Circuits and Systems (ICECS2K)*, pages 461–464, Beirut, Lebanon, December 2000.

[24] G. C.-H. Chuang and C.-C. J. Kuo. Wavelet Descriptor of Planar Curves Theory and Applications. *IEEE Transactions on Image Processing*, 5(1):56–70, May 1996.

[25] K. G. Coffman and A. Odlyzko. The Size and Growth Rate of the Internet. Technical Report 99-11, 21, 1999.

[26] I. Cox, M. Miller, T. Minka, T. Papathornas, and P. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Trans. On Image Processing*, 9(1):20–37, 2000.

[27] B. Cramariuc, I. Shmulevich, M. Gabbouj, and A. Makela. A New Image Similarity Measure Based on Ordinal Correlation. In *Proc. of the International Conference on Image Processing*, volume 3, pages 718–721, Vancouver, BC, Canada, September 2000.

[28] O. Cuisenaire. *Distance Transformations: Fast Algorithms and Applications to Medical Image Processing*. PhD thesis, Communications and Remote Sensing Laboratory, Université Catholique de Louvain, Louvain, Belgium, October 1999.

[29] L. da F. Costa and R. M. Cesar Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, 2001.

[30] A. Del Bimbo, V. Castelli, S.-F. Chang, and C.-S-Li. Guest editor's Introduction: Content-based Access of Image and Video Libraries. *Computer Vision and Image Understanding*, 75(1-2):1–2, 1999.

[31] L. Dunckley. *Multimedia Databases: An Object-Relational Approach*. Addison-Wesley, 2003.

[32] J. P. Eakins, J. M. Boardman, and M. E. Graham. Similarity Retrieval of Trademark Images. *IEEE Multimedia*, 5(2):53–63, April–June 1998.

[33] D. C. Fallside. XML Schema Part 0: Primer. Technical report, W3C Recommendation, http://www.w3.org/TR/xmlschema-0, May 2001.

[34] C. Faloutsos. *Searching Multimedia Databases by Content*. Kluwer Academic Publishers, 1996.

[35] D. Feng, W. C. Siu, and H. J. Zhang, editors. *Multimedia Information Retrieval and Management*. Springer, 2003.

[36] C. Fermuller and W.G. Kropatsch. Multi-Resolution Shape Description by Corners. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 271–276, June 1992.

[37] C. Fermuller and W.G. Kropatsch. Multiresolution Shape Description by Corners. In *MDSG94*, pages 539–548, 1994.

[38] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkhani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: The QBIC System. *IEEE Computer Magazine*, 28(9):23–32, September 1995.

[39] H. Freeman. Computer Processing of Line-Drawing Images. *ACM Computing Surveys*, 6(1):57–97, March 1974.

[40] M. Gabbouj, S. Kiranyaz, K. Caglar, B. Cramariuc, F. Alaya Cheikh, O.Guldogan, and E. Karaoglu. MUVIS: A Multimedia Browsing, Indexing and Retrieval System. In *Proc. of the IWDC 2002 Conference on Advanced Methods for Multimedia Signal Processing*, Capri, Italy, September 2002.

[41] M. Gabbouj, G. Morrison, F. Alaya Cheikh, and R. Mech. Redundancy Reduction Techniques and Content Analysis for Multimedia Services - The European COST 211 quat Action. In *Proc. of the Workshop on Image Analysis for Multimedia Interactive Services, (WIAMIS'99)*, pages 69–72, Berlin, Germany, May - June 1999.

[42] D. M. Gavrila. Pedestrian Detection from a Moving Vehicle. In *Proc. of the European Conference on Computer Vision*, pages 37–49, Dublin, Ireland, June - July 2000.

[43] D. M. Gavrila and V. Philomin. Real-time Object Detection using Distance Transforms. In *Proc. of the IEEE International Conference on Intelligent Vehicles*, pages 274–279, Stuttgart, Germany, 1998.

[44] Y. Gong. *Intelligent Image Databases: Towards Advanced Image Retrieval*. Kluwer Academic Publishers, 1998.

[45] L. Guan, T. Adali, S. Katagiri, J. Larsen, and J. Principe. Special Issue on Intelligent Multimedia Processing. *IEEE Transactions on Neural Networks*, 13(4), July 2002.

[46] E. Guldogan and O. Guldogan. Compression Effects on Content-based Multimedia Indexing and Retrieval Using Color. M.sc. thesis, Tampere University of Technology, Tampere, Finland, January 2003.

[47] C. Gurrin, A. F. Smeaton, H. Lee, K. McDonald, N. Murphy, N. O'Connor, and S. Marlow. Mobile Access to the Físchlár-News Archive. In F. Crestani, M. Dunlop, and S. Mizzaro, editors, *In Mobile HCI 2003 - 5th International Symposium on Human-Computer Interaction, Workshop on Mobile and Ubiquitos Information Access*, volume 2954, pages 124–142, Udine, Italy, September 2003. Springer. Series: Lecture Notes in Computer Science.

[48] M. H. Han and D. Jang. The use of Maximum Curvature Points for the Recognition of Partially Occluded Objects. *Pattern Recognition*, 23(1/2):21–23, 1990.

[49] R. M. Haralick, K. Shanmugam, and I. Dinstein. Texture Features for Image Classification. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-3(6):610–621, November 1973.

[50] Y. He and A. Kundu. 2D Shape Classification Using Hidden Markov Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1172–1184, 1991.

[51] C. Hildreth. The Detection of Intensity Changes by Computer and Biological Vision Systems. *Proc. of the Computer Vision, Graphics, and Image Processing*, 22:1–27, 1983.

[52] D. D. Hoffman and W. Richards. Parts of Recognition. Technical Report AIM-732, 1983.

[53] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image Indexing Using Color Correlograms. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPRŠ97)*, pages 762–768, San Juan, Puerto Rico, June 1997.

[54] J. Huang, S. Ravi Kumar, and M. Mitra. Combining Supervised Learning with Color Correlograms for Content-Based Image Retrieval. In *ACM Multimedia*, pages 325–334, 1997.

[55] T. S. Huang, S. Mehrotra, and K. Ramchandran. Multimedia Analysis and Retrieval System (MARS) Project. In *Proc. of the 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval*, 1996.

[56] A. Iftikhar, F. Alaya Cheikh, B. Cramariuc, and M. Gabbouj. Query by Image Content Using NOKIA 9210 Communicator. In *Proc. of the Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'01*, pages 132–137, Tampere, Finland, May 2001.

[57] A. Iftikhar, F. Alaya Cheikh, B. Cramariuc, and M. Gabbouj. Query by Image Content using Mobile Information Device Profile (MIDP) . In *Proc. of FINSIG'03 - 2003 Finnish Signal Processing Symposium*, pages 209–212, Tampere, Finland, May 2003.

[58] Y. Ishikawa, R. Subramanya, and C. Faloutsos. MindReader: Querying Databases Through Multiple Examples. In *Proc. of the 24th International Conference Very Large Data Bases, VLDB*, pages 218–227, New York City, USA, August 1998.

[59] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[60] J. J. Rocchio Jr. Relevance Feedback in Information Retrieval. In The SMART Retrieval System: Experiments in Automatic Document Processing. *Prentice Hall, Englewood Cliffs*, pages 313–323, 1971.

[61] R. L. Kashyap and R. Chellappa. Stochastic Models for Closed Boundary Analysis: Representation and Reconstruction. *IEEE Trans. on Information Theory*, 27(5):627–637, 1981.

[62] M. Kendall and J.D. Gibbons. *Rank Correlation Methods*. Edward Arnold, New York, 5th edition, 1990.

[63] P. Kerminen and M. Gabbouj. The Visual Goodness Evaluation of Color-based Retrieval Processes. In *Proc. of the EUSIPCO 2000*, pages 2153–2156, Tampere, Finland, September 5–8 2000.

[64] M. W. Koch and R. L. Kashyap. Using Polygon to Recognize and Locate Partially Occluded Objects. *IEEE Trans. PAMI*, 9:483–494, 1987.

[65] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scale. *Image and Vision Computing*, 10(8):557–565, October 1992.

[66] R. Koenen. Overview of the MPEG-4 Standard . Technical report, ISO/IEC JTC1/SC29/WG11 N4668,

http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm, March 2002.

[67] I. Kompatsiaris, E. Triantafillou, and M. G. Strintzis. Region-Based Colour Image Indexing and Retrieval. In *Proc. of the International Conference on Image Processing (ICIP2001)*, volume 1, pages :658 – 661, Thessaloniki, Greece, October 2001.

[68] M. Koskela. *Interactive Image Retrieval Using Self-Organizing Maps*. Ph.d. thesis, Helsinki University of Technology, Espoo, Finland, November 2003.

[69] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja. Self-Organizing Maps as a Relevance Feedback Technique in Content-Based Image Retrieval. *Pattern Analysis & Applications*, 4(2–3):140–152, June 2001.

[70] J. Laaksonen, M. Koskela, and E. Oja. PicSOM: Self-Organizing Maps for Content-Based Image Retrieval. In *Proc. of the International Joint Conference on Neural Networks (IJCNN'99)*, Washington, D.C., USA, July 1999.

[71] J. Laaksonen, M. Koskela, and E. Oja. PicSOM-Self-Organizing Image Retrieval with MPEG-7 Content Descriptors. *IEEE Transactions on Neural Networks*, 13(4):841–853, July 2002.

[72] J. T. Laaksonen, J. M. Koskela, S. P. Laakso, , and E. Oja. PicSOM - Content-based Image Retrieval with Self-Organizing Maps. *Pattern Recognition Letters*, 21(13–14):1199–1207, December 2000.

[73] C. Lee, W. Y. Ma, and H. J. Zhang. Information Embedding Based on UserŠs Relevance Feedback for Image Retrieval. In *SPIE IV International Conference on Multimedia Storage and Archiving Systems*, volume IV, pages 19–22, Boston, USA, September 1999.

[74] H. Lee, A. Smeaton, N. Murphy, N. O'Conner, and S. Marlow. Físchlár on a PDA: A Handheld User Interface to a Video Indexing, Browsing. In *UAHCI 2001 - International Conference on Universal Access in Human-Computer Interaction*, New Orleans, Luisiana, USA, August 2001.

[75] J. Lee, Y. Sun, and C. Chen. Multiscale Corner Detection by Using Wavelet Transform. *IEEE Transactions on Image Processing*, 4(1):100–104, January 1995.

[76] M. S. Lew, editor. *Principles of Visual Information Retrieval*. Springer-Verlag, 2001.

[77] J. C. Lin. The Family of Universal Axes. *Patter Recognition*, 29(3):477–485, March 1996.

[78] T. Lindeberg. Scale-Space for Discrete Signals. *IEEE Transactions on Pattern Analysis and Machine Vision*, 12(3):234–254, 1990.

[79] T. Lindeberg. Discrete derivative approximations with scale-space properties: A basis for low-level feature extraction. *Journal of Mathematical Imaging and Vision, (JMIV)*, 3:349–376, 1993.

[80] O. Guldogan M. Partio, E. Guldogan and M. Gabbouj. Applying Texture and Color Features to Natural Image Retrieval. In *Proc. of FINSIG'03 - 2003 Finnish Signal Processing Symposium*, pages 199–203, Tampere, Finland, May 2003.

[81] W. Y. Ma and B. S. Manjunath. Netra: A Toolbox for Navigating Large Image Databases. In *Proc. of the IEEE International Conference on Image Processing*, volume 1, pages 568–571, Washington, DC, USA, October 1997.

[82] W.-Y. Ma and B. S. Manjunath. NeTra: A Toolbox for Navigating Large Image Databases. *Multimedia Systems*, 7(3):184–198, 1999.

[83] S. MacArthur, C. Brodley, and C. Shyu. Relevance Feedback Decision Trees in Content-based Image Retrieval. In *IEEE Workshop on Content-based Access of lmage and Video Libraries*, pages 68–72, South Carolina, USA, June 2000.

[84] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2nd edition, 1999.

[85] B. S. Manjunath, T. Huang, A. M. Teklap, and H. J. Zhang. Special Issue on Image and Video Processing for Digital Libraries. *IEEE Transactions on Image Processing*, 9(1), January 2000.

[86] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

[87] B. S. Manjunath, P. Salembier, and T. Sikora, editors. *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons Ltd., 2002.

[88] J. M. Martinez. MPEG-7 Overview. Technical Report, ISO/IEC JTC1/SC29/WG11, http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, March 2003.

[89] R. Mech. Objective Evaluation Criteria for 2D-Shape Estimation Results of Moving Objects. In *Proc. of the Workshop on Image Analysis for Multimedia Services (WIAMIS'2001)*, pages 23–28, Tampere, Finland, May 2001.

[90] R. Mech and F. Marqués. Objective Evaluation Criteria for 2D-Shape Estimation Results of Moving Objects. *EURASIP Journal of Applied Signal Processing, Special Issue: Image Analysis for Multimedia Interactive Services* , 2002(4):401–409, April 2002.

[91] B. M. Mehtre, M. Kankanhalli, and W. F. Lee. Shape Measures for Content Based Image Retrieval: a Comparison. *Information Processing and Management*, 33(3):319–337, 1997.

[92] T. Minka and R. Picard. Interactive Learning Using a "Society of Models". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-1996)*, pages 447–452, San Francisco, CA, USA, June 1996.

[93] F. Mokhtarian and A. K. Mackworth. Scale Based Description of Planar Curves. In *CSCSI84*, pages 114–119, 1984.

[94] F. Mokhtarian and A. K. Mackworth. Scale Based Description and Recognition of Planar Curvesand Two-Dimensional Shapes. *IEEE Transactions on Pattern Analysis and Machine Intellegence*, 8(1):34–43, September 1986.

[95] F. Mokhtarian and A. K. Mackworth. A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves. *IEEE Transactions on Pattern Analysis and Machine Intellegence*, 14(8):789–805, August 1992.

[96] F. Mokhtarian and R. Suomela. Robust Image Corner Detection Through Curvature Scale Space. *IEEE Transactions on Pattern Analysis and Machine Intellegence*, 20(12):1376–1381, December 1998.

[97] G.A. Orban. *Neuronal Operations in the Visual Cortex*. Springer-Verlag, Berlin, 1984.

[98] T. O'Rourke and R. Stevenson. Human Visual System Based Wavelet Decomposition for Image Compression. *Visual Communication Image Representation*, 6:109–121, 1995.

[99] P. Villegas P., X. Marichal, and A. Salcedo. Objective Evaluation of Segmentation Masks in Video Sequences. In *Proc. of the Workshop on Image Analysis for Multimedia Services (WIAMIS'1999)*, pages 85–88, Berlin, Germany., May/June 1999.

[100] T. V. Papathomas. Special Issue on Visual-Perception: Guest Editorial. *International Journal of Imaging Systems and Technology*, 7(2):63–64, 1996.

[101] M. Partio. Content-based Image Retrieval using Shape and Texture Attributes. M.sc. thesis, Tampere University of Technology, Tampere, Finland, November 2002.

[102] M. Partio, B. Cramariuc, M. Gabbouj, and A. Visa. Rock Texture Retrieval using Gray Level Co-occurrence Matrix. In *NORSIG-2002, 5th Nordic Signal Processing Symposium*, On Board Hurtigruten M/S Trollfjord, Norway, October 2002.

[103] G. Pass, R. Zabih, and J. Miller. Comparing Images Using Color Coherence Vectors. In *Proc. of the 4th International ACM Multimedia Conference (ACM MM Š96)*, pages 65–73, Boston, MA, USA, November 1996.

[104] A. Pentland, R. W. Picard, and S. Sciaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.

[105] E. Persoon and K. S. Fu. Shape Discrimination Using Fourier Descriptors. *PAMI*, 8(3):388–397, May 1986.

[106] R. Picard. Toward a Visual Thesaurus. MIT Media Lab Perceptual Computing Technical Report 358, MIT Media Lab, 1995.

[107] M. K. Pietikainen, editor. *Texture Analysis in Machine Vision*, volume 40. World Scientific, series in machine perception and artificial intelligence edition.

[108] M. Qinghong. Texture Analysis with Applications to Content-based Image Indexing and Retrieval. M.sc. thesis, Tampere University of Technology, Tampere, Finland, May 1999.

[109] A. Quddus, F. Alaya Cheikh, and M. Gabbouj. Wavelet-Based Multi-level Object Retrieval in Contour Images. In *Proc. of the Very Low Bit rate Video Coding (VLBV'99) workshop*, pages 43–46, Kyoto, Japan, October 1999.

[110] A. Quddus and M. Fahmy. Fast Wavelet-based Corner Detection Technique. *Electronics Letters*, 35:287–288, February 1999.

[111] A. Quddus and M. Gabbouj. Wavelet-based Corner Detection Technique Using Optimal Scale. *Pattern Recognition Letters*, 23(1-3):215–220, January 2002.

[112] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proc. of the IEEE*, volume 77, pages 257–286, February 1989.

[113] I. Ragnelmam. Neighborhoods for distance transformations using ordered propagation. In *Computer Vision Graphics and Image Processing, Image Understanding*, volume 56, pages 399–409, Orlando, FL, USA, November 1992. Academic Press, Inc.

[114] P. Reijonen. Shape Analysis for Content-Based Image Retrieval. M.sc. thesis, Tampere University of Technology, Tampere, Finland, October 2001.

[115] A. Rosenfeld and M. Thursten. Edge and Curve Detection for Visual Scene Analysis. *IEEE Transactions on Conputers*, 20(5):562–569, 1972.

[116] Y. Rui, T. Huang, and S. Chang. Image Retrieval: Current Techniques, Promising Directions and Open Issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, April 1999.

[117] Y. Rui and T. S. Huang. A Novel Relevance Feedback Technique in Image Retrieval. In *ACM Multimedia (2)*, pages 67–70, 1999.

[118] Y. Rui, T.S. Huang, and S. Mehrotra. Content-based Image Retrieval with Relevance Feedback in MARS. In *Proc. of the IEEE International Conference on Image Processing, ICIP '97*, pages 815–818, Santa Barbara, California, USA, October 1997.

[119] Y. Rui, T.S. Huang, S. Mehrotra, and M. Ortega. Automatic Matching Tool Selection via Relevance Feedback in MARS. In *Proc. of the 2nd International Conference on Visual Information Systems*, pages 109–116, San Diego, California, December 1997.

[120] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-based Image Retrieval. *IEEE Trans. on CSVT*, 8(5):644–655, September 1998.

[121] Y. Rui, A. She, and T. Huang. Modified Fourier Descriptors for Shape Representation – A Practical Approach. In *First International Workshop on Image Databases and MultiMedia Search*, Amesterdam, Holland, August 1996.

[122] J. C. Russ. *The Image Processing Handbook*. CRC, Springer and IEEE Press inc., 3rd edition, 1999.

[123] E. Saber and A. M. Tekalp. Region-based Shape Matching for Automatic Image Annotation and Query-by-Example. *Journal of Visual Communication and Image Representation*, 8(1):3–20, March 1997.

[124] M. Safar, C. Shahabi, and X. Sun. Image Retrieval By Shape: A Comparative Study. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 141–144, NY, USA, August 2000.

[125] G. Salton. *Automatic Text Processing*. Addison-Wesley Publishing Company, 1989.

[126] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1983.

[127] S. Santini. *Exploratory Image Databases*. Academic Press, 2001.

[128] S. Santini and R. Jain. Similarity Measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.

[129] S. Scarlogg, L. Taycher, and M. La Cascia. Image Rover: A Content-based Image Browser for the World Wide Web. In *Proc. of the IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 10–18, Puerto Rico, June 1997.

[130] C. Schmid and R. Mohr. Local Grayvalue Invariants for Image Retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):530–535, August 1997.

[131] S. Sclaroff, L. Taycher, and M. LaCascia. ImageRover: A Content-Based Image Browser for the World Wide Web. Proc. of the IEEE Workshop on Content-based Access of Image and Video Libraries 5, 1997.

[132] I. Shmulevich, B. Cramariuc, and M. Gabbouj. A Framework for Ordinal-based Image Correspondence. In *Proc. of the X European Signal Processing Conference, EUSIPCO-2000*, pages 1389–1392, Tampere, Finland, September 2000.

[133] D.C. Dimitroff S.K. Chang, C.W. Yan and T. Arndt. An Intelligent Image Database System. *IEEE Transactions on Software Engineering*, 14(5):681–688, May 1988.

[134] J. R. Smith. *Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression*. Ph.d. thesis, Graduate School of Arts and Sciences, Columbia University, Columbia, USA, 1997.

[135] J. R. Smith and S.-F. Chang. Local Colour and Texture Extraction and Spatial Query. In *Proc. of the IEEE International Conference on Image Processing*, volume 3, pages 1011–1014, Lausanne, Switzerland, September 1996.

[136] J. R. Smith and S.-F. Chang. An image and video search engine for the world-wide web. In *Proc. of the SPIE Storage and Retrieval for Image and Video Databases*, volume 3022, pages 85–95, San Jose, CA, USA, February 1997.

[137] J. R. Smith and S.-F. Chang. Querying by color regions using the VisualSEEk content-based visual query system. Intelligent Multimedia Information Retrieval 159–173, MIT Press, 1997.

[138] M. Stricker and M. Orengo. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases III (I&ST/SPIE)*, volume 2420, pages 381–392, San Jose, CA, USA, February 1995.

[139] M. A. Stricker. Bounds for the Discrimination Power of Colour Indexing Techniques. In *Storage and Retrieval for Image and Video Databases II, (I&ST/SPIE)*, volume 2185, pages 15–24, San Jose, CA, USA, February 1994.

[140] M. Swain and D. Ballard. Colour indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[141] M. Swain, C. Frankel, and V. Athitsos. WebSeer: An image search engine for the world wide web. Technical report tr-96-14, University of Chicago Department of Computer Science, July 31 1996.

[142] H. Tamura and N. Yokoya. Image Database Systems: A Survey. *Pattern Recognition*, 17(1):29–49, 1984.

[143] C.H. Teh and R. T. Chin. On the Detection of Dominant Points on Digital Curves. *IEEE Trans. PAMI*, 11:859–872, 1989.

[144] P. J. Toivanen. New Geodesic Distance Transforms for Gray-Scale Images. *PRL*, 17(5):437–450, May 1996.

[145] S. Tong and E. Chang. Support Vector Machine Active Leaning for Image Retrieval. In *Proc. of the 9th ACM Conference on Multimedia*, Ottawa Canada, 2001.

[146] M. Trimeche. Shape Representations for Image Indexing and Retrieval. M.sc. thesis, Tampere University of Technology, Tampere, Finland, May 2000.

[147] M. Trimeche, F. Alaya Cheikh, and M. Gabbouj. Similarity Retrieval of Occluded Shapes Using Wavelet-Based Shape Feature. In *Proc. of the SPIE Conference on Internet and Multimedia Management Systems*, volume 4210, pages 281–289, Boston, USA, November 2000.

[148] M. Trimeche, F. Alaya Cheikh, M. Gabbouj, and B. Cramariuc. Content-based Description of Images for Retrieval in Large Databases: MUVIS. In *Proc. of the X European Signal Processing Conference, EUSIPCO-2000*, volume 1, pages 139–142, Tampere, Finland, September 2000.

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

[150] N. Vasconcelos and A. Lippman. Learning Over Multiple Temporal Scales in Image Databases. In Springer, editor, *Proc. the 6th European Conference on Computer Vision (ECCV)*, volume 1, pages 33–47, Dublin, Ireland, June - July 2000.

[151] R. Veltkamp and M. Tanase. Content-Based Image Retrieval Systems: A Survey. Technical Report 34, Utrecht University, Information and Computing Sciences, Utrecht, The Netherlands, 2001.

[152] R. C. Veltkamp. Shape Matching: Similarity Measures and Algorithms. In *Proc. the International Conference on Shape Modeling and Applications* , pages 188–197, Genova, Italy, May 2001.

[153] D. De Vleeschauwer, P. de Smet, F. Alaya Cheikh, R. Hamila, and M. Gabbouj. Optimal Performance of the Watershed Segmentation of an Image Enhanced by Teager Energy Driven Diffusion. In *Proc. of the VLBV98 workshop*, pages 137–140, Urbana, IL, USA, October 1998.

[154] R. Barber W. Niblack, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying Omages by Content Using Colour, Texture, and Shape. In *Proc. of the IS&T SPIE Storage and Retrieval for Image and Video Databases*, pages 173–187, San Jose, CA, USA, January 1993.

[155] A. Witkin. Scale Space Filtering. In *Proc. of the International Joint Conference on Artificial Intelligence*, volume 2, pages 1019–1023, Washington, DC, USA, August 1983.

[156] P. Wu and B. S. Manjunath. Adaptive Nearest Neighbour Search for Relevance Feedback in Large Image Database. In *Proc. of the 9th ACM Conference on Multimedia*, pages 89–97, Ottawa, Ontario, Canada, October 2001.

[157] Y. Wu, Q. Tian, and T. Huang. Discriminant-EM Algorithm with Application to Image Retrieval. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, volume 1, pages 222–227, South Carolina, USA, June 2000.