TA
1637
J37
2006

AUTOMATIC CONTENT-BASED IMAGE RETRIEVAL USING HIERARCHICAL CLUSTERING ALGORITHMS

by

KAMBIZ JARRAH
B.Eng. Electrical Engineering
Ryerson University, Canada, 2004

A thesis presented to
Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2006

© Kambiz Jarrah, 2006

UMI Number: EC53500

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI Microform EC53500
Copyright 2009 by ProQuest LLC
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

(Kambiz Jarrah)

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

(Kambiz Jarrah)

Borrower's Page

Ryerson University requires the signatures of all persons using or photocopying this thesis.
Please sign below, and give address and date.

[illegible]

ABSTRACT

AUTOMATIC CONTENT-BASED IMAGE RETRIEVAL USING HIERARCHICAL CLUSTERING ALGORITHMS

©Kambiz Jarrah 2006

Master of Applied Science
Department of Electrical and Computer Engineering
Ryerson University

The overall objective of this thesis is to present a methodology for guiding adaptations of an RBF-based relevance feedback network, embedded in automatic content-based image retrieval (CBIR) systems, through the principle of unsupervised hierarchical clustering. The main focus of this thesis is two-fold: introducing a new member of Self-Organizing Tree Map (SOTM) family, the Directed self-organizing tree map (DSOTM) that not only provides a partial supervision on cluster generation by forcing divisions away from the query class, but also presents an objective verdict on resemblance of the input pattern as its tree structure grows; and using a base-10 Genetic Algorithm (GA) approach to accurately determine the contribution of individual feature vectors for a successful retrieval in a so-called “feature weight detection process.”

The DSOTM is quite attractive in CBIR since it aims to reduce both user workload and subjectivity. Repetitive user interaction steps are replaced by a DSOTM module, which adaptively guides relevance feedback, to bridge the gap between low-level image descriptors and high-level semantics. To further reduce this gap and achieve an enhanced performance for the automatic CBIR system under study, a GA-based approach was proposed in conjunction with the DSOTM. The resulting framework is referred to as GA-based CBIR (GA-CBIR) and aims to import human subjectivity by automatically adjusting the search process to what the system evolves “to believe” is significant content within the query. In this engine, traditional GA operators work closely with the DSOTM to better attune with the apparent discriminative characteristics observed in an image by a human user.

Acknowledgments

I would sincerely like to thank my supervisors, Dr. Ling Guan and Dr. Sri Krishnan, for their kind encouragement, enlightening comments and suggestions, and compassionate support throughout the course of this thesis. This research work would have been impossible without their feedback, patience, and kindness.

In addition, I would like to thank the members of Ryerson Multimedia Research Laboratory (RML) and Signal Analysis Research Group (SAR) at Ryerson University for creating a friendly and congenial environment. Special appreciation to my friend and colleague, Mr. Matthew Kyan, visiting scholar from the School of Electrical and Information Engineering, University of Sydney, Australia, for many in depth discussions that have given inspiration to much of this work. Also, I would like to thank my editors, Ms. Phaedra Downer and Mr. Jason Argyris, for their kind comments on the structure of this thesis.

Much gratitude goes to my parents, my beautiful girlfriend, and the rest of my family. Without their love, support, and encouragement, the completion of this thesis would not be possible. Also, in memory of my grandfather who passed away right before completion of this thesis.

This work was funded by National Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant, Ryerson Graduate Student Scholarship, Canada Research Chair program, and Canada Foundation for Innovation (CFI).

Contents

Abstract	iv
Acknowledgements	v
List of Figures	viii
List of Tables	x
1 Introduction	1
1.1 General Background	1
1.2 Text-Based Retrieval of Visual Information	2
1.3 Content-Based Retrieval of Visual Information	3
1.4 Contribution of the Thesis	4
1.4.1 The Directed Self-Organizing Tree Map (DSOTM)	4
1.4.2 The Genetic Algorithm (GA) based Feature Weight Detection	5
1.5 Organization of Thesis	5
2 Background	7
2.1 Content-Based Image Retrieval (CBIR)	8
2.1.1 Visual Content Descriptors	11
2.1.1.1 Color Descriptors	11
2.1.1.2 Texture Descriptors	13
2.1.1.3 Shape Descriptors	20
2.1.2 Vector Normalization	25
2.1.3 Feature Selection	27
2.1.4 Proximity Measures	30
2.2 Interactive Approach in CBIR	31
2.2.1 Query Modification	35
2.2.2 Non-linear Radial Basis Function	36
2.3 Summary	38

3	Hierarchical Clustering Algorithms and Their Application in Automatic CBIR	40
3.1	Principles of Self-Organization	41
3.1.1	Synaptic Self-Amplification and Competition	42
3.1.2	Co-operation	42
3.1.3	Knowledge Through Redundancy	43
3.2	Unsupervised Learning and Self-Organization	44
3.2.1	The Self-Organizing Feature Map (SOFM)	44
3.2.2	The Self-Organizing Tree Map (SOTM)	46
3.3	The Directed Self-Organizing Tree Map (DSOTM)	51
3.4	Experimental Results: Unsupervised Learning Algorithms in the CBIR Application	60
3.5	Summary	64
4	Genetic Algorithm and Its Application in Automatic CBIR	66
4.1	The Genetic Algorithm (GA)	67
4.1.1	Genetic Operators	68
4.1.1.1	Selection	68
4.1.1.2	Reproduction	69
4.1.1.3	Replacement	71
4.1.1.4	Termination Criteria	72
4.1.2	The GA Algorithm	72
4.2	Genetic Algorithm and Automatic CBIR System (GA-CBIR)	74
4.2.1	The GA-CBIR Building Structure	76
4.2.2	The GA-based Feature Weight Detection Algorithm	80
4.3	Experimental Results: Feature Weight Detection in the GA-CBIR Application	82
4.4	Summary	86
5	Conclusions	93
5.1	Summary of Thesis	93
5.2	Future Research Extension	94
	Bibliography	97
A	List of Publications	108

List of Figures

1.1	Google and Yahoo image search engines	2
2.1	Rank-based CBIR system.	9
2.2	Example: CBIR system using image ranking mechanism	10
2.3	Color feature extraction: Color Histograms	14
2.4	Texture feature extraction: Wavelet moments	15
2.5	Gabor filter dictionary	18
2.6	Texture feature extraction: Gabor wavelets	19
2.7	Shape feature extraction: Fourier descriptors	20
2.8	Shape feature extraction: Hu's Seven Moment Invariants	24
2.9	Linear vector normalization	26
2.10	Gaussian vector normalization	26
2.11	Nonlinear Gaussian vector normalization	27
2.12	Interactive CBIR Engine - The flowchart	33
2.13	Query adaptation and RBF learning - The flowchart	34
2.14	The query modification model	36
2.15	Nonlinear Gaussian RBF	37
2.16	The RBF network model	38
3.1	The SOFM network model	44
3.2	The SOTM- vs. the SOFM-based data clustering	47
3.3	The SOTM Flowchart in the CBIR application	49
3.4	Polythetic Clustering approach in the SOTM application	51
3.5	The DSOTM- vs. the SOTM-based data clustering	53
3.6	The DSOTM Flowchart in the CBIR application	55
3.7	Ranking function, $\beta(\mathbf{z}, \mathbf{x}, t)$, in the DSOTM architecture	56
3.8	Example: the 2-D mapping of the DSOTM classifier	58
3.9	Timing Diagram of the DSOTM-based classification	59
3.10	Machine-controlled CBIR system	61
3.11	Machine-controlled CBIR system - A closer look	61
3.12	Example: Machine-controlled CBIR system using SOFM	63
3.13	Example: Machine-controlled CBIR system using SOTM	64
3.14	Example: Machine-controlled CBIR system using DSOTM	65

4.1	Genetic operator: Selection	68
4.2	Genetic operator: Single-point crossover	70
4.3	Genetic operator: Two-point crossover	70
4.4	Genetic operator: Uniform crossover	71
4.5	Genetic operator: Mutation	72
4.6	The Genetic Algorithm - The Flowchart	73
4.7	Example: GA-based feature weight detection	75
4.8	Feature weight detection in GA-CBIR engine	77
4.9	GA-CBIR system - The flowchart	79
4.10	GA-CBIR system - A closer look	79
4.11	The CBIR performance in presence of various crossover probabilities	83
4.12	Study of effects of the GA parameters on its convergence	92

List of Tables

2.1	Visual content descriptors used in this thesis.	11
2.2	Summary of content descriptors and their feature lengths.	25
3.1	Experimental results in terms of RR - The Automatic CBIR	62
4.1	Parameter setting for the GA	83
4.2	Experimental results in terms of RR - The GA-CBIR	85

Chapter 1

Introduction

1.1 General Background

WITH the recent developments in the computer technology, collecting visual information in digital format has become more and more feasible. Digital scanners and cameras as well as powerful personal computers with vast storage capabilities are some of the available tools that enable users to create and store visual data effortlessly and at very affordable prices. Moreover, the boost in hardware technology in recent years has enabled computers to manage and manipulate stored data through more complex algorithms and on larger blocks of data to facilitate maximizing efficiency and minimizing required processing time.

Multimedia processing is a comprehensible model of today's technological advance. Multimedia, in general, aims to provide individual users with high-levels of control and interaction while sustaining some degree of intelligence in terms of being able to formulate and imitate users' preferences through its learning paradigm. Transition from the traditional text-based to the modern content-based approaches for indexing and retrieval of tera-bytes of online information is one of the most evident examples of the integration of multimedia with computer technologies. This chapter provides a brief explanation of the trends behind conventional text-based retrieval of audiovisual data, mainly images, and the necessity of transition to the modern content-based image retrieval approaches in Sections 1.2 and 1.3. Sections 1.4 and 1.5 provide an overview of the contribution and organization of this thesis.

Figure 1.1: Query “Rose”: (a) Google and (b) Yahoo image search engines.

1.2 Text-Based Retrieval of Visual Information

The text-based retrieval is an extension of modern information retrieval (IR) systems that aim to annotate and retrieve textual documents automatically, without human intervention. Modern IR systems are dependent on text contents to index and retrieve documents instead of relying on a few standard textual clues as were used in traditional cataloguing systems [1]. Text-based retrieval systems, on the other hand, rely on both document contents and meta-data to search audiovisual collections. Meta-data provide crucial information about stored textual documents including file name, file type, file size, etc. Figure 1.1 illustrates two of the most frequently used text-based image search engines from Google [2] and Yahoo [3] corporations and demonstrates their behaviors with respect to the provided query. In this figure the top relevant images are retrieved based on the position of the query near the images, or in the image captions, or in the links to the images interpreted as the actual “flower” or a “person” identified as “Rose”. Text-based image retrieval engines, such as

Google and Yahoo, assume direct associations between descriptive key-words and image contents regardless of visual and conceptual discrepancies (low- or high-level) between the query and retrieved images.

Although manual annotation of audiovisual data files, as is used by the current text-based retrieval systems, has proven to be an effective model for content description, it suffers from several major drawbacks: It is a laborious method especially for indexing large visual databases while raising the possibility of unreliable image interpretations as well as inadequate content description. Content-based image annotation, on the other hand, has received considerable research and commercial interest in the recent years due to its ability to index audiovisual data directly based on their content information while providing new opportunities and challenges. Among those opportunities, is the possibility of integrating low-level audiovisual features (such as pitch, energy, color, texture, shape, layout, motion field) as well as mid-level features (such as prosody, trajectories) with the established keyword search. This represents a significant improvement in the field of multimedia processing.

1.3 Content-Based Retrieval of Visual Information

The term “CBIR” originated after the influential work of David Marr [4] whose work captured computer vision research for the last two decades. According to Marr, the primitive image features including edges, angles, color, spatial proximity, etc, are adequate for a machine to infer sufficient meaning for limited image understanding, pattern recognition, and retrieval. Even though machine perception still lags far behind human visual perception, current research in CBIR offers an improved architecture *via* minimizing the gap between image content understanding and machine visual perception.

Image content understanding, in the majority of the existing CBIR techniques, is limited to relatively low-level features that provide statistical measures of image contents and often fall short of higher-level interpretation and knowledge. Recently, researchers have begun to focus on bridging the semantic and conceptual gap by integrating the human perception and

scene interpretation models inspired from psychology and linguistics in the search process [5].

In his book, Forsyth et al. [6] defines machine visual perception (or vision) as an enterprise that uses statistical methods to disentangle data using models constructed with the aid of physics, geometry, and learning theory. Thus, content-based image retrieval relies on a solid understanding of image contents *via* computer vision and content understanding techniques.

1.4 Contribution of the Thesis

In order to reduce the human subjective error that has resulted from the over-involvement of users in the interactive CBIR systems and to tackle the obstacles in the development of modern multimedia retrieval systems, a combination of machine intelligence and optimization techniques is adapted in this work. Machine intelligence is critical for image retrieval application since it imports *objectivity* to the search process. On the other hand, optimization techniques (especially evolutionary process) are important for discovering significant contents embedded in an image and injecting some degree of *subjectivity* into the image retrieval process. Below is a summary of the contributions to this field:

1.4.1 The Directed Self-Organizing Tree Map (DSOTM)

The DSOTM is an unsupervised machine learning algorithm and is inspired by principles found in the Self-Organizing Feature Map (SOFM) architecture. DSOTM aims to classify input patterns while preserving the integrity of the image clusters, mainly in the query class. One of the primary advantages of this learning paradigm is its tree-like configuration that develops in a direction that prevents any unnecessary cluster centers to be generated in low- or zero-density regions of the feature space. As a result, it forms a more plastic structure in comparison with its two other counterparts, the SOFM and the Self-Organizing Tree Map (SOTM). In this thesis, DSOTM is incorporated with CBIR technology in order to perform

the required decision makings about the relevance of individual images through an automate image retrieval architecture.

1.4.2 The Genetic Algorithm (GA) based Feature Weight Detection

Image retrieval is distinct from other pattern recognition problems because of two main reasons: 1) high dimensionality of input patterns, and, 2) feature space sparing, resulting from small but highly inter-related samples. DSOTM can successfully address the second problem *via* its deformable structure and its node insertion mechanism. The first problem can be addressed by employing the genetic algorithm. The GA is one of the most popular optimization techniques among a set of evolutionary processes since it is quite effective in the meticulous global search of large and nonlinear spaces. Due to the ability of the GA to manage multi-constraint optimization problems and due to its capacity to handle closely related features, the GA is used for the purpose of feature weight detection in this thesis.

1.5 Organization of Thesis

The remainder of the thesis is organized in the following chapters:

Chapter 2 - Background: This chapter provides a comprehensive study on the previous developments in the area of CBIR. The mathematical representation of the various feature extraction techniques including color, texture, and shape content descriptors is presented here. In addition to touching on the pre-processing methods (including feature normalization and selection) this chapter includes an explanation of proximity measurement and query modification techniques accompanied with a brief explanation of one of the previously developed interactive CBIR engines.

Chapter 3 - Hierarchical Clustering Algorithms and Their Application in Automatic CBIR: This chapter introduces four major principles of self-organization: Synaptic

Self-Amplification, Synaptic Competition, Co-operation, and Knowledge Through Redundancy. This chapter then presents three fundamental realizations of these regulations: the Self-Organizing Feature Map (SOFM), the Self-Organizing Tree Map (SOTM), and the Directed Self Organizing Tree Map (DSOTM) algorithms. A comprehensive study on DSOTM, its structure, and its effectiveness in automating the search process in the CBIR engines is also discussed in this chapter.

Chapter 4 - Genetic Algorithm and Its Application in Automatic CBIR: An overview of the genetic algorithm, its operators, and its principles are covered in this chapter, along with the introduction of a new architecture of CBIR engines, namely, the GA-CBIR. In this design, GA interacts closely with DSOTM in order to discover significant contents within query images through the principle of feature weight detection.

Chapter 5 - Conclusions: Recommendations on the current approach followed by future research directions are presented in this chapter.

Chapter 2

Background

WITH the enormous growth of digital multimedia data in recent years, multimedia processing techniques and related technologies are also rapidly evolving. In considering the current research directions in this area, it is evident that multimedia processing provides a bridge between the multimedia, in the traditional sense, and signal processing models. In signal processing, the objective is to understand the behavior of a particular type of signal, which can be from any type of sensor (as opposed to traditional audiovisual alone). The trend in multimedia processing is to bring together concepts from signal processing in a way that fuses different modalities (from traditional audiovisual signals and beyond) into more semantic based models (higher level concepts) that aid in understanding the information. This trend is observable through the evolution from DCT-based compression of audiovisual signals (MPEG-1 and MPEG-2) toward content-based description standards for semantic multimedia management (MPEG-7) [7]. Despite many technical developments in this field, automation of certain processes - whether it be feature extraction and labeling, object detection and segmentation, content exploration and retrieval, etc - on some semantic level still remains a major challenge.

With the growth of digital visual collections in recent years, the need for effective search tools for accessing multimedia data is also rapidly rising. Content-based image retrieval (CBIR) is regarded as one of the most effective ways of accessing visual data [8] - [26] and has applications in numerous fields including distance education, business, entertainment,

law enforcement, medicine, etc. Along with this, the development of multimedia retrieval systems is widely recognized as one of the most promising fields in the area of multimedia data management. The main objective of this chapter is to provide a comprehensive study on modern approaches in the area of image indexing and retrieval. Section 2.1 presents an overall description of the structure of the CBIR system used in this work. A comprehensive study on the interactive approach in CBIR is provided in Section 2.2. Section 2.3 concludes this chapter with some remarks.

2.1 Content-Based Image Retrieval (CBIR)

With the development of the internet, visual information has been widely used in engineering and scientific applications. Images and video are being captured, categorized, and made available on large online databases. Therefore, it is extremely important to develop efficient techniques to characterize, search, and retrieve digital images in large visual digital libraries [27].

Content-based image retrieval (CBIR) - also known as query by image content (QBIC) and content-based visual information retrieval (CBVIR) - have shown to behave effectively in finding a set of images whose contents are similar to, or even match with, a given query from within a large image database. These systems rely on characterization of images based on their visual content. These visual contents are inspired by the human visual system and the way it interprets image contents. Capturing all aspects of characteristics of the human visual system is an important, but currently impossible, task in designing effective CBIR systems. Moreover, finding a set of descriptors that effectively represent the image content has always been an important research area in CBIR.

Visual content are generally captured by a multi-dimensional vector representation of an image (feature space) and can be extracted from images through certain mathematical and statistical analyzes. Features describing image content can be classified as either *General* or *Specific* based on their level of content interpretation. General features include low-level

Figure 2.1: Rank-based CBIR system.

descriptors such as color, texture, and shape that offer a general description of an image. These features may be similar among semantically different images with similar low-level feature distributions. Specific features, on the other hand, refer to more advanced (higher-level) descriptors and comprise a more detailed representation of an image. These features are application dependent and may include, for instance, human emotion, specifically, expressive or face annotative features.

Figure 2.1 illustrates the standard structure of a CBIR engine. Image indexing and Retrieval are two major components in the current CBIR technology. Features from all images in the database are extracted and stored in the indexing phase. Usually due to an extensive number of images in a digital library and, thus, exhaustive computational demand for extracting their features, these descriptors are pre-computed (off-line) and stored in a feature database. This will reduce the retrieval delay significantly. Retrieval phase can be further viewed as a combination of on-line query feature extraction, similarity matching, and image ranking stages. This phase is initialized by extracting query features on the fly. Subsequently, extracted features from the query image are being compared with all of the pre-computed feature vectors in the database and then are ranked based on their degree of resemblance with respect to the query. Top Q ranked images will be stored and then

Figure 2.2: A sample query (top-left) to demonstrate behavior of the rank-based CBIR system.

displayed back to the user.

The CBIR system with the above architecture simply neglects the semantic resemblance among target samples by searching and retrieving images according to degree of (statistical) similarities between the query and its neighboring images. Consequently, this system assumes direct association between statistical similarities and semantics of the query image and disregards inter-relationships among target images. Figure 2.2 clearly illustrates the problem of such an approach. In this figure, the query image is located in the top-left corner of the figure and the top 16 ($Q = 16$) images are ranked and retrieved - from left to right, and top to bottom - according to the decaying level of their likeness with respect to the query. It is evident that target images with high feature similarities may not be semantically similar to the query image due to the gap between low-level features used for image indexing and the high-level concepts used by human observers. This problem will be addressed shortly through introducing the relevance feedback-based machine learning technique. Before that, the four major components of all CBIR systems: visual description, vector normalization,

Table 2.1: Visual content descriptors used in this thesis.

Color Descriptors	Texture Descriptors	Shape Descriptors
Color Histograms	Wavelet Moments	Hu's Moment Invariants
Color Moments	Gabor Filters	Fourier Descriptors

feature subset selection, and proximity measurement, are presented in detail in the following subsections.

2.1.1 Visual Content Descriptors

Visual Contents provide a content description of still images or video streams and enable computers to browse, identify, and categorize visual data purely based on their content information, with or without the aid of textual clues.

The concept of visual content can be interpreted as low-level features such as color, shape, and texture. These features provide a global description of an image without regard to what it precisely presents within the scope of its object(s) of interest. In order to achieve a complete description of an image, some kind of object recognition and scene interpretation mechanism is necessary. Unfortunately, this requires the crucial initial step of image segmentation, which is also a problem in itself. Although significant progressions have been made in the area of image segmentation, an accurate characterization of objects within images has not yet been achieved.

Table 2.1 summarizes the features used in the feature extraction modules of the CBIR system introduced in this work. Following subsections provide a brief overview on each of these visual descriptors.

2.1.1.1 Color Descriptors

Color is one of the most important visual attributes for human perception and is widely used in computer vision applications including image or video indexing and retrieval. Color

features are relatively resistant to any type of translation, orientation, and size variation of images or their comprised objects. Color descriptors are usually defined by three independent distributions in various color spaces including RGB, HSV, YC_bC_r , etc.

Color histograms are the most conventional way for representing color properties of an image and are obtained by discretizing image colors and counting the number of times each color occurs in an image array.

This method has a number of major advantages that make it attractive for our indexing purposes: 1) Only those colors that are presented in both images are contributed toward the similarity matching process; 2) The similarity matching process is independent of the number of bins used in the color histograms; 3) Histograms are invariant to translations and rotation about the viewing axis, and change only slowly under the change of the angle of view, change in scale and occlusion; and 4) Histograms, by themselves, do not include spatial information; thus, images with very different layouts may have similar histograms [28].

According to what Swain and Ballard described [28], the concept of similarity among the histograms of two images is achieved by the *Histogram Intersection* method. Histogram Intersection matches the color histogram of a query image with histograms of each of the models in the database. The higher the match value, the better the fit to query color contents. This model will be further discussed in section 2.1.4.

High volume of colors, used in computers to represent images, is one of the major drawbacks of this feature extraction scheme. Such a high number of colors can result in an inaccurate similarity estimation. Therefore, to reduce number of colors, a color space quantization method is required. Decision on the type of quantization process is an important issue regarding color histograms. A uniform quantization may be appropriate for the perceptually uniform color spaces (i.e., HSV), whereas, for perceptually non-uniform color spaces (i.e., RGB) a non-uniform quantization method is a better choice.

In this work, 48 - bin color histograms are used to describe the color contents of images in the HSV color space, where H and S are uniformly quantized into 16 and 3 regions, respectively. The V component is discarded due to its sensitivity to the lighting condition

as described in [29, 30]. Figure 2.3 illustrates the process of the color histogram generation.

In addition, a 9 - dimensional feature vector from mean, μ , standard deviation, σ , and skewness, ς , of the color distribution of each channel in the RGB color space was also generated to represent *color moment* descriptors [31].

2.1.1.2 Texture Descriptors

Texture is one of the most difficult and complex features to define when compared with other low-level descriptors, such as color and shape. Texture refers to the homogenous or nonhomogeneous property of visual patterns, and results from the presence of several colors or intensities in an image. Texture carries information about an object and specifies its relationship with its surroundings. For instance, texture can provide significant information about two overlapping objects by decoding information about their relative orientation and spatial depth.

Measuring similarities between two textural images is a difficult task. Two images can be considered to have equivalent textural information when both have identical spatial color arrangements while not necessarily containing similar color components.

There are several techniques available to extract textural information from an image. In his book, Manjunath et al. [7] has introduced three texture descriptors that are currently used in the MPEG-7 standard:

- a.* Homogenous texture descriptor (HTD),
- b.* Texture browsing descriptor (TBD), and
- c.* Edge histogram descriptor (EHD).

The HTD is an effective model in characterizing homogenous texture regions and offers a quantitative description of texture regions through the mean energy and energy deviation of their frequency channel sets. The TBD provides a visual characterization of texture regions and is inspired by the human perceptual ability in differentiating the regularity, coarseness, and directionality of these areas. This descriptor is quite effective in browsing applications

(a)

(b)

(c)

Figure 2.3: Color Histogram generation: (a) Original Image, (b) HSV Transformed Image, and (c) Color Histograms of the Original Image

(a)

(b)

Figure 2.4: Wavelet transform provides a spatial-frequency informative view at various resolution levels for the purpose of texture characterization: (a) Original Image, (b) Sub-images resulting from 3-level wavelet decomposition.

and coarse classification of textures. The EHD captures the spatial distribution of edges in an image, while supplying a mean to detect and capture image local-edge distributions. Moreover, because of EHD's tendency to characterize local image regions, this method is particularly useful in matching regions with partially non-uniform and non-homogenous texture.

Research on using wavelet transform to capture image textural information has been widely increasing after the introduction of wavelet transform in the early 1980s [32] - [35]. Wavelet is a method of representing images in various resolution levels (i.e., sub-images) and providing a spatial-frequency view at each level. This algorithm can therefore aid in evaluating distinct characteristics of an image by means of extracting its textural information. Wavelet possesses a spatial localization property which preserves a logical spatial relationship among image textural regions [36]. Also, due to the practical consideration that wavelet forms the core technology for the JPEG2000 standard [37], wavelet moments seem to be attractive for the purpose of image indexing and retrieval. In this work, a 20 - dimensional feature vector from mean, μ , and standard deviation, σ , of a three-level wavelet decomposed image (see Figure 2.4) was generated to decode its textural information [38].

In his paper, Ma et al. [39] proposed another method for texture representation of images based on Gabor wavelet transform technique. In this method, an image can be considered as a mixture of different textural regions and the image features associated with these regions can be used for search and retrieval purposes. These texture features can be found by calculating the mean, μ , and standard deviation, σ , of the Gabor filtered image.

Gabor filters are basically composed of a group of wavelets, in which each wavelet captures energy at a specific frequency and direction. Texture features are extracted from these energy distributions [40].

The 2 - dimensional Gabor transform function, $g(x, y)$, and its Fourier transform, $G(u, v)$, for a given image, $I(x, y)$, with size of $P \times Q$ can be calculated using Equations 2.1 and 2.2:

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jWx\right], \quad (2.1)$$

$$G(u, v) = \exp\left\{-\frac{1}{2}\left[\frac{(u - W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\}, \quad (2.2)$$

where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$ are scaling parameters. In particular, the impulse response of Gabor filters in the spatial domain, $g(x, y)$, is a 2-D Gaussian-shape kernel modulated by a complex sinusoidal plane-wave. The sinusoidal modulator is known as a

carrier and the Gaussian-shape kernel is called an envelope; W in the above equations defines the spatial frequency of the sinusoidal carrier.

Gabor functions form a complete, but not orthogonal, basis set and provide a localized frequency description of a signal. The nonorthogonality of the Gabor wavelet implies that there is redundant information in the filtered images, which must be reduced.

Gabor wavelet is a collection of self-similar functions defined through Equation 2.3:

$$g_{mn}(x, y) = a^{-m}G(x', y'), \quad (2.3)$$

where m and n are integers that specify the *scale* and *orientation* of the wavelets respectively. The scaling factor, a^{-m} , ensures that the energy is independent of m . x' and y' can be calculated as follows:

$$x' = a^{-m}(x \cos \theta + y \sin \theta), \quad (2.4)$$

$$y' = a^{-m}(-x \sin \theta + y \cos \theta), \quad (2.5)$$

where $\theta = n\pi/K$ and K is the total number of orientations. Variables in the above equations are defined as:

$$a = \left(\frac{U_h}{U_l}\right)^{-\frac{1}{S-1}}, a > 1, \quad (2.6)$$

$$\sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}}, \quad (2.7)$$

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right)[U_h - 2\ln\left(\frac{\sigma_u^2}{U_h}\right)][2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}]^{-\frac{1}{2}}, \quad (2.8)$$

where U_l and U_h are the lower and upper center frequencies of interest, and S is the number of scales in the multi-resolution decomposition. The following filter parameters were employed in this thesis: $U_l = 0.05$, $U_h = W = 0.4$, $K = 6$, and $S = 4$ [39].

Figure 2.5: Contours of the Gabor filter dictionary with Gabor wavelets of four scales and six orientations.

As it was mentioned before, the nonorthogonality of the Gabor wavelet implies redundancy in the response of the filter. This redundancy can be reduced by designing a filter in such a way that the half-peak magnitude of the filter responses, in the frequency spectrum, touch each other as shown in Figure 2.5.

Consequently, for a given image, $I(x, y)$, its Gabor Wavelet Transform can be extracted as:

$$W_{mn}(x, y) = \int I(x_1, y_1) g_{mn}^*(x - x_1, y - y_1) dx_1 dy_1, \quad (2.9)$$

where $*$ indicates the complex conjugate of Gabor wavelet itself. The mean, μ_{mn} , and the standard deviation, σ_{mn} , of the magnitude of the transformed coefficients can be calculated by Equations 2.10 and 2.12:

$$\mu_{mn} = \int \int |W_{mn}(xy)| dx dy, \quad (2.10)$$

$$\sigma_{mn} = [\int \int (|W_{mn}(x, y)| - \mu_{mn})^2 dx dy]^{1/2}. \quad (2.11)$$

(a)

(b)

(c)

(d)

Figure 2.6: Applying Gabor wavelet transform with a set of basis functions consist of Gabor wavelets spanning four scales and six orientations on an image: (a) Original Image (b) Gabor filters in spatial domain (c) Gabor filters in frequency domain, and (d) Gabor Transformed Image.

For Gabor feature extraction purposes, a matrix of statistical descriptive features can be created where its columns and rows are constructed from 48 - dimensional feature vectors using μ_{mn} and σ_{mn} for different images in the database. Figures 2.6(b) and (c) represent Gabor filters in the spatial and frequency domains respectively, while Figure 2.6(d) is an illustration of Gabor transformed image from the original image in part (a) [40].

Figure 2.7: Boundary parameters of an edge image.

2.1.1.3 Shape Descriptors

Shape provides powerful visual clues about the notion of similarity within objects and can be instantly detected by a human observer. These visual clues are identified by the topological closeness of lines and edges in space. Objects can be solely recognized by their shape information. This characteristic of the shape feature makes it essentially distinct from other types of descriptors including color and texture. Moreover, representation of shape requires some kind of regional information about an image, while both color and texture capture the global attribute of an image.

Shape features can be classified into two distinct categories: *boundary-based* or *region-based*. Boundary-based shape descriptors represent an object based on the explicit representation of its outer boundaries. Region-based descriptors represent the shape in more detail by taking both the objects' boundary and internal regions into consideration. Region-based descriptors are usually able to capture and describe complex attributes of objects with multiple disconnected or overlapping regions [7].

Fourier descriptors (FD) [41] - [49] and Hu's seven moment invariants (HSMI) [50, 51] are among the most popular shape-based image descriptors and have been used in various applications, such as recognition, indexing, and retrieval. Although FDs are boundary-based shape descriptors, they are not directly invariant to image transformations. On the

other hand, Hu's seven moment invariants are region-based image descriptors and have these invariance properties by their very nature [38].

Consider an image boundary region with K samples reside on the coordinate (x_k, y_k) where $k = 1, \dots, K - 1$. For each pair of (x_k, y_k) , a set of complex variables can be defined:

$$u_k = x_k + jy_k. \quad (2.12)$$

Figure 2.7 illustrates these boundary parameters associated with an edge image in the Cartesian coordinate. The FD for these K u_k points on the boundary are defined by Equation 2.13. Usually, a small number of FD descriptors provide sufficient discriminatory information about the boundaries of an object.

$$FD_l = \sum u_k \exp(-j \frac{2\pi}{K} lk), l = 0, \dots, K - 1. \quad (2.13)$$

Fourier shape parameters were extracted using Sobel filters for the purpose of edge detection due to their low computational cost. The resulting edge parameters were then converted from the Cartesian coordinate to the polar coordinate and the Fast Fourier Transform (FFT) was then applied and the low frequency components were extracted to form a 10 - dimensional feature vector. One of the benefits of this coordinate conversion is to take advantage of the scale invariance property of the polar coordinate system.

HSMI shape features are defined indirectly through 2 - dimensional *geometric moments* described in terms of Riemann integrals:

$$m_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^i y^j I(x, y) dx dy, \quad i, j = 1, 2, \dots \quad (2.14)$$

where m_{ij} is the $(i + j)^{th}$ order Regular moment of the continues image function, $I(x, y)$.

Geometric moments are popular features for pattern recognition since they offer an equivalent representation of an image and provide rich information about its shape, size, and orientation [52].

An image can be reconstructed from its moments [53]. The zero-order moment, m_{00} , represents the total image power and can be used in conjunction with the first-order moments,

m_{10} and m_{01} , to locate the centroid of the image $I(x, y)$; while the second-order moments, m_{20} , m_{02} , and m_{11} , characterize size and orientation of an image.

Regular moments must be normalized in such a way that invariance properties can be established since they can not be directly used for characterizing the shape of objects. Invariance to position of an object can be achieved by using central moments of an image. Central moments, m_{ij} are defined as:

$$m_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^i (y - \bar{y})^j I(x, y) dx dy, \quad (2.15)$$

where $\bar{x}$ and $\bar{y}$ are centroid of the image and are defined as follows:

$$\begin{aligned} \bar{x} &= m_{10}/m_{00}, \\ \bar{y} &= m_{01}/m_{00}. \end{aligned} \quad (2.16)$$

Scale and contrast invariance can be achieved through normalized central moments. The normalized central moments, η_{ij} , are defined as:

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{(i+j+2)/2}}, \quad i, j = 2, 3, \dots \quad (2.17)$$

Using different combinations of the normalized central moments based on the theory of algebraic invariants, Hu introduced seven moments:

$$\begin{aligned} \vartheta_1 &= \eta_{20} + \eta_{02}, \\ \vartheta_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \\ \vartheta_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \\ \vartheta_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \\ \vartheta_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], \\ \vartheta_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ &\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}), \\ \vartheta_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]. \end{aligned} \quad (2.18)$$

Calculating the moments of any order is possible using the algebraic invariants method, but computing the higher order of Hu's moments is much more complex. Likewise, the extraction of image shapes from higher order moments will be complex [51, 52].

In this work, the HSMI shape descriptors were extracted by converting color images into binary segmented images and then extracting shape parameters from resulting images using Equation 2.18. The binary image segmentation was performed through the following processes, as shown in Figure 2.8:

1. Resize the input image to a uniform number of pixels;
2. Convert the image to a gray scale image;
3. Create a binary image using a proper threshold value;
4. Detect edges of the original image using a proper filter;
5. Fill-out the regions (closing operation) of the edge image in (4);
6. Subtract the filled image region in (5) from the edge image in (4) and dilate the resulting image;
7. Add the dilated image to the gray scale image;
8. Create a binary image from the image in (7);
9. Perform morphological (Erosion and Dilation) operations to remove spurs, restore, and fill internal holes of the image in (8);
10. Perform the logic AND operation on image in (9) and (3) and resize the resulting image to its original size;
11. Label connected objects in the binary image in (10); and
12. Select region of interests from the labeled regions.

Table 2.2 summarizes content descriptors and their respective vector lengths.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.8: Propose binary image segmentation process: (a) Steps 1 and 2, (b) Step 3, (c) Step 4, (d) Steps 5 to 8, (e) Steps 9, and (f) Steps 10 to 13.

Table 2.2: Summary of content descriptors and their feature lengths.

Color Features	Texture Features	Shape Features
Color Histograms (48)	Wavelet Moments (20)	Hu's Moment Invariants (7)
Color Moments (9)	Gabor Filters (48)	Fourier Descriptors (10)

2.1.2 Vector Normalization

Features normalization is one of the most important steps before the similarity matching process. Individual feature vectors, used to provide low-level representation of images, are fused together to produce a complete set of features. The resulting feature vectors are not suitable for similarity matching since their distinct components (features) might have various magnitudes and dynamic ranges. Due to this variety in the feature vectors, the similarity measure will be biased toward the features with larger values. As a result, those features with lower values might even be discarded.

Vector normalization is a process of readjusting feature components to a range that ensures their equal contribution and significance to the process of similarity matching. This range is normally defined in $[0, 1]$ or $[-1, 1]$ by proper data scaling.

Consider an $M \times P$ feature set, $\mathbf{F}$, consisting of M image with the vector length of P :

$$\begin{aligned}\mathbf{F} &= [\mathbf{x}'_1, \dots, \mathbf{x}'_m, \dots, \mathbf{x}'_M]^T \\ &= [x'_{m,i}] \quad m = 1, \dots, M, \quad i = 1, \dots, P,\end{aligned}\tag{2.19}$$

where $\mathbf{x}'_m = [x'_{m,1}, \dots, x'_{m,i}, \dots, x'_{m,P}]^T$ represents feature vector of image m in the database.

One method for normalization is to linearly readjust the entries in each column, $\{x'_{m,i}\}_{m=1}^M$, to a same dynamic range of $[0, 1]$ under the influence of the maximum and minimum feature values of the sequence as is illustrated by Equation 2.20:

$$\mathbf{F}_{m,i} = \frac{x'_{m,i} - \min_i}{\max_i - \min_i}.\tag{2.20}$$

In this equation, $\max_i$ and $\min_i$ refer to largest and smallest feature value of the i^{th} column of matrix $\mathbf{F}$, respectively.

One of the major drawbacks of this method is its dependency on two extreme feature values. This dependency will not ensure an equal dynamic range for all the members in feature space by *linearly* biasing a larger portion of the range to higher feature values and leaving the rest of the elements with a very narrow range. For example, consider the following 3×3 input data and its normalized counterpart:

$$\begin{array}{ccc} \begin{bmatrix} 0.1000 & 0.0800 & 0.2000 \\ 0.4000 & 50.000 & 0.3300 \\ 0.0500 & 0.3000 & 0.1500 \end{bmatrix} & \begin{bmatrix} 0.0010 & 0.0006 & 0.0030 \\ 0.0070 & 1.0000 & 0.0056 \\ 0.0000 & 0.0050 & 0.0020 \end{bmatrix} \\ (a) & (b) \end{array}$$

Figure 2.9: Linear normalization: (a) Input data (b) The normalized sequence to the range of $[0, 1]$.

Clearly, this normalization puts more emphasis on outliers (i.e., 50) by devoting most of the $[0, 1]$ range to them and wrapping the rest in a very slight range.

This problem can be addressed by applying a Gaussian-based normalization scheme. By making the assumption that sequence $\{x'_{m,i}\}_{m=1}^M$ is a Gaussian sequence and distributed evenly around the mean, the following process normalizes features linearly to zero mean and unit standard deviation as is illustrated by Equation 2.21:

$$\mathbf{F}_{m,i} = \frac{x'_{m,i} - \mu_i}{\sigma_i}, \quad (2.21)$$

In the above equation, μ_i and σ_i are mean and standard deviation of the i^{th} feature vector in the database respectively. Figure 2.10 illustrates the matrix of our example after performing the Gaussian normalization:

$$\begin{array}{ccc} \begin{bmatrix} 0.1000 & 0.0800 & 0.2000 \\ 0.4000 & 50.000 & 0.3300 \\ 0.0500 & 0.3000 & 0.1500 \end{bmatrix} & \begin{bmatrix} -0.4402 & -0.5812 & -0.2870 \\ 1.1446 & 1.1547 & 1.1121 \\ -0.7044 & -0.5735 & -0.8251 \end{bmatrix} \\ (a) & (b) \end{array}$$

Figure 2.10: Gaussian normalization: (a) Input data (b) The normalized sequence to the range of $[-1, 1]$.

In the case that data are not evenly distributed around the mean, nonlinear normalization methods (i.e., exponential) can be employed to map data within specified intervals. Equation 2.22 illustrates one of the popular candidates of this type that compresses data in the range of $[0, 1]$.

$$h_{m,i} = \frac{x'_{m,i} - \mu_i}{k\sigma_i}, \quad \mathbf{F}_{m,i} = \frac{1}{1 + \exp(-h_{m,i})}. \quad (2.22)$$

This equation consists of two parts. The first part is a linear Gaussian normalization function that follows the $3 - \sigma$ rule for $k = 3$. According to the $3 - \sigma$ rule, approximately 99% (distance of 3 times the standard deviation for normal random variables) of the entries fall in the range of $[-1, 1]$. The second part is an exponential function that exponentially wraps data in the range of $[0, 1]$. For small values of $h_{m,i}$, $\mathbf{F}_{m,i}$ is a linear approximation of $x'_{m,i}$, whereas large values of $h_{m,i}$ (values away from the mean) are exponentially condensed toward the mean [53].

Figure 2.11 illustrates the same matrix after normalization. It is evident that the presence of outliers will no longer influence other feature elements [24].

$$\begin{array}{cc} \begin{bmatrix} 0.1000 & 0.0800 & 0.2000 \\ 0.4000 & 50.000 & 0.3300 \\ 0.0500 & 0.3000 & 0.1500 \end{bmatrix} & \begin{bmatrix} 0.4636 & 0.4517 & 0.4761 \\ 0.5942 & 0.5951 & 0.5916 \\ 0.4416 & 0.4524 & 0.4317 \end{bmatrix} \\ (a) & (b) \end{array}$$

Figure 2.11: Nonlinear Gaussian normalization: (a) Input data (b) The normalized sequence to the range of $[0, 1]$.

2.1.3 Feature Selection

Feature selection refers to the task of identifying, recognizing, and selecting an effective subset of features to represent pattern similarities from a larger set of redundant or even irrelevant features. In other words, the feature selection principle involves discarding certain features that do not necessarily contribute significant information as to how relevant an image

is in the classification process. Moreover, the choice of features to represent the patterns affects several aspects of the pattern recognition problem such as accuracy, required learning time, and the necessary number of samples. It is usually thought that a better classifier can be achieved by applying more features to describe individual images. It has been observed that beyond a certain point, the inclusion of additional features leads to worse, rather than better, performance results.

Multiple criteria influence the feature subset selection process including the number of features and accuracy of classification. There exist several popular feature selection algorithms. The preliminary work in this field started in the early 1960s. The previous approaches were based on probabilistic measurements of class distinctions by ignoring the interactions among features. Therefore, they usually produce an unsatisfactory selection of a subset of features.

The branch-and-bound feature selection algorithm [54] is claimed to be an effective scheme for discovering the optimal subset of features. One of the major constraints of this method is the monotony of the cost function, which makes it essentially impractical for a relatively small number of feature samples. The Wrapper method [55] is another choice of selecting the feature subset. However, it is very slow and computationally expensive because the whole data set needs to be analyzed repeatedly to produce a feature set [56]. Bhatti et al. [57] have investigated the combination of the Sequential Forward Selection (SFS) with the General Regression Neural Network (GRNN) for feature selection. Although it was proven to be effective, GRNN also suffers from high computational complexity [58]. Principle Component Analysis (PCA) is another popular technique for feature dimensionality reduction [59]. In this method, feature dimensionality is reduced by eliminating less significant data with smaller eigenvalues. Each eigenvalue represents the contribution of individual eigenvectors (principal component vectors) to the total variation in the data set. Even though PCA is an effective method for feature dimensionality reduction, useful information may be lost while truncating data in the PCA domain. The stepwise method is another candidate for the process of feature subset selection. This method is a combination of forward and backward procedures that, at each step, intends to achieve maximum interclass Mahalanobis

distance by constantly adding one feature to, or removing it from, the current feature subset. As Wang concluded in his work [56], the stepwise method is a suboptimal feature selection algorithm and, thus, an optimal feature subset can not be guaranteed.

Feature selection provides a hard-decision on the tradeoff between computational complexity and feature selection efficiency. On the other hand, it eliminates as much features as possible to achieve maximum classification accuracy without regard to the direct or indirect interactions among features. As a result, many useful traits might be unwillingly lost during this process. Instead, feature weight detection provides an estimate of the importance of features through its soft-decision mechanism. This trait is quit important for image processing applications since it avoids eliminating important information embedded in their inter-related features. The purpose of feature weight detection, as a general principle of feature selection, is to de-emphasize (not eliminate) certain features that do not necessarily contribute significant information to how relevant an image is in the retrieval process. Previously in [27] we had investigated the application of an exhaustive search method to search and discover optimum feature weights among all the possible combinations. Although optimal features are guaranteed, this method suffers from high computational costs.

In this work, a GA based approach is proposed to measure the importance of individual features and the degree of their contribution toward improving the retrieval process and overall performance of the system. Due to the ability of the GA to manage multi-constraint optimization problems (i.e., feature weight detection) along with its capacity to handle closely related features, several methods based on the GA for feature selection have been explored.

Activities in this area are purely application dependent and can be classified in categories including: Synthetic Aperture Radar (SAR) airborne imagery [60]; image classification [61]; classifier fusion systems [62]; gender classification [63]; feature transformation [64]; handwritten character recognition [65], etc. In the above applications, the proposed approaches have employed different combinations of GAs with different bases as well as different neural network classifiers (if applicable) to detect fitness values, which are varied depending on the nature of the problem.

The advantage of genetic algorithms-based feature selection techniques lie in their ability to deal with multiple criteria including classification accuracy and feature cost measurement. This makes them particularly attractive in the design of pattern classifiers in many practical domains. However, the main disadvantage of this approach lies in defining a cost function that is able to take all possible tradeoffs between accuracy and detected feature weights into consideration and producing an optimal feature set.

2.1.4 Proximity Measures

Proximity is a measure of similarity between two points in P -dimensional space and contains information about the degree of their closeness with respect to each other. In the image retrieval domain, however, proximity provides statistical (not semantic) clues of similarity between query image and other test images based on a linear combination of their corresponding low-level similarities. This metric will then be used for ranking images according to their relevance with respect to the query image.

Distances are discussed extensively on the literature in the context of clustering and classification [53]. Several widely used proximity evaluation schemes such as Euclidean distance, the City-block distance, and the Histogram intersection method will be discussed below.

The Euclidean distance between two P -dimensional vectors $\mathbf{x}'_1 = [x'_{1,1}, \dots, x'_{1,i}, \dots, x'_{1,P}]^T$ and $\mathbf{x}'_2 = [x'_{2,1}, \dots, x'_{2,i}, \dots, x'_{2,P}]^T$ is defined by:

$$d_E(\mathbf{x}'_1, \mathbf{x}'_2) = \left[\sum_{i=1}^P (x'_{1,i} - x'_{2,i})^2 \right]^{1/2}. \quad (2.23)$$

In general, Euclidean distance provides a very effective proximity measurement in image retrieval application. However, it is a computationally expensive technique, especially when the floating point data type is involved, since it requires P floating point multiplications, where P is the vector dimension.

To overcome this costly operation, the City-block distance is usually employed. This metric is defined by:

$$d_{CB}(\mathbf{x}'_1, \mathbf{x}'_2) = \sum_{i=1}^P |x'_{1,i} - x'_{2,i}|, \quad (2.24)$$

where $|\cdot|$ represents the absolute value of the difference between two points. In City-block metric, the distance between two points depends only on the points themselves and not on the path followed to travel between the points. In Euclidean distance, however, the distance is defined by the shortest path connecting them. Due to the low computational complexity, the City-block method was used for image comparisons and classification purposes in this thesis.

The Histogram intersection method is used to locate the common parts of objects by intersecting two images under similarity evaluation. The histogram intersection of two images composed of P -bin histograms is defined by:

$$d_{hist}(\mathbf{x}'_1, \mathbf{x}'_2) = 1 - \sum_{i=1}^P \frac{\min(x'_{1,i}, x'_{2,i})}{\min(|x'_1|, |x'_2|)}, \quad (2.25)$$

The Histogram intersection algorithm, proposed initially by Swain and Ballard in [28], was used as a metric to compare color contents of images after generating their color histograms.

The next section provides an overview on human-controlled interactive CBIR systems and studies their effective behavior in incorporating the semantic information in the retrieval process.

2.2 Interactive Approach in CBIR

As illustrated earlier in this chapter (see Figure 2.2), image retrieval based on the statistical representation of images and the linear approximation of similarity is unable to completely articulate the users' requirements on semantic levels. This is due to the gap between low-level features used to describe individual images and the high-level concept required to interpret their contents by the user. Past efforts to bridge this gap emphasized simulating human perception of visual contents *via* the human-computer interaction scheme (HCI): a learning

mechanism that allows retrieval systems to adapt to the users' needs by tuning the proximity matching process toward semantic levels using low-level visual features.

The tuning in the system's behavior is achieved by adapting a relevance feedback (RF) scheme accompanied by a weighted distance metric during the matching stage. In this process, different weights are assigned to individual components in a feature vector and are modified according to the users' preferences (i.e., the user directly teaches the system about her/his interpretations of an image). The weighted Euclidean is defined as:

$$d_E(\mathbf{x}'_1, \mathbf{z}) = \left[\sum_{i=1}^P w_i (x'_{1,i} - z_i)^2 \right]^{1/2}, \quad (2.26)$$

where z_i is the i^{th} component of the query feature vector, $\mathbf{z} = [z_1, \dots, z_i, \dots, z_P]^T$, and w_i is a weight parameter corresponding to the same component.

The idea behind weighted distance is to re-weight the feature space in favor of relevant features. The weight parameters can be estimated from the standard deviation σ_i of the features in the relevant set selected by the user at each RF iteration process [24], [66], [67]; or through optimization techniques (i.e., genetic algorithm) in the proposed automatic RF CBIR system [68].

The relevance feedback approach is an extended application of the modern information retrieval (IR), proposed by Salton and McGill [69], in the image retrieval process. In IR systems, each document is represented by a set of key words and terms. These terms are then concatenated in a set of vectors (a.k.a. "Vector Model") and are then made available for search and retrieval. Some of the well-known implementations of this approach are Multimedia Analysis and Retrieval System (MARS) [8], PicToSeek [21], DrawSearch [25], and Viper [26]. In all the above systems, some kind of query refinement strategy (i.e., feature weighting) have been adapted to interactively create a new query with the goal of optimizing the search process.

Although above implementations have offered some improvement in the process of image retrieval, they still suffer from a limited degree of adaptivity. It is due to the incapability of distance measurement techniques used in above systems to adequately model perceptual

Figure 2.12: An adaptive Human-controlled relevance feedback system [66].

differences as seen by the human user.

To overcome this problem, Muneesawang et al. [66] proposed a new architecture for the RF-based CBIR systems as is depicted in Figure 2.12. This system takes advantage of an adaptive technique based on non-linear radial-basis function (RBF) model [70] for learning the users' notion of similarity between images. In this process, the user is provided with a set of retrieved images and is asked to select those with the highest (semantic) similarity with respect to the query image. Feature vectors extracted from selected images are then used as training examples to determine centers and widths of different RBF units in the network. Using the RBF-based learning model offers further adaptability to the retrieval system to refine the search to different users and various types of images rather than enforcing a fixed metric for comparisons.

The RBF-based learning process in the CBIR engine of Figure 2.12 starts by presenting a set of Q retrieved images that are ranked according to their degree of statistical resemblance to the query image based on a predefined distance metric (i.e., City-block or Euclidean distance). The user then (subjectively) classifies each image into either relevant or irrelevant

Figure 2.13: Query adaptation and RBF learning.

samples (based on their semantic resemblance to the query image). The training vectors associated with selected images are used to form an $M \times P$ feature matrix, $\mathbf{R}$, consisting of M images marked as relevant with the vector length of P . Similarly, the training feature vectors of irrelevant images are also used to construct an $N \times P$ feature matrix, $\mathbf{N}$, where $N + M = Q$ as are illustrated by Equations 2.27 and 2.28 respectively:

$$\begin{aligned} \mathbf{R} &= [\mathbf{x}'_1, \dots, \mathbf{x}'_m, \dots, \mathbf{x}'_M]^T \\ &= [x'_{m,i}] \quad m = 1, \dots, M, \quad i = 1, \dots, P, \end{aligned} \quad (2.27)$$

$$\begin{aligned} \mathbf{N} &= [\mathbf{x}''_1, \dots, \mathbf{x}''_n, \dots, \mathbf{x}''_N]^T \\ &= [x''_{n,i}] \quad n = 1, \dots, N, \quad i = 1, \dots, P. \end{aligned} \quad (2.28)$$

In the above equations, $\mathbf{x}'_m = [x'_{m,1}, \dots, x'_{m,i}, \dots, x'_{m,P}]^T$ and $\mathbf{x}''_n = [x''_{n,1}, \dots, x''_{n,i}, \dots, x''_{n,P}]^T$ represent feature vectors of the m^{th} relevant and n^{th} irrelevant image in the database, respectively. A new query based on the selected images is subsequently created to adapt the search to the class that that query is embedded in. The selected query is then used for the next retrieval iteration where the similarity matching is performed non-linearly and through the RBF network units, described by Equations 2.32 and 2.33. This interactive process continues until the algorithm converges (i.e., until a satisfactory result is achieved). This process is summarized in the flowchart of Figure 2.13.

The following subsections provide overviews on the query remodeling and RBF-based

similarity matching techniques.

2.2.1 Query Modification

In the process of image retrieval there are situations where the selected query can not entirely reflect users' preferences because of the uncertainty in image interpretations as a result of ambiguous image contents (i.e., the presence of several objects of interest). Under such circumstances, the system often generates trivial or even irrelevant retrieval results due to its incapability of extracting all the required information that leads it to converge toward the query (relevant) class. Moreover, query modification enables the system to readjust query location more toward a best representative class with the objective of retrieving more relevant images at subsequent RF iterations. These modifications are carried out based on information (or preferences) provided by the user from earlier iterations of RF.

Several schemes have been studied for the purpose of query modification in the literature. Figure 2.14 illustrates one of the proposed approaches. In this method, the query position is tuned to the center of mass of relevant samples *via* calculating the mean value of each column (feature elements) of the relevant matrix $\mathbf{R}$ in Equation 2.27 as is described by Equation 2.29. Since the relevant group indicates the users' preference, the modified query will also reasonably satisfy this preference [71].

$$\hat{\mathbf{z}} = \frac{1}{M} \sum_{m=1}^M \mathbf{x}_m'. \quad (2.29)$$

A better method for the query modification is realizable in situations where the matrix $\mathbf{R}$ contains a small subset of the actual relevant class. In this situation, the previous query modification scheme will not perform adequately since sparse data resolution can extensively impact the modified query by diverging it from the true position of the relevant cluster center. To overcome this problem, the new scheme modifies the query based on the information extracted from both relevant and irrelevant sub-samples (i.e., both $\mathbf{R}$ and $\mathbf{N}$ matrices). As a result, the query is adjusted to a new position by shifting it away from the irrelevant group and more toward the relevant image cluster. This process is illustrated by Equation 2.30.

Figure 2.14: Query modification Model.

$$\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k-1} + \alpha_R \left(\frac{1}{M} \sum_{m=1}^M \mathbf{x}'_m - \hat{\mathbf{z}}_{k-1} \right) - \alpha_N \left(\frac{1}{N} \sum_{n=1}^N \mathbf{x}''_n - \hat{\mathbf{z}}_{k-1} \right). \quad (2.30)$$

In this equation, α_R and α_N are small positive constants, $\hat{\mathbf{z}}_{k-1}$ is the modified query at the previous iteration, and $\hat{\mathbf{z}}_k$ is the query under modification. The second term in the above equation represents the relevant image cluster, whereas, the third term indicates the irrelevant group. In practice, the constants in Equation 2.30 are chosen such that $\alpha_R \gg \alpha_N$; thus, a larger relocation is allowed toward the relevant samples while a slighter realignment is permitted away from the irrelevant group.

2.2.2 Non-linear Radial Basis Function

Linear association between distance and similarity, according to the traditional distance metrics, is not an optimal way of modeling human perceptual similarity. Instead, non-linear models can be used to reflect this resemblance since pattern recognition and classification of visual content in the human visual system is performed on a non-linear basis [72].

Radial Basis Function (RBF) is an attractive technique for simulating the human per-

Figure 2.15: Nonlinear Gaussian RBF centered at zero with the width of one.

ception. RBF is a kernel function that has an outstanding approximation capability for non-linear proximity evaluation. One of the major properties of RBF is its localization capability: the trait that is determined through its exponentially decaying (or growing) behavior with respect to the distance from a mean point [70]. Figure 2.15 illustrates an example of a Gaussian-shape RBF located at zero with the width of one (i.e., $\mu = 0$ and $\sigma = 1$).

The Gaussian-shaped RBF in Figure 2.15 is generated using Equation 2.31. In this equation, μ is center of RBF and σ denotes its width:

$$G(x) = \exp\left[-\frac{(x - \mu)^2}{2\sigma^2}\right]. \quad (2.31)$$

In this work, a one-dimensional Gaussian RBF is associated with each component of image feature vector and is used for the purpose of the proximity evaluation between query, $\mathbf{z}$, and input image, $\mathbf{x}$, feature vectors. On the other hand, each RBF unit provides a nonlinear mapping of distance versus similarity where the highest similarity is achieved when $\mathbf{z} = \mathbf{x}$. This process is illustrated by Equation 2.32 and subsequently in Figure 2.16:

Figure 2.16: Nonlinear Similarity Measures using RBF Network.

$$S(\mathbf{z}, \mathbf{x}) = \sum_{i=1}^P G_i(x_i - z_i) = \sum_{i=1}^P \exp\left[-\frac{(x_i - z_i)^2}{2\sigma_i^2}\right], \quad (2.32)$$

where σ_i , $i = 1, \dots, P$ are tuning parameters in the form of RBF width centered at the query position. Each tuning parameter linearly approximates the relevance of individual features through Equation 2.33. The small value of σ_i reflects high relevance, whereas a larger value of σ_i declares low resemblance of the i^{th} feature component.

$$\sigma_i = \eta \max |x_i - z_i|, \quad (2.33)$$

In this equation, η is an additional factor to ensure a large output for each exponential RBF unit, G_i . In particular, highly relevant features allow a higher sensitivity to the similarity matching process in Equation 2.32, whereas poorly relevant features have a minimal affect in this process.

2.3 Summary

The goal of image retrieval is to find, from a large image database, a set of images whose contents are similar to a given query. Each image in the database is considered as a vector point in the feature space that characterizes its color, shape, and textural information.

Traditional Rank-Based search engines rely on linear similarity functions, such as Euclidean distance, to decide on the degree of resemblance between the query and its neighboring images while neglecting the semantic association among retrieved images. This type of search engine will not perform satisfactorily as the result of the gap between low-level features and high-level concepts. In order to reduce this gap, relevance feedback-based machine learning techniques were introduced. The resulting search engine is capable of adapting the users' preferences (semantics) in the similarity matching process.

There are two problems, however, associated with this learning approach. First, the system requires a high degree of user workload in providing feedback samples through many cycles of relevance feedback before convergence. Second, the potential for human (subjective) error is introduced when users provide relevance judgment to retrieved images.

In view of the above problems, an unsupervised learning algorithm, namely, the Directed self-organizing tree map (DSOTM) will be introduced in the next chapter. The resulting search engine aims to minimize both human subjectivity and workload by replacing the required interactions with the DSOTM learning unit. Comprehensive experiments have demonstrated the feasibility of the proposed unsupervised scheme.

Chapter 3

Hierarchical Clustering Algorithms and Their Application in Automatic CBIR

AMONG the many properties of artificial neural networks (ANN), learning capability is of primary importance: an iterative process of network adaptations that enables synaptic weights to be adjusted according to the environment in which the network is embedded. On the other hand, learning in the ANN context is viewed as the process of updating the network architecture (i.e., connection strength) with respect to underlying patterns or features with or without the guidance of a teacher. The former is called *Supervised* and the latter is called *Unsupervised* Learning. A combination of these two learning paradigms results in a third type, *Hybrid* Learning [73].

In supervised or active learning, the teacher is regarded as a set of desired targets or training samples that inject prior knowledge of the optimum response of the network to inputs and provides guidance on synaptic weight adjustments under the influence of an error correction mechanism provided by the feedback loop: that is, a mechanism that seeks to minimize variations between the actual and desired response of the network.

In contrast, in unsupervised or self-organized learning, the network seeks to develop an internal representation of salient features in the input pattern by automatically adjusting the synaptic weights under the influence of quality of learning: a measure of the network's tuning capability versus statistical regulation of the input data. The competitive learning

rule plays an important role in the principle of self-organization. In competitive learning, the output neurons actively compete over some limited resources (e.g., input samples); only one neuron with the greatest access to total resources (strongest synapse) is declared to be the winner (the winner-takes-all principle). Thus over the time, strong synapses can grow, while weaker ones tend to disappear.

The main objective of this chapter is to provide a method for incorporating the competitive learning principles into the process of image retrieval. Section 3.1 provides qualitative arguments to describe the basic principles of self-organization. This will be followed by introductory material on Self-Organizing Feature Map (SOFM) and Self-Organizing Tree Map (SOTM) in Section 3.2. Section 3.3 provides an overview on a new member of SOTM family, the Directed Self-Organizing Tree Map (DSOTM) and studies its application in automating the image retrieval systems. A comprehensive study on the performance of the proposed CBIR engine, integrated with the above classifiers is presented in Section 3.4. Section 3.5 summarizes this chapter.

3.1 Principles of Self-Organization

As previously mentioned, unsupervised learning consists of repeated presentation of the patterns to the network and readjustment of the network's synaptic weights under the influence of the error correction mechanism provided by the feedback loop. In his book [59], Haykin identifies four major principles of self-organization:

- a.* Synaptic Self-Amplification,
- b.* Synaptic Competition,
- c.* Co-operation,
- d.* Knowledge through Redundancy.

3.1.1 Synaptic Self-Amplification and Competition

The principle of Synaptic Self-Amplification is, in fact, expressed through Hebb's postulate of learning [74], which is supported by neurobiological evidence. This states that (on the cellular level) when two cells are within significant proximity enabling one to excite another, and furthermore, do so persistently, some form of physiological/metabolic growth process results. This process works to enhance the firing cell's efficiency in triggering the second cell. In other words (in neural network terms), a synaptic path evolves and strengthens between the two neurons (memory cells), so that future associations occur much more readily. This action of strengthening the association between two nodes functions as a correlation between their two states.

Typically, the Hebbian learning rule that follows from this principle would eventually drive synapses into saturation were it not for the second principle, wherein some *competition* occurs for limited resources. These properties combined, have led to the modified Hebbian *adaptive* rule, as proposed by Kohonen [75]:

$$\Delta w_{k^*} = \alpha \varphi(k^*, x_i) [x_i - w_{k^*}], \quad (3.1)$$

where w_{k^*} is the synaptic weight vector of the *winning* neuron k^* , α is the learning rate, and $\varphi(.)$ represents some scalar response to the firing of neuron k^* (activation).

The activation function $\varphi(.)$ is the result of the second principle (competition). Neurons generally compete to see which is most representative of a given input pattern presented to the network. Some form of discriminative function oversees this process (for example, choosing a neuron as a *winner* if its synaptic vector minimizes Euclidean distance over the set of all neurons). This process is often termed *Competitive Learning*.

3.1.2 Co-operation

Hebb's postulate is also suggestive of the lateral or associative aspect of the way knowledge is then captured in the network, i.e., not just through the *winner*, but also through nearby

neurons in the output layer. This property is generally implemented in self-organizing architectures by virtue of the way in which nodes are interconnected. Often the strength of connection is not considered, but rather a simple link functions as an indicator of which other nodes in the network will more readily be associated with a winning node. Thus a local neighborhood is defined, and it is through this that knowledge may be imparted. Local adaptation usually follows the simple Kohonen update rule, whereby a portion of information is learned by the winning node, with neighboring nodes extracting lesser portions from the same input.

3.1.3 Knowledge Through Redundancy

Although not explicitly encoded, the final principle of self-organization implicitly results from the action of the first three. When exposed, any order or structure inherent within a series of activation patterns represents redundant information that is ultimately encoded by the network as knowledge. In other words, the network will evolve such that similar patterns will be captured and encoded by similar output nodes, while neighboring nodes organize themselves according to these dominant redundancies, each in turn focusing on and encoding lesser redundant patterns across the input space.

Architecture such as Kohonen's *Self-Organizing Feature Map* (SOFM) [75] represents one of the most fundamental realizations of these principles. Kong's *Self-Organizing Tree Map* (SOTM) [76, 77] structure also benefits from the self-organization principles listed above.

The remainder of the chapter is devoted to present a methodology for guiding adaptations of an RBF-based relevance feedback network, embedded in automatic content-based image retrieval (CBIR) systems, through the principle of unsupervised hierarchical clustering using the above classifiers.

Figure 3.1: A two-dimensional SOFM model with one input [75].

3.2 Unsupervised Learning and Self-Organization

3.2.1 The Self-Organizing Feature Map (SOFM)

The SOFM is constructed from an artificial topographical map that learns through neurologically-motivated self-organization. SOFM was inspired by the way the different sensory inputs (motor, visual, auditory, etc.) are mapped onto corresponding areas of cerebral cortex in the human brain [78]. In this context, SOFM seeks to present a topologically ordered model of incoming multi-dimensional patterns by transforming them into a one- or two-dimensional discrete map (lattice) and dividing them into groups of similar patterns.

Associative connections linking prototypes within SOFM-based clustering algorithms are generally responsible for their innate ability to infer an *ordered* or *topologically-preserved* mapping of the underlying data space. Associations among nodes are advantageous as they help guide the evolution of such networks, and may also assist in formulating post-processing strategies or for extracting higher-level properties of any clusters discovered (e.g., inter-cluster relationships). This type of property is often used for the visualization of high-dimensional data - where multivariate data (of a dimension greater than three) are mapped onto a two-dimensional grid such as the lattice of a SOFM. Since topology is preserved, neighboring nodes in the lattice are often representative of related properties in the original data [79].

Figure 3.1 illustrates a two-dimensional SOFM model with one input. SOFM's self-organization (clustering) process starts by randomly initializing the synaptic weights in the network. Once the network is properly initialized, the training phase begins by repeatedly presenting the input samples (usually either randomly or in batch mode) to the network. Each such presentation creates *competition* among the neurons and slightly adjusts connection strengths through the principle of *Synaptic Self-Amplification*. The particular neuron with the smallest distance metric to the incoming sample is declared to be the winner of the competition. The *Co-operation* process is subsequently initialized by the definition of a lateral distance around the winning node: a decaying neighborhood of excited neurons. Moreover, the presence of *redundancy* in the input space is required for the creation of a topologically ordered map in the SOFM structure by providing the necessary knowledge for the learning process of the network [59].

The clustering process of SOFM is summarized in the following steps [59]:

Initialization: A root node $\{\mathbf{w}_j\}_{j=1}^J$ is chosen from the available set of input vectors $\{\mathbf{x}_k\}_{k=1}^K$ in a random manner. J is the total number of neurons in the SOFM map (lattice) and K is the total number of input vectors (i.e., images);

Similarity Matching: A new data point, $\mathbf{x}$, from the input space is randomly selected and the best-matching (winning) neuron, j^* , is found by minimizing the predefined Euclidean distance criterion in Equation 3.2:

$$\mathbf{w}_{j^*}(t) = \arg \min_j \|\mathbf{x}(t) - \mathbf{w}_j(t)\|, \quad j = 1, 2, \dots, J; \quad (3.2)$$

Updating: The synaptic weight vectors of all neurons are adjusted according to the reinforcement learning rule:

$$\mathbf{w}_{j^*}(t+1) = \mathbf{w}_{j^*}(t) + \alpha(t)h_{j^*}(t)[\mathbf{x}(t) - \mathbf{w}_{j^*}(t)], \quad (3.3)$$

where $\alpha(t) = \alpha(t_0) \exp(-t/\tau)$ is the learning rate that decays monotonically with time, $0.01 \leq \alpha(t) \leq \alpha(t_0)$, and $\alpha(t_0) = 0.1$; $h_{j^*}(t) = \exp[-\|\mathbf{x}(t) - \mathbf{w}_{j^*}(t)\|/2\sigma^2(t)]$ denote the

decaying topological neighborhood function centered on the winning centroid and measure the degree to which excited neurons in the vicinity of the winning neuron participate in the learning process, $\sigma(t) = \sigma(t_0) \exp(-t/\kappa)$, and τ and κ are time constants with values of 1000 and $1000/\log \sigma(t_0)$ respectively;

Continuation: The *Similarity Matching* step is repeated until maximum number of iterations is reached, and/or no noticeable changes in the feature map are observed;

Relevance Identification: If $\mathbf{w}_{j^*} = \arg \min_j \|\mathbf{z} - \mathbf{w}_j\|$ and $N_R = \{\cup_{k=1}^K \mathbf{x}_k | \arg \min_j \|\mathbf{x}_k - \mathbf{w}_{j^*}\|\}$, that is, if the closest center to a set of N_R input samples is also the closest center to the query, **then** mark $\{\mathbf{x}_k\}_{k=1}^K \in N_R$ as relevant samples; **and** mark the rest of the samples as irrelevant.

In the above SOFM-based approach, topology is imposed rather than inferred (*via* a fixed predefined lattice of nodes). Such fixed topology makes the SOFM structure more prone to distortion and less capable of converging on an appropriate number of classes and, thus, incapable of representing the appropriate relationship within the underlying sparse data space. SOTM-based learning, however, is quite capable of dealing with sparse data spaces as is explained in more detail in the next section.

3.2.2 The Self-Organizing Tree Map (SOTM)

The SOTM was first proposed in 1995 [76, 77] and later developed in [80, 87]. Similar to the Kohonen's SOFM, SOTM tends to follow both the self-organization and competitive learning principles discussed in [75]; however unlike SOFM, SOTM tends to grow a more dynamic topology (more plastic than SOFM) that not only extracts global intuition from an input pattern space but also injects some degree of localization into the discriminative process, such that maximal discrimination becomes a priority at any given resolution (or number of classes). Compared to SOFM, the deformable tree-like structure of SOTM enables it to learn faster and more efficiently, while retaining more stable and more secure attributes that

Figure 3.2: Self-organizing data clustering; (a) performed by SOTM, no nodes converge to join the areas of zero data density; (b) performed by SOFM, nodes converge to areas of zero data density [30].

lead to the maintenance of its existing information [73]. SOTM's ability to span and force the division in the extremes of the data in the early stages, delaying the division of most similar aspects until the later stages of learning, makes it very sensitive to the most dominant differences in the data and, thus, less prone to classification errors and more attractive to the retrieval applications than the SOFM algorithm [81].

Figure 3.2 provides a vivid comparison of the clustering capabilities of the SOTM and SOFM on a two-dimensional feature space consisting of x and y data positions [30]. In this figure, the input vectors are uniformly distributed within many rectangular squares. The SOTM's clustering performance in Figure 3.2(a) shows that there are no neurons lying in a zero-density area. In contrast, although the SOFM's topology exhibits the distribution of the structured input vectors, it also introduces several false representations outside of the distribution of the input space, as shown in Figure 3.2(b). Therefore, classifying a high-dimensional input space with a sparse data structure using the SOTM algorithm results in the formation of a suitable network structure that prioritizes the spanning of data across the

different stages of learning.

Although classification using SOFM appears to be robust and effective, which tolerates even very high input vector dimensionalities [75], the lattice topology of the SOFM network makes it undesirable for clustering sparse data spaces due to the concentration of a fraction of nodes in the map that results from the best-matching unit computation [59]. SOFM has nodes that can easily get trapped in regions of zero density and, therefore, it can simply lose its ability to represent the underlying topology of the input pattern and, consequently, can change the true boundaries of high density classes by pulling some of the samples away from the real clusters between which they fall and allocating them to the middle node. The tree structure of SOTM, however, is more suitable for determining the high-density regions.

In many ways, the SOTM process draws inspiration from Adaptive Resonance Theory (ART) applied to an SOFM-type framework. SOTM sustains SOFM's property of topology preservation while maintains some degree of plasticity to adapt to any change in the input pattern distribution in greater detail. It also retains ART's ability to dynamically create new output neurons according to a global threshold setting [76]. ART [82]–[84] has a sufficient supply of output units (prototypes) that are used when required. Through the so-called vigilance test, the similarity of each input to the existing prototypes is examined; whereas in competitive learning the nearest prototype in the outer layer is cited. If the input falls within the tolerance of the winning prototype, then *resonance* is said to occur, resulting in the refinement of the winning prototype. Otherwise, if there is still sufficient capacity, a new prototype is formed.

Node or prototype insertion in the SOTM is driven by a top-down process that effectively explores the data space from the *outside in*. The tree structure of the SOTM is constructed through the random selection of an isolated root node (center) and by repeated presentation of the rest of the patterns to the network. The closest existing prototype to the center with respect to current proximity matching is declared to be the winner. Each such presentation of input patterns slightly modifies the winning node's position in the network: a position that eventually evolves toward the center of the mass of the current class. This gradual

Figure 3.3: The SOTM Flowchart in the CBIR application.

adaptation of the node's position is controlled by an exponentially decreasing function called the *learning rate*. The learning rate resets to its initial value each time a new center is generated. Therefore, sufficient time is given to the network to adapt itself to the presence of new samples, thus, the tree grows larger and the similarity measurement tends to be more accurate. The generation of new centers (tree branches) is controlled by a global vigilance threshold $H(t)$ that is used to evaluate the proximity of each new input sample against the winning node. A “hard decision” is made: if the input is distant from the winner beyond the threshold, a new prototype is spawned as a child node to the current winner, and the process continues with the next input, thus, over time, a tree structure evolves [81]. Otherwise, the input is deemed *significantly similar* and the winner is updated toward the input.

The hierarchical control function [76], $H(t)$, in the SOTM algorithm is, in fact, regarded as a dynamic vigilance function that evaluates proximity at decreasing levels over time and forms a hierarchical partitioning of the data. This hierarchical approach naturally attempts to maximize discrimination across the entire data space in early phases of *discovery* before considering finer differences in the data.

The algorithm for generating the map is summarized in a simplified flowchart depicted in Figure 3.3. The details associated with each of the main components are given in the following steps [27, 38, 76]:

Initialization: A root node $\{w_j\}_{j=1}^J$ is chosen from the available set of input vectors $\{x_k\}_{k=1}^K$ in a random manner. J is the total number of centroids (initially set to 1) and K

is the total number of input vectors (i.e., images);

Similarity Matching: A new data point, $\mathbf{x}$, is randomly selected from the input space and the best-matching (winning) neuron, j^* , is found through the minimization of the predefined Euclidean distance criterion in Equation 3.4:

$$\mathbf{w}_{j^*}(t) = \arg \min_j \|\mathbf{x}(t) - \mathbf{w}_j(t)\|, \quad j = 1, 2, \dots, J; \quad (3.4)$$

Updating: If $\|\mathbf{x}(t) - \mathbf{w}_{j^*}(t)\| \leq H(t)$, where $H(t)$ is the hierarchy function used to control the levels of the tree and decreases in time according to $H(t+1) = \lambda H(t)$ where λ is the threshold decreasing constant and $0 < \lambda < 1$, **then** assign $\mathbf{x}(t)$ to the j^{th} centroid, and adjust the synaptic vector according to the reinforced learning rule:

$$\mathbf{w}_{j^*}(t+1) = \mathbf{w}_{j^*}(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}_{j^*}(t)], \quad (3.5)$$

where $\alpha(t)$ is the learning rate, which decays monotonically with time, $\alpha(t) = \alpha(t)/(\alpha(t)+1)$, $0.01 \leq \alpha(t) \leq \alpha(t_0)$, and $\alpha(t_0) = 0.1$; **Else** form a new centroid node starting with $\mathbf{x}$ and increment j^* by 1;

Continuation: The *Similarity Matching* step is repeated until the maximum number of iterations is reached, the maximum number of clusters is generated, and/or no noticeable changes in the feature map are observed;

Relevance Identification: If $\mathbf{w}_{j^*} = \arg \min_j \|\mathbf{z} - \mathbf{w}_j\|$ and $N_R = \{\bigcup_{k=1}^K \mathbf{x}_k | \arg \min_{j^*} \|\mathbf{x}_k - \mathbf{w}_{j^*}\|\}$, that is, if the closest center to a set of N_R input samples is also the closest center to the query, **then** mark $\{\mathbf{x}_k\}_{k=1}^K \in N_R$ as relevant samples; **and** mark the rest of the samples as irrelevant.

In attempts described in the above steps, SOTM tends to classify input vectors into several different clusters (classes), specifying the centroid of each. This polythetic approach assumes that there may be multiple irrelevant clusters, each likely to be different not only

Figure 3.4: Polythetic Clustering Approach using Multi-Class SOTM: From left to right: original data set; clustering of relevant class using SOTM at the first iteration; updated clustering using SOTM at subsequent iterations.

to the relevant (query) class, but also from each other. Capturing a single relevant class and multiple irrelevant classes allows for a more non-linear partitioning of images into both relevant and irrelevant groupings. Once identified, information from the relevant cluster may be used to modify the query so as to better encapsulate the spread of relevant images in any subsequent search iterations (see Figure 3.4). If the root node is initialized as the query vector, any new centroids so discovered typically represent groups of statistically irrelevant images [85].

3.3 The Directed Self-Organizing Tree Map (DSOTM)

One major advantage of the SOTM over classical unsupervised learning methods, such as SOFM and K-mean algorithms, is its ability to work well with sparsely distributed data [80]. SOTM is chosen in the current application, as the problem in image retrieval have different characteristic than other data classification applications: First, the training data set required by relevance feedback learning algorithms is very small, e.g., a few to tens of samples. Also, the feature space is of a very high dimension consisting of a combination of color, shape, and texture features. These tend to form sparsely distributed data. Secondly, a problem is caused by an unbalanced data distribution between relevant and irrelevant samples in the training set. It is expected that, after the first iteration of relevance feedback, the relevant items are retrieved more than irrelevant ones, and thus, the majority of relevant items will introduce

an unbalanced space to the resulting clusters. To solve this problem, the SOTM allows for a focus and maximization of discrimination within sub-regions of the (unbalanced) training data *via* competition among its hierarchically-discovered nodes. This efficient allocation and breakdown of class relationships minimizes classification errors compared to that achieved through SOFM, which unfold across data space, often leading to distortion within sparse interstices.

Despite the above advantages of the use of SOTM-based classifiers, there are two major drawbacks associated with the algorithm: it inappropriately decides on the relevant number of classes; and often loses track of the true query position. The decision about which clusters are relevant in the SOTM is postponed until after the algorithm has converged (see subsection 3.2.2). This is largely due to the absence of an innate controlling process in the algorithm to influence cluster generation around the query center - the SOTM clusters entirely independently. The loss of a sense of query location within the input space can have undesired effects on the true structure of the relevant class and can force the SOTM algorithm to spawn new clusters, form unnecessary boundaries within the query class, and make erroneous decisions about resemblance of the samples as illustrated in Figure 3.5. Therefore, retention of some degree of supervision for the prevention of unnecessary boundaries from forming around the query class appears to be vital.

Due to the above limitations, a new member of the SOTM family, the Directed SOTM (DSOTM) is proposed in this thesis. The DSOTM algorithm not only provides a partial supervision on cluster generation by forcing divisions away from the query class but also makes a gradual decision about the resemblance of the input patterns by constantly modifying each sample's membership during the learning phase of the algorithm. As a result, a more robust topology with respect to the query, as well as a better sense of likeness, can be achieved [68].

In addition, DSOTM relies on the query position as the a priori center of the relevant class and updates memberships according to this knowledge. The synaptic vector adjustments in the DSOTM are not simply limited to the winning node; the algorithm also constantly modifies all the centers according to the query position. If the winning center is not the

(a)

(b)

(c)

Figure 3.5: 2-D mapping: (a) Input pattern with five distinct clusters, (b) 14 generated centers using SOTM, and (c) five generated centers using DSOTM. The SOTM forms a boundary near the query (triangle) contaminating relevant samples, whereas some supervision is maintained in the DSOTM case, preventing unnecessary boundaries from forming.

query center, vector adjustments will affect both the winning node and the relevant center's position by moving the winning node (the center of the irrelevant class) toward the irrelevant samples and moving the relevant center toward the query center. Thus, as more samples are exposed to the network, the DSOTM algorithm will learn less from irrelevant samples and more from relevant ones by maintaining the integrity of a relevant center near the original query position. This action helps to foster a sense of *generalization* in the query class across feature subtleties that may objectively be deemed as discriminative by an independently operating SOTM. The DSOTM *desensitizes* partitioning in the region of a query.

The algorithm for generating the DSOTM map is summarized in a simplified flowchart depicted in Figure 3.6. Details associated with each of the main components are given in the following steps [68, 81]:

Initialization: A root node $\{\mathbf{w}_j\}_{j=1}^J$ is chosen from the available set of input vectors $\{\mathbf{x}_k\}_{k=1}^K$ in a random manner. J is the total number of centroids (initially set to 1) and K is the total number of input vectors (i.e., images);

Similarity Matching: A new data point, $\mathbf{x}$, is randomly selected and the best-matching (winning) centroid, j^* , is found through the minimization of the predefined Euclidean distance criterion in 3.6:

$$\mathbf{w}_{j^*}(t) = \arg \min_j \|\mathbf{x}(t) - \mathbf{w}_j(t)\|, \quad j = 1, 2, \dots, J; \quad (3.6)$$

Updating: If $\|\mathbf{x}(t) - \mathbf{w}_{j^*}(t)\| \leq H(t)$, where $H(t)$ is the hierarchy function used to control the levels of the tree and decays exponentially over time from its initial value, $H(t_0) > \sigma_{\mathbf{x}}$, according to $H(t+1) = \lambda H(t) \exp(-t/\rho)$, where λ is the threshold constant, $0 < \lambda < 1$, and $\rho = \max(t)/\log_{10}(H(t))$. The proposed threshold function is empirically established to decay faster than the one employed in the SOTM architecture. As a result, the network is given a better opportunity to generate the required centers at its initial training phase and learn from them at the later stages of training. Alternatively, the preliminary training phase in the DSOTM algorithm is prioritized with the node generation process while the later

Figure 3.6: The DSOTM Flowchart for the CBIR application.

stages are dominated with learning about the existing information. *then* $\mathbf{x}(t)$ is assigned to the j^{th} centroid, and the synaptic vector is adjusted according to the reinforced learning rule:

$$\mathbf{w}_{j*}(t+1) = \mathbf{w}_{j*}(t) + \alpha(t)\beta(\mathbf{z}, \mathbf{x}, t)[\mathbf{x}(t) - \mathbf{w}_{j*}(t)], \quad (3.7)$$

where $\alpha(t)$ is the learning rate, which decays exponentially over time as more neurons are allocated, $\alpha(t) = \alpha(t_0) \exp(-t/\max(t))$, $0.01 \leq \alpha(t) \leq \alpha(t_0)$, and $\alpha(t_0) = 0.1$; and $\beta(\mathbf{z}, \mathbf{x}, t)$ is the exponential ranking function that measures the similarity between query feature vector,

Figure 3.7: The magnitude of $\beta(\mathbf{z}, \mathbf{x}, t)$ represents the similarity between the input vector $\mathbf{x}$ and the query $\mathbf{z}$, where the highest similarity is attained when $\mathbf{x} = \mathbf{z}$.

$\mathbf{z}$, and input feature vector, $\mathbf{x}$, at an automatic relevance feedback iteration from the previous search operation as is indicated in Equation 3.8 [27]:

$$\beta(\mathbf{z}, \mathbf{x}, t) = \sum_{i=1}^P G_i(x_i - z_i) = \sum_{i=1}^P \exp\left[-\frac{(x_i - z_i)^2}{2\sigma_i^2}\right]. \quad (3.8)$$

In this equation, P is the total number of features, $\sigma_i = \eta \max_i |x_i - z_i|$ is the tuning parameter, and η is an additional factor to ensure a large output for $\beta(\mathbf{z}, \mathbf{x}, t)$. A large value of $\beta(\mathbf{z}, \mathbf{x}, t)$ indicates a high relevance of the feature vector compared to the respective query feature at time t as is illustrated in Figure 3.7. As a result, the synaptic vectors are adjusted so that they learn more from statistically similar inputs and less from statistically irrelevant ones. *Else* form a new centroid node starting with $\mathbf{x}$, reset the learning rate to its initial value (i.e., $\alpha(t_0) = 0.1$), and increment j by 1;

Cluster Adjustment and Relevance Identification: If $\|\mathbf{x}(t) - \mathbf{w}_{j^*}(t)\| \leq H(t)$ and $\arg \min_{j^*} \|\mathbf{z} - \mathbf{w}_{j^*}(t)\|$, that is, if the closest center to the current input data is also the closest center to the query, *then* mark $x(t)$ as a relevant sample *and* update its centroid (winning neuron) toward the query position according to the degree of resemblance of the sample using:

$$\mathbf{w}_{j^*}(t+1) = \mathbf{w}_{j^*}(t) + \alpha(t)\beta(\mathbf{z}, \mathbf{x}, t)[\mathbf{z} - \mathbf{w}_{j^*}(t)]; \quad (3.9)$$

else mark $x(t)$ as an irrelevant sample *and* update its centroid using Equation 3.7. Subsequently, find and move center of relevant class further toward the query center using Equation 3.9;

Continuation: The *Similarity Matching* step is repeated until the maximum number of iterations is reached, the maximum number of clusters is generated, and/or no noticeable changes in the feature map are observed.

The *Cluster Adjustment and Relevance Identification* step in the DSOTM algorithm imposes some constraints on cluster generation near the query position and, thus, avoids unnecessary boundaries to be formed around it. As a result, a better sense of relevance measurements can be achieved as the tree structure develops. Moreover, the growth of the DSOTM is biased *via* the ranking function to learn more from input vectors deemed to be similar to the query itself, and less from images far from the query [86]. This promotes the generation of multiple irrelevant classes, while maintaining necessary plasticity in the relevant class [27].

As discussed by Kyan et al. in [87], $H(t)$ provides a mechanism for the hierarchical exploration of the input space with different resolutions over time. It appears to be crucial to have a frontier on decaying the $H(t)$ below the smallest meaningful resolution across the feature space to avoid the indefinite growth of both SOTM and DSOTM networks in absence of the *user's defined* maximum desired number of neurons to be allocated. Beyond this limit, the network can grow unboundedly and unnecessary centers (clusters) can be generated until the number of classes equals (or even exceeds) the number of samples in the feature database. The threshold function used in the DSOTM architecture can be automatically adapted to a reasonable resolution before its existing topology reaches a stable condition as illustrated in Figure 3.5(c). This is due to restrictions imposed by the query position as previously discussed. The threshold function of the SOTM algorithm, however, is unable to adequately

Figure 3.8: 2-D mapping of DSOTM classification: (a) Input pattern with five distinct clusters, query (relevant class center) is illustrated in a blue triangle, (b) Generation of the second center after 15 epochs, (c) Generation of the third center after 17 epochs, (d) Generation of the fourth center after 20 epochs, (e) Generation of the fifth center after 40 epochs, and (f) Final clustering

Figure 3.8 - Cont'd: results after 100 epochs. The dynamic threshold function, (represented in the red ellipsoid) evaluates proximity at a decreasing level during time and forms a hierarchical partitioning of the input pattern on (a). The threshold, $H(t)$, is responsible for creating the boundary allowing the algorithm to decide whether the currently chosen neuron (memory) vector is significantly different from the input (stimulus) or not. Thus, it naturally attempts to maximize discrimination across the entire data space in the early phases of discovery before considering finer differences in the data.

Figure 3.9: Timing Diagram of the DSOTM-based classification of Figure 3.8

decide on the appropriate number of centers and, thus, over-classifies the input space in Figure 3.5(b). The smallest resolution was defined to be the Euclidean sum of the smallest resolution of all feature sets included in the feature space [87].

A vivid illustration of the step-by-step classification using the above algorithm is illustrated in Figure 3.8, while Figure 3.9 represents the timing diagram of the waveforms governing the evolution of DSOTM for the same example. The random selection of a vector from the input space (stimulus) is essential to the learning process and the convergence of the algorithm as it gives currently existing neurons (in the memory of the network) an adequate opportunity to compete and be excited by the closest stimuli in the Euclidean sense. The decaying learning factor, $\alpha(t)$, ensures that the neurons eventually settle (for higher value

of $\alpha(t)$) and attuned (for a smaller value of $\alpha(t)$) toward the center of mass of the dense regions (clusters) in the feature space as depicted in Figure 3.8(d). The decaying hierarchical function, $H(t)$, controls the levels of the tree and the generation of new neurons: if the current stimulus falls within the range defined by $H(t)$'s ellipsoid boundary, then the input sample is considered significantly similar to the winning neuron; conversely, if the stimulus happens to occur outside this resemblance boundary, then the network will spawn a new neuron (create a new memory), attaching it to the closest representative (parent) prototype. To be able to provide adequate time for the algorithm to search through all the samples in the data space before considering a new range of candidates for neuron generation, a stepped exponential function with a fixed radius was employed as $H(t)$ (see Figure 3.9). The use of such hierarchical controlling function allows time for the existing neurons to readjust to the best representation of the input space and, thus, generates more robust clusters [87].

3.4 Experimental Results: Unsupervised Learning Algorithms in the CBIR Application

The block diagram of the computer-controlled CBIR engine is illustrated in Figure 3.10 and in more detail in Figure 3.11. In this chapter a relatively simple (non-optimized) CBIR architecture is employed to compare ability of the proposed hierarchical clustering algorithm, the DSOTM, for the sole purpose of data classification with two other techniques: SOFM and SOTM. A more complex (optimized) architecture will be presented in the next chapter.

In Figure 3.11, the second unit of the *Initial Search* module, *Feature Extraction 1*, deals with the calculation of features from a high volume image database. Consequently, a standard set of content descriptors such as MPEG-7 standard features (see Chapter 2), is extracted in this module to provide a more generic and rapid interface for existing databases based on the chosen standard. Extracted features are used to retrieve the most similar images (in a relative sense) based on a predefined distance metric. The top Q images are then displayed back to the user through an *Interface* block. Subsequently, the user may request

Figure 3.10: Machine controlled CBIR system [27].

Figure 3.11: Machine-controlled CBIR system - A closer look [88].

an automatic search which operates independently. Upon this request, control of the system is switched to the *Automatic Search* module, wherein another set of features with higher perceptual quality are extracted from the top Q retrieved images from the initial search, using *Feature Extraction 2*. Although computation of those features could be intensive, this module allows for the use of more proprietary or specialty features. Such a module may

Table 3.1: Experimental results in terms of RR - The Automatic CBIR

Classifier	Set A	Set B	Set C	Average
Rank-Based	37.8%	39.2%	39.8%	<i>38.9%</i>
SOFM	51.2%	49.1%	52.3%	<i>50.9%</i>
SOTM	52.1%	50.6%	54.4%	<i>52.4%</i>
DSOTM	54.8%	56.4%	58.6%	<i>56.6%</i>

enhance perceptual discrimination beyond that which might otherwise be possible through standard features alone. These features are then used as seeds to train the *Unsupervised Classifier*. A new query, based on the selected images from the previous iterations, is then selected to represent the relevant class through the *Query Modification* module.

In this work, Color Histograms, Color Moments, Wavelet Moments, and Fourier Descriptors were used in the *Feature Extraction 1*, whereas Hu’s Seven Moment Invariants (HSMI) and Gabor Descriptors accompanied with Color Histograms and Color Moments were used in *Feature Extraction 2*.

In this work, the simulations were carried out using a subset of the Corel image database [89] consisting of nearly 12000 JPEG color images, covering a wide range of real-life photos, from 100 different categories. Each category consisted of 100 visually associated objects to simplify measurements of the retrieval accuracy (RR) during the experiments. Three sets of 120 images were drawn from the database to form sets A , B , and C . Each set consists of randomly selected images such that no two images were from the same class. Retrieval results were statistically calculated from each of the three sets. In the simulations, a total of 16 ($Q = 16$) most relevant images were retrieved to evaluate the performance of the retrieval.

The experimental results are illustrated in Table 3.1. In the SOFM and SOTM algorithms, the maximum number of allowed cluster generation was set to C , $C < Q$, where Q is the total number of retrieved images from the initial search; C was empirically set to eight. A 4×2 grid topology was used in the SOFM structure to locate the eight possible cluster centers (a fixed topology). A “hard decision” on the resemblance of the input samples was

Figure 3.12: Machine-controlled CBIR system using SOFM.

made: if the sample was closer to one center than any other centers, in terms of a predefined distance metric, it was considered to be part of the cluster represented by that center.

Figures 3.12 - 3.14 illustrate screenshots of the system's behavior using the same query sample as Figure 2.2. SOTM has shown effective behavior in minimizing human interactions and automating the search process by efficiently classifying an unknown and non-uniform data space into more meaningful clusters. Moreover, the flexible tree-like topologies of the SOTM allow data exploration through a top-down and outside-in process making it essentially sensitive to the most dominant differences in the data and, therefore, less prone to classification errors and more attractive to the retrieval applications.

Despite the advantages of the use of SOTM-based classifiers, the retention of some degree of supervision to prevent unnecessary boundaries from forming around the query class appears to be crucial. In the DSOTM algorithm, decisions about the association of input pattern to query image are gradually made as each sample is presented to the system. The DSOTM also contains a controlling mechanism that keeps track of the query center by forc-

Figure 3.13: Machine-controlled CBIR system using SOTM.

ing the center of the relevant class to remain in the vicinity of the query position. Therefore, it can dynamically control the generation of new centers and can determine the relevance of input samples, with respect to the query, as the tree structure grows. On the other hand, it limits the synaptic vector adjustments according to its reinforced learning rules and constrains cluster generation through prevention of the spawning of redundant centers around the query position, as this part of the map is already occupied by a relevant class center.

3.5 Summary

The framework for a novel unsupervised clustering algorithm based on the DSOTM was introduced in this chapter. The performance of the proposed method was compared with two other conventional clustering methods, SOFM and SOTM, by measuring the performance of the computer controlled CBIR system depicted in Figure 3.11. The main focus of this chapter was to introduce a new member of SOTM family, the Directed SOTM (DSOTM)

Figure 3.14: Machine-controlled CBIR system using DSOTM.

which not only provides a partial supervision on cluster generation by forcing divisions away from the query class, but also presents a flexible verdict on resemblance of the input pattern as its tree structure grows. Comprehensive comparisons with SOFM and SOTM algorithms demonstrate the feasibility of the proposed scheme.

Although DSOTM is proven to be an effective solution for enhancing the performance of image retrieval systems, its performance is highly dependant on the query position and the combination of features that are used to tokenize the query image. These issues will be addressed in detail in Chapter 4.

Chapter 4

Genetic Algorithm and Its Application in Automatic CBIR

DATA classification is a process of labeling an unknown pattern space into several sensible clusters (groups) to provide easier means of access, to give a more comprehensive description, and to offer a more meaningful conclusion of data. Classification is one of the most fundamental steps for successful pattern recognition and has been inspired by the process of information management in the human brain. The human brain categorizes huge amounts of information into clusters to avoid processing them as single entities.

As explained in Chapter 2, the typical CBIR systems, however, neglect such principles by simply processing the database as a whole and by treating each and every target image separately and ranking them according to their degree of similarity (association) with respect to the query image. This process is carried out regardless of any dissimilarity (disassociation) that may exist between them. The proposed automatic image retrieval system, however, differs from such rank-based engines by integrating data clustering principles into the retrieval process, thereby exploiting the DSOTM algorithm (see Chapter 3).

The main focus of this chapter is to present a method for optimizing relevance identification in CBIR systems through the principle of feature weight detection. The novelty of this scheme lies in the use of a base-10 Genetic Algorithm method to measure importance of individual features and their degree of contribution toward improving the retrieval process and overall performance of the system in a so-termed feature weight detection process.

GA offers a particularly attractive approach since it is generally quite effective in the rapid global search of large and nonlinear spaces. Section 4.1 provides an overview on the genetic algorithm technique used for the purpose of feature weight detection in this work. A detailed outline on the proposed genetic-based CBIR engine (GA-CBIR) is presented in Section 4.2, and comprehensive experiments of the proposed methodology are provided in Section 4.3. The chapter is summarized in Section 4.4.

4.1 The Genetic Algorithm (GA)

The Genetic Algorithm [90, 91] is one of the most commonly used optimization techniques amongst a set of evolutionary computational algorithms and was inspired by Darwin's theory of evolution [92]:

Individuals having advantageous variations are more likely to survive and reproduce than those without the advantageous variations.

The formal idea of the algorithm was developed by John Holland in the late 1960s and early 1970s and extended by Goldberg in the late 1980s [90]. The basic concept of the algorithm is to model Darwin's evolutionary process where members of a population of *potential solutions* compete with each other for survival based on their inherent behavior. These potential solutions are usually encoded as strings called chromosomes, and each chromosome is associated with a fitness value that indicates its optimality with respect to the current optimization criterion. The fitness value of a solution will affect its probability of being selected for further evolution. Typically, chromosomes are binary strings of a fixed length, but other forms of strings are also used to encode the chromosomes. In this work, we use a base-10 representation of chromosomes. The goal of the evolutionary process is to generate new candidate solutions using the genetic operators: Selection, Crossover, and Mutation [61]-[63], [93].

Figure 4.1: Probabilistic Selection Mechanism Provided by GA Roulette Wheel. Chromosome 4 has the highest chance to be selected for the reproduction process.

4.1.1 Genetic Operators

The genetic operators are described in the following:

4.1.1.1 Selection

A *Roulette Wheel Selection* method is used to select chromosomes for reproduction in proportion to their score as indicated by a fitness function. The *Roulette Wheel* is a probabilistic selection method: a technique in which a chromosome with higher fitness score in the population has a better chance to be selected for reproduction since it occupies a bigger slot on the Roulette Wheel (see Figure 4.1).

The Roulette Wheel selection process is summarized as follows [64]:

- a.* Each chromosome in the population, C_m , is associated with a fitness value, $F_m = F(C_m)$ and the total fitness of the current population, F_{sum} is calculated:

$$F_{sum} = \sum_{m=1}^M F_m, \quad (4.1)$$

where M is the population size and is kept constant during the GA operation;

- b.* A random number n , between 0 and F_{sum} is generated:

$$n = F_{sum} \times rand, \quad (4.2)$$

where $rand$ is a normalized pseudorandom number and $0 \leq rand \leq 1$;

c. The fitness of each individual is added to F_{sum} , i.e., $F_{sum} = F_{sum} + F(C_m)$, and the chromosome is selected if the new F_{sum} is greater than n :

$$Select(C_m) \mid_{n \geq (F_{sum})} . \quad (4.3)$$

In GA, we adopted the Roulette Wheel selection mechanism in selecting the parents for *reproduction*, where candidates with higher fitness scores are more likely to be selected. On the other hand, once the best individual is found, it can always be retained and can survive into future generations.

4.1.1.2 Reproduction

Crossover and Mutation are used to generate and manipulate a diverse number of individuals in the current population by allowing candidate solutions with higher fitness values a better chance to act as parents to generate offspring. In this way, they can pass desired traits of their own to the next generation. At the same time, they are able to evolve further themselves to become even better solutions [64].

A. Crossover

Crossover is the main reproduction operator to generate new individuals in the GA. The main idea of this operation is to combine segments of chromosomes from selected parents to form a new individual, which is similar to the crossover process in genetics. Through this process, a new individual can inherit useful characteristics from its parents, which eventually leads to a higher fitness value. The crossover operation can be performed through a variety of different methods that specify how the GA combines two individuals (i.e., parents) to form a new (and more fit) offspring for the next generation. The three commonly used methods are Single-point, Two-point, and Uniform crossovers [91].

Figure 4.2: Single-point crossover operation.

- In *Single-point crossover* a single crossover position is chosen randomly, and segments of the parents are exchanged according to this chosen position as illustrated in Figure 4.2.
- In *Two-point crossover* two crossover positions are chosen randomly and segments of the parents are exchanged according to the crossover points to generate new individuals as illustrated in Figure 4.3.

Figure 4.3: Two-point crossover operation.

- In *Uniform crossover*, each gene has the same probability of being selected and exchanged from the parent chromosomes to generate new children as illustrated in Figure

Figure 4.4: Uniform crossover operation.

4.4.

B. Mutation

Changing the value of a gene in a randomly chosen position along with a chromosome generated from the crossover operation is called mutation. Generally, mutation is used to maintain genetic diversity from one generation to the next through the constant generation of new genes. This helps the algorithm to avoid local minima by preventing the population from becoming too similar to each other [94]. Thus, mutation is regarded as a supplemental reproduction operator in GA. The probability of mutation is usually very small; therefore, the newly created individuals are not much different than their original parents (Figure 4.5).

4.1.1.3 Replacement

A selective mechanism to produce the new generation by replacing the current population with the parent chromosomes and the newly generated offspring is called replacement. This process aims to select the best individuals as well as to maintain diversity in the population. Elitism is the replacement mechanism that is used in this work. Elitism specifies the number of the fittest individuals and guarantees their survival for the next generation. Elitism is important for the generation of better solutions since it selects and breeds the best parents,

Figure 4.5: Mutation Operation.

using them to replace the worst individuals into the next generation. As a result, the algorithm converges more quickly toward optimal solutions.

4.1.1.4 Termination Criteria

There are several criteria that can contribute to the termination of the algorithm. One of the most widely used approaches is to reach a maximum number of possible generations. This can force the algorithm to stop even if it is not converged completely. Our approach is to stop the algorithm when there is no significant improvement in the best fitness value as perceived for a specific number of seconds. This provides enough opportunity for the algorithm to explore the population for the optimal (or near-optimal) solutions for the feature weight detection problem.

4.1.2 The GA Algorithm

Figure 4.6 outlines the relationships between the GA's operators described above. The main steps of the algorithm are as follows:

Initialization: The algorithm is started with the random selection of an initial population of chromosomes of size M . Randomization is crucial for the convergence of the algorithm

Figure 4.6: The Genetic Algorithm Flowchart.

since it provides an equal opportunity for the chromosomes to contribute to the search process.

Fitness Evaluation: The fitness of individual chromosomes in the current population is measured and they are scored accordingly. The fitness function is an application-dependent cost function that will be minimized through the GA operation.

Selection: The fittest members of the current population (parents) are selected for reproduction. This selection is a linear search through a Roulette Wheel with slots weighted in proportion to the chromosomes' fitness scores.

Reproduction: New individuals (children) are created using the genetic operators - Crossover and Mutation. The reproduction part of GA is a process in which individual strings are copied according to their fitness to create a mating pool. As a result, chromosomes with higher fitness scores will have a greater chance to reproduce.

Replacement: A new population is generated by replacing the offsprings as well as the fittest parents with the less fit members of the current population.

Continuation: The *Fitness Evaluation* step is repeated until no noticeable improvement in the best fitness value is perceived for a user-defined number of seconds.

4.2 Genetic Algorithm and Automatic CBIR System (GA-CBIR)

Although the use of the DSOTM classifier in conjunction with the process of query modification was perceived to be an effective solution for enhancing the performance of image retrieval systems, multiple criteria can still influence the optimality of this automatic approach.

First, direction in the growth of the DSOTM topology is entirely dependent on the location of the query in the space. If the query is located on the boundaries of the relevant group, some of the irrelevant samples might be mistakenly classified as part of the relevant class. In addition, the actual shape of the relevant class might also be jeopardized by generating unnecessary centers within this region. Figures 4.7(a) and 4.7(b) illustrate the effect of the query location on the DSOTM classification. From this figure, it is evident that process of classification based on *statistical similarity* alone can not entirely reflect the actual *semantic resemblance* between target images. On the other hand, genetic algorithm is quite capable of biasing the search process toward what is believed to be the best representative and distinctive traits among different image classes. This characteristic of GA enables CBIR systems to tailor the retrieval process to the users' subjectivity by encoding the semantic information from the image categories and using them for both classification and query modification processes. Figures 4.7(c) and 4.7(d) illustrate the DSOTM classification after the extraction of common traits (features) from the query class and the modification of the query position toward the best representative location of such traits using the GA.

Secondly, the choice on *number* and *variety* of features used to represent images and the degree of their *contributions* to successfully differentiate and distribute samples within their particular clusters can directly affect the DSOTM classification. The high dimensionality of low-level features resulting from image indexing might impose an unfavorable influence on the classification process. It is usually thought that a better classifier can be achieved by applying more features to describe individual images. It has been observed that beyond a certain point, the inclusion of additional features leads to a worse rather than a better classification result [93]. This is due to the de-constructive effect of redundant or even

(a)

(b)

(c)

(d)

Figure 4.7: GA-based feature weight detection and its effect on enhancing performance of the DSOTM classification: (a) Input space with the query located on the boundaries of the relevant group; (b) Classification result using DSOTM of the input pattern in part (a); (c) The modified query position using the GA-based feature weight detection principle; and (d) Classification result using DSOTM of the input pattern in (c).

irrelevant features on the classification process. Hence, de-emphasis of the features that do not necessarily contribute significant information about the relevance of an image to the retrieval process can bias the classifier even further toward the discovery of common *conceptual* traits among image groups (i.e., the query class) and, thus, improves the retrieval

process as illustrated in Figure 4.8. In this figure, the contribution of individual features is accurately determined by the GA and through the principle of feature weight detection. The bottom figure represents the retrieval of a similar query after the process of GA feature weight detection. As expected, the GA operation appears to identify the features important in aiding what humans appear to attentively discriminate, by the detection of higher weights that emphasize both texture and shape features as opposed to color features, which has a moderate contribution to the DSOTM classification.

The main focus of this section is therefore to provide an enhanced method for reducing the gap between low-level descriptors and high-level concepts (i.e., semantics), such that a CBIR system can achieve further improved performance while in automatic retrieval mode, through the principle of feature weight detection. A genetic algorithm (GA) -based approach is used in conjunction with the Directed Self-Organizing Tree Map (DSOTM). The resulting framework is referred to as GA-based CBIR (GA-CBIR), and aims to generalize the image retrieval application to real-life photos stored in more generic digital archives. GA-CBIR mimics the human-feedback mechanism proposed in [30, 88] by automatically adapting the search process to what the system evolves to believe is significant content within the query. In this engine, DSOTM is a major component of the GA analysis as it works closely with traditional GA operators to achieve a more robust perceptual classification - one that is more in tune with the apparent discriminative characteristics observed in an image by a human user [95].

4.2.1 The GA-CBIR Building Structure

The block diagram of the proposed GA-CBIR system is illustrated in Figure 4.9 and in more detail in Figure 4.10. In the *Weight Detection* module of the GA-CBIR engine, a GA-based approach is used to measure the importance of individual features and the degree of their contribution toward improving the retrieval process and overall performance of the system. To avoid the need for the encoding and decoding of the chromosomes a base-10 GA is employed, which operates similarly to the binary GA described previously except that

(a)

(b)

Figure 4.8: The degree of contribution of individual feature descriptors can affect both the

Figure 4.8-Cont'd: DSOTM classifier and the CBIR system: (a) 62.5% retrieval rate for the query image (on top-left corner) using the DSOTM classifier without feature weight detection (with the same degree of feature weights); and (b) 100% retrieval rate for the same query using feature weight detection. Color feature offers the least contribution, while texture and shape features provide maximum amount of information for the DSOTM classifier.

genes vary between 0 and 9, and there is an extra digit for the sign representation [96]. Each chromosome carries information about a given number of traits: the feature weights, in our case, are the actual parameters to be varied to optimize the fitness function. These genes are combined with each other to create an initial population of M chromosomes in the range of $[m, 1]$, where m is a user defined negative value specifying a boundary for the navigation of the GA algorithm. The empirical range for m is found to be $[-1, 0]$. This range will give the GA a variety of options to use in the exploration of the negative range of real numbers, and to properly emphasize or de-emphasize features to a greater extent, as input feature vectors are initially normalized to the range of $[0, 1]$.

In the CBIR engine of Figure 4.10, the second unit of the *Initial Search* module, the *Feature Extraction 1*, deals with calculating features from a high volume image database. Consequently, a standard set of content descriptors (for example, MPEG-7), is extracted to provide a more generic and rapid interface to existing databases based on the chosen standard. Extracted features are used to retrieve the most similar images (in the relative sense) based on a predefined distance metric (i.e., City-block or Euclidean distance). The top Q images are then displayed back to the user through an *Interface* block. Subsequently, the user may request an automatic search, which operates independently. Upon this request, control of the system is switched to path (a) of the *GA-based automatic search* module, wherein another set of features (usually of higher perceptual quality) are extracted from the top Q retrieved images from the initial search, using *Feature Extraction 2*. Although computation of such features could be intensive, they become feasible at this stage since they are only computed for a few retrieved images from the previously rank-based process. This module allows for the use of more proprietary or specialty features which may enhance perceptual discrimination beyond that which might otherwise be possible through standard

Figure 4.9: Machine-controlled GA-based CBIR system [68, 81].

Figure 4.10: GA-CBIR System [68, 81].

features alone. These features are then used as seeds to train the unsupervised classifier.

To integrate the *Weight Detection* module into the retrieval process, the system is first transferred to the distance measurement phase to identify the top Q similar images. The system's response to different weights is subsequently quantified through the measurement of the ability of an unsupervised classifier (i.e., the DSOTM) in correctly classifying selected images by taking the path (b). A new query, based on selected images from the previous iterations, is then adapted to substantially represent the relevant class through the *Query Modification* module. The system's response to identified weights, in terms of retrieval accu-

racy, is detected by the system based on the pre-classified database in the *Fitness Evaluation* block. A new population is then generated in the *Replacement* module through the preparation of the most fit chromosomes for breeding through *Selection* and *Reproduction* blocks. The detected feature weights and final retrieval results are then displayed back to the user upon the convergence of the algorithm.

Since the DSOTM seeks to provide a better conclusion regarding the resemblance of input samples to one another while the GA aims to select and reproduce a more fit population of candidate solutions, a DSOTM-GA combination can effortlessly replace the relevance feedback learning process to achieve a fully-automated retrieval engine. The GA thus attempts to mimic human attention by working and coordinating the search subjectively, by feeding and evaluating results from the more objective DSOTM. This is possible because of the highlighting of dominant characteristics of images through the principle of feature weight detection (see Figure 4.8). Such a detection mechanism seeks to emphasize certain characteristics in an image (i.e., its color, texture, and/or shape) that might provide significant information to the DSOTM for a more robust classification as illustrated in Figure 4.7. Assigning proper weights for individual features can significantly increase the performance behavior of CBIR systems, reducing the need for human supervision.

4.2.2 The GA-based Feature Weight Detection Algorithm

A chromosome in the base-10 GA is defined as $C = \{C_{Color}, C_{Texture}, C_{Shape}\}$, where C is any real number and is defined in the range of $m \leq C \leq 1$. The length of the chromosome depends on the desired precision of the weights, in terms of digit decimal fraction. These chromosomes are used as feature weights for the current query and are the main variables to be manipulated by GA operators with the aim of enhanced retrieval results. The process of GA weight detection is summarized as follows:

Initialization: A random initial population $P_0 = \{C_1, \dots, C_m, \dots, C_M\}^T$ is created, where M is the population size and T is the transpose factor;

Evaluation: The fitness value for each individual chromosome is computed in the current population and they are scored accordingly. The process of fitness calculation for each chromosome is as follows:

- a. Let $z = [z_c, z_t, z_s]$ represent the selected query image described by its *color*, *texture*, and *shape* feature vectors;
- b. The top Q statistically relevant images are selected by minimizing a predefined distance criterion;
- c. A new query feature vector is created, z_{New} , where $z_{New} = [C_{Color} \times z_c, C_{Texture} \times z_t, C_{Shape} \times z_s]$;
- d. The training matrix, $\mathbf{R}$, is constructed from the relevant class identified by the DSOTM algorithm (see Chapter 2);
- e. The query position is modified to the center of mass of the relevant class using the method described in Chapter 2;
- f. The response of the GA-CBIR system is investigated to find the presence of new feature weights. Performance is measured based on the number of images retrieved from Q that fall into the query class;
- g. The precision of the retrieval, $P_r(N_c)$, is detected in terms of the retrieval rate (RR), where $P_r(N_c) = \frac{N_R}{N_c} \in [0, 1]$, N_R denotes the number of relevant images retrieved, and N_c denotes the number of retrieved images [30];
- h. The fitness function, $F(w)$, is calculated using Equation 4.4:

$$F(w) = e^{-P_r(N_c)}. \quad (4.4)$$

The fitness function in the above equation is computed by measuring the system's performance against various feature weights and its minimization through the determination of the inverse of this values since function maximization is essentially the minimization of the inverse of the same function. Also, the use of an exponential operator can discriminate the system's performance non-linearly and greater than a linear function;

Selection: A Roulette Wheel method is used to select chromosomes for reproduction in proportion to their scores as indicated by the fitness function;

Reproduction: More fit individuals are created for the next generation through the exchange of genetic materials of the most fit chromosomes in the mating pool (crossover) and the random manipulation of their genes (mutation);

Replacement: The parents are replaced with their newly-created children and the few most fit parents are allowed to pass on to the next generation without any genetic modification.

The above procedure enables GA to search among possible solutions in the provided range and to select the optimum weight for each feature vector in order to maximize the discrimination with respect to significant query content and eventually maximize the *RR*.

4.3 Experimental Results: Feature Weight Detection in the GA-CBIR Application

A number of experiments were conducted to compare the behaviors of the automatic CBIR and GA-CBIR engines using the SOFM, SOTM, and DSOTM clustering algorithms. Simulations were carried out on the same subset of the Corel image database used in Chapter 3.

Table 4.1 illustrates the adapted GA parameter settings in our experiments. A base-10 GA algorithm with an initial population of 50 chromosomes, defined within the range of $[-1, 1]$, with a maximum of 100 iterations, was employed. The size of the initial population was decided empirically and kept constant throughout the entire process of feature weight detection. Figure 4.11 shows the effect of the employment of various crossover probabilities on the automatic CBIR performance. The crossover probability was defined as the probability of the occurrence of crossover between any two selected chromosomes. In this figure, the highest *RR* was archived at the 0.8 level. Therefore, a two-point crossover operation with a probability of 80% was utilized. The mutation probability, on the other hand, is the

Table 4.1: Parameter setting for the GA

Parameter	Setting
Population Size	50
Crossover Type	Two-point
Crossover Probability - P_C	80%
Mutation Probability - P_M	5%
Selection Mechanism	Roulette Wheel
Termination Criterion	Best score does not change over 20 seconds

probability of the flipping of a gene to another value. The mutation probability was set to 5%. The Roulette Wheel method was used to select chromosomes in proportion to their score as indicated by the fitness function. Consequently, a chromosome with higher fitness score would have a better chance of being selected for reproduction.

Figure 4.11: Comparing CBIR performance in terms of RR for varying crossover fractions.

Figure 4.12 illustrates 2-D plots of the effects of the GA parameters (i.e., population size as well as crossover and mutation probabilities) on the process of feature weight detection. Figure 4.12 is a plot of feature weights vs. the GA-CBIR performance in terms of RR . The x and y axes represent feature weights (chromosomes) and the z axis represents the system's performance against applied weights. The zero level in this figure represents the performance of the automatic CBIR engine, structured around the DSOTM learning algorithm alone. The different combinations of feature weights can enhance or deteriorate retrieval results with respect to the zero level, and are represented by the local and global maxima and minima. The global maximum represents the optimum retrieval rate specifying the best combination of feature weights characterized by the GA. For instance, the function under optimization in Figure 4.12 has one global maximum characterized by feature weights of (15, 20), one global minimum described by feature weights of (15, 5), and various other local variations for each feature value in the range of $[0, 30]$ correspond with x and y coordinates (features). Having a large solution space as well as a significant number of maxima and minima (fluctuations), as depicted by this function, makes it difficult for standard gradient-based methods to locate the global maximum (or minimum) without getting trapped in local solutions. Conversely, the GA is quite capable of searching highly nonlinear and large spaces for an optimal solution.

Decisions about population size represent a trade-off between the accuracy and the convergence time of the GA optimization process. A large initial population is generally effective for the GA operation as it allows the potential solutions to span more of the search space and attain more accurate results; however, by increasing the population size, the processing time of the algorithm can be significantly prolonged. Figures 4.12(c)-4.12(e) provide comparison of population size vs. accuracy of the GA's final solutions under fixed crossover and mutation probabilities. As a result, by increasing the size of the population from 20 to 50, and then to 100, the GA can be provided with greater opportunities to search as many regions as possible within the solution space and to converge on the optimal (global) solution.

Figures 4.12(f)-4.12(k) illustrate the ineffective behavior of the GA when two extreme (high and low) crossover probabilities confront. A crossover probability of 100% denotes that

Table 4.2: Experimental results in terms of RR - The GA-CBIR

Classifier	Set A	Set B	Set C	Average
SOFM	61.2%	59.9%	63.6%	61.6%
SOTM	66.8%	65.7%	67.5%	66.7%
DSOTM	71.2%	68.8%	72.3%	70.8%

all the children, besides elite individuals, are crossover offspring, while a crossover probability of 0% signifies all the mutated children in the population.

In the case of dominant crossover children, the algorithm is unable to reproduce new genes since there is no possibility for mutation. Thus, after generating the best individuals in preliminary iterations, subsequent generations are filled with only the best individuals. Having a population of the best individuals hardly improves the fitness criteria and, consequently, the algorithm terminates prematurely. In the situation of dominant mutated offspring, the algorithm is unable to combine the enhanced genes resulting from the random changes in the mutation operation, to improve the fitness criteria since there is no chance for crossover. Having a genetically enhanced population - but not inclusive of the best individuals - will never improve the fitness value over time and the algorithm will therefore stop unexpectedly.

The simulations used here were carried out using a subset of the Corel image database [89] consisting of nearly 12000 JPEG color images, covering a wide range of real-life photos, from 100 different categories. Each category consisted of 100 visually associated objects to simplify measurements of the retrieval accuracy (RR) during the experiments. Three sets of 120 images were drawn from the database to form sets A , B , and C . Each set consists of randomly selected images such that no two images come from the same class. Retrieval results were statistically calculated from each of the three sets. In the simulations, a total of 16 ($Q = 16$) most relevant images were retrieved to evaluate retrieval performance. The experimental results are illustrated in Table 4.2. DSOTM outperformed both SOFM and SOTM in the GA-CBIR framework. The effectiveness of the use of genetic-based algorithm in the structure of CBIR systems was also evident from these results.

4.4 Summary

This chapter demonstrated the framework for a novel computer-controlled CBIR system based on the DSOTM and the GA feature weight detection technique. The resulting framework is referred to as GA-based CBIR (GA-CBIR), and aims to close the gap between low-level descriptors and high-level concepts (i.e., semantics) by mimicking human attention through the highlighting of this dominant characteristics of images through the principle of feature weight detection. This comprehensive study illustrated the feasibility of the proposed GA feature weight detection scheme in conjunction with the DSOTM, SOTM, and SOFM classifier techniques.

(a) *Function to be optimized.*

(b) *Top view of the function in (a).*

(c) Contour map: Population size: 20, $P_C = 80\%$, and $P_M = 5\%$.

(d) Contour map: Population size: 50, $P_C = 80\%$, and $P_M = 5\%$.

(e) Contour map: Population size: 100, $P_C = 80\%$, and $P_M = 5\%$.

(f) Contour map: Population size: 20, $P_C = 100\%$, and $P_M = 5\%$.

(g) Contour map: Population size: 50, $P_C = 100\%$, and $P_M = 5\%$.

(h) Contour map: Population size: 100, $P_C = 100\%$, and $P_M = 5\%$.

(i) Contour map: Population size: 20, $P_C = 0\%$, and $P_M = 5\%$.

(j) Contour map: Population size: 50, $P_C = 0\%$, and $P_M = 5\%$.

(k) Contour map: Population size: 100, $P_C = 0\%$, and $P_M = 5\%$.

Figure 4.12: Effects on the population size and crossover and mutation probabilities on the GA-based optimization.

Chapter 5

Conclusions

5.1 Summary of Thesis

THE framework for a novel unsupervised clustering algorithm which is able to closely mimic the process thought to be at work in the human brain [59, 97] was introduced in the first part of this work. It was observed that the new unsupervised model, DSOTM, is able to provide an effective representation of the nature of the data by assigning centers to the image groups and preventing any center to be allocated in low- or zero-density regions in the data space. The strength of the proposed method is in its nominal dependency on the number of training seeds (samples) and its architecture that is capable of undergoing continuous deformation during its training phase. The training phase in this algorithm is controlled by a top-down process that performs the search globally at the initial phases of the training process and localizes it in the later stages of the search.

The proposed approach was applied to the feedback loop of the CBIR system in order to provide an objective measurement of a resemblance of images with respect to the supplied query and to automate the search process. In this application, the decision on association of input pattern to query image is gradually made as each sample is presented to the system. It also contains a controlling mechanism that keeps track of the query center by forcing the center of relevant class to remain in the vicinity of the query position. Therefore, it dynamically controls generation of new centers and determines the relevance of input samples, with respect to the query, as its tree structure grows. In other words, it constrains

cluster generation by preventing redundant centers to be spawned around the query position since this part of the network is already occupied by relevant class center. The experimental results also illustrate a promising performance by utilizing the DSOTM in the structure of automatic CBIR engines.

The second part of the thesis was involved in optimizing the overall performance of the CBIR engine using an evolutionary process, the genetic algorithm. As it was mentioned previously, the process of image retrieval can be viewed as an optimization problem since the CBIR performance is highly sensitive to dominant characteristics of images highlighted by the GA through the principle of feature weight detection. The feature weight detection mechanism seeks to emphasize certain characteristics in an image (i.e., its color, texture, and shape) that might provide significant information to the DSOTM for a more robust classification. The GA thus attempts to mimic human attention by working and coordinating the search subjectively, feeding and evaluating results from the more objective DSOTM.

5.2 Future Research Extension

Developments in artificial intelligence systems in recent years has led to the adoption of alternative approaches by a number of perceptual theorists. The perceptual theories are now adapted in the form that allow computers to directly interpret sensory information in the same manner that a human observer might [97]. David Marr's presumption [4] on direct perception comports well with current theories on machine perception. According to Marr, all the information required for direct perception is embedded in the stimulus input in the form of some simple dimensions (i.e., boundaries and edges, line endings, particular patterns where stimuli meet, etc.) In order for computers to interpret or analyze this high-volume of information, which is usually accessible in the form of mathematical steps or equations, additional calculation of those aspects of the stimuli that contain the most useful information (i.e., feature extraction) is required.

The result of current research in the field of computational intelligence, which is primarily influenced by Marr's perception theory, has proven inadequate to model the human

perceptual sensory system solely on the basis of low-level information contained in the stimuli (images in our case) at any one moment in time. In addition to the above low-level (i.e., logical) information available to our sensory system, which is processed in a *bottom-up* model, the information from our previous experiences, our expectations, and our cognitive processing strategies are influential in drawing a meaningful conclusion regarding the nature of a stimulus input. Moreover, human visual perception may involve other sources of information, some non-visual in nature, which may arise from our past and are processed through a *top-down* processing model [97].

The process of image retrieval is heavily dependent on machine perception and, thus, is reliant on the way images are interpreted for the retrieval system. Combining cognitive theories with the process of feature extraction, classification, and similarity matching is crucial for future success in this field. Although DSOTM has facilitated our approach to achieve a more robust classifier through its *top-down* processing architecture, it still suffers from dependencies on numerous controlling factors, namely, the learning rate and the threshold. This dependency can make the classifier to be less adaptive to the true nature of the data, and thus, unsuccessful in some applications. Integrating some kind of *bottom-up* approach with its top-down process to adjust the algorithm's learning as well as its localization capability, such as the one we know is utilized in the human brain for day-to-day information classification, can make this classifier more adaptive to any type of data, with different densities and distributions.

Measuring the DSOTM's clustering performance in terms of the quality of generated clusters, which is critical for guiding the GA to select a better combination of feature weights, without human supervision is essential for achieving a fully automatic retrieval engine. Even though, the amount of human supervision required in the GA-CBIR engine is less than would be required in any typical relevance feedback type CBIR system, the need for a fully automated retrieval process is still vital.

There is no doubt that avoiding an integration of the cognitive processes with machine learning algorithms will patch but not fully solve the image retrieval problem. Certainly

this requires further research into the realms of human cognition, psychology, and neurology - realms that have gone relatively unexplored with respect to the society of computer intelligence.

Bibliography

- [1] H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," *IBM Journal of Research and Development*, pp. 309-317, October 1957.
- [2] Google, <http://www.google.ca>.
- [3] Yahoo, <http://www.yahoo.com>.
- [4] D. Marr, *Vision: a computational investigation into the human presentation and processing of visual information*, W.H. Freeman and Company, New York, 1982.
- [5] J. Luo, M. Boutell, and C. Brown, "Pictures are not taken in a vacuum - an overview of exploiting context for semantic scene content understanding," *IEEE Signal Processing Magazine*, Vol. 23, Issue 2, pp. 101-114, Mar 2006.
- [6] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Pearson Education Inc., 2004.
- [7] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7 Multimedia Content Description Interface*, J. Wiley & Sons Ltd., Chichester, England, 2002.
- [8] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in *Proc. of IEEE Int. Conf. Image Processing*, pp. 815-818, Santa Barbara, CA, 1997.
- [9] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.18, pp. 837-842, Aug. 1996.

- [10] P. Muneesawang and L. Guan, "A nonlinear RBF model for interactive content-based image retrieval," in *Proc. of 1st IEEE Pacific-Rim Conf. Multimedia*, pp. 188-191, Sydney, Australia, Dec. 2000.
- [11] Y. Zhuang, X. Liu, and Y. Pan, "Apply semantic template to support content-based image retrieval," in *Proc. of SPIE Storage Retrieval Multimedia Databases*, pp. 422-449, San Jose, CA, Dec. 1999.
- [12] P. Muneesawang and L. Guan, "Interactive CBIR using RBF-based relevance feedback for WT/VQ coded images," in *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Vol. 3, pp. 1641-1644, Salt Lake City, UT, May 2001.
- [13] N. Vasconcelos and A. Lippman, "Bayesian representations and learning mechanisms for content-based image retrieval," in *Proc. of SPIE Storage Retrieval for Multimedia Database*, Vol. 3972, pp. 43-54, Jan. 2000.
- [14] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papatthoman, and P. N. Yianilos, "The Bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments," *IEEE Trans. Image Processing*, Vol. 9, No. 1, pp. 20-37, Jan. 2000.
- [15] C. S. Lee, W. Y. Ma, and H. J. Zhang, "Information Embedding Based on User' Relevance Feedback for Image Retrieval," in *Proc. SPIE Int. Conf. Multimedia Storage and Archiving Systems IV*, pp. 19-22, Boston, Sept. 1999.
- [16] M. E. Wood, N. W. Campbell, and B. T. Thomas, "Iterative refinement by relevance feedback in content-based digital image retrieval," in *Proc. of the 6th ACM Int. Conf. on Multimedia*, pp. 13-20, Bristol, U.K., Sept. 1998.
- [17] I. El-Naqa, M. N. Wernick, Y. Yang, and N. P. Galatsanos, "Image retrieval based on similarity learning," in *Proc. of the Int. Conf. on Image Processing*, Vol. 3, pp. 722-72, Vancouver, BC, Canada, 2000.

- [18] R. Brunelli and O. Mich, "Image retrieval by examples," *IEEE Trans. on Multimedia*, Vol. 2, pp. 164-171, Sept. 2000.
- [19] Y. Rui and T. Huang, "Optimizing learning in image retrieval," in *Proc. of IEEE int. Conf. On Computer Vision and Pattern Recognition*, pp. 236-245, Hilton Head, SC, USA, Jun. 2000.
- [20] A. Celentano and E. Di Sciascio, "Feature integration and relevance feedback analysis in image similarity evaluation," *Journal of Electronic Images*, Vol. 7, No. 2, pp. 308-317, 1998.
- [21] T. Gevers and A. W. M. Smeulders, "PicToSeek: Combining color and shape invariant features for image retrieval," *IEEE Trans. on Image Processing*, Vol. 9, pp. 102-119, 2000.
- [22] J. Peng, B. Bhanu, and S. Qing, "Probabilistic feature relevance learning for content-based image retrieval," *Computer Vision and Image Understanding*, Vol. 75, No. 1, pp. 102-119, Jan. 2000.
- [23] G. Ciocca and R. Schettini, "Using a relevance feedback mechanism to improve content-based image retrieval," in *Proc. of the 3rd Int. Conf. on Visual Information and Information Systems*, Vol. 1614, pp. 105-114, Springer-Verlag, London, UK, 1999.
- [24] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 644-655, Sept. 1998.
- [25] E. Di Sciascio and M. Mongiello, "DrawSearch: A tool for interactive content-based image retrieval over the net," in *Proc. of SPIE Storage and Retrieval for Image and Video Databases VII*, Vol. 3656, pp. 561-572, 1999.
- [26] H. Mller, W. Mller, S. Marchand-Maillet, and D. M. Squire, "Strategies for positive and

- negative relevance feedback in image retrieval,” in *Proc. IEEE Int. Conf. on Pattern Recognition*, Vol. 1, pp. 1043-1046, Barcelona, Spain, Sep. 2000.
- [27] K. Jarrah, M. Kyan, I. Lee, and L. Guan, “Application of image visual characterization and soft feature selection in Content-Based Image Retrieval,” in *Proc. of SPIE Multimedia Content Analysis, Management, and Retrieval (SPIE’06)*, Vol. 6073, pp. 101-109, San Jose, California, USA, Jan. 2006.
 - [28] M. J. Swain and D. H. Ballard, “Indexing Via Color Histograms,” in *Proc. of IEEE 3rd Int. Conf. on Computer Vision*, pp. 390-393, Osaka, Dec 1990.
 - [29] Y. Rui, *Efficient indexing, browsing and retrieval of image/video content*, PhD Thesis, University of Illinois, 1999.
 - [30] P. Muneesawang, *Retrieval of Image/Video Content by Adaptive Machine and User Interaction*, PhD Thesis. University of Sydney, Australia, 2002.
 - [31] M. Stricker and M. Orengo, “Similarity of color images,” in *Proc. of SPIE Storage and Retrieval for Image and Video Databases III*, Vol. 2420, pp. 381-392, San Jose, USA, Feb. 1995.
 - [32] J. R. Smith and S. F. Chang, “Automated binary texture feature sets for image retrieval,” in *Proc. of IEEE Int. Conf. on Acoustics, speech and Signal Processing*, Vol. 4, pp. 2239-2242, Atlanta, GA, USA, May 1996.
 - [33] T. Chang and C. C. J. Kuo, “Texture analysis and classification with tree-structured wavelet transform,” *IEEE Trans. on Image Processing*, Vol. 2, No. 4, pp. 429-441, Oct. 1993.
 - [34] M. H. Gross, R. Koch, L. Lippert, and A. Dreger, “Multiscale image texture analysis in wavelet spaces,” in *Proc. of 1st IEEE Int. Conf. on Image Processing*, Vol. 3, pp. 412-416, Nov. 1994.

- [35] A. Kundu and J. L. Chen, "Texture classification using qmf bank-based subband decomposition," *Journal Computer Vision, Graphics and Image Processing*, vol. 54, No. 5, pp. 369-384, Sep. 1992.
- [36] P. Muneesawang, and L. Guan, "Multiresolution-histogram indexing and relevance feedback learning for image retrieval," in *Proc. of IEEE Int. Conf. on Image Processing*, Vol 2, pp. 526-529, Vancouver, Canada, Sept. 2000.
- [37] ISO/IEC JTC 1/SC 29/WG 1, ISO/IEC FDIS 15444-1: *Information technology - JPEG 2000 image coding system: core coding system [WG 1 N 1890]*, Sep. 2000.
- [38] K. Jarrah, P. Muneesawang, I. Lee, and L. Guan, "Minimizing Human-Machine Interactions in Automatic Image Retrieval," in *Proc. of IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE'04)*, Vol. 3, pp. 1589-1592, Niagara Falls, Canada, May 2004.
- [39] W. Y. Ma and B. S. Manjunath, "A comparison of wavelet transform features for texture image annotation," in *Proc. of IEEE Int. Conf. on Image Processing*, Vol. 2, pp. 256-259, Oct. 1995.
- [40] D. Zhang, A. Wong, M. Indrawan, G. Lu, "Content-Based Image Retrieval Using Gabor Texture Features," in *Proc. of 1st IEEE Pacific-Rim Conference on Multimedia*, pp. 392-395, Sydney, Australia, Dec. 2000.
- [41] E. Persoon and K. S. Fu, "Shape Discrimination using Fourier Descriptors," *IEEE Trans. on System, Man and Cybernetics*, Vol. 7, No. 3, pp. 170-179, Mar. 1977.
- [42] R. Chellappa and R. Bagdazian, "Optimal Fourier Coding of Image Boundaries," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 1, pp. 102-105, Jan. 1984.
- [43] C. C. Lin and R. Chellappa, "Classification of Partial 2D Shapes using Fourier Descrip-

- tors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 5, pp. 686-690, 1987.
- [44] C. S. Lin and C. L. Hwang, "New Forms of Shape Invariants from Elliptic Fourier Descriptors," *IEEE Conf. on Pattern recognition*, Vol. 20, No. 5, pp. 535-545, Elsevier Science Inc., New York, NY, USA, 1987.
 - [45] A. Krzyzak, S. Y. Leung and C. Y. Suen, "Reconstruction of two-dimensional Patterns from Fourier Descriptors," in *Proc. of IEEE Conf. on Machine Vision and Applications*, Vol. 2, No. 2, pp. 123-140, 1989.
 - [46] K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger, "Application of Affine-Invariant Fourier Descriptors to Recognition of 3-D Objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 7, pp. 640-647, Jul. 1990.
 - [47] H. Kauppinen, T. Seppanen and M. Pietikainen, "An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 2, pp. 201-207, Feb. 1995.
 - [48] B. M. Mehtre, M. S. Kankanhalli and W. F. Lee, "Shape measures for Content Based Image Retrieval: A Comparison," in *Proc. of IEEE Cnf. on Information Processing & Management*, Vol. 33, No. 3, pp. 319-337, Jun. 1997.
 - [49] Y. Rui, A. C. She, and T. S. Huang, "Modified Fourier descriptors for shape representation - a practical approach," in *Proc. 1st Int. Workshop on Image Database and Multimedia Search*, Amsterdam, The Netherlands, Aug. 1996.
 - [50] Q. Chen, E. Petriu, and X. Yang, "A Comparative Study of Fourier Descriptors and Hu's Seven Moment Invariants for Image Recognition," in *Proc. of IEEE Canadian Conf. on Electrical and Computer Engineering*, Vol. 1, pp. 103-106, Niagara Falls, Canada, May 2004.

- [51] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. on Information Theory*, Vol. IT-8, pp.179-187, Feb. 1962.
- [52] S. Paschalakis and P. Lee, "Pattern recognition in gray level images using moment based invariant features," in *Proc. IEEE Conf. on Image Processing and its Applications*, No.465, pp.245-249, 1999.
- [53] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, 2nd Edition, Elsevier Science, 2003.
- [54] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. on Computer*, Vol. 26, No. 9, pp. 917-922, Sep. 1977.
- [55] R. Kohavi and G. H. John, "Wrappers for feature subset selection," in *Proc. IEEE Int. Conf. on Artificial Intelligence*, Vol. 97, No. 1-2, pp. 273-324, Dec. 1997.
- [56] Y. Wang, *Recognizing Human Emotional State from Audiovisual Signals*, M.A.Sc Thesis, Ryerson University, Canada, 2005.
- [57] W. Bhatti, Y. Wang, and L. Guan, "A neural network approach for human emotion recognition in speech," in *Proc. of IEEE Int. Symposium on Circuits and Systems*, Vol. 2, pp. 181-184, Vancouver, Canada, May 2004.
- [58] D. F. Specht, "A general regression neural network," *IEEE Trans. on Neural Networks*, Vol. 2, No. 6, pp. 568-576, Nov. 1991.
- [59] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Ed. Prentice Hall, New Jersey, 1999.
- [60] Y. Marcoz, and M. Torres-Torriti, "Feature Selection using Genetic Algorithms in SAR airborne imagery," July 2001, Available at: [http:// www.ifrance.com/silkswift/ga/](http://www.ifrance.com/silkswift/ga/).
- [61] W. R. Smart and M. Zhang, "Classification Strategies for Image Classification in Genetic Programming," in *Proc. of IEEE Int. Conf. on Image and Vision Computing*, pp. 402-407, Palmerston North, New Zealand, 2003.

- [62] L. I. Kuncheva, L. C. Jain, "Designing classifier fusion systems by genetic algorithms," *IEEE Trans. on Evolutionary Computation*, Vol. 4, No. 4, pp. 327 - 336, Nov. 2000.
- [63] S. Zehang, Y. Xiaojing, G. Bebis, S. J. Louis, "Neural-network-based gender classification using genetic search for Eigen-Feature selection," in *Proc. of IEEE International Joint Conference on Neural Networks*, pp. 2433-2438, May 2002.
- [64] C. I. Chiu, *Feature Transformation Using Evolutionary Computation for Multimedia Information Content Characterization*, M.A.Sc Thesis, City University of Hong Kong, Hong Kong, 2004.
- [65] L. S. Oliveira and R. Sabourin, "A Methodology for Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Digit String Recognition ," *Int. Journal of Pattern Recognition and Artificial Intelligence*, Vol. 17, No. 6, pp. 903-929, 2003.
- [66] P. Muneesawang, L. Guan, "A Non-Linear RBF Model For Adaptive Content-Based Image Retrieval," *Int. Symp. on Multimedia Information Processing*, pp: 188-191, University of Sydney, Australia, Dec. 2000.
- [67] Y. Li, *An Effective Shape Descriptor for Content-based Image Retrieval*, M.A.Sc Thesis, Ryerson University, Toronto, Canada, 2006.
- [68] K. Jarrah, M. Kyan, S. Krishnan, and L. Guan, "Computational intelligence techniques and their Applications in Content-Based Image Retrieval," *IEEE Int. Conf. on Multimedia & Expo (ICME'06)*, pp. 33-36, Toronto, Canada, Jul. 2006.
- [69] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, New York, McGraw Hill, 1983.
- [70] T. Sigitani, Y. Liguni, and H. Maeda, "Image interpolation for progressive transmission by using radial basis function networks," *IEEE Trans. on Neural Networks*, Vol. 10, No. 2, pp. 381-390, 1999.

- [71] P. Muneesawang, H. S. Wong, J. Lay, and L. Guan, "Learning and Adaptive Characterization of Visual Contents in Image Retrieval Systems," *Handbook on neural networks for signal processing*, Y.H. Hu and J.-N. Hwang, eds, pp. 11.1-11.29, CRC Press, 2001.
- [72] T. Amin, *Application of Laplacian Mixture Model to Image and Video Retrieval*, M.A.Sc thesis, Ryerson University, 2004.
- [73] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial Neural Networks: A Tutorial," *IEEE Computer Magazine*, Vol. 29, Issue 3, pp. 31-44, March 1996.
- [74] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, New York: Wiley, 1949.
- [75] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, Vol. 43, No. 1, pp. 59-69, January 1982.
- [76] H. S. Kong, *The Self-Organizing Tree Map, and its Applications in Digital Image Processing*, PhD Thesis, University of Sydney, Australia, 1998.
- [77] H. Kong and L. Guan, "Detection and removal of impulse noise by a neural network guided adaptive median filter," in *Proc. of IEEE Int. Conf. on Neural Networks*, Vol. 2, pp. 845-849, Perth, Australia, 1995.
- [78] D. H. Hubel and T. N. Wiesel, "Functional Architecture of Macaque Monkey Visual Cortex," in *Proc. of the Royal Society of London*, Vol. 198, pp. 1-59, London, 1977.
- [79] G. G. Yen, and Z. Wu, "Ranked Centroid Projection: a Data Visualization Approach for Self-Organizing Maps," *Proc. of International Joint Conference on Neural Networks*, Vol. 3, pp. 1587-1592, Montreal, Canada, Aug. 2005.
- [80] J. Randall, L. Guan, X. Zhang, and W. Li, "Investigations of the self-organizing tree map," *Proc. Of Int. Conf. on Neural Information Processing*, Vol 2, pp. 724-728, Nov. 1999.

- [81] K. Jarrah, S. Krishnan, and L. Guan, "Automatic Content-Based Image Retrieval Using Hierarchical Clustering Algorithms," *International World Congress on Computational Intelligence (WCCI'06)*, pp. 6564-6569, Vancouver, Canada, Jul. 2006.
- [82] S. Grossberg, *Studies of Mind and Brain*, Reidel Press, Dordrecht, Holland, 1982.
- [83] G. Carpenter and S. Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network," *IEEE Computer Magazine*, Vol. 21, No. 3, pp. 77-87, Mar. 1988.
- [84] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "ARTMAP: Supervised real-time learning and classification of non-stationary data by a self-organizing neural network," *Neural Networks*, Vol. 4, No. 5, pp. 565-588, 1991
- [85] I. Lee, P. Muneesawang, and L. Guan, "Automatic Relevance Feedback for Distributed Content-Based Image Retrieval," *ICGST Int. Journal on Graphics, Vision and Image Processing*, Vol. 5, No. 4, 2005.
- [86] P. Muneesawang, and L. Guan, "Automatic Machine Interactions For Content-Based Image Retrieval Using a Self-Organizing Tree Map Architecture," *Tran. on Neural Networks*, Vol. 13, No. 4, pp. 821-834, Jul. 2002.
- [87] M. Kyan, L. Guan and S. Liss, "Refining competition in the self-organising tree map for unsupervised biofilm image segmentation," *Tran. on Neural Networks, Elsevier*, Vol. 18, No. 5-6, pp 850-860, Aug. 2005.
- [88] P. Muneesawang and L. Guan, "Minimizing user interaction by automatic and semi-automatic relevance feedback for image retrieval," in *Proc. of IEEE Int. Conf. on Image Processing*, Vol.2, pp. 601-604, Rochester, USA, Sep. 2002.
- [89] Corel Gallery Magic 65000, <http://www.corel.com>, 1999.
- [90] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York, Jan. 1989.

- [91] M. Mitchell, *An Introduction to Genetic Algorithms*, Cambridge MA: The MIT Press, Feb. 1998.
- [92] C. Darwin, *The Origin of Species By Means of Natural Selection or the Preservation of Favored Races in the Struggle for Life*, The Book League of America, New York, 1929. Originally published in 1859.
- [93] J. Yang and V. Honavar, "Feature Subset Selection Using a Genetic Algorithm," *IEEE Intelligent Systems*, Vol. 13, No. 2, pp. 44-49, Mar. 1998.
- [94] Wikipedia the Free Encyclopedia, http://en.wikipedia.org/wiki/Main_Page.
- [95] M. Kyan, K. Jarrah, P. Muneesawang, and L. Guan, "Strategies for unsupervised multimedia processing: Self-Organizing Trees and Forests," *IEEE Computational Intelligence Magazine*, Vol. 1, No. 2, pp. 27-40, May 2006.
- [96] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," in *Proc. of ACM SIGGRAPH*, pp. 203-212, ACM Press, New York, NY, USA, Aug. 2001.
- [97] S. Coren, L. M. Ward, J. T. Enns, *Sensation and Perception*, 6th Ed., John Wiley & Sons Inc., USA, 2004.

Appendix A

List of Publications

Journal Publications

- M. Kyan, K. Jarrah, P. Muneesawang, and L. Guan, "Strategies for unsupervised multimedia processing: Self-Organizing Trees and Forests," *IEEE Computational Intelligence Magazine*, Vol. 1, No. 2, pp. 27-40, May 2006.

Conference Publications

- K. Jarrah, S. Krishnan, and L. Guan, "Automatic Content-Based Image Retrieval Using Hierarchical Clustering Algorithms," *International World Congress on Computational Intelligence (WCCI'06)*, pp. 6564-6569, Vancouver, Canada, Jul. 2006.
- K. Jarrah, M. Kyan, S. Krishnan, and L. Guan, "Computational intelligence techniques and their Applications in Content-Based Image Retrieval," *IEEE Int. Conf. on Multimedia & Expo (ICME'06)*, pp. 33-36, Toronto, Canada, Jul. 2006.
- K. Jarrah, M. Kyan, I. Lee, and L. Guan, "Application of image visual characterization and soft feature selection in Content-Based Image Retrieval," in *Proc. of SPIE Multimedia Content Analysis, Management, and Retrieval (SPIE'06)*, Vol. 6073, pp. 101-109, San Jose, California, USA, Jan. 2006.
- K. Jarrah, P. Muneesawang, I. Lee, and L. Guan, "Minimizing Human-Machine Interactions in Automatic Image Retrieval," in *Proc. of IEEE Canadian Conf. on*

Electrical and Computer Engineering (CCECE'04), Vol. 3, pp. 1589-1592, Niagara Falls, Canada, May 2004.