

IMAGE SEGMENTATION AND CONTENT  
BASED IMAGE RETRIEVAL

By

BADRINARAYAN RAGHUNATHAN

Bachelor of Engineering

University of Madras

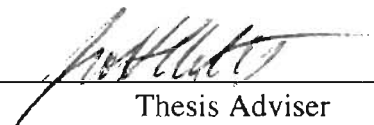
Madras, Tamil Nadu, India

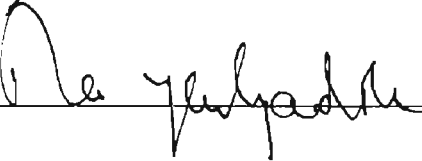
1998

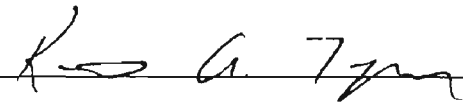
Submitted to the Faculty of the  
Graduate College of the  
Oklahoma State University  
in partial fulfillment of  
the requirements for  
the Degree of  
MASTER OF SCIENCE  
July, 2000

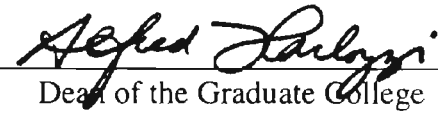
IMAGE SEGMENTATION AND CONTENT  
BASED IMAGE RETRIEVAL

Thesis Approved:

  
\_\_\_\_\_  
Thesis Adviser

  
\_\_\_\_\_

  
\_\_\_\_\_

  
\_\_\_\_\_  
Dean of the Graduate College

## **ACKNOWLEDGEMENTS**

The past two years at graduate school in Oklahoma State University have been very enjoyable and intellectually fulfilling. I have been exposed to the best in terms of coursework and research. For completion of this thesis I owe thanks to a number of people. Firstly I owe an unmeasurable amount of gratitude to my adviser Dr. Scott Acton for being a friend, philosopher and guide in my research, academics and personal life. One quote I will always remember from Dr. Acton is that “It is easier to be mean than it is to be nice”. My colleagues have made my days at the Oklahoma Imaging laboratory fun-filled and professionally-fulfilling. These include the systems guy – Joseph Bosworth, Zhongxiu Hu, David Cary, Junaid Ahmed, Yongjian Yu, Nilanjan Ray and Khaled Al-Mughadhawi. I thank them for their friendship and wish them the best of luck in their future endeavors. I would like to thank Dr. Keith Teague and Dr. Rao Yarlagadda for being on my thesis committee, taking time to review my thesis and suggesting corrections. I would also like to thank Lucent technologies and NASA for having supported me during my graduate study. Ending on a personal note I would like to specially thank my parents and my younger brother for their love, affection and constant encouragement during all these years of my education.

## TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	1
1.1 Overview.....	1
II. LITERATURE SURVEY.....	6
2.1 Background.....	6
2.1.1 QBIC.....	7
2.1.2 PHOTOBOK.....	7
2.1.3 BLOBWORLD.....	9
2.1.4 NETRA.....	9
III. IMAGE SEGMENTATION.....	12
3.1 Introduction.....	12
3.1.1 Background on Image Segmentation.....	12
3.2 Gabor Filter Approach to Texture Image Segmentation.....	15
3.2.1 Overview.....	15
3.2.2 Gabor Filter Design Parameters.....	20
3.3 A Simplified Gabor Decomposition Model.....	26
3.4 Scaled Gabor Decompositions for Texture Image Segmentation.....	29
3.4.1 Background on Image Morphology.....	29
3.4.2 Area Morphology.....	32
3.4.2.1 A Fast Algorithm for Area Morphology.....	34
3.4.2.1.1 Marker Image Creation using the Open Filter...34	
3.4.2.1.2 Reconstruction by Geodesic Dilation.....	35
3.5 Paradigms for Segment Extraction Segmentation.....	37
3.5.1 Fuzzy C-Means Clustering.....	37
3.5.2 Level Set Analysis.....	42
3.5.2.1 Connected Component Combinatorial Analysis.....	46
3.5.2.2 Complexity Analysis.....	54
3.6 Chapter Summary.....	55

<b>Chapter</b>	<b>Page</b>
IV FEATURE EXTRACTION.....	58
4.1 Introduction.....	58
4.2 Color Feature Extraction.....	59
4.2.1 Color Models.....	60
4.2.1.1 The RGB Color Model.....	61
4.2.2 Preprocessing for Color Feature Extraction.....	62
4.2.2.1 Background on Diffusion.....	64
4.2.2.2 The M-CAD Scheme for Color Smoothing.....	67
4.2.3 Coarse Color Quantization for Color Feature Extraction.....	71
4.3 Texture Feature Extraction.....	76
4.4 Shape Feature Extraction.....	78
4.4.1 Boundary Models.....	79
4.4.2 Contour FSDs.....	82
4.4.3 Radius FSDs.....	82
4.4.4 Curvature FSDs.....	83
4.4.5 Affine Invariant FSDs.....	83
4.5 Chapter Summary .....	88
V SIMILARITY ANALYSIS.....	89
5.1 Introduction.....	89
5.2 Color.....	90
5.3 Texture.....	92
5.4 Shape.....	92
5.5 A Logical Hierarchical Framework for Similarity Analysis.....	93
VI APPLICATIONS.....	102
6.1 Introduction.....	102
6.2 CBR for Remotely Sensed Imagery.....	103
6.3 CBR for Automated Inspection of Circuit Boards.....	105
6.3.1 Overview.....	105
6.3.2 Preprocessing and Primary Feature Extraction.....	107
6.3.3 Formulation of Higher Level Features.....	110
6.3.4 Query Formulation and CBR.....	112
6.3.5 Analysis.....	115
VII CONCLUSIONS.....	118
7.1 Summary.....	118
BIBLIOGRAPHY.....	120

## LIST OF TABLES

<b>Table</b>	<b>Page</b>
1. Properties of FDS under various geometric transformations.....	81
2. Intra-board correlation factor behavior for 90 possible chip combinations.....	115

## LIST OF FIGURES

<b>Figure</b>	<b>Page</b>
1. An Image and its ideal segmentation.....	3
2. A general layout of our proposed CBR engine.....	5
3a. Surface plot of magnitude of real part of a Gabor filter in the spatial domain.....	18
3b. Texture extraction using Gabor filters.....	19
4. $\eta$ peak contour of a 2-D Gabor filter in the $uv$ plane.....	21
5. 2-D Gabor filter in 4 evaluated along a line passing through the origin and center frequency.....	22
6a. Frequency domain representation of Gabor filterbank .....	23
6b. Frequency domain representation of Gabor filterbank .....	24
6c. Frequency domain representation of Gabor filterbank .....	25
6d. Frequency domain representation of Gabor filterbank .....	25
7 a. Real part of a 2-D gabor function plotted as a surface.....	27
7 b. Imaginary part of a 2-D Gabor function plotted as a surface.....	27
8a. The original coins image and AOC scaled versions.....	36
8b. The AOC and Gaussian scale-cubes.....	37
9 a. Original image.....	40
9 b. Luminance space of figure 9a.....	40
9 c. Scaled Gabor response of figure 9b.....	40

<b>Figure</b>	<b>Page</b>
9 d. FCM result of figure 9c.....	40
9 e. Overlaid segmentation of figure 9a.....	40
10. Illustrative figure of inability of Area Open-Close to remove at times sub-scale connected components.....	41
11. Concept of Level lines.....	42
12a. Original Zebra image and its Luminance space.....	48
12 b. Gabor response and scaled Gabor response of zebra image.....	48
12 c. LLGG of scaled Gabor response in figure 12 b.....	49
12 d. Morphologically filtered version of figure 12 c.....	49
12 e. Sampled Level sets.....	50
12 f. Segmentation map and overlaid segmentation.....	51
13 a. Original Synthetic texture image and its Gabor response.....	51
13 b. Scaled Gabor response.....	52
13 c. LLGG of scaled Gabor response in figure 13 b.....	52
13 d. Morphologically filtered version of figure 13 c.....	53
13 e. Sampled Level sets from figure 13d.....	53
13 f. Segmentation map and overlaid segmentation.....	54
14. Segmentations of zebra image by FCM clustering and level set analysis.....	56
15 a. Median, WMMR and AOC scaled representations of garden image .....	56
15 b. Clustering results for figure 15 a.....	57
15 c. Superiority of AOC over other scaling techniques.....	57
16. RGB color cube.....	62



<b>Figure</b>	<b>Page</b>
17 a. Original image, M-CAD result, Global coarse color quantization result.....	69
17 b. Original image, M-CAD result, Global coarse color quantization result.....	70
17 c. Original image, M-CAD result, Global coarse color quantization result.....	71
18. Global and local color feature extraction.....	74
19. Global and local color feature extraction.....	75
20. Global and local color feature extraction.....	76
21 a. Original image, highlighted segment, segment contour.....	84
21 b. Contour FSDs of contour in figure 21 a.....	85
21 c. Radius FSDs of contour in figure 21 a.....	86
21 d. Affine invariant FSDs of contour in figure 21 a.....	87
22. Query and shape retrieval results via FSDs.....	94
23. Query and best results by local shape feature matching.....	95
24 a. Query and results by global histogram matching.....	95
24 b. Query and results by global dominant colors matching.....	96
24 c. Query and best results by local dominant colors matching.....	96
25. Comparative analysis of retrieval performance of global color pdf and local color pdf features.....	97
26. Comparative analysis of retrieval performance of global color pdf and local color pdf features.....	98
27 a. Query and results via global color histogram matching.....	99
27 b. Query and retrieval by local color pdf matching.....	99

<b>Figure</b>	<b>Page</b>
27 c. Query and retrieval by hierarchical matching.....	100
28 a. Query and retrieval by global histogram matching.....	100
28 b. Query and retrieval by local color pdf matching.....	101
28 c. Query and retrieval by hierarchical matching.....	101
29. A sample LANDSAT TM band 3 image.....	103
30 a. Query and best results in remote sensing CBR.....	104
30 b. Query and best results in remote sensing CBR.....	105
31. Highlighting the difference between a bad board and a good "reference" board database.....	106
32. Off-line processing.....	107
33. Configuration of system library.....	107
34. A segmentation template with ROI's (chips) numbered.....	108
35. Formulating a Gaussian model of intensity for a chip.....	109
36. Eliminating the effects of background temperature via image statistics (chip model) .....	110
37. Pairwise feature scatterplot with linear regression.....	111
38. FCBs with priority factors for various chips (ROI) .....	112
39. Query board and best match from CBR using primary features.....	113
40. Query board and best match from CBR using higher level features.....	113
41. Query and results with cutoff parameter for CBR by primary features.....	114
42. Query and results with cutoff parameter for CBR by higher level features.....	114

**Figure**

**Page**

43. Standard deviation of inter-board correlation from best to worst matches  
for different query boards.....116

# CHAPTER I

## INTRODUCTION

### 1.1 Overview

The onset of the digital age has seen the advent of the Internet and associated multimedia technologies. The emphasis has been on using these technologies to disseminate information. Consequently we have multimedia databases of images, video, audio and text. Some of these include digital image databases in photography, medicine, engineering, science and the entertainment industry. Due to increases in the number of satellites, available bandwidth, and in commercial applications, these databases have been expanding rapidly.

The preferred method of access to these databases has been via specification of a query and consequent retrieval of results similar to the query in terms of semantics. Access to some of these databases has been inhibited due to their large sizes. Current access technologies perform satisfactorily for text databases and the same has been extended to databases of images and video. As part of this implementation, the various image and video entities in the database are annotated by text keyword descriptors relating to their semantics. These are termed metadata. The retrieval results based on this technology for image and video databases have been far from satisfactory, since plain

text descriptors offer limited encapsulation of entity semantics. Thus, issues like access automation and metadata relevancy limit the effectiveness of this retrieval technology.

In order to remedy these defects of conventional approaches recent research has focussed on content based retrieval (CBR) approaches to manage and utilize large image databases. The emphasis has been on encapsulating the semantics of the database entities in their descriptors. This is accomplished via image processing. As part of this approach various mathematical models for image properties like color and texture have been formulated. The eventual goal is to assess similarity between images by means of their semantic properties arising from image processing parameters like color and texture. Early approaches were based on global models of color and texture. Such image models might not be suitable for CBR. For example an image of a red Ferrari might match up to a red flowerbed in terms of a global color model. Thus these primitive image models do not retain the semantic image information in them.

The challenges lie in the gap between high level semantic information and low level features. Researchers have turned to biological vision systems to tackle this problem. It is a known fact that researchers have for long attempted to emulate biological vision systems given that these are the best known working vision systems. Irving Biederman in 1987 proposed a theory of human image understanding [13]. According to this theory the perceptual recognition of an image involves a process of the segmenting the image at regions of deep concavity into simple geometric components. In image processing parlance this can be construed as extraction of homogeneous regions in an image, i.e. segmentation. These regions might be homogeneous in terms various image properties like color and texture. Figure 1 illustrates this concept of image segmentation.

Thus rather than just annotating an image of a zebra with the text keyword “zebra” we would like to annotate it with zebra (object/segment) specific information like texture pattern and shape information. These features cannot be obtained via a global model since these are segment specific.



**Figure 1.** An image and an ideal segmentation

Clearly segmentation is the key to CBR; accurate segmentations would result in semantically meaningful results. A segmentation algorithm for CBR would need to work on real world images, which are typically rich in color and texture. Hence relatively few researchers have attempted to address this issue of segmenting an image for CBR [47] [48] [34]. However most of these approaches have involved human interaction and various heuristics like edge linking, region merging which we aim to reduce. Some of these segmentation mechanisms achieve a certain degree of automation; however, the quality of their results has yet to be established. Another drawback in these approaches is that they seek a complete partition of the image. Such an approach might result in an over segmentation resulting in semantically needless comparisons for retrieval. Rather what is

required is only extraction of objects of interest. This approach yields lower comparison complexity and faster retrieval performance.

This thesis presents a computational approach to segmentation and consequent feature extraction for CBR. Chapter II briefly reviews the various approaches to image segmentation and outlines a well-motivated approach for CBR image segmentation. A unique area morphological scaling technique is described and two segment extraction paradigms are documented. Thus segments corresponding to regions of homogeneous texture/color are extracted for semantic feature extraction. In chapter III we outline various global and local feature extraction frameworks for CBR. As part of these feature extraction frameworks color, texture and shape features are computed. A hierarchical matching strategy and segment matching strategies are described in chapter IV. Various methods of numerical computation of feature similarity are also given. Experimental results using a database of natural and flower imagery are given. The results using the local features are contrasted with those derived from conventional global features. These retrieval results highlight the impact of segmentation on retrieval quality. The actual segmentation and feature extraction are done off-line during ingestion of images into database while the query specification, retrieval are performed on line. Chapter V discusses issues like end-users of such CBR systems and other CBR applications. Apart from the prototypical image searching application we present real world CBR applications for remote sensing and automated inspection. The various CBR applications are outlined and results demonstrating the successful extension of CBR to real world problems are given. In chapter VI we comment on our experimental results and discuss





## CHAPTER II

### LITERATURE SURVEY

#### 2.1 Background

The nineties have seen the explosive growth of the Internet and consequently multimedia rich databases. As briefly mentioned above, conventional image database management tools have been based on text metadata. In an effort to yield more efficient searches the focus has shifted to content based management of these databases. Consequently CBR has emerged as one of the research frontiers in image processing and the next generation of multimedia content based search engines [17] [34] [42]. Recently CBR tools for diverse applications like automated inspection, database management and web based searches [49] [48] have emerged.

Past CBR engines include QBIC [40], Photobook [42], Virage [7], Blobworld [12], VisualSeek [55], and Netra [34]. The earlier CBR engines provided for querying of images based on various image models mostly global in nature while some used manually/automated extracted regions of interest. A more detailed account of some of these CBR engines is necessary to highlight our objectives and contributions.

### **2.1.1 QBIC (Query By Image Content) [40]**

Query By Image Content (QBIC) from IBM research was one of the first CBR engines in the research community. The various features used by QBIC for image retrieval corresponded to global and local models. QBIC involved object extraction mechanisms, which can be construed as a type of segmentation. These segmentation routines involved heuristics and a significant amount of human interaction. Some of these heuristics involved pixel selections for flood filling, interactive outlining using snakes. QBIC allowed for image searches based on color, texture, shape and metadata information. The texture features were measures of coarseness, contrast and directionality of patterns and were computed over moving windows. The color features essentially were histograms in various color spaces like RGB, YCbCr, CIE - Lab, MTM (Mathematical Transform to Munsell). The shape features involved both heuristics and traditional shape features. The shape features used were from the traditional pattern recognition arena like area, circularity, eccentricity, major axis orientation, compactness and moment invariants. In addition to these quantitative image features the database images were annotated with metadata, more specifically text keyword descriptors. QBIC allowed for different types of queries based on these features. These ranged from query by sketch to user specified weighted combinations of different features.

### **2.1.2 PHOTOBOK [42]**

Photobook from MIT focussed on retrieval by semantic indexing of image/video content. One of the interesting things about Photobook was its emphasis was

on semantics preserving compression. The compression was achieved by representing the images in the database by coefficients representing image features. Some of these features were based on a Karhunen-Loeve transform. These features were used for the twin purposes of CBR and reconstruction. The authors sought to argue that such a large set of feature parameters would allow for more flexible query formulation and better retrieval results. The principle features used were shape, texture and a database specific eigen-faces feature. The authors address the issue of segmentation via identification of model instances, for indexing in the database. In other words extract content (predefined) specific information/features for the purpose of CBR. The mechanisms introduced for this purpose used “clustering in conjunction with affine motion models” and a method of keyframe extraction [20] [58]. The Brodatz database was used for demonstration of texture retrieval, while a face database was used for retrieval of faces. A database of airplanes (single object) was used for similar shape retrieval. The texture features were based on a “Wold” decomposition model [42]. The 2-D Wold-like decomposition can be considered to be “a superposition of three mutually orthogonal components: a purely indeterministic field, a generalized-evanescent field and a harmonic field” [42]. Shape similarity was assessed by means of “the strain energy required to align two shape feature sets” [42]. These feature sets were based on a finite element model [42]. The authors sought to argue that their features are semantics preserving by visual evaluation of the quality of their results.

### **2.1.3 BLOBWORLD [12]**

Blobworld was a product from the Berkeley digital library initiative. For purposes of retrieval images were modeled as a set of elliptical blobs with texture, color and spatial properties. These blobs were derived from a color/texture space segmentation. Segmentation was automatically performed by “iteratively modeling the joint distribution of color and texture with a mixture of Gaussians ” [12]. The color distribution was derived from the *hue-saturation-value (hsv)* space while the texture distribution was modeled as a neighborhood property. One of the interesting things about Blobworld was that the scale parameter for texture description/modeling was selected automatically based on an analysis of the extent of dominant orientations in a neighborhood. The blobs were used for computing spatial properties and refinement of queries, while the color/texture properties were derived from the segmentations. For similarity analysis the user assigned the various feature weights. The authors compared their method to the color indexing method of Swain and Ballard [56] and concluded that their blob representation had superior results because of its content specific representation. The quality of results was assessed qualitatively.

### **2.1.4 NETRA [34]**

Netra was a product of research under the Alexandria digital library initiative [34]. The focus was on a robust segmentation algorithm as the basis for CBR. For the purposes of retrieval, segment specific shape, color and texture features were used. The segmentation was based on a propagation of edge energy flow in terms of texture/color.

Edges were extracted from this process and were linked by a heuristic edge-linking process to generate the segmentations. A heuristic region merging algorithm was incorporated for the purpose of reduction of number of regions. The texture features incorporated were derived from a Gabor filterbank, while the shape features were based on the Fourier shape descriptors. Netra also involved a color quantization scheme, which reduced the color depth. Color features were extracted from this reduced color space. The system was demonstrated on prototype databases of sunsets, flowers and natural scenery. The authors did not provide for evaluation of the results on the grounds of lack of ground truth.

Among other CBR engines were commercial products like the Virage image search engine from Virage corporation [7]. This was a CBR engine based on text metadata and global features. From the above survey it can be seen that some techniques are not suited for a practical application where an image database contains a large number of images. Thus more recent CBR frameworks have focussed on a segment specific feature representation for querying. Some of them suffer from drawbacks due to incorporation of only specific features. A few CBR engines have achieved a certain degree of automation, which is necessary as current image databases contain a large number of images. However they incorporate a level of human interaction and various heuristics like interactive outlining, edge linking and region merging, which we seek to avoid. Most segmentation mechanisms serve to partition the image completely. We differ in this aspect. We believe that when a user specifies a query, the image contains a/an object(s) of specific interest. Hence we strive for a segmentation extracting the most significant objects of interest in an image. This we feel would provide improved retrieval results and

a well-motivated system. Another advantage of this approach is that it reduces the amount of computation to be performed for segment specific similarity analysis.

## CHAPTER III

### IMAGE SEGMENTATION

#### 3.1 Introduction

Higher level image processing tasks like image understanding require the isolation of different objects in an image [13]. This isolation of image objects is termed as image segmentation. It is segmentation that enables an image processing system to organize raw data in a manner such that processing can focus on specific regions and objects in the scene rather than the entire collection of raw data. In this chapter we chiefly outline our segmentation mechanism for CBR. Section 3.1.1 gives a brief review of image segmentation. Section 3.2 outlines the Gabor filter approach to image segmentation while section 3.3 describes a simplified Gabor decomposition model for image segmentation. Section 3.4 elaborates on a scaling methodology for textured image segmentation and section 3.5 documents two paradigms for segment extraction.

##### 3.1.1 Background on Image segmentation

A vast literature exists on segmentation of digital imagery. Segmentation is based on the minimum level of detail acceptable; to be designated an object. The level of detail is tuned by an image processing parameter : *scale*. Classical approaches to image segmentation have been mainly edge-based and region based. Most edge-based

segmentation approaches have attempted to achieve segmentations via creating a scale-space and then detecting edges [36] in this scale-space. Edges are defined to be sudden, sustained changes in average image intensity that extend along a contour, while a scale-space is a set of filtered representations of an image proceeding from fine to coarse. The fine level in a scale-space has a lot of detail while in the coarse representation the detail has been smoothed out. Usually most of these edge-based approaches require a heuristic edge-linking algorithm to link edges and yield closed contours for the purpose of segment extraction. In contrast region based segmentation approaches yield closed contours. Some of these region based segmentation algorithms are multidimensional clustering and region growing [28]. Clustering involves grouping of pixels based on intensity while region growing involves growing of regions in the interior of objects until the boundaries of the regions correspond to the edges of the objects being segmented. It can be seen that region growing segmentation algorithms too involve the concept of scale enunciated above.

Traditionally the segmentation approaches outlined above have been based on gray-scale/color, which do not work well with textured images, particularly those containing macrotextures. Textures whose primitives are large are referred to as macrotextures. The net result is that the detail in the textures is captured as segmentation. Such a segmentation would be an erroneous portrayal of texture perception. Texture is characterized by a spatial distribution of grayscale/color in a given neighborhood. Conventional segmentation algorithms fail when applied to textured images, as they are based on point/pixel definitions. Thus, on the other hand, we have purely texture based approaches to solve this problem. Early texture approaches were based on primitives like image statistics. The most successful among these were those of gray-level co-occurrence



matrices. The primary motivation was to capture the spatial distribution of pixels which researchers at that time felt contributed to the perception of texture. A gray-level co-occurrence matrix  $P(i, j)$  defined for a displacement vector  $\mathbf{d} = (dx, dy)$ , is the number of pixels separated by  $\mathbf{d}$  having gray levels  $i$  and  $j$ . Various features like energy, contrast and homogeneity are defined in terms of algebraic operations on the entities of these gray-level co-occurrence matrices. These features were used for purposes of texture classification and discrimination typically using a linear classifier [30]. Gray-level co-occurrence matrices have performed well in case of microtextures [30]. Textures whose primitives are small are referred as microtextures. However these statistical methods of texture analysis were largely heuristic and seldom performed well for general imagery [32].

A series of advances extending from the late sixties to mid eighties in vision psychophysics and physiology had a profound impact on subsequent texture research in image processing and computer vision. Various researchers characterized cortical cell responses of some mammalian (cat) vision systems as being a function of spatial frequency [18] [29] [14] [19]. More specifically these vision systems were akin to independent linear channels sensitive to narrow ranges of spatial frequencies. Marcelja [35] pointed out the similarity between the cortical receptive field profiles and Gabor's 1-D elementary signal [27]. Daugman [21] extended the 1-D principle of the time-frequency uncertainty principle to 2-D and proved that the 2-D versions of the Gabor's elementary signals achieved the lower bound on this joint spatio-temporal uncertainty principle. Various researchers in the image processing community have successfully used this biological Gabor model of texture description to segment and classify highly oriented

juxtaposed textures like those of the Brodatz image database. The typical approach has been to generate multiple filtered representations of an image via a Gabor filterbank tuned to various texture frequencies and generating a segment map via postprocessing and linear classification of these filtered representations [16].

However the drawback in most of these color/texture approaches has been in their lack of general applicability. Most researchers have in these approaches used specifically tuned parameters to achieve good results for specific images. A majority of these require considerable parameter tuning and heuristics. A segmentation algorithm for CBR would need to be robust, automated and require little or no parameter specification. Needless to say the resultant segments would need to be in reasonable concordance with a perceptual depiction of objects in the image. Thus it is apparent that ready application of the segmentation algorithms in the literature is not appropriate for CBR.

## **3.2 Gabor Filter Approach to Texture Image Segmentation**

### **3.2.1 Overview**

Biological systems pertaining to aural and visual perception are known to use both time and frequency information for purposes of perception. Various researchers have thus motivated joint time-frequency analysis for texture analysis and segmentation. Joint space-frequency analysis techniques aim to overcome the drawbacks of conventional spatial and Fourier analysis techniques. In case of texture analysis the drawbacks of the spatial texture analysis techniques have been highlighted earlier.

Conventional Fourier analysis techniques are not suited for local analysis as they provide a global decomposition. Ideally these Fourier based techniques would be suitable for single frequency component (monochromatic) signals. However common test signals/images consist of multiple components, which renders interpretation of Fourier domain representations limited in nature. Thus Fourier based techniques are not jointly localized in the space and frequency domains.

A standard measure of localization of a 2-D function is the product of the bandwidths in spatial and frequency domains, which represents the joint resolution in space and frequency domains. The effective bandwidth of a function is given by the square root of the variance of its energy distribution. For a 2-D function  $f(x, y)$  and its Fourier transform  $f(u, v)$ , the effective bandwidths in the spatial and frequency domains are given by

$$(\Delta x)^2 = \frac{\iint x^2 f(x, y) f^*(x, y) dx dy}{\iint f(x, y) f^*(x, y) dx dy} \quad (1)$$

$$(\Delta y)^2 = \frac{\iint y^2 f(x, y) f^*(x, y) dx dy}{\iint f(x, y) f^*(x, y) dx dy} \quad (2)$$

and

$$(\Delta u)^2 = \frac{\iint u^2 F(u, v) F^*(u, v) du dv}{\iint F(u, v) F^*(u, v) du dv} \quad (3)$$

$$(\Delta v)^2 = \frac{\iint v^2 F(u, v) F^*(u, v) du dv}{\iint F(u, v) F^*(u, v) du dv} \quad (4)$$

The Heisenberg uncertainty principle imposes a lower bound on the joint resolution in spatial and frequency domains i.e.

$$(\Delta t \Delta f) \geq (\frac{1}{2} \pi) \quad (5-1)$$

$$(\Delta x \Delta u)(\Delta y \Delta v) \geq (\frac{1}{4} \pi^2) \quad (5-2)$$

where  $\Delta t, \Delta f$  are function (1-D) bandwidths in the time and frequency domains while  $\Delta x, \Delta y, \Delta u, \Delta v$  are function (2-D) bandwidths in the spatial and frequency domains.

In 1946 D. Gabor in his classic monograph [27] showed that a class of functions now termed as Gabor functions achieves the maximum possible joint resolution in spatial and frequency domains. These Gabor functions or Gabor elementary functions were defined to be a Gaussian function modulated by a sinusoid. Daugman [21] showed that the 2-D versions of the Gabor functions achieved the lower bound on the joint spatio-temporal uncertainty principle.

Gabor functions are Gaussian modulated sinusoids in the space domain and hence shifted Gaussians in the frequency domain. The general form of a 2-D Gabor function  $h(x, y)$  is given by

$$\begin{aligned} h(x, y) &= g(x, y) \exp[2\pi j(Ux + Vy)], \\ g(x, y) &= \left( \frac{1}{2\pi\sigma^2} \right) \exp\left[ -\frac{x^2 + y^2}{2\sigma^2} \right] \end{aligned} \quad (6)$$

where  $g(x, y)$  is a 2-D Gaussian function with scale parameter  $\sigma$ . The pair  $(U, V)$  gives the center frequencies of the 2-D Gabor filter. The 2-D frequency response of the Gaussian  $g(x, y)$  is given by

$$g(u, v) = \exp[-2\pi^2\sigma^2(u^2 + v^2)] \quad (7)$$

The frequency response of the 2-D Gabor function is just a shifted version of the above equation. It is given by

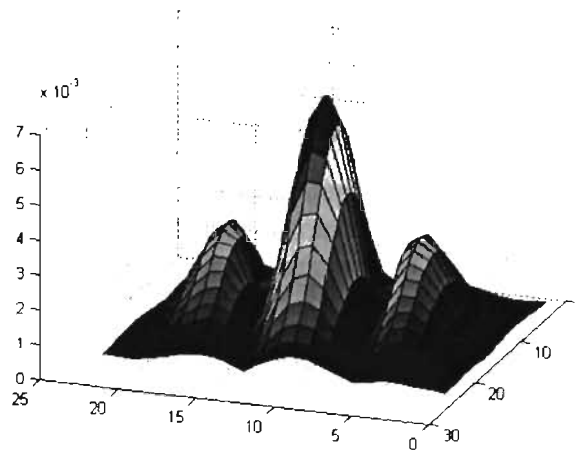
$$h(u, v) = \exp\left[-2\pi^2\sigma^2\left((u-U)^2 + (v-V)^2\right)\right] \quad (8)$$

The center frequencies  $U$ ,  $V$  can be represented in polar form by a radial center frequency  $F$ , an orientation  $\Theta$

$$F = \sqrt{U^2 + V^2} \text{ Cycles/pixel or cycles/sample,}$$

$$\Theta = \text{Tan}^{-1}\left(\frac{V}{U}\right) \text{ degrees or radians measured w.r.t. the } u \text{ axis} \quad (9)$$

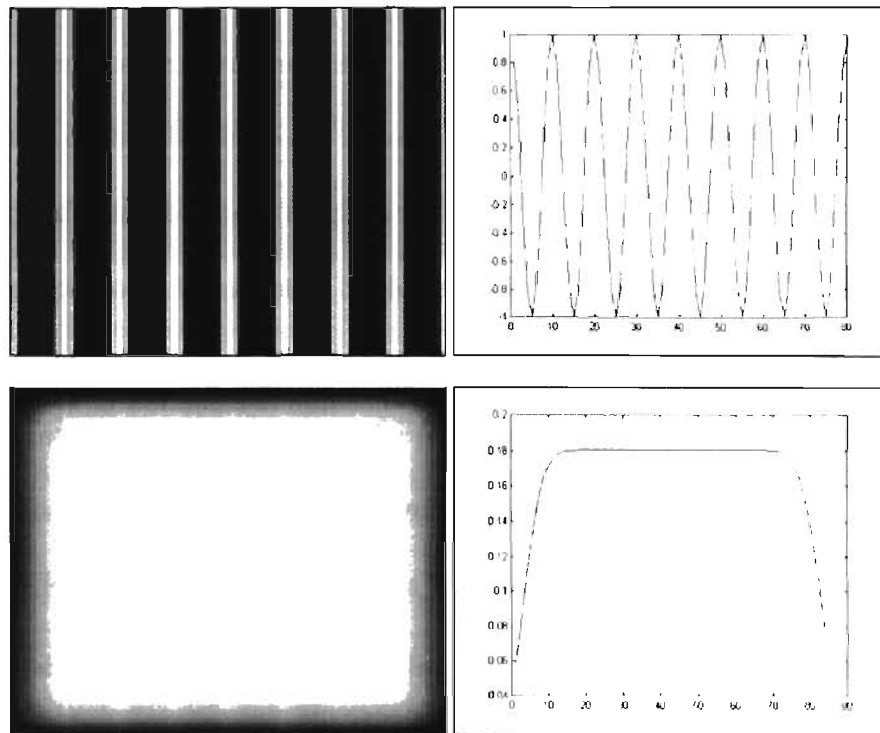
Since 2-D Gabor function achieves the upper bound of the 2-D space-frequency uncertainty principle, it optimizes the balance between localization in the spatial and frequency domains and achieves maximum possible joint resolution. Figure 3a shows the magnitude of the real part of a Gabor filter in the spatial domain.



**Figure 3 a.** Surface plot of magnitude of real part of a Gabor filter in the spatial domain

If an image were to be filtered by a Gabor function, the complex magnitude of the filtered image will be maximized over regions with texture oriented at

the center frequencies of the Gabor filter. This behavior of the Gabor filter is illustrated in figure 3 b. A particular texture can be extracted from an image by applying a bank of Gabor filters tuned to the dominant frequencies in that texture. Thanks to human visual system's natural high frequency roll-off response, a bank of Gabor filters tuned to a few



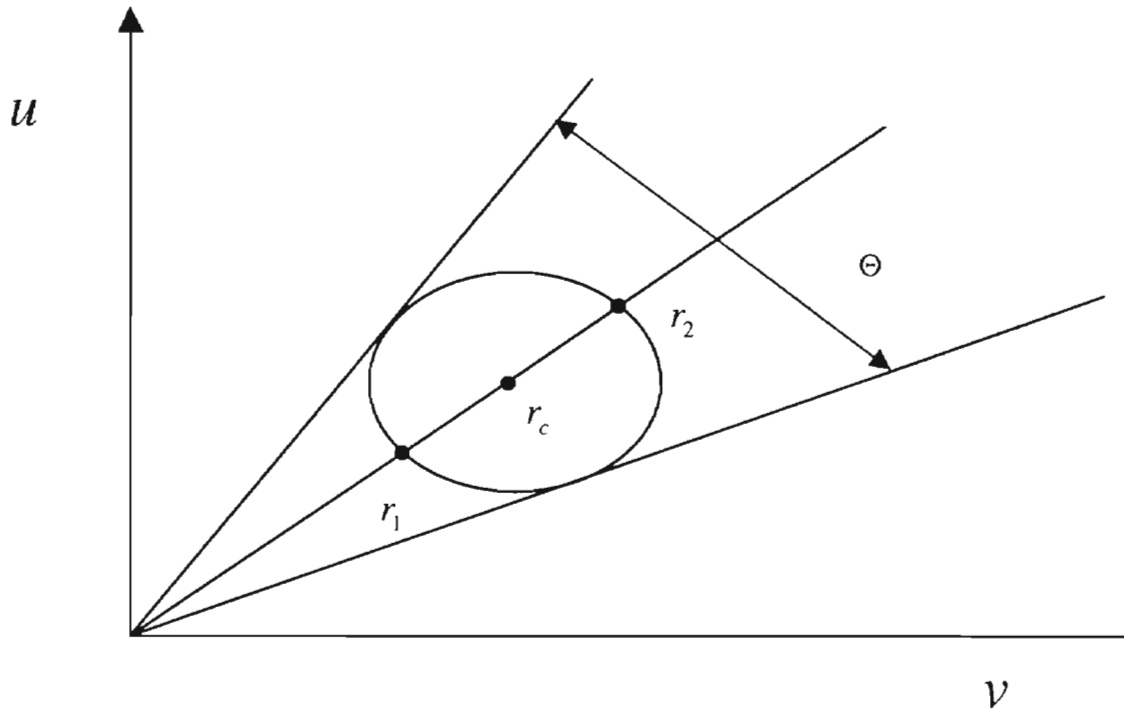
**Figure 3 b.** Row one – Periodic texture image (cosine in horizontal direction 0.1 cycles/pixel), 1-D slice of periodic texture image. Row two – Periodic texture image filtered with Gabor filter ( $U=0.1$  Cycles/pixel,  $V=0$  Cycles/pixel), 1-D slice of Gabor filtered image.

low/intermediate frequencies is usually enough to capture effectively the texture layout in an image for segmentation purposes.

### 3.2.2 Gabor Filter Design Parameters

By varying the center frequency, orientation and the scale parameter of the Gabor filter, different image features can be localized in the spatial and frequency domains. The center frequency and consequently the orientation of the Gabor filter determines the pattern of texture the filter is tuned to, while the scale parameter determines the extent of the filter in space. A large scale parameter implies that the output of the filter will be affected by more neighboring pixels, while a small scale parameter implies that the output will be affected by fewer neighboring pixels. Thus the scale parameter intuitively corresponds to the size of the texture primitives. Hence the Gabor design parameters intuitively correspond to the frequency of the texture, the size of the texture primitives and the orientations of the texture primitives.

Consider the  $\eta$  peak contour of a 2-D Gabor filter in the  $uv$  plane shown in figure 4. The  $\eta$  peak contour corresponds to the portion of the Gabor filter, which is  $\eta$  times its peak response. The axes correspond to the  $u$  and  $v$  frequency axis respectively while  $r_c$  corresponds to the radial center frequency of the Gabor filter. The filter magnitude response is at a fraction  $\eta$  of the peak response at the radial frequencies  $r_1$  and  $r_2$ . The orientation bandwidth  $\Theta$  is the angle between the two tangents to the  $\eta$  peak contour, pass through the frequency origin. Figure 5 shows the filter evaluated on a line from the frequency origin through its center frequency  $r_c$ .



**Figure 4.**  $\eta$  peak contour of a 2-D Gabor filter in the  $uv$  plane

The  $\eta$  peak radial octave bandwidth of a Gabor filter is defined by

$$B = \log_2 \left( \frac{r_2}{r_1} \right) \quad (10)$$

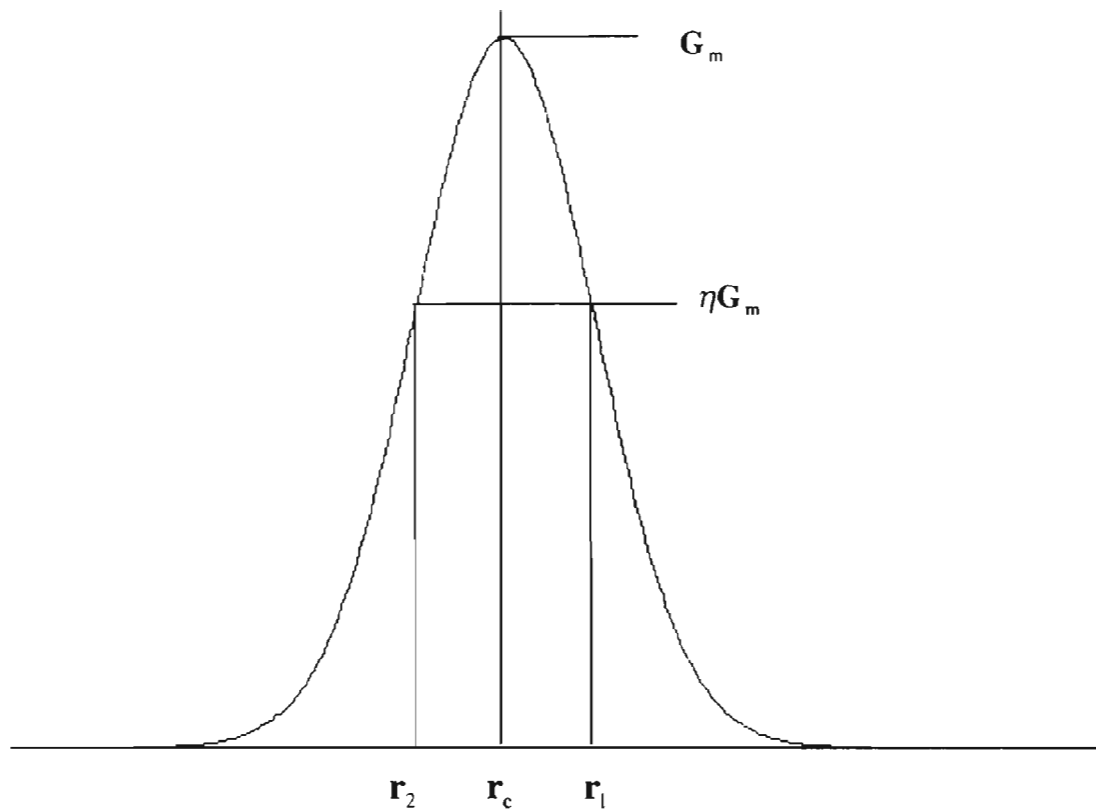
Evaluating the 1-D signal in figure 5 for  $r_2$  and  $r_1$  we have,

$$e^{(-2\pi^2\sigma^2(r_2-r_m)^2)} = \eta, \quad (11)$$

$$r_2 = r_m + \frac{\sqrt{-\ln \eta}}{\sqrt{2\pi\sigma}},$$

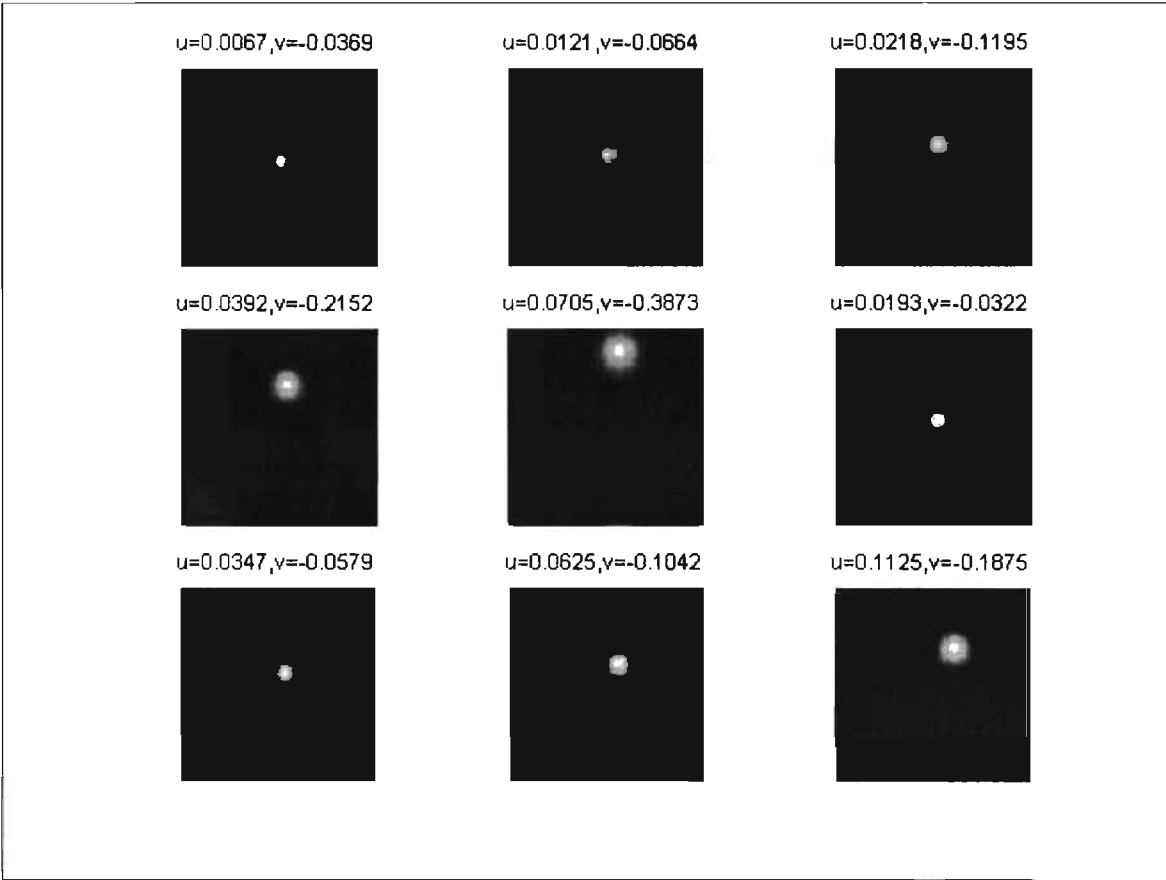
$$r_1 = r_m - \frac{\sqrt{-\ln \eta}}{\sqrt{2\pi\sigma}}. \quad (12)$$



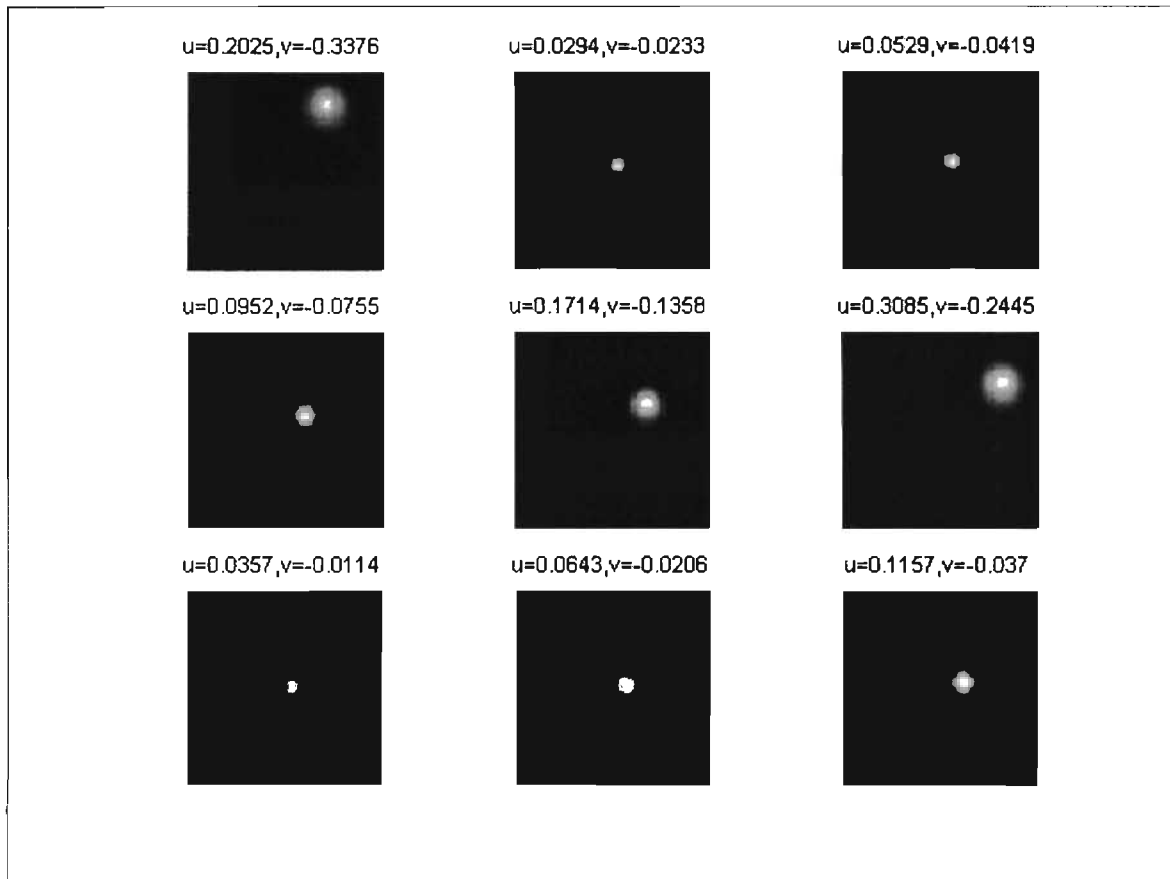


**Figure 5.** 2-D Gabor filter in figure 4 evaluated along a line passing through the origin and center frequency

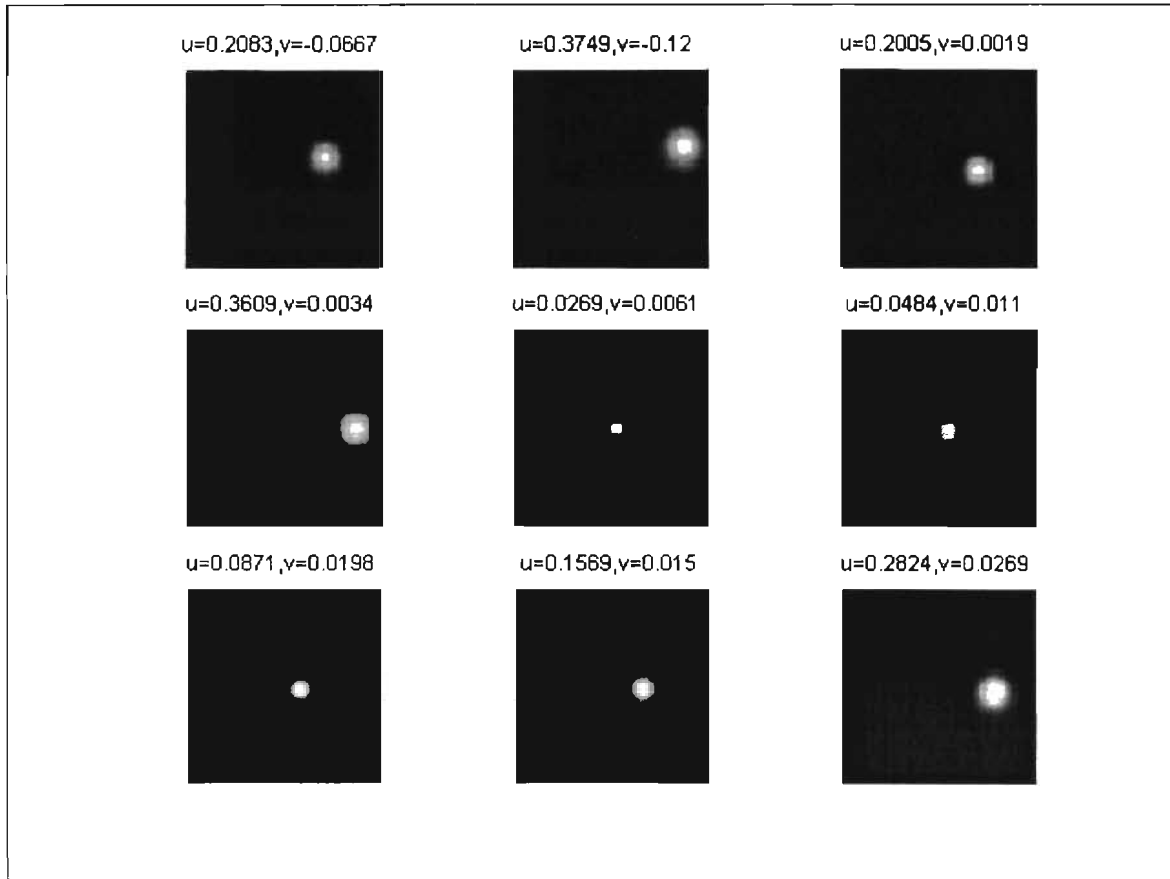
The  $\eta$  peak radial octave bandwidth of the Gabor filter is specified in our framework as a design parameter. In our experiments we have usually used design parameters of  $B = 1$  octave and  $\eta = 0.5$ . We use a Gabor filterbank similar to the one documented in [31]. Figures 6 a,b,c,d illustrate this particular Gabor filterbank.



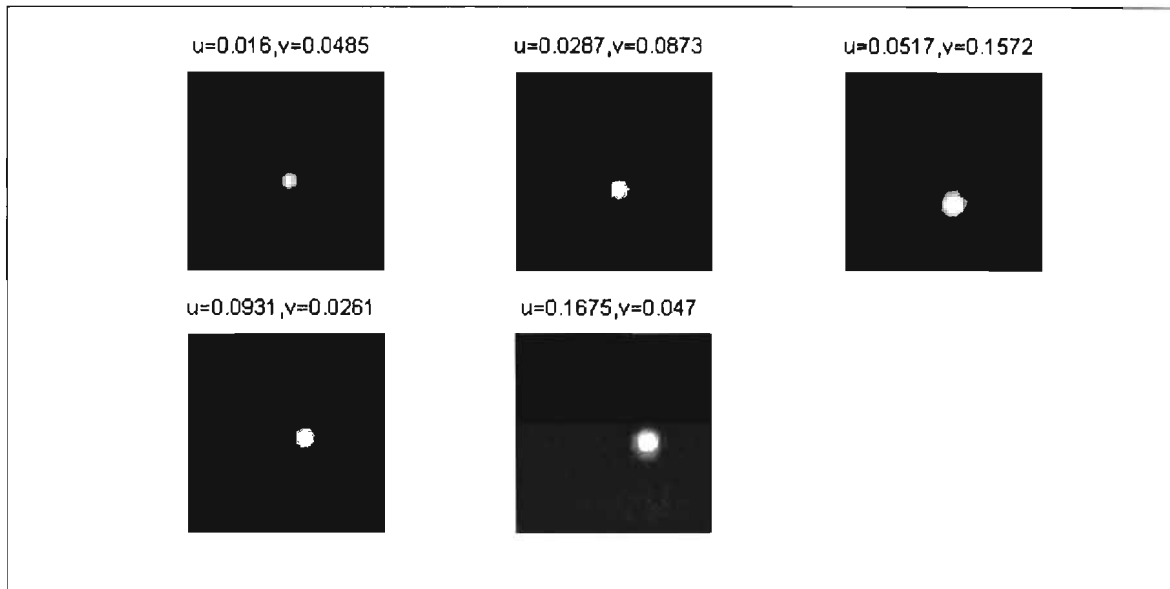
**Figure 6 a.** Frequency domain representation of Gabor filterbank in [39]



**Figure 6 b.** Frequency domain representation of Gabor filterbank in [39]



**Figure 6 c.** Frequency domain representation of Gabor filterbank in [39]



**Figure 6 d.** Frequency domain representation of Gabor filterbank in [39]

The 2-D Gabor filterbank approach provides a versatile model for texture description and optimizes the balance between localization in the spatial and frequency domains. Moreover, 2-D Gabor filters are known to mimic the biological perception of texture. As part of this approach images are decomposed into multiple oriented spatial frequency channels.

### 3.3 A Simplified Gabor Decomposition Model

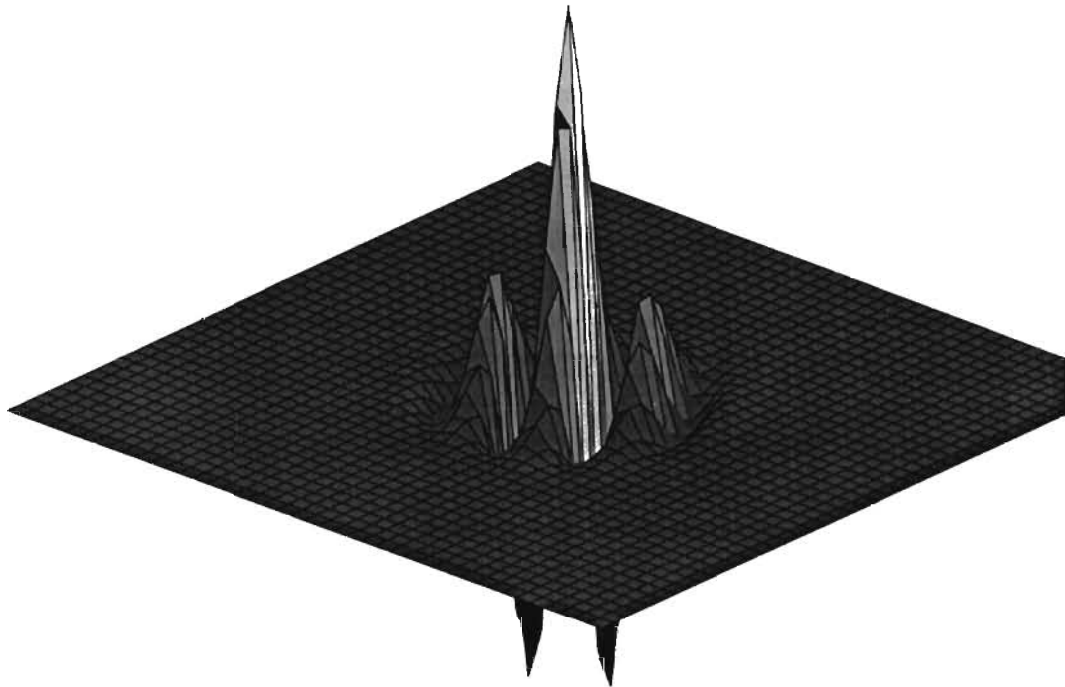
The real part of the Gabor function (6) is given by

$$h_r(x, y) = g(x, y) \cos(Ux + Vy) \quad (13)$$

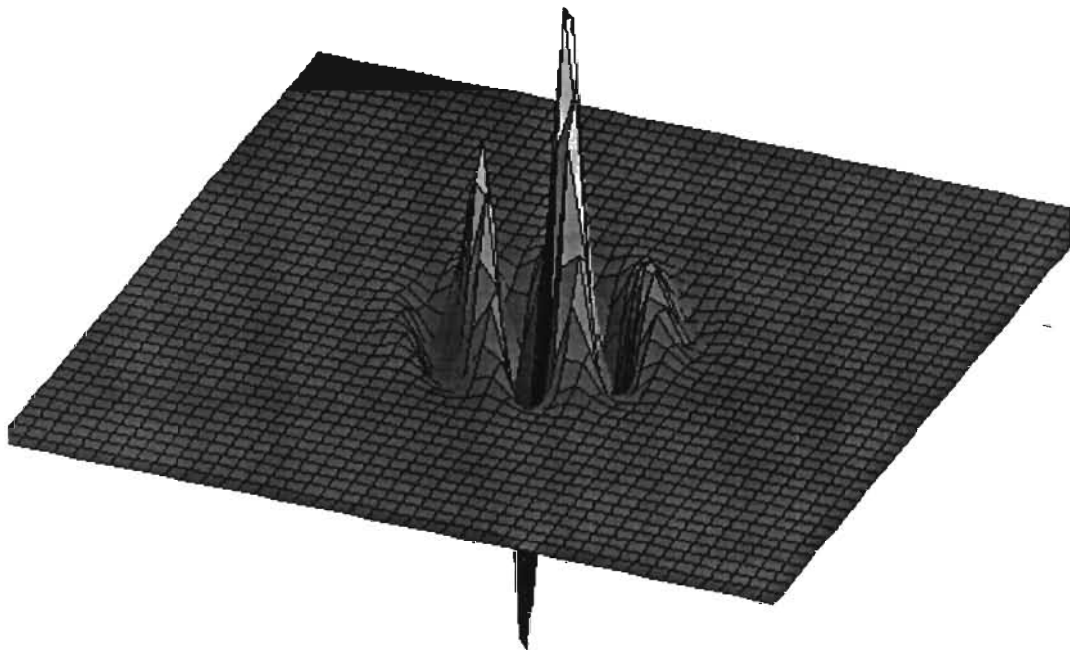
while the imaginary part of is given by

$$h_i(x, y) = g(x, y) \sin(Ux + Vy) \quad (14)$$

Equations 13 and 14 are out of phase w.r.t. each other by  $\frac{\pi}{2}$  along the direction of  $\Theta$ . Figures 7 a,b show equations 13 and 14 as a surface. In [16] the real and imaginary parts of the Gabor responses have been used to distinguish boundaries between phase shifted regions within a texture, illusory contours. However such instances are more prevalent among synthetic superpositions of highly oriented textures like the Brodatz database [17] and synthetic textures. Hence we argue that the usage of the magnitude of the Gabor response suffices for our goal of extracting objects of the most significant interest from the image (rather than partitioning the image completely). Moreover our target image databases consist of general imagery in which instances of highly oriented textures are extremely rare.



**Figure. 7 a.** Real part of a 2-D Gabor function plotted as a surface  $U=0.2$  cycles/pixel,  $V=0.1$  cycles/pixel



**Figure 7 b.** Imaginary part of a 2-D Gabor function plotted as a surface  $U=0.2$  cycles/pixel,  $V=0.1$  cycles/pixel

A Gabor expansion of any  $n$ -dimensional function is a weighted sum of appropriately shifted Gabor functions [27]. The weights are usually signal-specific describing the weight of each Gabor response.

$$I(x, y) = \sum_{m,n,U,V} C_{mnUV} I_h(x, y), \quad (15-1)$$

$$I_h(x, y) = I(x, y) * h(x - x_m, y - y_n) \quad (15-2)$$

where  $I(x, y)$  is the original image and  $I_h(x, y)$  is its Gabor response. The expansion coefficients are computed in a 4-D grid whose dimensions correspond to  $u, v, x$  and  $y$ . A simple aggregation of various Gabor responses would be an erroneous approximation as Gabor functions are fundamentally non-orthogonal. Two functions  $f(x)$  and  $g(x)$  are said

to be orthogonal on the interval  $a \leq x \leq b$  if  $\int_a^b f(x)g(x)dx = 0$ .

Basstians [10] developed an analytical technique for determining the coefficients in a Gabor expansion. Zeevi and Porat extended this concept to 2-D [60]. The expansion coefficients  $C_{mnUV}$  for all possible (discrete) center frequencies, orientations and grid shifts are sufficient for complete reconstruction of the image  $I(x,y)$ . However we are interested in extracting segments of homogeneous texture rather than a faithful reconstruction of the image. Hence we do not require the expansion coefficients for a Gabor expansion. Our goal is to extract segments of homogeneous texture directly via the Gabor responses. We term this approach a simplified Gabor decomposition for analysis and extraction of segments of homogeneous texture i.e.

$$S_i(x, y) \approx \bigcup_{m,n,U,V} \{I_h(x, y)\}. \quad (16)$$

Here  $S_i(x, y)$  constitutes the various Gabor filter responses of the image  $I(x, y)$ . The simplified Gabor decomposition contains regions of homogeneous texture as locally maximized regions. It is our goal to extract these regions and classify them as representative of the textured objects in the image. Thus we motivate a scaling methodology and paradigms for extraction for these segments.

### **3.4 Scaled Gabor Decompositions For Textured Image Segmentation**

Segmentation results based solely on the Gabor decomposition may contain insignificant or spurious regions in terms of area, which lead to errors and increased computational burden in CBR. Our emphasis within the context of CBR has been on matching large scale regions that are similar in terms of texture. The solution presented here overcomes the difficulties with small scale regions by utilizing scaled versions of the Gabor decompositions. Hence we argue that extraction of segments satisfying a minimum scale requirement results in semantic extraction of objects of interest and reduced computational complexity for similarity analysis. Hence we use scaled Gabor decompositions for segment extraction.

#### **3.4.1 Background on Image Morphology**

Scaling mechanisms have been used in image processing to remove detail and to reduce the information content of an image, while retaining the essential features [15].



The choice of the scaling technique depends on the application. The main requirement we lay down for scaling in CBR is strong edge localization w.r.t. the Gabor filter responses.

We use an area morphological technique for scaling purposes. True to its terminology it does involve the “study of shape”; however it differs considerably from traditional image morphology theory. The area operators affect the image by removing connected components within the image level sets that do not satisfy the minimum area criterion. Through the area morphological approach an image can be decomposed into a number of thresholded binary representations called level sets. This can be likened to conventional morphology with a liquid structuring element of fixed area; however operating on image level sets. Thus sub-scale objects can be removed from an image using a truly scalable nonlinear filter. Various researchers have shown in the literature the attractive properties of the area morphological filters [1] [50]. These include feature causality, Euclidean invariance, and edge preservation through scale.

Area morphology has been used in the literature for purposes of image restoration [37], segmentation [47] and document processing [39]. The drawback in normal implementations of area morphology is the large computation time given the sequential nature of the level set and connected component analysis. This has precluded it from being more widely used. Recently there has been an advent of fast area morphological algorithms, which allow for their inclusion in multimedia frameworks for content based retrieval [47] and object based coding [5]. The following section is a brief review of morphological filters and fast algorithms for area morphology.

The basis for morphological operations are the erode and dilate functions. The erode operator is given by

$$J = Erode[I, B] = I \ominus B,$$

$$J(i, j) = Min\{B \circ I(i, j)\} \quad (17)$$

while the dilate operator is given by

$$J = Dilate[I, B] = I \oplus B,$$

$$J(i, j) = Max\{B \circ I(i, j)\} \quad (18)$$

where the operator  $B$  is referred to as the structuring element,  $I$  is the image and  $\oplus$  and  $\ominus$  are the dilation and erosion operators respectively. Thus dilation and erosion are windowed maxima and minima respectively. The standard morphological operators open and close are derived from the erode and dilate functions. The open operator is a concatenation of the erode and dilate operators in that order. It can be expressed as

$$J = OPEN[I, B],$$

$$= I \circ B = (I \ominus B) \oplus B \quad (19)$$

while the close operator is a concatenation of the dilate and erode operators in that order and can be expressed as

$$J = CLOSE[I, B],$$

$$= I \bullet B = (I \oplus B) \ominus B \quad (20)$$

where  $\circ$  and  $\bullet$  are the open and close operators respectively.

These standard morphological filters are relatively inexpensive to implement considering that they decompose into erode and dilate operators. Open filtering can be modeled as fitting the structuring element under the topological image surface while the process of fitting the structuring element from above the topological image surface can be modeled as close filtering. However the application of these morphological operators does not guarantee removal of sub-scale objects as open removes bright objects that

cannot be fit entirely by the structuring element. Hence open cannot remove certain sub-scale bright objects. Hence the inherent grayscale bias of these morphological operators is an additional drawback of morphological scaling. [54] [41] [8] give an exhaustive overview and description of standard morphology and scale-space theory.

### 3.4.2 Area Morphology

The area open-close operator in contrast to standard morphological operations does not involve a structuring element. As briefly described above it guarantees a minimum scale by removing sub-scale connected components within level sets. For an image  $I$  with discrete domain  $\mathbf{D} \subset \mathbf{Z}^2$  and image location  $p \in \mathbf{D}$ , a level set  $\mathbf{B}_l$  at level  $l$ , (where  $l \in \mathbf{Z}$ ) is defined by

$$\begin{aligned} \mathbf{B}_l(p) &= 1 \text{ if } I(p) \geq l, \\ &= 0 \text{ otherwise} \end{aligned} \quad (21)$$

Within a level set  $\mathbf{B}_l$ , the connected component  $\mathbf{C}_{\mathbf{B}_l}$  at  $p$  is given by:

$$\mathbf{C}_{\mathbf{B}_l}(p) = \{q : \exists P_{l \geq l}(p, q)\}, \quad (22)$$

where  $P_{l \geq l}(p, q)$  is an unbroken path between image locations  $p$  and  $q$  for which each element obeys  $\mathbf{B}_l(\cdot) = 1$  (satisfying the level set equation). The neighboring pixels in this path are defined by 4-connectivity in our case. For a level set  $\mathbf{B}_l$ , we can define the area open operation  $\mathbf{B}_l \circ (s)$  by

$$(x, y) \in \mathbf{B}_l \circ (s) \text{ if } |\mathbf{C}_{\mathbf{B}_l}(x, y)| \geq s, \quad (23)$$

where  $|C_{B_I}(x, y)|$  is the cardinality (area) of the connected component at location  $(x, y)$ , and  $s$  is the minimum area. On the other hand, the area open implies that  $(x, y) \notin \overset{\circ}{\bullet} (B_I)$  if  $|C_{B_I}((x, y))| < s$ . Similarly we can define the area close operation  $B_I \bullet (s)$  by

$$(x, y) \in \overset{\circ}{\bullet} (B_I) \text{ if } |C_{S_I}^{\sim}(x, y)| \geq s, \quad (24)$$

where  $|C_{S_I}^{\sim}(x, y)|$  is the cardinality (area) of the connected component at location  $(x, y)$ ,  $s$  is the minimum area and it may be noted that this is defined on the complement of the level set  $C_{B_I}$ . Similarly the area close implies that  $(x, y) \notin \bullet (B_I)$  if  $|C_{B_I}((x, y))| < s$ . Hence the area open operator removes bright sub-scale objects while area close removed dark sub-scale objects within the level sets of the image. For grayscale imagery, each level set is area open-closed independently, and the final grayscale result is computed by a stacking operation. The reconstructed area open-close image at scale  $s$  is thus given by

$$(I \circ (s)) \bullet (s) = \sum_{l=0}^{L-1} \{(B_l \circ (s)) \bullet (s)\} \quad (25)$$

This image is seen to contain only connected components of a minimum size and above within each of its level sets. [50] elaborates on the various other interesting properties of AOC. These include feature causality with scale and more importantly the level lines (edges of connected components within the level sets) are not distorted with increased scale. This bodes well in terms of edge localization. The significance of this property for our CBR will be shown shortly.

### 3.4.2.1 A Fast Algorithm for Area Morphology

Traditionally these AOC operations have been cumbersome and time-consuming due to connected component labeling at each level set. The drawback of computational cost has been overcome by means of fast algorithms for this area morphological process. Given a fast algorithm for the area open algorithm, we can produce an area close result by employing the Boolean complement of the input level sets used in the area open operation. The fast algorithms presented in the literature are of two main types (1) Pyramidal [57] and (2) Marker image creation and reconstruction [2]. We use the second method in view of no restrictions on minimum scale parameter and its minimal overall complexity. Essentially this involves the creation of a *marker* image  $M$  by opening the original image. Then connected components within the level sets that partially survive the open operation are fully reconstructed w.r.t. the original image. An area opening by reconstruction is not considered to be equal to an ideal area opening. This is because some super-scale connected components may not survive the area opening thanks to the open filter. For example a single pixel width region of area 30 pixels is removed by opening with a 3x3 square structuring element, although it exceeds the minimum scale. Hence these fast algorithms are regarded as approximate algorithms. However the strong point of these fast algorithms is that they vastly improve upon computational cost required [2].

#### 3.4.2.1.1. Marker image creation using the open filter.

The marker image is created via the traditional open filter (equation 17),  $I \circ B$ . The structuring element corresponds to the minimum scale

requirements. This would ensure that connected components of insufficient area do not survive the filtering process. Again as mentioned above some super-scale objects may not survive this process. Some solutions in the literature have used a combination of opening with all possible shapes of structuring elements in order to match the actual area open operation, while some have used a pyramidal basis for creation of the marker image. However we used the traditional open filter for marker image creation due to its optimum overall performance [2].

#### 3.4.2.1.2. Reconstruction by geodesic dilation.

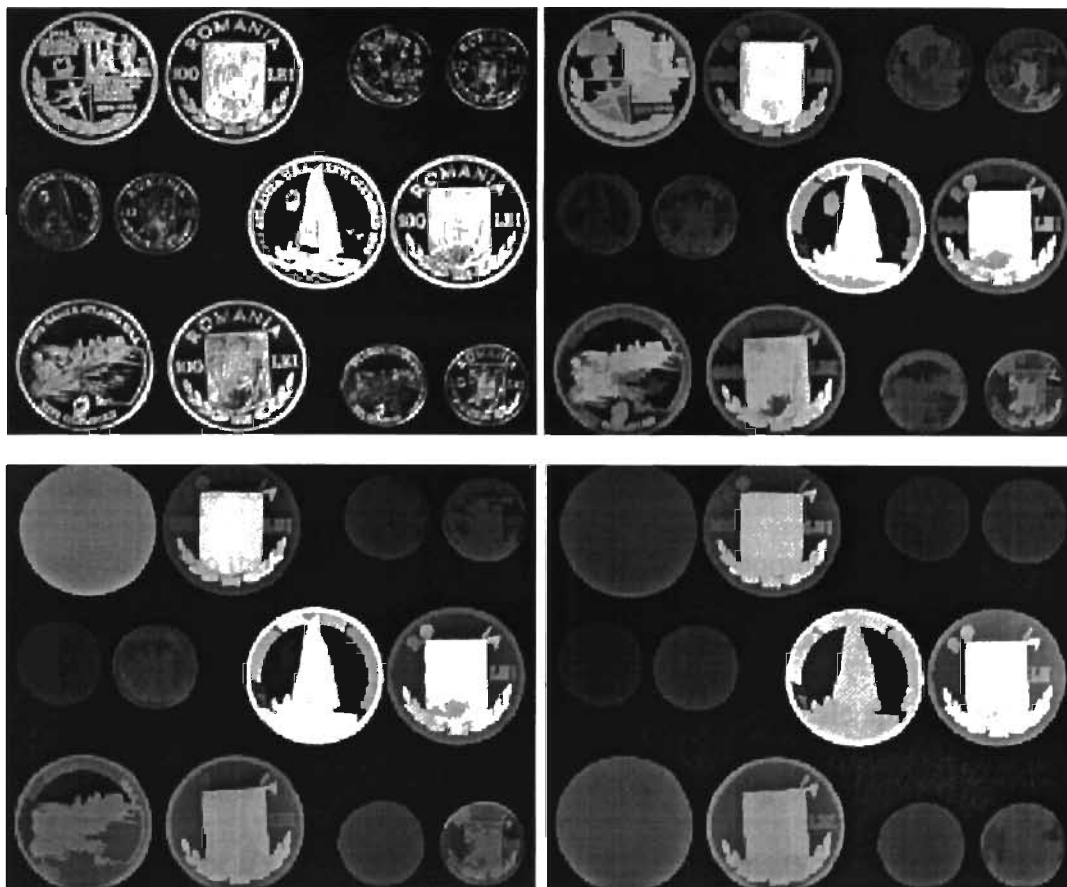
Reconstruction of entire connected components within level sets is based on the partial connected components (within the level sets) and pixel intensities in the input image  $I$ . These partial connected components are obtained from the marker image  $M$ . The reconstruction is done by selectively dilating these components one pixel at a time. If a connected component in the marker image has a 4-connected neighboring pixel that has lower intensity than that of the input image, then that particular pixel is updated to the minimum intensity between the dilated image and the intensity in the input image. This can be expressed as :

$$R_i(p) = \min\left[\left(\mathbf{R}_{i-1} \oplus \mathbf{K}^+\right)(p), I(p)\right] \quad (26)$$

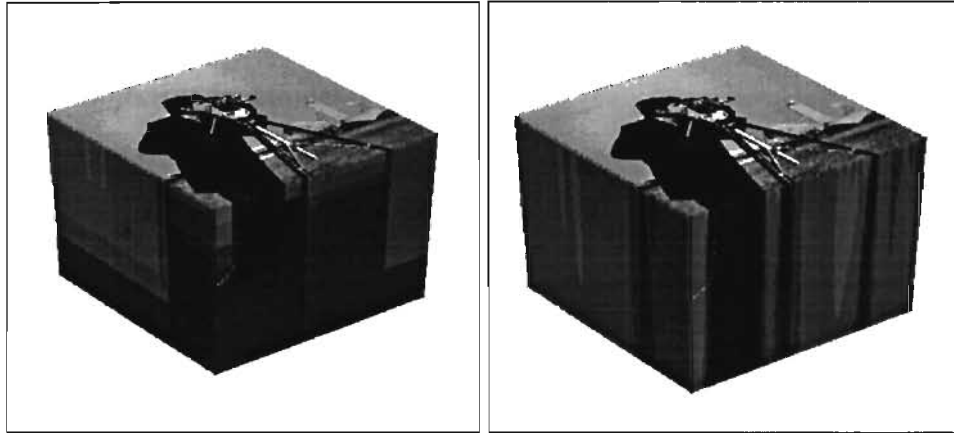
where  $p \in D$ ,  $R_0 = M$ , and  $\mathbf{K}^+$  is a  $3 \times 3$  circular structuring element with the origin at the center. The update in the above equation will stabilize when each of the marked connected components is reconstructed, requiring  $T$  total updates per pixel. The value  $T$  is bound by the maximum geodesic distance between a boundary pixel of a connected component in  $\mathbf{M}$  and the boundary in the reconstructed connected component in  $\mathbf{R}$ . Figure 8 a. illustrates area morphological scaling with various minimum area/scale

parameters. Figure 8 b shows the scale-cubes for AOC scaling and conventional Gaussian filtering/scaling. A scale-cube is similar to a scale-space. It can be seen that the AOC scale-cube exhibits less or no edge/feature drift through scale while this is not the case with the linear Gaussian scale-space.

The minimum scale  $s$  (area parameter) is specified for each of the images in our database by means of conventional metadata. Each of the Gabor filter responses are scaled by means of the area morphological scaling technique described above. These scaled Gabor decompositions (responses) are utilized for segment extraction, which is described in the next section.



**Figure 8 a.** The original coins image and AOC scaled versions (area 5x5, 10x10, 15x15)



**Figure 8 b.** The AOC and Gaussian Scale-cube

### 3.5 Paradigms For Segment Extraction

#### 3.5.1 Fuzzy C-Means Clustering

The scaled Gabor decompositions contain homogeneous regions signifying regions of homogeneous texture. We utilize the Fuzzy C-Means (FCM) clustering technique to extract segments from the scaled Gabor decompositions. The FCM technique operates by grouping together pixels in a multidimensional space based on their distance with respect to their pixel intensities. The number of dimensions in this space is equal to the number of classes specified for the purpose of classification.

For each position  $(x, y)$  in the input image, each scale  $s$  and each texture layer  $t$ , we have intensities  $I(x, y, s, t)$  in the 4-D texture / scale space. These intensities are used to cluster vectors for each position  $(x, y)$  using the FCM algorithm. We can consider a vector  $\mathbf{I}(x, y)$  as the evolution of the pixel intensity at  $(x, y)$  through scale  $s$  and texture  $t$ . The fuzzy clustering technique is based on minimizing an objective functional



that quantifies the distance between cluster centers and the data within the various clusters. This objective functional is given by

$$J_m(U, \mu) = \sum_{\Omega} \sum_{i=1}^C (u_i(x, y))^m \|d_i(x, y)\|^2 \quad (27)$$

Here,  $U$  is the fuzzy  $C$  – class partition of the texture/scale- space, where  $C$  is the number of classes.  $\mu$  is the set of cluster centers, and  $\Omega$  is the domain over which the clustering is done, i.e.  $(x, y) \in \Omega$ . Given a texture / scale space vector  $\mathbf{I}(x, y)$  at location  $(x, y)$ , the measure

$$\|d_i(x, y)\| = \|\mathbf{I}(x, y) - \mu_i\| \quad (28)$$

is the distance between the texture / scale space vector and the  $i^{\text{th}}$  cluster center  $\mu_i$ . The distance is weighed by the fuzzy membership value (of each texture / scale space vector)  $u_i(x, y)$  corresponding to  $i^{\text{th}}$  class. The fuzzy exponent  $m$  has the range  $m \in [1, \infty]$ . The objective functional is iteratively minimized. This iteration is subject to the conditions

$$\sum_{\Omega} \sum_{i=1}^C u_i(x, y) = 1,$$

$$0 < \sum_{\Omega} u_i(x, y) < |\Omega|,$$

$$\text{and } u_i(x, y) \geq 0. \quad (29)$$

At each iteration the fuzzy membership value for each texture / scale-space vector  $\mathbf{I}(x, y)$  is computed by

$$u_i(x, y) = 1 / \left[ \sum_{\Omega} \sum_{j=1}^C \left( \frac{d_i(x, y)}{d_j(x, y)} \right)^{2/(m-1)} \right] \quad (30)$$

Initially this membership value is computed using a uniformly-distributed random number generator. At each iteration, the cluster center  $\mu_j$  is updated according to

$$\mu_j = \frac{\sum (u_i(x, y))^m \mathbf{I}(x, y)}{\sum_{\Omega} (u_i(x, y))^m} \quad (31)$$

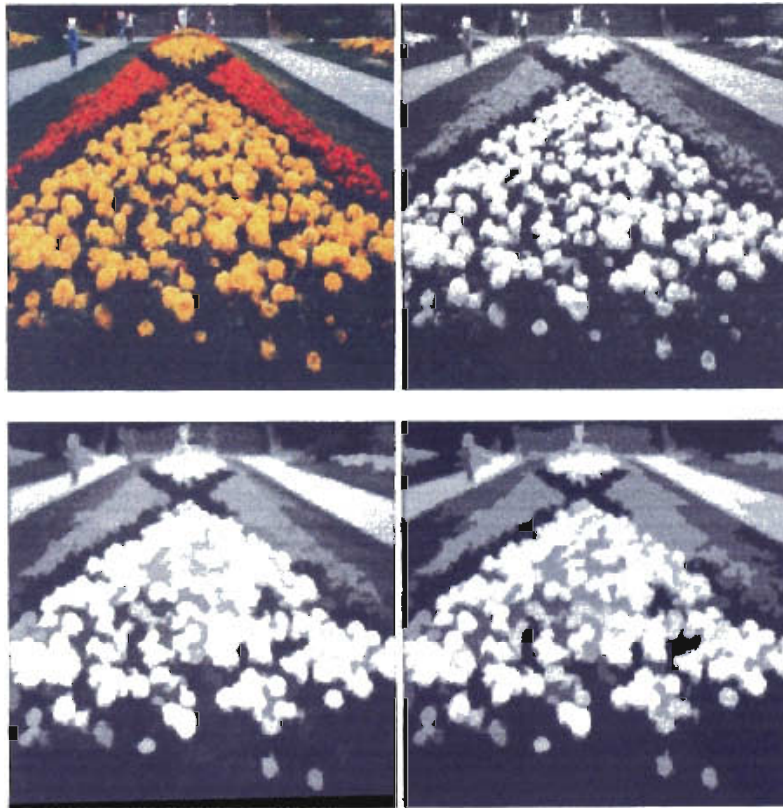
The clustering proceeds in this iterative fashion until convergence, where convergence is defined by insignificant changes in the observed objective functional between two consecutive iterations.

The resultant segmentation provides regions that are homogeneous in terms of texture and are significant in terms of minimum scale. In contrast to the solutions which use heuristic region merging, the segmentation from texture / scale space does not require post-processing, nor knowledge of the number of significant regions. The motivation for multi-scale, multi-Gabor response clustering has been that it improves classification by clustering members of similar objects more effectively than a fixed scale classifier [1]. Figure 9 illustrates results via scaled multi-Gabor response clustering.

However this approach suffers from a drawback. This is due to the inability of AOC to eliminate at times sub-scale connected components, which are in the middle of a staircase distribution of intensity. This particular sandwiched connected component would be part of connected components corresponding to either the first or last steps of the staircase distribution in certain level sets. It would never be an independent connected component. If the connected component(s) it belonged to, was super-scale then it would never be discarded. As a result of which the clustering results sometimes contain certain sub-scale regions, which are a computational irritation in CBR.

As a remedy we would have to resort to an additional post-processing AOC operation to remove these sub-scale region. Figure 10 illustrates this particular drawback of the inability of AOC to eliminate at times sub-scale connected components.

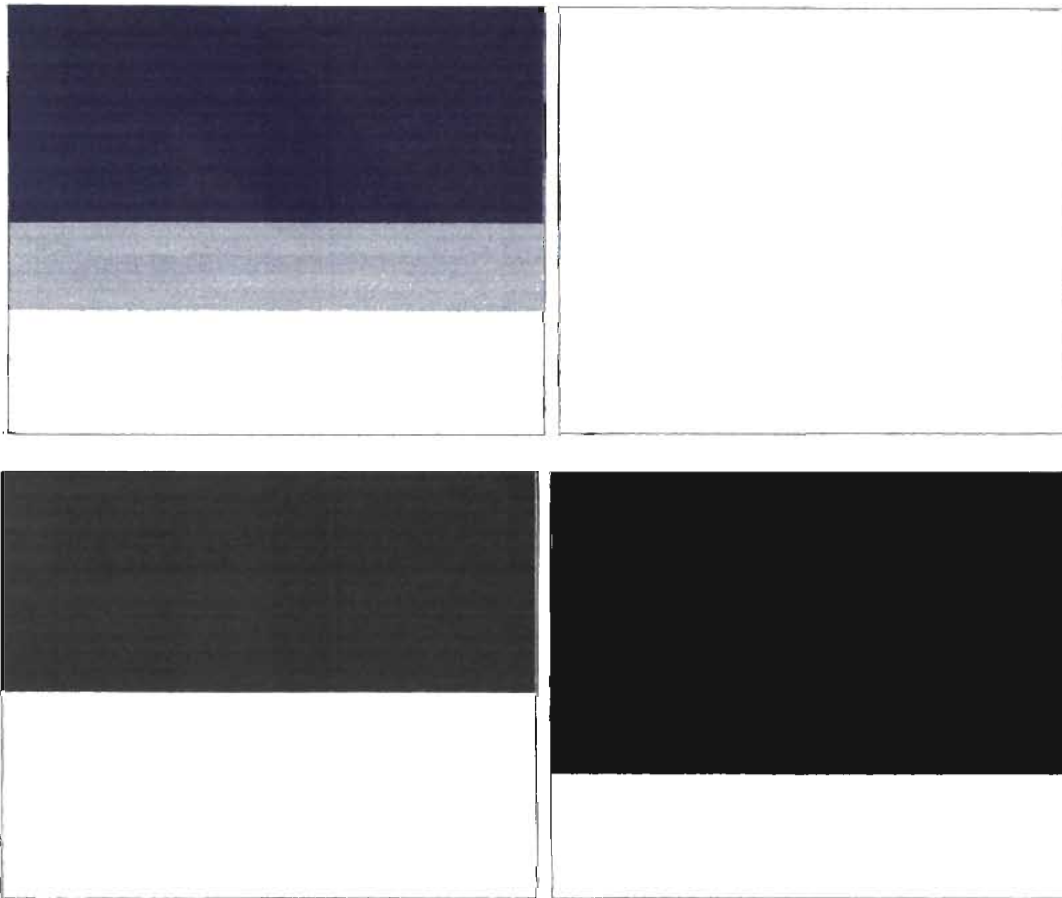
However if we were to use just the connected components corresponding to the level sets then we would be guaranteed segments of a minimum scale. This is because the AOC guarantees minimum scale connected components with all the level sets. Hence we motivate a level set analysis of the scaled Gabor decompositions for segmentation.



**Figure 9 a,b,c,d.** Original image, Luminance space, Scaled Gabor response , FCM result



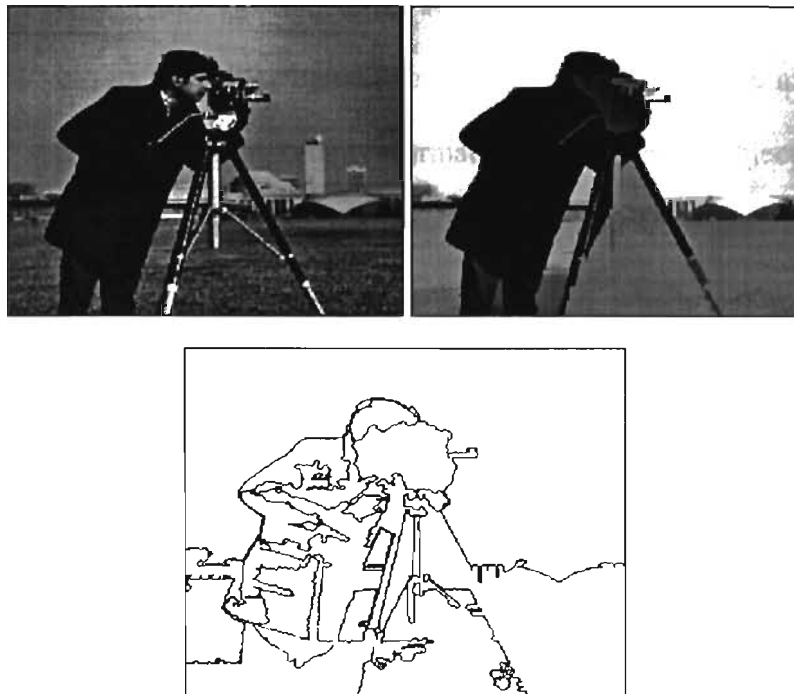
**Figure 9 e.** Overlaid segmentation Garden image



**Figure 10.** Shows a inability of AOC to eliminate at times sub-scale connected components (a) contains 3 connected components of gray levels 0, 1, 2 and areas of 30 , 10 and 20 respectively. (b) shows the level set at gray level 0. It encompasses the whole image. (c). shows the level set at gray level 1(d) shows the level set of gray level 2. None of the level sets contain the connected component at gray level 1 which is sub-scale when compared to the other two connected components.

### 3.5.2 Level Set Analysis

Conventional scale-space segmentation methodologies use an edge-based technique to extract segments from scaled images. The drawback with this technique is that the difference in pixels intensities (edges) is sensitive to changes in illumination. Recent psychovisual studies have shown that the human eye is essentially sensitive to ordering of gray levels in an image rather than the differences between them [37]. The concept of level lines has recently been motivated for various image processing tasks like disocclusion [37]. Level lines are boundaries pixels in level sets. The level lines are invariant to changes in illumination hence offer considerable advantages over edge based methods. Hence the extraction of segments using level lines is well motivated. Figure 11 illustrates the concept of level lines.



**Figure 11.** Original cameraman image, scaled version (area =  $10 \times 10$ ), significant Level lines.

The level sets of scaled Gabor decompositions contain connected components satisfying a minimum scale parameter. We consider these to be prospective segments and would like to extract them for feature extraction purposes. However extraction of all these connected components would lead to a redundant segmentation as many of them are self-contained within other connected components or overlapping with them. Hence a combinatorial analysis is required to extract non-overlapping, non-redundant segments.

For a gray-scale image with intensity values ranging from 0 to 255 there would be a maximum of 256 level sets. However a great many of them differ very insignificantly from the next lower level set. Hence the level sets could be sub sampled to reduce number of level sets being analyzed. The level step parameter is a measure of this sampling step-size. Given a linear level set stepsize  $\Delta b$ , the level sets would range from 0 to 256 in steps of  $\Delta b$ . The magnitude of the constant stepsize  $\Delta b$  determines how effectively the level sets are sub-sampled. However a constant step-size might skip some important level sets containing important information in the form of objects with strongly localized boundaries. We want to sample only the important level sets. These important level sets are defined to contain the important information, which may not conform to a constant stepsize. A more important aspect of this is that this non-linear level step size has to be specified or determined in an automatic manner. The level lines concept enunciated above is an important indicator of important object information in a level set in terms of object edge localization. We propose to use the level lines gradient for determining the level sets to be sampled for purposes of segment extraction.

Consider a level set  $\mathbf{B}_l$

$$\begin{aligned}\mathbf{B}_l(p) &= 1 \text{ if } I(p) \geq l, \\ &= 0 \text{ Otherwise}\end{aligned}\tag{32}$$

where image location  $p \in \mathbf{D}$ , for the discrete domain  $D \subset \mathbf{Z}^2$  and  $l \in \mathbf{Z}$ .

Within these level sets we define the boundary pixels of connected components as level lines. We define the boundary as pixels on the boundary rather than one pixel outside/inside. We assume 4-neighborhood for connected component labeling. Given a connected component  $C_{\mathbf{B}_l}(p)$ , its boundary pixels satisfy the following conditions:

$$\begin{aligned}N_h(p_c) &= \{C_{\mathbf{B}_l}(p_c \pm \nabla p)\}, \\ |N_h(p_c) > 0| &\geq 1\end{aligned}\tag{33}$$

where  $p_c \in C_{\mathbf{B}_l}(p)$ ,  $N_h(p_c)$  is the set of neighborhood pixels for each pixel in  $C_{\mathbf{B}_l}(p)$ ,  $|N_h(p_c)|$  refers to the cardinality of the set of neighborhood pixels and  $\nabla p$  is the minimum resolution step size of the image (usually  $\nabla p = 1$  pixel). In other words a pixel is said to be a boundary pixel if it has at least one off pixel (0 intensity) as a neighbor (4-N in our case). We define the gradient at these boundary pixels  $N_h(p_c)$  as

$$G_N(p_c) = |G_{N_x}| + |G_{N_y}|,\tag{34}$$

$$\text{where } G_{N_x} = \frac{N_h(x_c + 1, y_c) - N_h(x_c - 1, y_c)}{2}, \quad G_{N_y} = \frac{N_h(x_c, y_c + 1) - N_h(x_c, y_c - 1)}{2}$$

$G_{N_x}$  and  $G_{N_y}$  correspond to the gradients in the  $x$  and  $y$  directions respectively.

Granulometry has traditionally been used in image processing to describe the size distribution of particles in an image [28]. We use granulometry to parameterize

the level sets. This parameterization will enable us to extract significant level sets for segmentation. The parameter we use is cumulative gradient from level lines. This parameter is recorded for each level set.

We term this metric the Level Lines Gradient Granulometry (LLGG). The LLGG metric for a level set  $\mathbf{B}_l$  is given by

$$\mathbf{L}_g(\mathbf{B}_l) = \sum_{c=1}^{cc} G_N(p_c) \quad (35)$$

where  $G_N(p_c)$  is the cumulative level lines gradient of a connected component  $p_c$  and  $cc$  is the number of connected components in the level set  $\mathbf{B}_l$ . The LLGG measure for a complete image is given by

$$\mathbf{L}_g(l) = \mathbf{L}_g(\mathbf{B}_{l\Delta b}) \quad (36)$$

where  $l$  is the level set number and usually ranges from 0 to  $G$  (maximum number of graylevels in the image ( $G=256$  usually)) in steps of  $\Delta b$ .  $\Delta b$  is a constant level step parameter ( $\Delta b=1$  usually).

The LLGG metric at each level set is essentially the cumulative absolute gradient of the level lines in the level set. This metric tracks the behavior of the level lines across the level sets. The granulometry can be regarded as a 1-D signal whose peaks correspond to level sets containing important objects with strongly localized boundaries. We extract these peaks morphologically to eliminate redundancies between sampled level sets. This process is analogous to determining a variable level step size  $\Delta b$ .

Consider a 1-D LLGG signal  $\mathbf{L}_g(B_l)$ . The peaks in this 1-D signal would need to be defined within a neighborhood  $n$ . Computationally the peaks are extracted by matching equal amplitude-position points between the original signal  $\mathbf{L}_g(B_l)$  and its



dilated version  $L_g(B_l) \bullet s_n$ , where  $s_n$  is a 1-D structuring element of size  $n$ . However exceptions have to be made for purely flat regions in the original signal  $L_x(B_l)$  else all the points in those regions will match with the corresponding region in the dilated signal. Very closely separated peaks ( in terms of both dimensions) would correspond to adjacent level sets with important information. However this also means information redundancy because of the adjacent nature of the level sets. Hence it is important to compute the peaks within an adequate neighborhood. The particular neighborhood parameter is a design parameter in our experiments. Increasing the neighborhood size decreases the number of sampled level sets. This decreases the computation time required in combinatorial analysis of the connected components collected from these non-linearly sampled level sets.

### 3.5.2.1 Connected Component Combinatorial Analysis

The candidate connected components for segmentation are collected from the level sets specified by the non-linear LLGG metric. In view of redundancies among these connected components a combinatorial analysis of the same is required. The combinatorial analysis determines overlapping and redundant connected components. Given two connected components at locations  $p$  and  $r$

$$C_{Bl}(p) = \{q : \exists P_{lzl}(p, q)\},$$

$$C_{Bl}(r) = \{q : \exists P_{lzl}(r, q)\},$$

The neighboring pixels in this path are defined by 4-connectivity in our case.  $C_{Bl}(p)$  and  $C_{Bl}(r)$  are said to overlap if they satisfy the condition

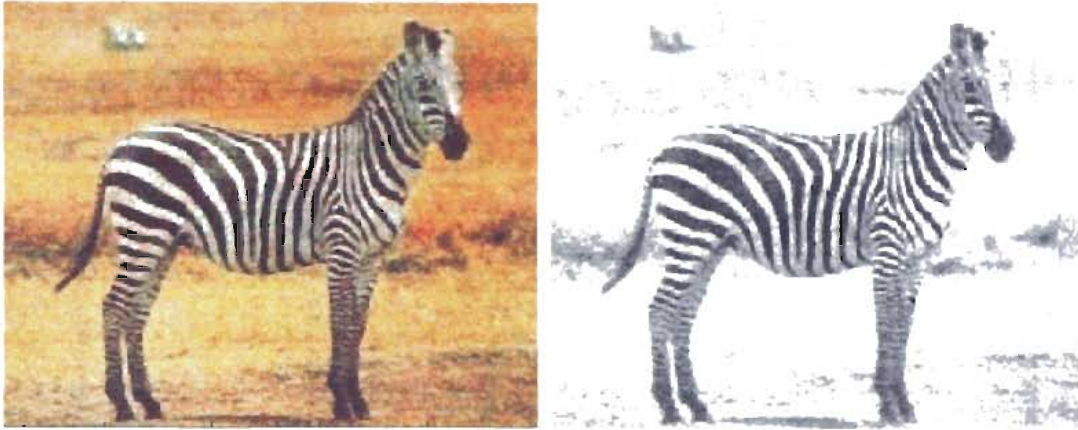
$$C_{Bl}(p) \cup C_{Bl}(r) \neq \emptyset \quad (37)$$

Equation 37 implies that the two connected components  $C_{Bl}(p)$  and  $C_{Bl}(r)$  overlap. Thus in case of spatial overlap these connected components are compared to minimize total internal variance. In other words the connected component with minimum total interval variance among the two overlapping connected components overrides the other. The total internal variances of the connected components  $C_{Bl}(p)$  and  $C_{Bl}(r)$  are given by

$$\sigma^2(C_{Bl}(p)) = \sqrt{\left( \frac{1}{|C_{Bl}(p)|} \sum_p (J_p - \mu_{J_p})^2 \right)} \quad (38)$$

$$\sigma^2(C_{Bl}(q)) = \sqrt{\left( \frac{1}{|C_{Bl}(q)|} \sum_q (J_q - \mu_{J_q})^2 \right)} \quad (39)$$

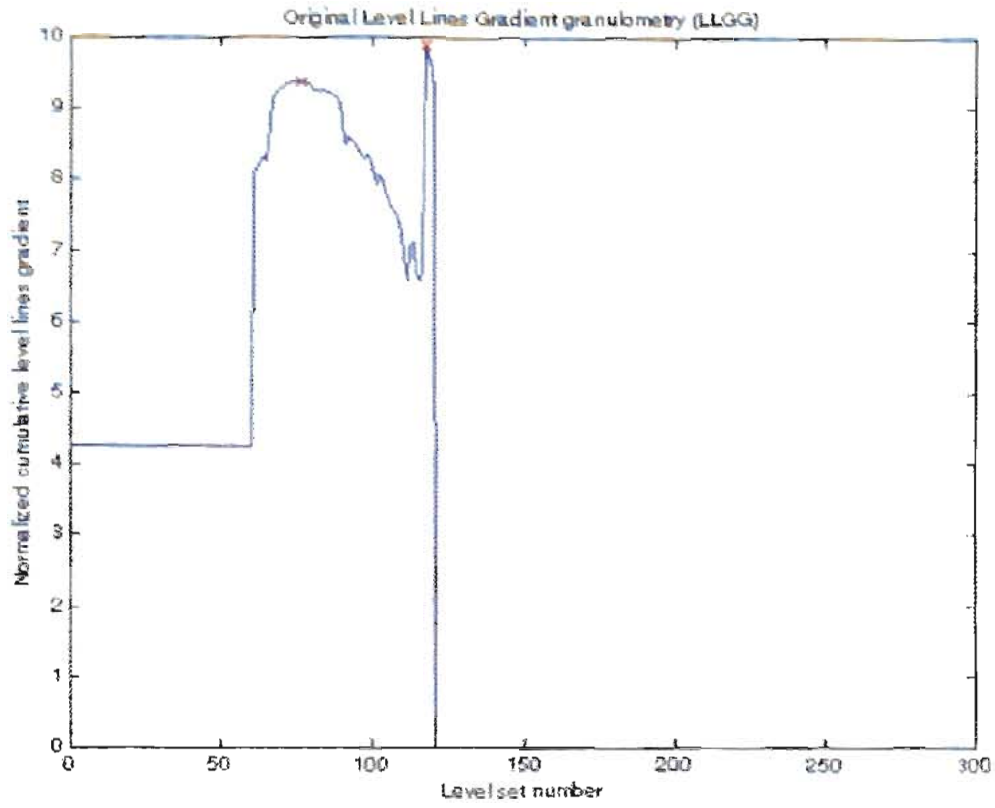
where  $|C_{Bl}(p)|$  and  $|C_{Bl}(q)|$  are the cardinality of the connected components,  $J_p$  and  $J_q$  are their intensities from the scaled Gabor decomposition at locations  $p$  and  $q$  respectively,  $\mu_{J_p}$  and  $\mu_{J_q}$  are the mean intensities of these connected components from the scaled Gabor decomposition. Figures 12 and 13 show segmentations via combinatorial analysis of scaled Gabor decompositions.



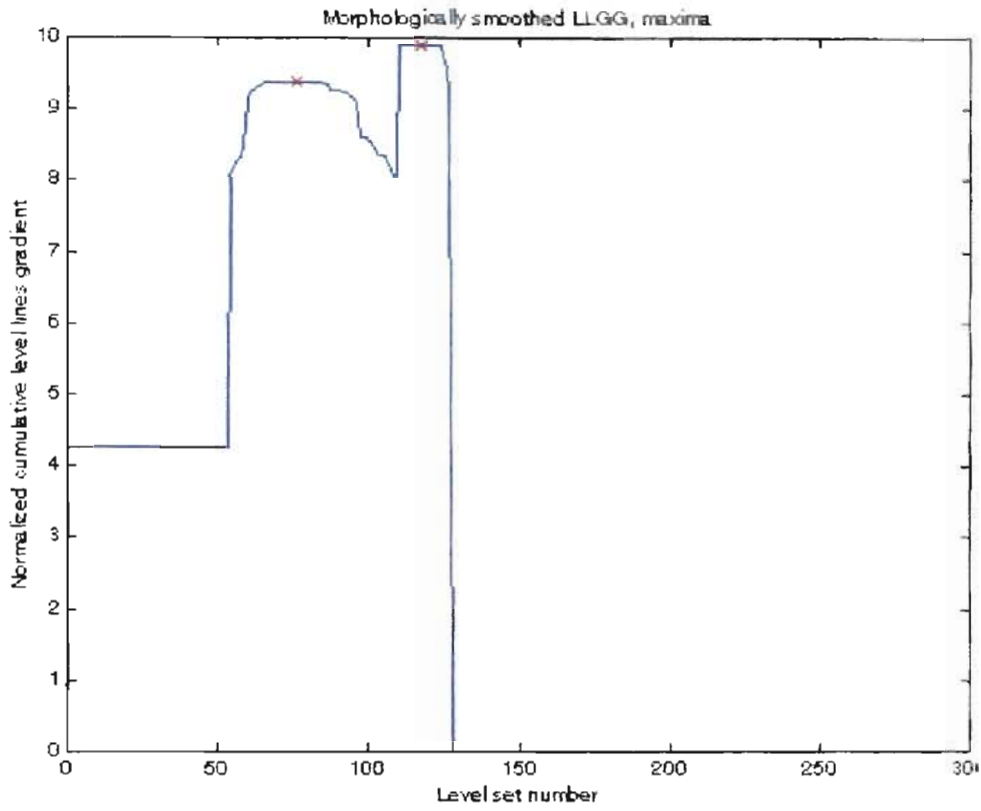
**Figure 12a.** Original Zebra image, luminance space



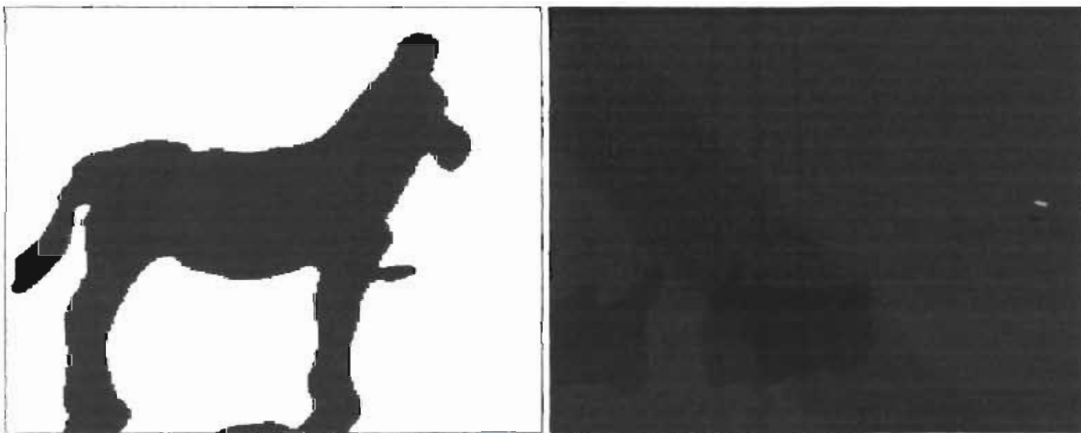
**Figure 12 b.** Gabor response  $u=0.03$  cycles/pixel,  $v=0.02$  cycles/pixel, Scaled Gabor response  $s=45 \times 45$



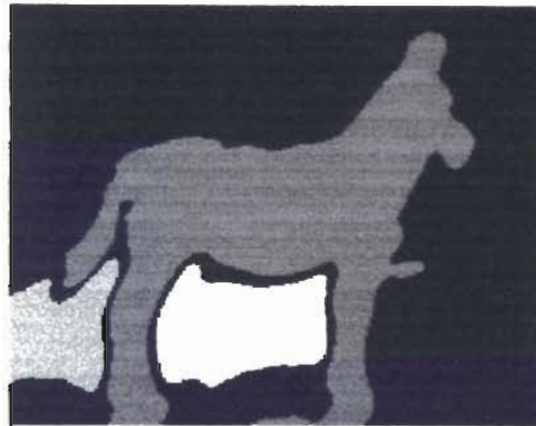
**Figure 12 c.** Image LLGG granulometry of scaled response in figure 12 (b)  
Local Peaks marked in red (neighborhood=15)



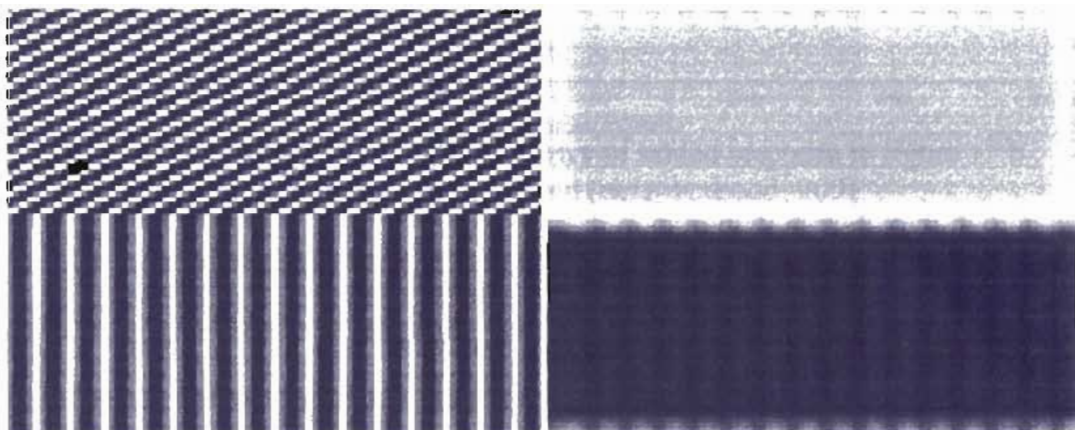
**Figure 12 d.** Morphologically filtered Image LLGG granulometry of scaled response in figure 12 (b) Local Peaks marked in red (neighborhood=15)



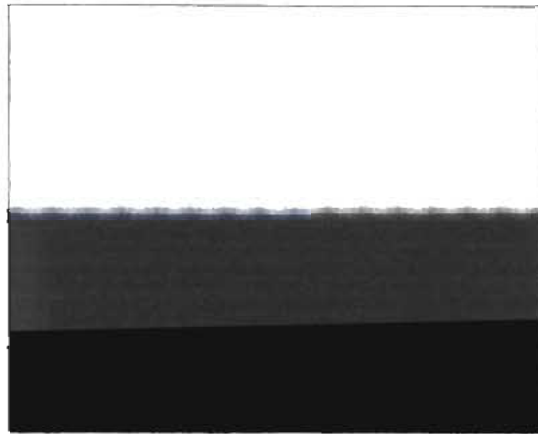
**Figure 12 e.** Sampled level sets from figure 12 d.



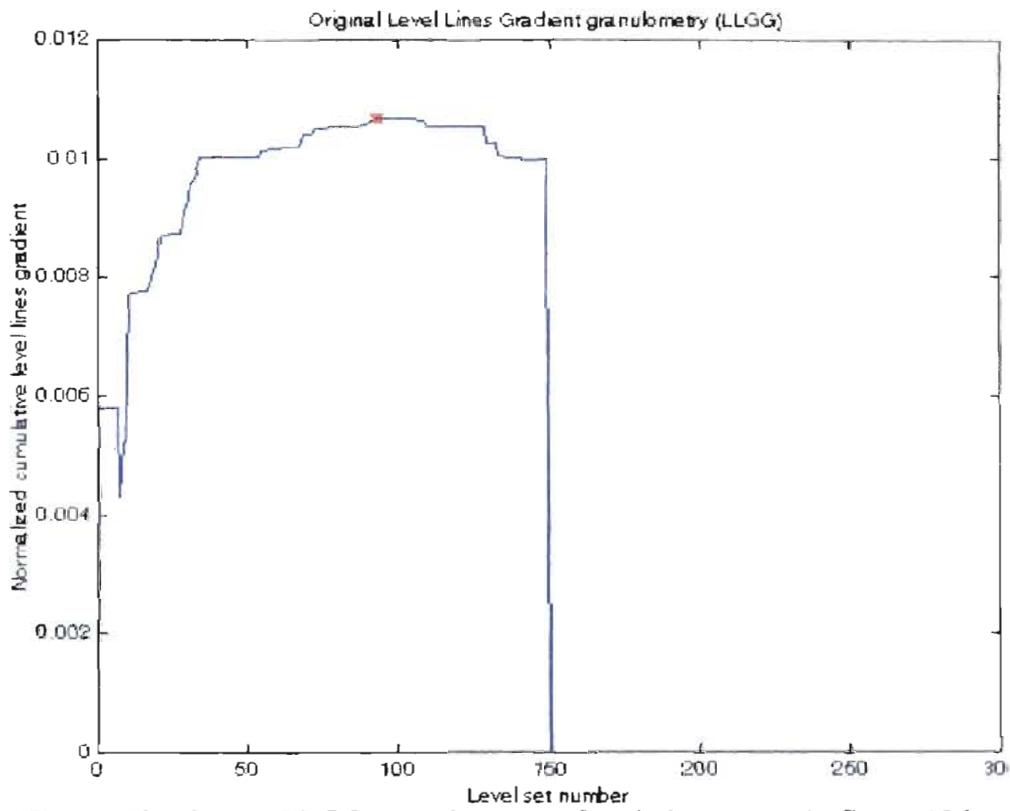
**Figure 12 f.** Segmentation map, overlaid segmentation



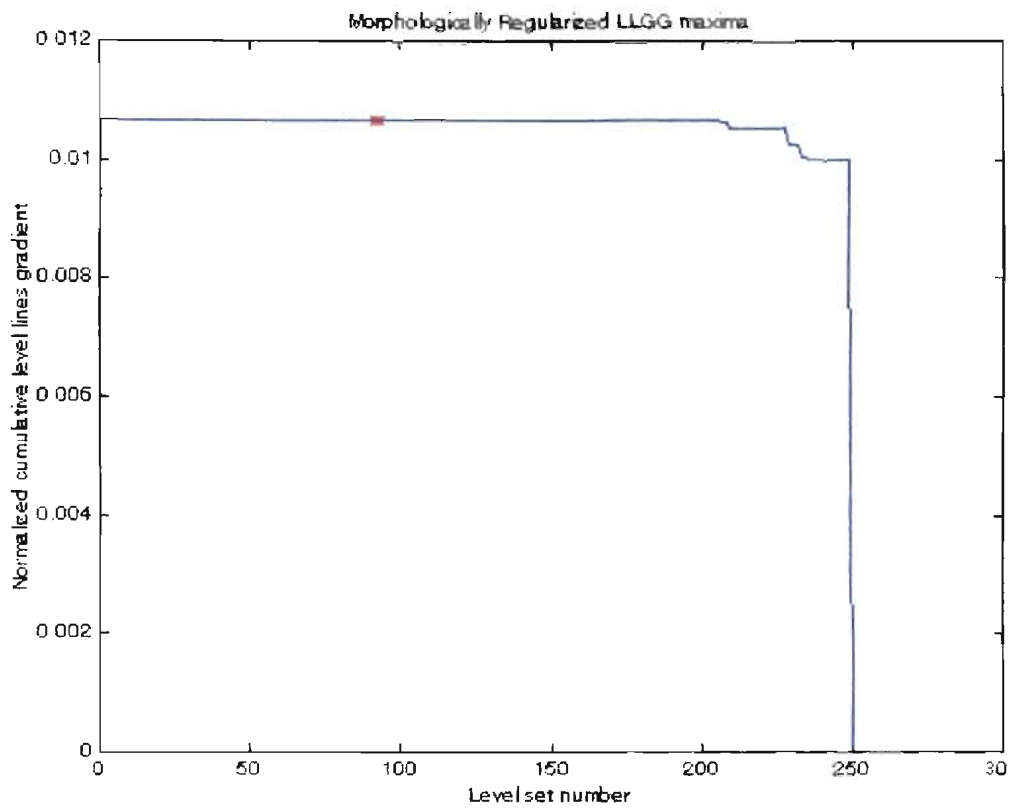
**Figure 13 a.** Original synthetic texture image, Gabor response  $u=0.3$  cycles/pixel,  $v=0.2$  cycles/pixel



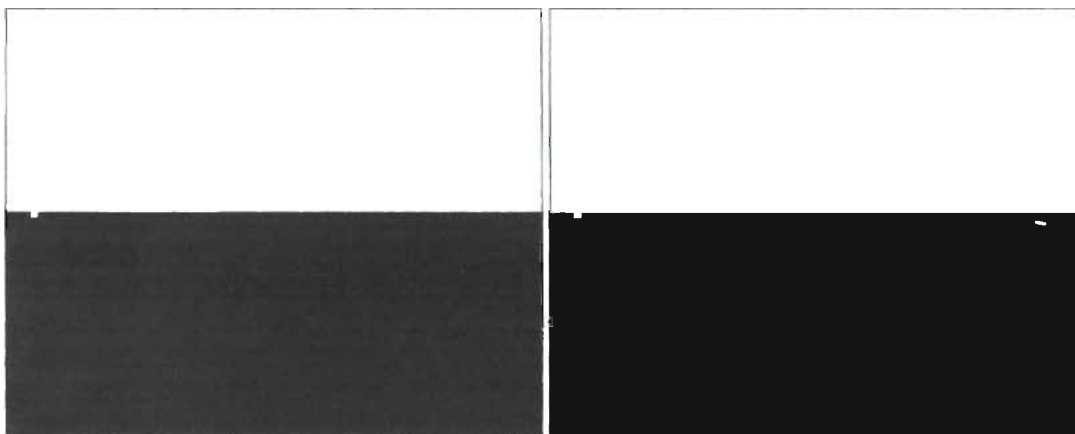
**Figure. 13 b.** Scaled Gabor response (from figure 13 a) Area = 15x15



**Figure 13 c.** Image LLGG granulometry of scaled response in figure 13 b  
Local Peaks marked in red (neighborhood=200)

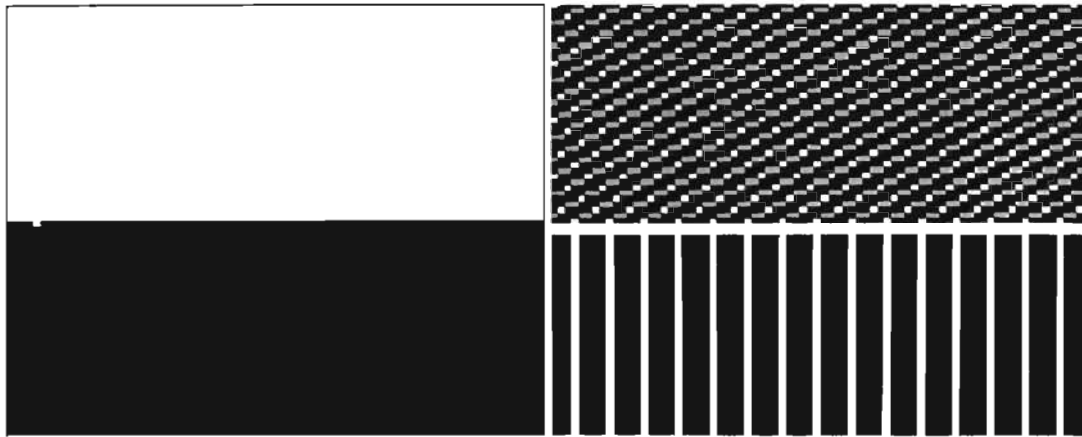


**Figure 13 d.** Morphologically filtered Image LLGG granulometry of scaled response in figure 13 b, Local Peaks marked in red (neighborhood=200)



**Figure 13 e.** Sampled level sets from figure 13 d.





**Figure 13 f.** Segmentation map, overlaid segmentation from figure 13 e.

### 3.5.2.2 COMPLEXITY ANALYSIS

The scaling methodology consists of marker image creation and reconstruction by geodesic dilation. For marker image creation using the open filter the computational complexity is  $2aN$  comparisons where  $a$  is the area of the structuring element and  $N$  is the size of the structuring element. Hence it can be written as  $O(aN)$ . In the case of reconstruction 4 comparisons per pixel are required for dilation and 1 for computing the minima, which gives a total of  $5TN$  operations.  $T$  is the total number of reconstruction steps per pixel. Hence the computational complexity can be written as  $O(TN)$ . This can be reduced to  $O\left(N^{\frac{3}{2}}\right)$ .

Connected component labeling is sequential in nature, hence its complexity can be written as  $O(\beta n)$  where  $\beta$  is a constant and  $n$  is the number of pixels in the image. Given  $n_c$  number of connected components for a combinatorial analysis. The number of comparisons required for the first connected component is  $n_c - 1$  i.e. it is compared with each of the other connected components (excluding itself). Progressing

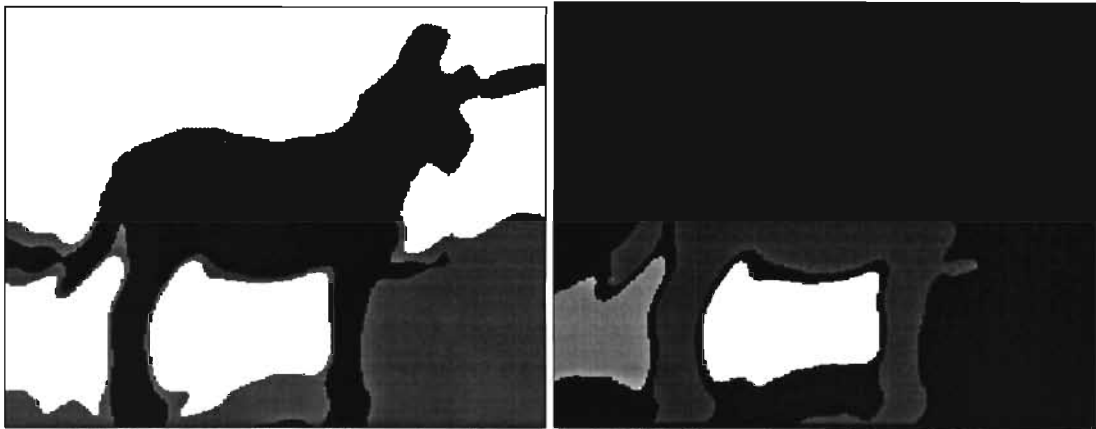
similarly for all connected components the total number of comparisons required is  $n_i!$ . Thus the complexity of connected component combinatorial analysis can be written as  $O(n!)$ .

### 3.6 Chapter Summary

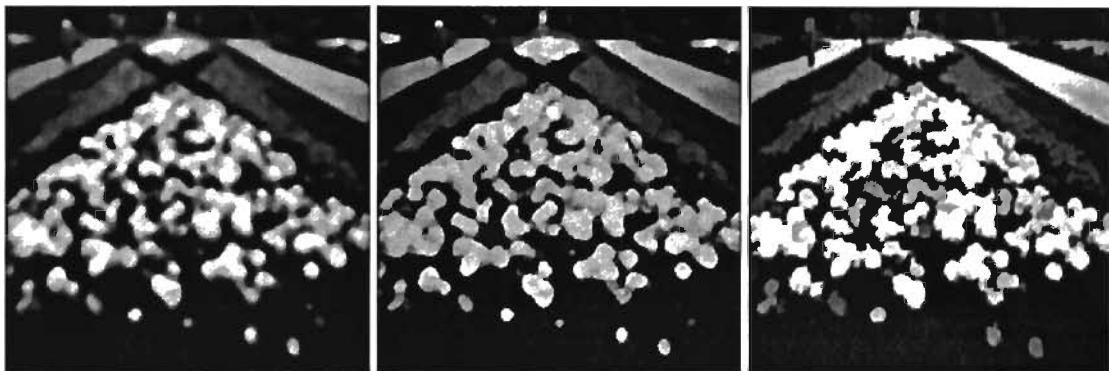
Ideally one would expect a comparison for the segmentation methods outlined in this chapter. However we are unable to do so as the segmentation via the level set analysis of scaled Gabor responses results in a partial segmentation. While regular segmentation methods provide a complete partition of the image. Thus a comparison between such segmentation methods would be analogous to comparing apples and oranges! However we do provide two such segmentation results in figure 14 for visual assessment of the segmentation quality. We can provide comparisons for the segmentation via fuzzy clustering of scaled Gabor responses. These results are elaborated upon in detail in [47]. These comparisons are to show the superiority of the AOC scaling method over other scaling methods. For this purpose we use the Weighted Majority with Minimum Range (WMMR) filter and the median filter as comparative scaling methods. The details regarding formulation of these scaling methods can be obtained in [47]. The segmentations are achieved by fuzzy clustering of scaled Gabor responses. The only difference being in the method of scaling. The LLGG metric enunciated earlier in section can be considered as a measure of edge strength/localization. Figure 15 a shows the scaling results from the three scaling methods while figure 15 b shows the resultant segmentations. Figure 15 c shows the LLGG metric for the segmentations via the three

scaling paradigms mentioned. Figure 15 c clearly shows the superiority of the AOC segmentation over those obtained via the median and WMMR scaling methods. This superiority is measured in terms of edge strength/localization.

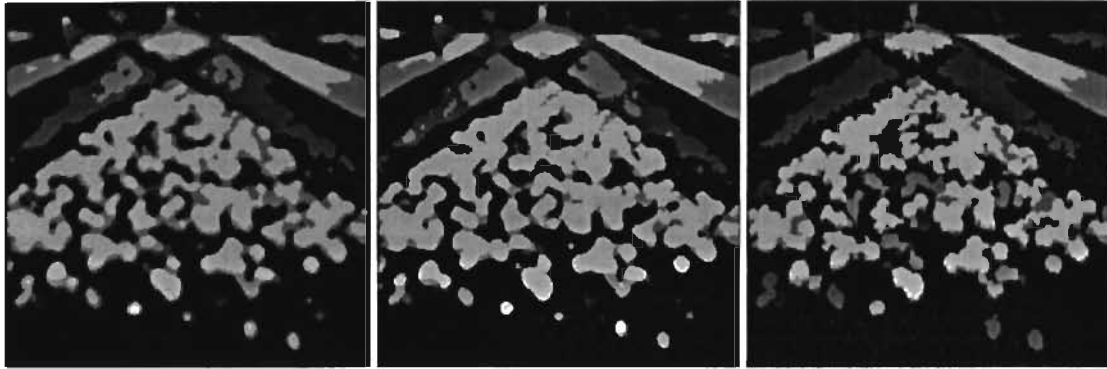
Thus in this chapter we have outlined a well-motivated approach to image segmentation for CBR. A Gabor filter based framework has been outlined for texture region extraction. A novel area morphological scaling method has been described. Two paradigms for segment extraction from the scaled Gabor decompositions have been described. Qualitative and quantitative results have been shown, which establish the superiority of our segmentation and scaling techniques.



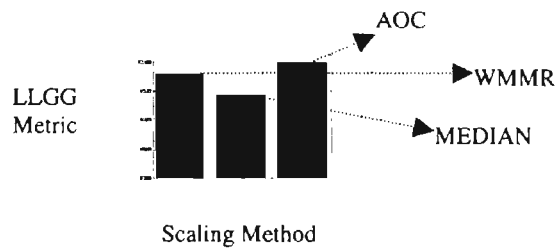
**Figure 14.** Segmentation of the zebra image by FCM clustering and Level set analysis of scaled Gabor decompositions



**Figure 15 a.** Median, WMMR and AOC scaled representations of garden image in figure 9 ( $7 \times 7$ ,  $a = 49$ )



**Figure 15 b.** Clustering results for the median, WMMR, and AOC texture spaces in figure 15 a (4 classes)



**Figure 15 c.** Superiority of AOC over WMMR and MEDIAN filters via the LLGG metric computed over the various (scaling techniques) segmentations(Figure 15 a,b).

## CHAPTER IV

### FEATURE EXTRACTION

#### 4.1 Introduction

Conventional CBR techniques have focussed on using text keywords as features. These type of retrieval mechanisms are seen to be ineffective in view of pressing issues like access automation and metadata relevancy. Hence there has been a renewed thrust on the usage of segment specific features for CBR. The focus of our research has been on formulation and demonstration of a robust segmentation algorithm for CBR. This chapter elaborates on the various features computed for CBR.

A recent thrust on CBR research has focussed on the usage of mathematical models of image properties like color, texture and object shapes, as criteria for similarity analysis. More specifically these have been global features pertaining to color and texture. The drawbacks of these global features have been described in the first chapter of this manuscript, and the advantages of segment specific features have been highlighted. Ideally one would envision a purely segment specific matching strategy for CBR. However modern digital image libraries contain a large number of images, making an exhaustive segment specific matching process computationally intensive. In order to achieve a tradeoff between retrieval quality and retrieval time we propose to use a judicious combination of global and local features for CBR.

We extract global features pertaining to color and texture. More importantly we extract local features pertaining to color, texture and shape. A novel color preprocessing scheme is used to smooth image artifacts, thus aiding in efficient feature extraction. A coarse color quantization aids in determining the dominant colors for the global color features. We now elaborate on the specifics of the feature extraction framework for CBR.

## 4.2 Color feature extraction

The use of color for retrieval is motivated by two principal factors. First, in automated image analysis, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, in image analysis performed by human beings, the motivation for color is that the human eye can discern thousands of color shades and intensities, compared to about only two-dozen shades of gray [28]. Although the process followed by the human brain in perceiving is a color physiopsychological phenomenon that is not yet fully understood, the physical nature of color can be expressed on a formal basis supported by experimental and theoretical results.

In 1666, Sir Isaac Newton discovered that when a beam of sunlight is passed through a glass prism the emerging beam of light is not white, but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other [28].

- Basically, the colors that human beings perceive in an object are determined by the nature of the light reflected from the object [28]. Characterization of light is central to the science of color. If the light is achromatic, its only attribute is its intensity. Achromatic light is what viewers see on a black and white television set. Thus the term gray level

refers to a scalar measure of intensity that ranges from black, to grays, and finally to white. Three basic quantities radiance, luminance and brightness are used to describe the quality of a chromatic light source. Radiance is the total amount of energy that flows from the light source, and is usually measured in watts (W). Luminance, measured in lumens(lm), gives a measure of the amount of energy an observer perceives from a light source. Brightness embodies the achromatic notion of intensity and is one of the key factors in describing color sensation. Owing to the structure of the human eye, all colors are seen as variable combinations of the three so-called *primary colors* red (R), green (G), and blue (B). The primary colors can be added to produce the secondary colors of light magenta (red plus blue), cyan (green plus blue), and yellow (red plus green). Mixing the three primaries, or a secondary with its opposite primary color, in the right intensities produces white light.

#### 4.2.1 Color Models

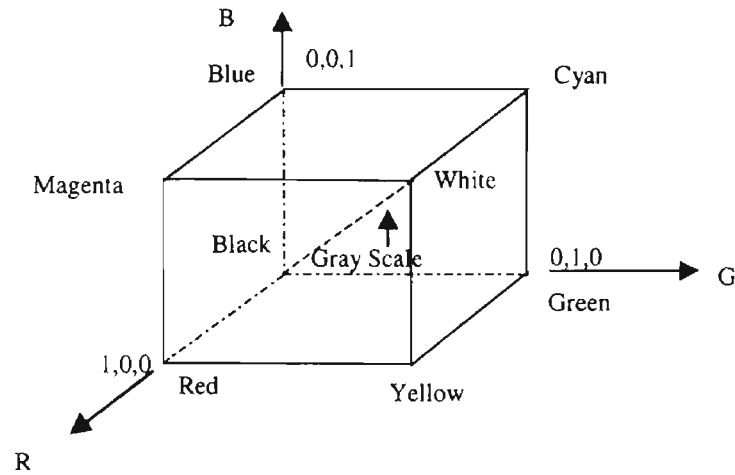
The purpose of a color model is to facilitate the specification of colors in some standard, generally accepted way. In essence, a color model is a specification of a 3-D coordinate system and a subspace within that system where each color is represented by a single point. Most color models in use today are oriented either toward hardware (such as for color monitors and printers) or toward applications where color manipulation is a goal (such as in the creation of color graphics for animation). The hardware-oriented models most commonly used in practice are the RGB (red, green, blue) model for color monitors and a broad class of color video cameras; the CMY (cyan, magenta, yellow)

model for color printers; and the  $YC_bC_r$  model, which is the standard for color TV broadcast [28]. In the third model the  $Y$  corresponds to luminance, and  $C_b$  and  $C_r$  are two chromatic components. Among the models frequently used for color image manipulation are the RGB model and the HSV (hue, saturation, value) model. In our experiments we have used the RGB model for color feature extraction. The  $YC_bC_r$  and the CMY color models are more suited for TV transmission and printing purposes respectively.

#### 4.2.1.1 The RGB color model

In the RGB model, each color appears in its primary spectral components of red, green and blue. This model is based on a Cartesian coordinate system. The color subspace of interest is the cube shown in figure 16 in which RGB values are at three corners; cyan, magenta and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin. In this model, the gray scale extends from black to white along the line joining these two points, and colors are point on or inside the cube, defined by vectors extending from the origin. For convenience, the assumption is that all color values have been normalized so that the cube shown in figure 16, is the unit cube.





**Figure 16.** RGB color cube. Points along the main diagonal have gray values, from black at origin to white at point (1,1,1).

That is, all values of  $R$ ,  $G$  and  $B$  are assumed to be in the range  $[0,1]$ . Images in the RGB color model consist of three independent image planes, one for each primary color. The RGB system is used for display in monitors. Thus it makes sense to use the RGB model for image processing when the images themselves are naturally expressed in terms of this model. Moreover, most color cameras used for acquiring digital images utilize the RGB format, making this an important model in image processing. For instance most multispectral satellite imagery consist of various bands. Some of these bands include R, G, B and infrared [28].

#### 4.2.2 Preprocessing for color feature extraction

Most artifacts in real world images are due to errors from noisy transmission channels, network bit errors and block based coding schemes. A plain color feature extraction scheme for CBR can produce erroneous features in view of these artifacts. The

remedy we propose is preprocessing to eliminate these artifacts. More specifically we propose to smooth these noisy images for semantic color feature extraction. This problem can be categorized under the broad category of image restoration [9]. This area of research has been one of the most explored areas in image processing.

One of the most popular smoothing techniques can be broadly grouped under variational formulations. This involves setting up an energy functional involving the input image and minimizing it via the principles of variational calculus yielding Euler equation(s). These Euler equations are solved to get the desired image. Most of the variational approaches are iterative in nature hence ill-suited for CBR. [38] gives an excellent review of various variational formulations in image processing.

Some of the other formulations involve various low pass filtering techniques like Gaussian filtering. Some of the earliest scale-space formulations advocated Gaussian filtering for this purpose [59]. However Gaussian filtering is known to suffer from various drawbacks of feature drift across scale i.e. edge localization problems.

Mathematical morphology is a smoothing technique based on the fundamental erode and dilate operators which are windowed maxima and minima operators, respectively. A more detailed review has already been undertaken in the chapter III. A point worth repeating here is that these morphological filters smooth noise, preserve edges and eradicate impulses. However on the downside there are minor points like the gray-level bias of morphological filters. The various advantages and drawback of morphological filters are elaborated upon in [15].

Anisotropic diffusion is another smoothing technique, which offers edge preservation through the image smoothing process. This is achieved via intraregion smoothing and inhibiting interregion smoothing (diffusion at edges). Anisotropic diffusion has been widely implemented in image and video processing architectures in view of its attractive image processing properties. Another point in favor of anisotropic diffusion is that the localized computation of anisotropic diffusion allows efficient implementation on a locally interconnected computer architecture. [3] gives an exhaustive survey of anisotropic diffusion methods. However plain implementations of anisotropic diffusion are ill-suited for CBR as they are restricted in their ability to remove impulse noise and small objects [52]. Thus we have seen how various image smoothing mechanisms are ill-suited for ready application to CBR. We propose a technique, which combines the strengths of morphology and diffusion based image smoothing techniques. We now give a brief background on diffusion and elaborate on our image smoothing technique for color feature extraction for CBR.

#### 4.2.2.1 Background on diffusion

Diffusion is the physical process resulting from random motion of molecules by which there is a net flow of matter from a region of high concentration to a region of low concentration. One of the most common examples is the transfer of heat. The traditional Gaussian filter has been shown to be a formulation of an isotropic diffusion process governed by

$$\frac{\partial \mathbf{I}_t}{\partial t} = \Delta \mathbf{I}_t \quad (40)$$

where  $\mathbf{I}$  is the image,  $t$  is the time instant and  $\Delta \mathbf{I}_t$  is the Laplacian of  $\mathbf{I}_t$ . Thus to evolve one pixel of  $\mathbf{I}$ , we have the following Partial Differential Equation (PDE):

$$\frac{\partial \mathbf{I}_t(x)}{\partial t} = \Delta \mathbf{I}_t(x) \quad (41)$$

As the equation indicates this is an iterative process. The edges from the image processed by isotropic diffusion suffer from artifacts such as corner rounding and from edge localization errors (deviations in detected edge position from the true edge position). Hence researchers have pursued anisotropic diffusion as an alternative to smooth images and detect edges [3].

As briefly mentioned earlier anisotropic diffusion inhibits smoothing at edges, thus yielding intraregion region smoothing rather than interregion smoothing. This is achieved via the formulation of a diffusion coefficient. Perona and Malik, in their landmark paper [43] advocated the concept of anisotropic diffusion. They formulated a non-linear diffusion coefficient, which inhibited smoothing at image edges. The anisotropic diffusion formulation in [43] is given by

$$\frac{\partial \mathbf{I}}{\partial t} = \text{div}[c \nabla \mathbf{I}] \quad (42)$$

where  $\text{div}$  is the divergence operator,  $\nabla \mathbf{I}$  is the image gradient, and  $c$  is the diffusion coefficient. The diffusion coefficient is allowed to vary with the image gradient. Thus the success of the algorithm depends on the formulation of this diffusion coefficient  $c$ . One formulation in [43] used a diffusion coefficient  $c$  of the form

$$c = e^{-\left(\frac{\|\nabla \mathbf{I}\|}{k}\right)^2} \quad (43)$$

where  $k$  is a gradient threshold. A low gradient ( $\nabla I$ ) value drives up (approaches unity) the value of the diffusion coefficient while a high gradient ( $\nabla I$ ) value drives down (approaches zero) the value of the diffusion coefficient. This inhibits smoothing across edges while promoting smoothing in relatively (w.r.t. edge threshold  $k$ ) flat regions with a low gradient. [43] gives a discrete version of the traditional continuous framework described above. The discrete iterative solution is given as

$$\mathbf{I}_{i,j,t+\Delta t} = \mathbf{I}_{i,j,t} + \lambda(c_N \nabla_N + c_S \nabla_S + c_E \nabla_E + c_W \nabla_W) \quad (44)$$

where  $\mathbf{I}_{i,j,t+\Delta t}$  is the image pixel value at location  $(i,j)$  time  $t$ ,  $\nabla_N, \nabla_S, \nabla_E, \nabla_W$  are image gradients in the north, south, east and west directions respectively,  $c_N, c_S, c_E, c_W$  are the corresponding diffusion coefficients,  $\Delta t$  is the time step and  $0 \leq \lambda \leq \frac{1}{4}$ .

In spite of advantages pertaining to edge locality; anisotropic diffusion is unable to remove small scale features with high contrast. Thus anisotropic diffusion is ill-suited to remove heavy-tailed noise and thus more importantly in this case ill-suited as a sampling prefilter. These small scale regions can contribute to erroneous color features for a particular region. Various such problems of anisotropic diffusion are elaborated upon in [52]. One possible solution summarized in [3] uses a modification of the gradient image used in computing the diffusion coefficients. Typically these have been Gaussian convolved versions of the gradient image. Even though this method denoises the image, it possesses certain drawbacks wherein the edges are softened and blotching artifacts appear in the smoothed result. Moreover the introduction of the linear Gaussian filter reintroduces edge movement thus defeats the primary motivation of the non-linear anisotropic diffusion process. Segall and Acton in [52] introduced a morphological diffusion coefficient which was motivated by the need to smooth small

scale objects while minimizing feature drift , identification of regions without removing important high frequency content. The morphological diffusion coefficient is of the form

$$c = e^{-\left(\frac{\|\nabla(I \circ K) \circ K\|}{k}\right)^2} \quad (45)$$

where  $\mathbf{I}$  is the original image,  $k$  is the gradient threshold and  $(\mathbf{I} \circ \mathbf{K}) \circ \mathbf{K}$  is the result of applying the morphological close-open filter to the original image  $\mathbf{I}$ . This particular formulation allows via anisotropic diffusion the elimination of sub-scale (less than the structuring element  $\mathbf{K}$ ) objects, at the same time maintaining edge locality. Edge quality metrics of morphological anisotropic diffusion are given in [52].

#### 4.2.2.2 The M-CAD scheme for color smoothing

Accurate color feature extraction requires that smoothing be performed in the color domain. All the smoothing methods described above are for gray-scale. However these methods have been adapted in the literature for multispectral/color data [4] [51]. Pope and Acton in [45] extended the modified gradient concept of [52] to the multispectral diffusion methods in [4] and [51]. [45] also provided comparisons, which corroborated the fact that the diffusion method of [4] with modified gradients exhibited superior performance in terms of edge localization and elimination of impulse noise. More technical data regarding our motivation for using the modified gradient vector distance dissimilarity diffusion method of [45] can be obtained in [45]. We now elaborate on our color smoothing paradigm for color feature extraction.

An RGB color image  $\mathbf{I}_M$  ( $M$ =multispectral) can be expressed in terms of its red  $\mathbf{I}_R$ , green  $\mathbf{I}_G$  and blue  $\mathbf{I}_B$  bands. A multispectral gradient can be defined for this RGB color image  $\mathbf{I}$  [24]. For example the multispectral gradient in the southern direction can be expressed as

$$\nabla_r(x, y, t) = \sqrt{(\mathbf{I}_R(x, y-1, t) - \mathbf{I}_R(x, y, t))^2 + (\mathbf{I}_G(x, y-1, t) - \mathbf{I}_G(x, y, t))^2 + (\mathbf{I}_B(x, y-1, t) - \mathbf{I}_B(x, y, t))^2} \quad (46)$$

The gradient terms for the other directions can be defined similarly. The diffusion coefficient for our color diffusion paradigm can be defined as follows.

$$c_M = e^{-\left(\frac{\|\nabla S_{I_M}\|}{k}\right)^2} \quad (47)$$

where  $c_M$  is the diffusion coefficient for band  $M$  of the image  $\mathbf{I}_M$  and  $\nabla S_{I_M} = \nabla(\mathbf{I}_M \bullet K) \circ K$  is the gradient of the result of applying the morphological close-open filter to band  $M$  of the image  $\mathbf{I}_M$ . Hence we term our color smoothing scheme the integrated morphological-color anisotropic diffusion (M-CAD) scheme for color smoothing. The diffusion equation for the M-CAD scheme can be written as

$$\frac{\partial \mathbf{I}_M}{\partial t} = \text{div}[c_M \nabla \mathbf{I}_M] \quad (48)$$

where  $c_M$  is the modified diffusion coefficient for the band  $M$  of the color image  $\mathbf{I}_M$  and  $\nabla \mathbf{I}_M$  is the band  $M$  gradient. The discrete version of equation 48 is similar to equation 46. Figure 17 shows a few results from our M-CAD scheme. Before going on to the numerical formulation of the color features extracted, we would like to describe briefly a

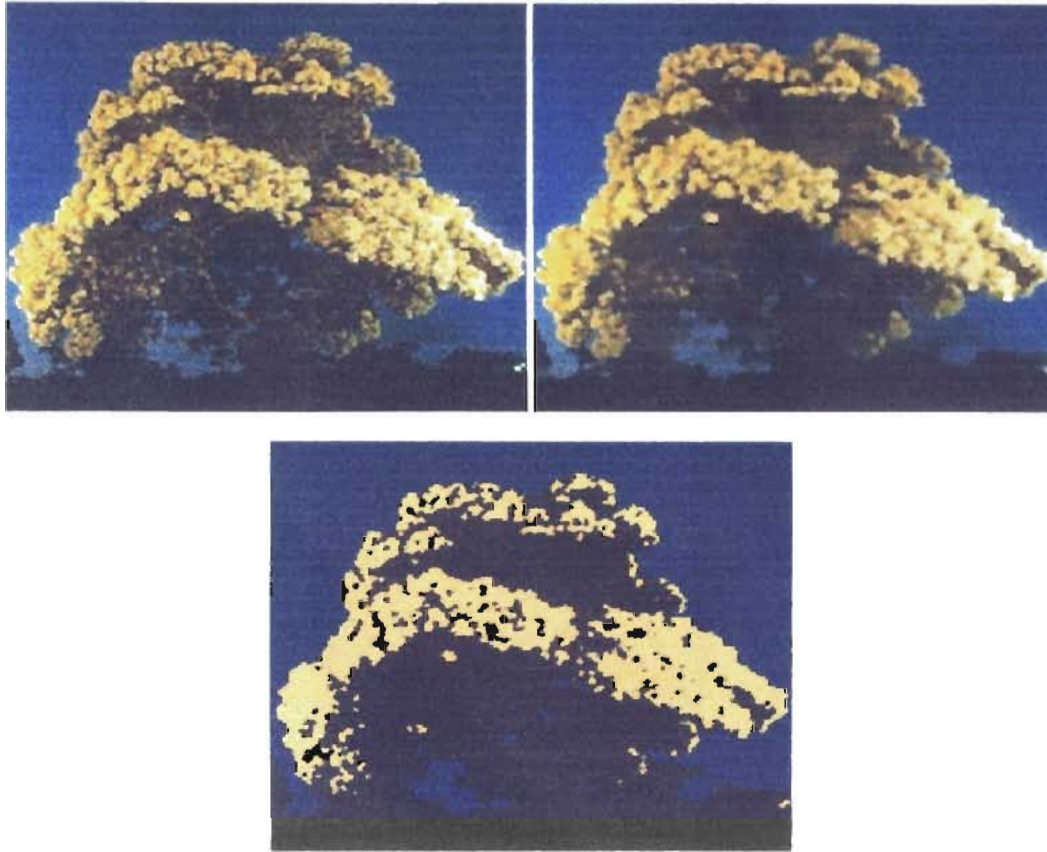
global coarse color quantization scheme used to reduce greatly the color depth in a color image.



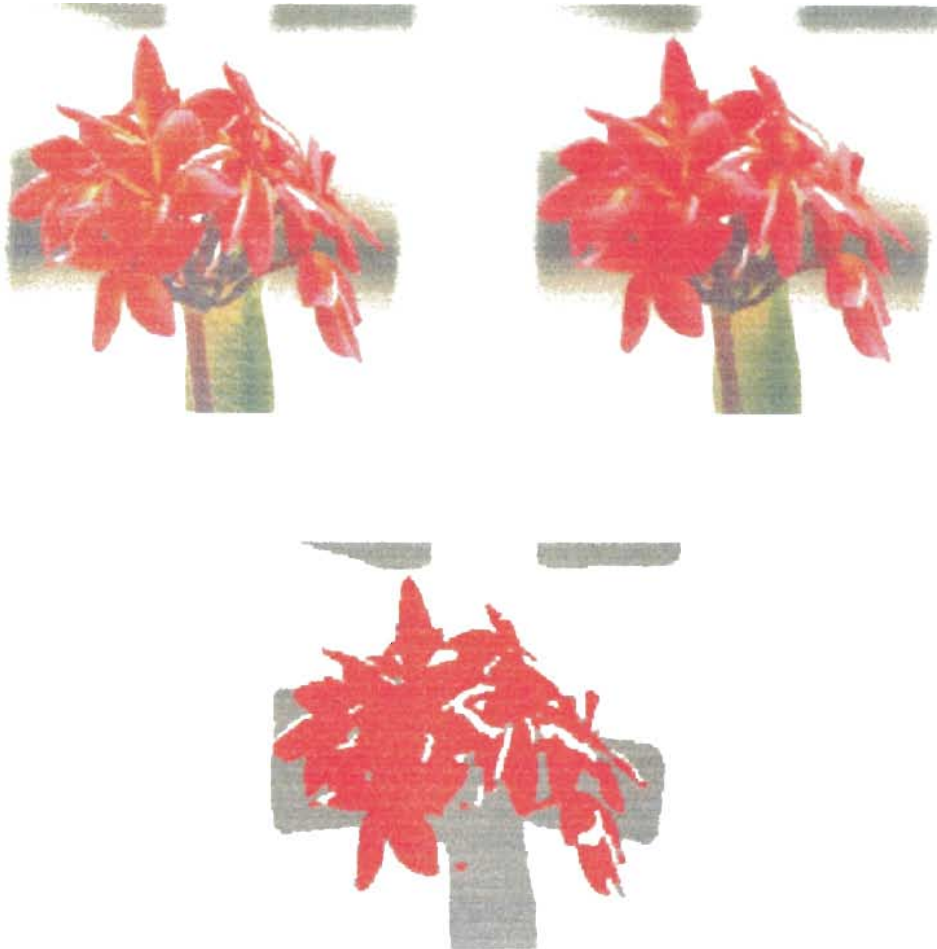
**Figure 17 a.** Original image, M-CAD result, global coarse color quantization (level of quantization=1)

Digitized by Sreeja Srinivasan, IIT Bombay





**Figure 17 b.** Original image, M-CAD result, global coarse color quantization (level of quantization=1)



**Figure 17 c.** Original image, M-CAD result, global coarse color quantization (level of quantization=1)

#### **4.2.3 Coarse color quantization for color feature extraction**

Traditionally color quantization research has focussed on reducing color depth for efficient TV/video transmission. A few researchers have focussed on using color quantization for the purposes of segmentation and CBR [23]. Hence traditionally color quantization has been defined as a technique used to reduce the bit rate by using

fewer bits. However our goal is to obtain an estimate of the globally/locally dominant colors, use them for similarity analysis in CBR.

For the purpose of color quantization we operate in the RGB color space. The color depth is reduced by clustering in the multidimensional color space. This particular clustering step requires as input the smoothed color image and the number of classes/clusters. The smoothed color image is generated using the M-CAD color smoothing scheme outlined earlier. The number of clusters  $C$  is specified as metadata for each image in our digital library. The clustering itself is achieved via the FCM technique described in detail in Chapter III. Thus each image is partitioned into a fixed number of clusters. The next step involves determining the top  $n$  3-tuples in each class in the classified image. i.e. 3-tuples with the highest frequency of occurrence. Each 3-tuple refers to a unique color in the multidimensional color space.  $n$  can be termed as the level of quantization. Typically in our experiments we use  $n=1,2,3$  in order to obtain a coarse color quantization. Given the original image  $\mathbf{I}_M$ , the M-CAD smoothed image  $S_{I_M}, \hat{S}_{I_M}$  the FCM result, the global color feature can be expressed as

$$C_x(f) = \sum_m \left\| \hat{S}_{I_M}(P_f) \right\|_m \quad (49)$$

where  $1 \leq f \leq C, 1 \leq m \leq n$ ,  $\left\| \hat{S}_{I_M}(P_f) \right\|_m$  refers to the  $m^{\text{th}}$  ranked 3-tuple among the pixels  $P_f$  (belonging to class  $f$ ) in the classified image  $\hat{S}_{I_M}$ , Thus  $C_x(f)$  records the top  $n$  3-tuples in each of the classes in the classified image. A threshold  $T$  is fixed as the minimum distance between any of the collected 3-tuples. This prevents recording of redundant colors in terms of perception (light green and lighter green). The global color feature records the dominant colors in the image by means of a coarse color quantization.

Search spaces in large digital image libraries can be cut down considerably by use of this global color feature. The local color feature is of two types. The first type of local color feature records the dominant 3-tuples (colors) in a segment in a manner similar to the computation of the global color feature. The second local color feature is the color probability density function (pdf) confined to the segment of interest. The levels of quantization are different for both the local features. The dominant local color feature has a high quantization level i.e.  $n=1,2,3$ . While the local color pdf has either a low quantization level or none at all, in hope of greater discrimination abilities. The dominant local color feature can be expressed as

$$C_l = \sum_m \|\hat{S}_{l,m}(P_l)\|_m \quad (50)$$

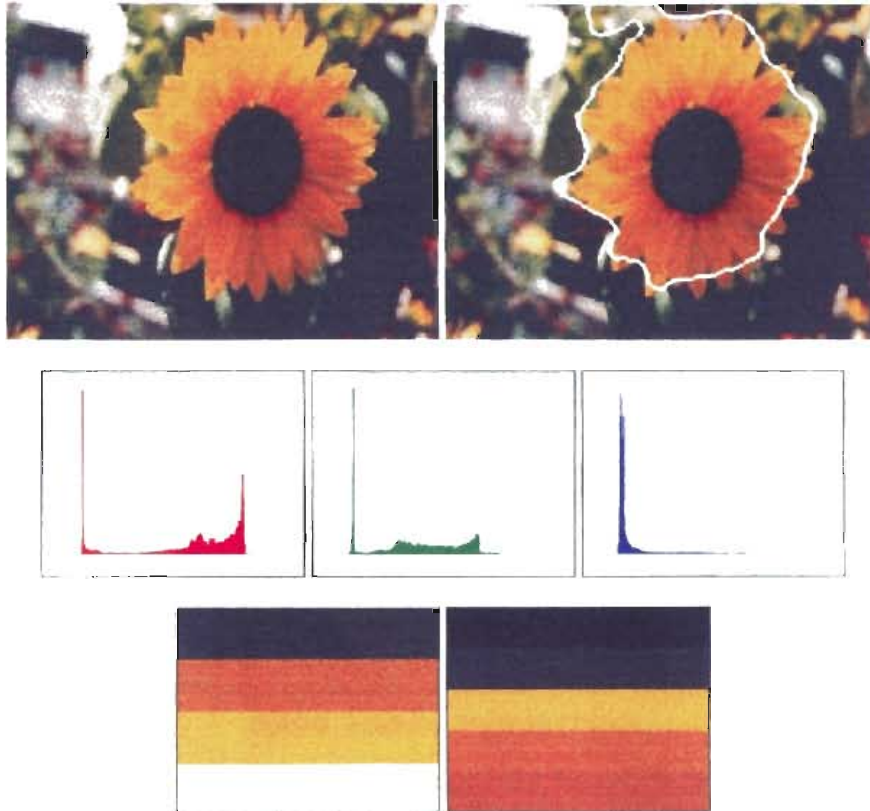
where  $1 \leq m \leq n$ ,  $\|\hat{S}_{l,m}(P_l)\|_m$  is the  $m^{\text{th}}$  highest (in terms of occurrence) 3-tuple among the pixels  $P_l$  (pixels in the segment  $l$ ) and thus  $C_l$  records the dominant 3-tuples(colors) in the segment  $l$ . The local color pdf can be written as

$$C_l(f) = |\hat{S}_{l,m}(P_l)|_f \quad (51)$$

where  $1 \leq f \leq f_{\max}$ ,  $f_{\max}$  is the maximum band intensity level,  $|\hat{S}_{l,m}(P_l)|_f$  is the cardinality of intensity level  $f$  among the pixels  $P_l$  confined to the segment  $l$  and is computed band-wise. Figure 17 also shows some results from the coarse color quantization scheme described here.

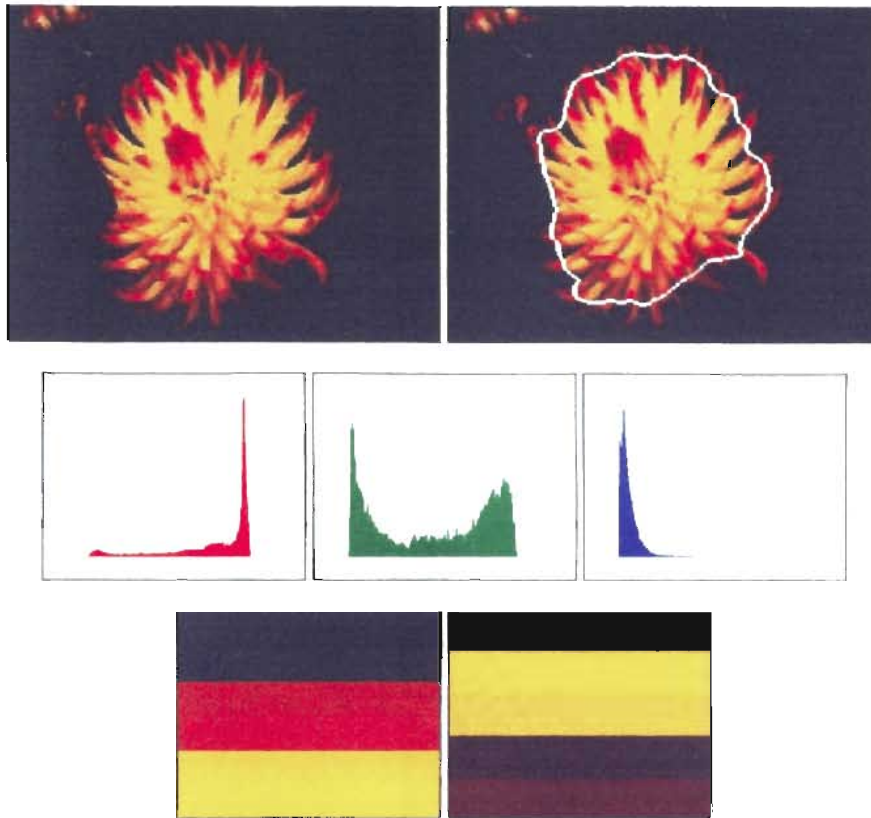
Thus we have demonstrated a subsystem for color feature extraction. Briefly summarizing the process of color feature extraction involves color smoothing using the M-CAD scheme to denoise images, quantizing the color depth in the smoothed

image for efficient global and local color feature extraction. Figures 18, 19 and 20 illustrate local and global color feature extraction.

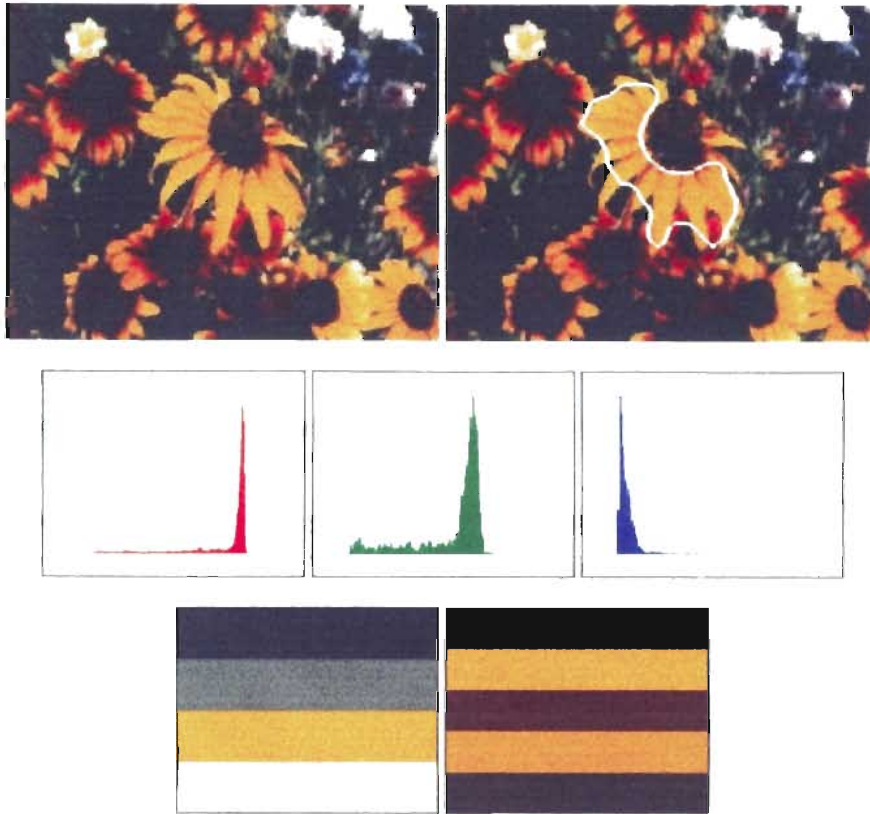


**Figure 18.** Original Image, highlighted segment, Local color pdf, Global dominant colors, Local dominant colors

Copyright © 2011, IGI Global. All rights reserved. This book is published by IGI Global.



**Figure 19.** Original Image, highlighted segment, Local color pdf, Global dominant colors, Local dominant colors



**Figure 20.** Original Image, highlighted segment, Local color pdf, Global dominant colors, Local dominant colors

### 4.3 Texture feature extraction

As described earlier, texture is a neighborhood property rather than a point property. The optimality of Gabor filters for textured region extraction has been described in great detail in chapter III. We propose to use a bank of Gabor filters for texture feature extraction. More specifically the Gabor responses of an image is used to compute its texture features. Equation 15-2 in chapter III gives the equation of the Gabor response of an image. The texture features defined are based on the mean, standard deviation and entropy of the Gabor responses. Intuitively these correspond to the strength of the Gabor response. A high strength would indicate the presence of texture at the

particular center frequency of the Gabor filter. At the local level these feature computations are confined to the extent of the segment. We term the two texture features extracted the statistic and stochastic texture features. The global statistic texture feature  $T_g$  can be expressed as

$$T_g = \mu_{I_h} + j\sigma_{I_h} \quad (52)$$

where the real part  $\mu_{I_h}$  is the mean of the magnitude of the Gabor response  $I_h$  while the imaginary part  $\sigma_{I_h}$  is the standard deviation of the magnitude of the Gabor response  $I_h$ .

Similarly the local statistic texture feature can be expressed as

$$T_l = \mu_{I_h(p_l)} + j\sigma_{I_h(p_l)} \quad (53)$$

where  $p_l$  refers to pixels confined to the extent of segment  $l$ . The statistic texture feature is expressed as a complex number for purpose of ease in similarity computation. The stochastic texture feature is the entropy of the Gabor response. The entropy of the Gabor response is a measure of its complexity, i.e. amount of information. The global stochastic texture feature can be written as

$$T_g = - \sum_{k=0}^{G-1} p_{I_h}(k) \log_2 [p_{I_h}(k)] \quad (54)$$

where  $G$  is the maximum number of gray levels in the image luminance map (usually 256),  $p_{I_h}(k)$  is the probability of gray level  $k$ , which numerically is the reciprocal of the cardinality of its occurrence. Similarly the local stochastic texture feature can be written as

$$T_l = - \sum_{k=0}^{G-1} p_{I_h(p_l)}(k) \log_2 [p_{I_h(p_l)}(k)] \quad (55)$$

where as before  $p_l$  consists of the pixels in the segment  $l$  of the image  $\mathbf{I}$ .



These statistic and stochastic texture features are extracted for the various Gabor responses derived via the Gabor filterbank.

#### 4.4 Shape feature extraction

Shape is another important cognitive local property for CBR. Classification and recognition of 2-D shapes regardless of their position, size and orientation is an important problem in areas of application such as content based retrieval (CBR), target recognition and scene analysis. The various methods found in the literature for 2D shape description are:

1. Fourier analysis of some function derived from the boundary (Fourier Shape Descriptors (FSDs)) [33] [44] [6].
2. Global statistical approaches based on methods of moments [11] [46].
3. Autoregressive (AR) models. [25] [53].

For real-time applications and processing involving large image databases (1) is very attractive due to the usage of the Fast Fourier Transform (FFT), moreover FSDs were found to have better shape classification results when compared to other methodologies mentioned above [33].

The FSDs make use of a basic concept that the position of a point along a closed contour is a periodic function. Traditionally Fourier series have been used to approximate contours. The resolution of the contour approximation is determined by the number of terms in the Fourier series. However we propose to use the geometric transform invariant

terms of the Fourier series as shape features. Before going on to the formulation of the shape features we describe a few boundary models for contour modeling.

#### 4.4.1 Boundary Models

A boundary is considered as a closed sequence of successive boundary pixel coordinates  $(x_i, y_i)$ . The various representations derived from the boundary coordinate chain are

1. Complex coordinate function
2. Centroidal distance function
3. Closed boundary curvature function

A complex coordinate function is simply the coordinates of boundary pixels in an object centered coordinate system, represented as complex numbers:

$$z_i = (x_i - x_c) + j(y_i - y_c) \quad (56)$$

where  $(x_i, y_i)$  are the coordinates of the contour points and  $(x_c, y_c)$  is the centroid of the closed contour.

The centroidal distance function expresses the distance of boundary points from the centroid  $(x_c, y_c)$  of the object.

$$r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad (57)$$

A curvature function is a function of angular changes of a boundary tangent. Curvature at a boundary point  $i$  can be expressed as a differentiation of successive tangent values calculated in a window  $w$ .

$$C_i = \tan^{-1} \left[ \frac{y_i - y_{i-w}}{x_i - x_{i-w}} \right] - \tan^{-1} \left[ \frac{y_{i-1} - y_{i-1-w}}{x_{i-1} - x_{i-1-w}} \right] \quad (58)$$

where  $I=(0,1,\dots,N-1)$ ,  $N$ = number of boundary points. Note that all the above three models are periodic since the functions are closed. All models are w.r.t. an object centered frame of reference hence the functions are translation invariant i.e. insensitive to the placement of the object in the image plane.

Fourier transformation of a boundary signal function generates a complete set of complex numbers; the FSDs. To facilitate the use of the FFT the boundary functions are sampled to  $2^n$  samples. These descriptors represent the shape of the object in the frequency domain. The lower frequency descriptors contain information about the general shape, while the higher frequency descriptors contain information about the smaller details. The sub-sampled boundary can be expressed as

$$u(n + \Delta n) = x(n + \Delta n) + jy(n + \Delta n) \quad (59)$$

where  $n=0,\dots,N-1$ ,  $N$  = number of points on the boundary,  $\Delta n$  is the sub-sampling step size. For a closed boundary periodic with period  $N$ , the discrete Fourier transform (DFT) representation is as follows

$$u(n) = \frac{1}{N} \sum_{k=0}^{N-1} a(k) e^{(j2\pi kn/N)} \quad 0 \leq n \leq N-1 \quad (60)$$

$$\text{where } a(k) = \sum_{n=0}^{N-1} u(n) e^{(-j2\pi kn/N)} \quad 0 \leq k \leq N-1$$

The complex coefficients  $a(k)$  are referred to as the FSDs of the boundary.

The main issues for FSDs w.r.t. CBR are invariance w.r.t.

1. Translation
2. Scaling
3. Starting Point (Order of contour traversal)
4. Rotation
5. Affine Transformation

Table 1 gives the various properties of FSDs under various geometric transformations.

An object centered coordinate system eliminates the effects of translation.

TRANSFORMATION	BOUNDARY	FOURIER DESCRIPTORS
Identity	$U(n)$	$a(k)$
Translation	$U(n)+u_0$	$a(k)+u_0\delta(k)$
Scaling	$\alpha u(n)$	$\alpha a(k)$
Starting Point( Contour Trace)	$U(n-n_0)$	$a(k) e^{-j2\pi n_0 k/N}$
Rotation	$U(n) e^{j\theta_n}$	$a(k) e^{j\theta_n}$

**Table 1.** Properties of FDS under various geometric transformations

From the above table it can be seen that the absolute value of the FSDs i.e.  $|a(k)|$  is invariant to starting point, and rotation, while  $\frac{a(k)}{|a(k)|}$  is invariant to scaling. However the actual FSD formulation differs based on the boundary models. More details regarding appropriate weighting of Fourier coefficients for geometric transform invariance can be found in [33].

#### 4.4.2 Contour FSDs (Complex Coordinate boundary function)

The contour Fourier descriptors are obtained via FFT transformation of the complex coordinate boundary representation. The descriptors from the negative frequency axis are essential as the original function is complex in nature. The dc component depends only on the position of the shape and hence is discarded. The absolute value of the first non-zero frequency component is used to scale the absolute values of the other descriptors to obtain the features. The feature vector for the contour FSDs is

$$S_l = \left[ \frac{|F_{-(N/2-1)}|}{|F_1|} \dots \frac{|F_{-1}|}{|F_1|} \frac{|F_2|}{|F_1|} \dots \frac{|F_{N/2}|}{|F_1|} \right]^T \quad (62)$$

where  $F_l$  denotes the  $l$ th component in the Fourier spectrum and  $l$  denotes the segment  $l$ .

#### 4.4.3 Radius FSDs (Centroidal distance function)

The centroidal distance function is transformed by the FFT algorithm to yield the radius FSDs. Because of the real nature of the centroidal distance function the FSDs are the same in the positive and negative frequency axes. The normalized FSDs are obtained by dividing the absolute values of the positive frequency component by the absolute value of the DC component. The dc component is discarded as it depends only on the position of the shape. The feature vector for radius FSDs is

$$S_l = \left[ \frac{|F_1|}{|F_0|} \dots \frac{|F_{N/2}|}{|F_0|} \right]^T \quad (63)$$

where  $F_i$  denotes the  $i$ th component in the Fourier spectrum and  $l$  denotes the segment  $l$ .

#### 4.4.4 Curvature FSDs (Curvature boundary function)

The curvature FSDs are similar to the radius FSDs given above except that the curvature boundary function is used instead of the centroidal distance function. The FSDs for curvature FSDs is similar to those of the radius FSDs given in equation 63.

#### 4.4.5 Affine invariant FSDs

A transformation is called affine when any straight line in the first image is mapped onto a straight line in the second image, while parallelism is preserved. Affine transformations can be decomposed into a linear (matrix) transformation and a translation. Examples of non-rigid affine transformations are both uniform and non-uniform scaling, and shearing. To achieve affine invariance the FSDs are subjected to a normalization procedure to remove the effects of affine transformations. For affine invariance the complex coordinate boundary function is used to formulate the FSDs. Affine invariance is achieved by applying a normalization procedure to remove the effects of affine transformations [6]. The affine invariant FSDs are

$$S_i = \left[ A_{-(N-2)} \dots A_{-2} \| A_1 \dots A_{N-2} \right]^T \quad (64-1)$$

where the  $A_i$  are computed from the equation below.

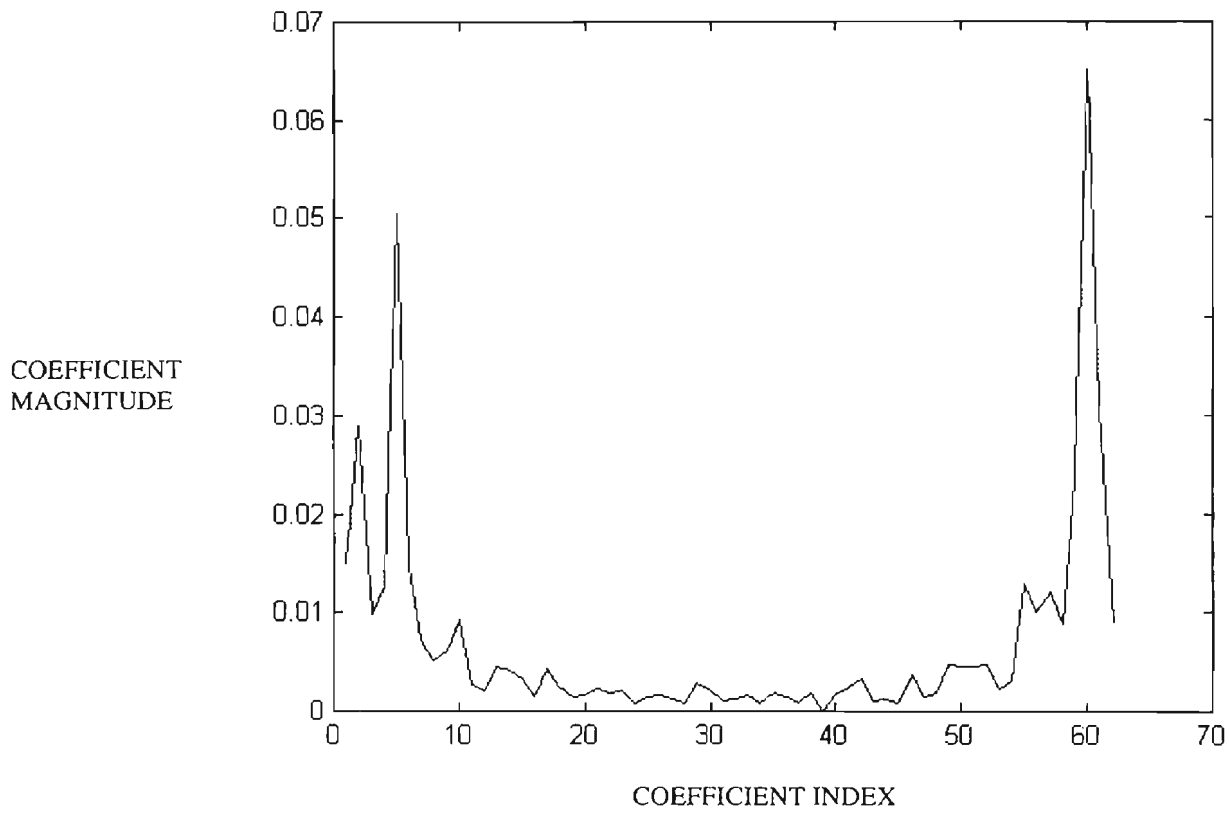
$$A_i = \frac{U_i V_p^* - V_i U_p^*}{U_p V_p^* - V_p U_p^*} \quad i = -\left(\frac{N}{2}-1\right), \dots, \frac{N}{2}, \quad (64-2)$$

where  $U$  = Fourier transformation of the  $x$  components of contour points,  $V$  = Fourier transformation of the  $y$  components of contour points and  $p$  is a constant (e.g. 1)

These affine invariant FSDs can prove to be particularly useful in CBR for remotely sensed imagery [48]. The affine invariant shape feature has been incorporated with a view of matching remote sensing images of overlapping land areas (w.r.t. a query image) taken at different satellite positions (inferring a change in viewpoints, *viz* affine transformations). Figure 21 illustrates FSDs for a given segment shape.



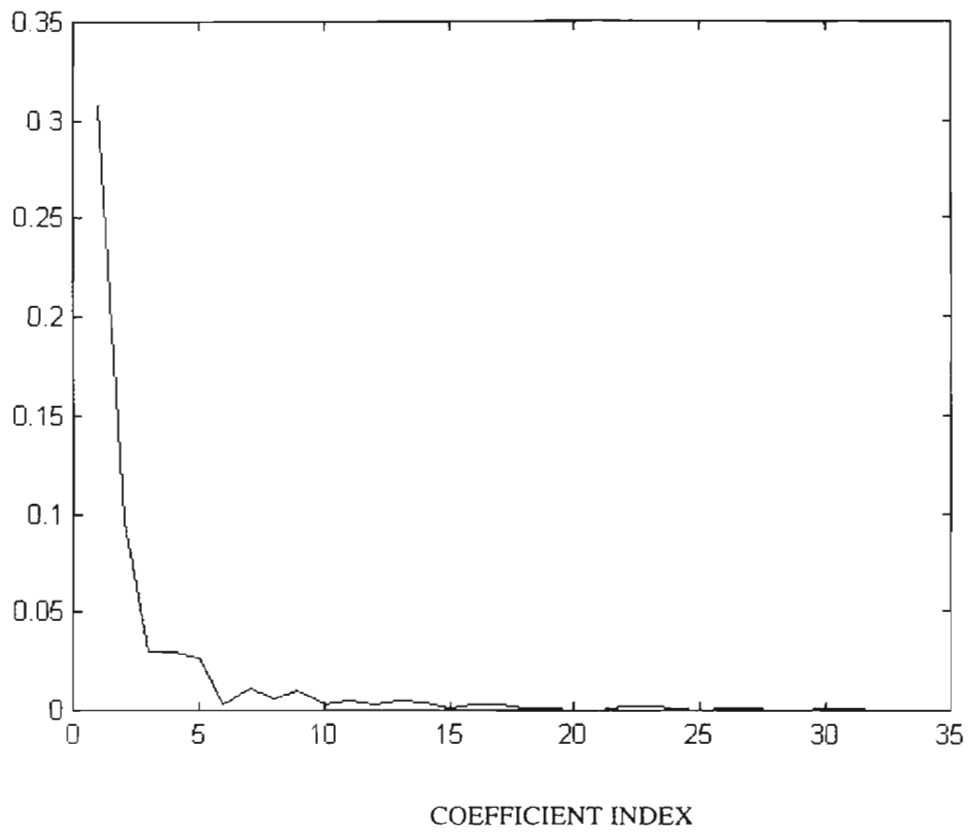
**Figure 21 a .** Original image, highlighted segment, Segment contour



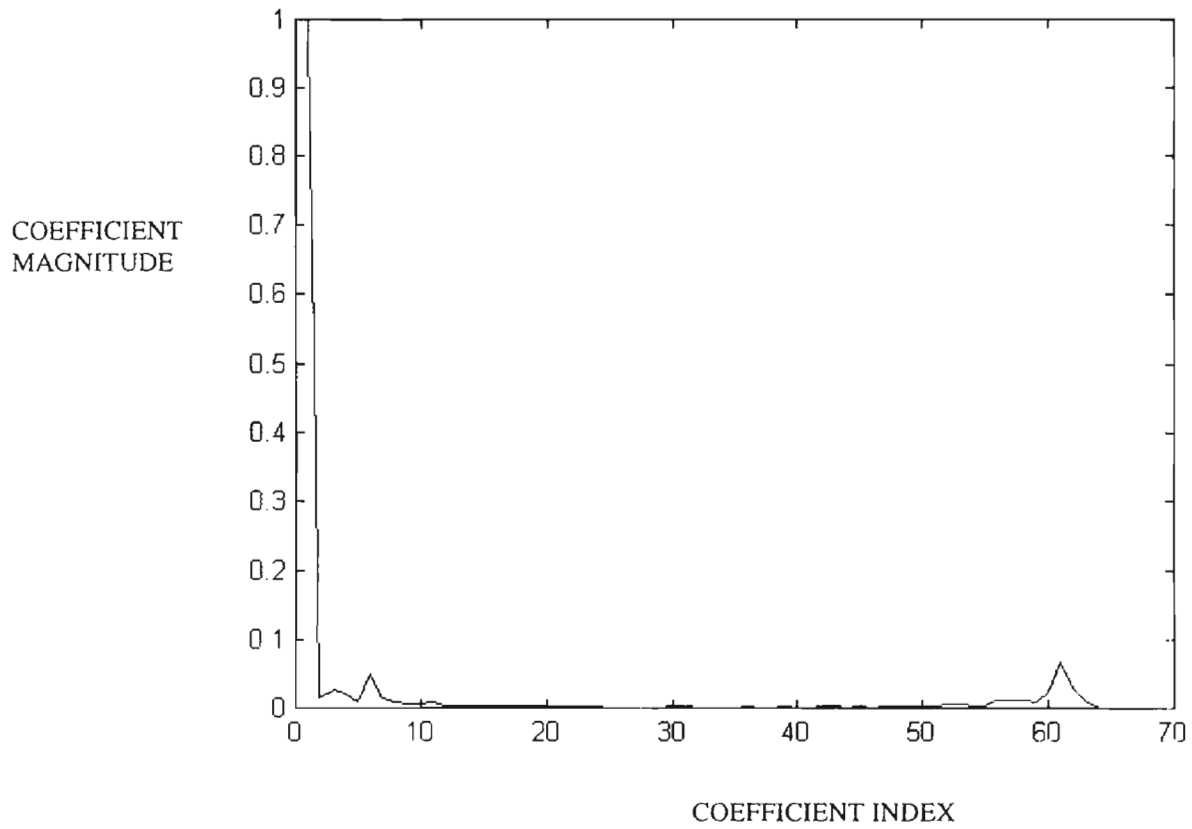
**Figure 21 b.** Contour FSDs for segment in figure 21 a.



COEFFICIENT  
MAGNITUDE



**Figure 21 c.** Radius FSDs for segment in figure 21 a.



**Figure 21 d.** Affine invariant FSDs for segment in figure 21 a.

## 4.5 Chapter Summary

Summarizing briefly, we have demonstrated various frameworks for feature extraction for CBR. An M-CAD scheme for color image smoothing has been outlined. The smoothed color images are used for computation of global and local color features. The Gabor responses from the Gabor filterbank are used to derive statistic and stochastic texture features. The shapes of local segments are encoded via FSDs. This involves Fourier transformation of the boundary representations. The number of points on the contour are subsampled to  $2^n$  (usually 64) points in order to facilitate usage of the FFT algorithm for FSD computation. The invariance of FSDs to various geometric transformations is achieved by appropriate coefficient weighting.

## CHAPTER V

### SIMILARITY ANALYSIS

#### 5.1 Introduction

The concept of CBR depends upon the comparison of two images and their degree of similarity. Traditionally researchers have judged similarity/dissimilarity by a similarity/dissimilarity metric. We judge similarity by means of the magnitude of feature errors. We use the term similarity/similarity measure in our experiments for the purpose of coherence. Hence by our definition of similarity we would ideally like to obtain a low similarity metric for images similar to the query image and a high similarity metric for images relatively less similar to the query image. The next logical step lies in determining the parameters based on which the similarity metric can be computed. Irving Biedermann in [13] proposed a recognition-by-components theory. He theorized that an objects is analyzed as a whole constituted of individual sub-objects. This can be construed as analogous to a segmentation. Thus we have adopted a segment based matching strategy for CBR. A scale based segmentation technique has been demonstrated for general imagery. A large number of segments would increase the number of comparisons for similarity analysis. The segmentation technique demonstrated here provides only significant segments. The significance of segment is assessed via a unique LLGG metric

described in chapter II. The reduced number of segments decreases number of comparisons to be made for segment based similarity analysis.

We have experimented with three types of segment similarity assessment techniques: The first of these is a one to one technique, wherein a query image and query segment are specified for as part of query formulation. For a given image the best matching segment contributes to the similarity metric for that image. In other words one to one matching is analogous to finding the best matching segment w.r.t. to a given query segment. The second type of matching technique is a one to many technique. The similarity is assessed based on all the constituent segments in the other images. The similarity metric for a given image would be computed by aggregation of all its segment feature errors w.r.t. the query segment feature. The third type of matching is a many to many matching technique. As the name indicates a similarity metric for an image is assessed by aggregation of feature errors of all possible segment pairs between the query image and the target image(s). We now elaborate on the actual numeric computation of feature based similarity analysis.

## 5.2 Color

Given two global dominant color features  $C_{g1}$  and  $C_{g2}$ , the global color similarity metric can be written as

$$\Delta C_g = \sum_{m_1, m_2} \sqrt{(r_{m_1g1} - r_{m_2g2})^2 + (g_{m_1g1} - g_{m_2g2})^2 + (b_{m_1g1} - b_{m_2g2})^2} \quad (65)$$

where  $1 \leq m_1 \leq n_1, 1 \leq m_2 \leq n_2$ ,  $n_1, n_2$  are the levels of color quantization described earlier.

Thus  $\Delta C_g$  is a measure of the Euclidean distance between the dominant colors

$$\left( (r_{m_1 g_1}, g_{m_1 g_1}, b_{m_1 g_1}), (r_{m_2 g_2}, g_{m_2 g_2}, b_{m_2 g_2}) \right).$$

Similarly given two local dominant color features  $C_{l1}$  and  $C_{l2}$ , the local dominant color similarity metric can be written as

$$\Delta C_l = \sum_{m_1, m_2} \sqrt{(r_{m_1 l1} - r_{m_2 l2})^2 + (g_{m_1 l1} - g_{m_2 l2})^2 + (b_{m_1 l1} - b_{m_2 l2})^2} \quad (66)$$

where  $1 \leq m_1 \leq n_1, 1 \leq m_2 \leq n_2$ ,  $n_1, n_2$  are the levels of color quantization described earlier,

$\Delta C_l$  is a measure of the Euclidean distance between the dominant local colors

$$\left( (r_{m_1 l1}, g_{m_1 l1}, b_{m_1 l1}), (r_{m_2 l2}, g_{m_2 l2}, b_{m_2 l2}) \right).$$

Given two local color pdfs  $C_{l1}(f)$  and  $C_{l2}(f)$ , the local color pdf similarity metric can be written as

$$\Delta C_l = \frac{1}{f_{\max}} \sum_f (C_{l1}(f) - C_{l2}(f))^2 \quad (67)$$

where  $f_{\max}$  is the maximum intensity level that  $f$  can attain. In addition to the color similarity metric mentioned above matching strategies similar to the segment matching strategies can be applied. For example a one to one color tuple matching technique would ensure only similar color tuples contribute to color feature errors.

### 5.3 Texture

Given two global statistic texture features  $T_{g1}$  and  $T_{g2}$ , the global statistic texture feature similarity metric can be written as

$$\Delta T_g = |T_{g1} - T_{g2}|^2 \quad (68)$$

where  $\Delta T_g$  is the absolute value squared of the error between the complex texture features.

Similarly given two global stochastic texture features  $T_{s1}$  and  $T_{s2}$ , the global stochastic texture feature similarity metric can be written as

$$\Delta T_g = (T_{s1} - T_{s2})^2 \quad (69)$$

Since the local and global texture features are essentially the same numeric concept, the similarity metric can be computed in a similar manner. These texture similarity metrics are computed over all Gabor responses and are summed up to give the final texture similarity metric. The total global texture similarity metric can be expressed as

$$\Delta T_{gf} = \sum_n \Delta T_{ng} \quad (70)$$

where  $n$  is the number of filters in the Gabor filterbank. The local texture similarity metric can be expressed similarly.

### 5.4 Shape

Various researchers have focussed on using a reduced number of coefficients in the FSDs for shape similarity computation. However we have not attempted any such

methods since it goes beyond the scope of our research. Given two FSDs  $F_1$  and  $F_2$ , the shape similarity metric can be expressed as

$$\Delta F = \sum_n (F_1(n) - F_2(n))^2 \quad (71)$$

where  $n$  is the number of points in the sub-sampled contour. The above expression can be used for all types of FSDs described earlier.

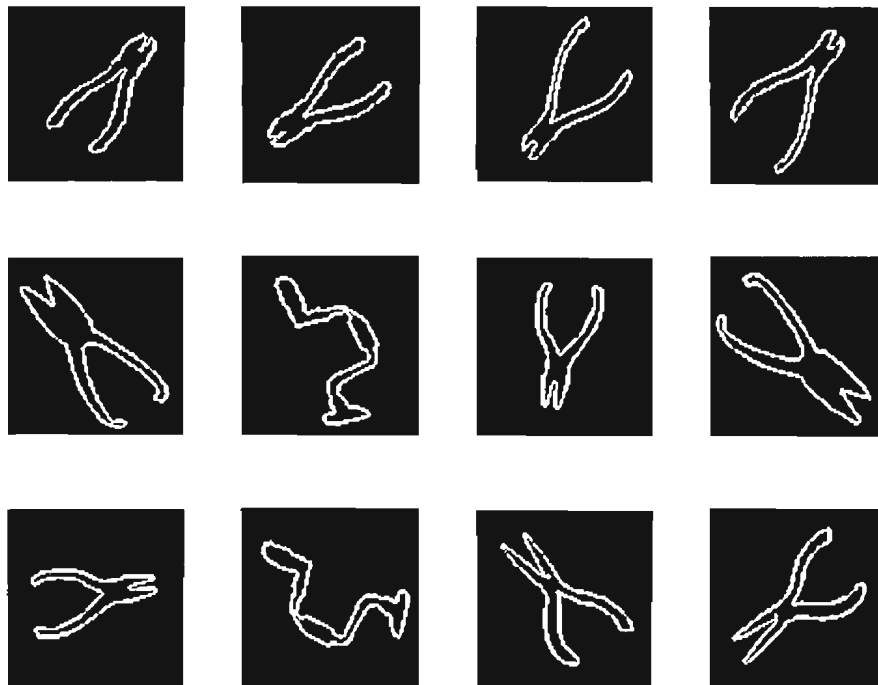
### 5.5 A Logical hierarchical framework for similarity analysis

As mentioned briefly before a purely segment based matching strategy would be inefficient for large digital image libraries. We propose to use a judicious combination of global and local feature to achieve a tradeoff between quality of retrieval and retrieval time. We use global features corresponding to color and texture to trim the initial search space. This reduced search space is input to the next stage of matching. We term this matching strategy the hierarchical framework for similarity analysis. The number of stages in this hierarchical framework is a user specified parameter in our experiments. The user also specifies the various features and cutoff parameters to be used in hierarchical matching.

We have formulated and described various feature similarity metrics. Combination of these various feature similarity metrics is an open research issue. The issues to be addressed are not just numeric weighting of the various features, but fundamental issues like the combination of diverse features like color, texture and shape. Perhaps psychological understanding of how the human brain perceives similarity as a combination of diverse features will provide an answer to this fundamental research issue.



We propose a logical hierarchical framework for combining these diverse features. The logical hierarchical framework for similarity analysis consists of multiple stages. The various stage parameters are specified by the user. Matches from one stage are ordered and a cutoff is imposed on them. This reduced search space is input to the next stage. Figure 22 shows query and retrieval by shape features in a database of tools. Figure 23 shows query and retrieval by shape features in a database of natural and flower imagery. Figure 24 shows differences in retrieval quality based on global histogram, global color, local color matching. Figures 25 and 26 illustrate differences in retrieval quality based on global color pdf and local color pdf. Figures 27 and 28 show the benefits of hierarchical matching in CBR.



**Figure 22.** Query and shape retrieval results via Fourier shape descriptors  
Ordering – Best to worst –Top to bottom – Left to right



**Figure 23.** Query and best results by local shape features  
 Ordering – Best to worst –Top to bottom – Left to right



**Figure 24 a.** Query and results by global histogram matching



**Figure 24 b.** Query and results by global dominant colors



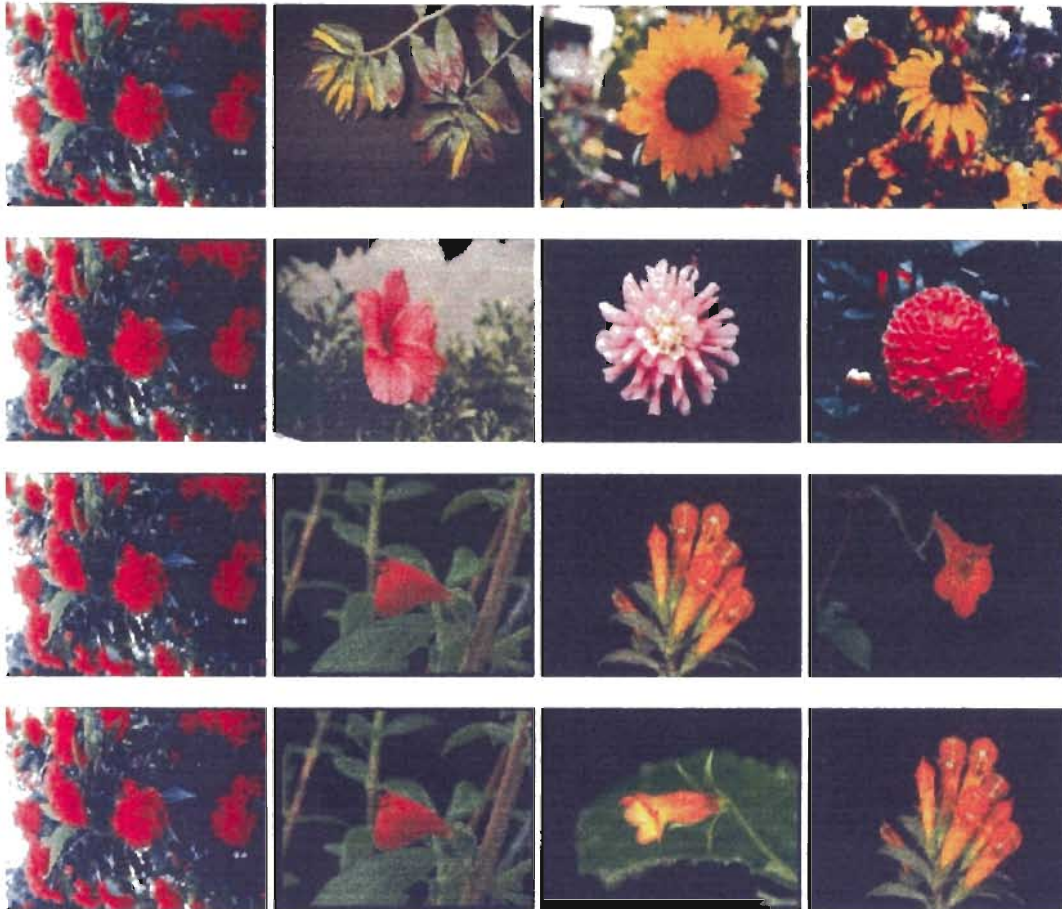
**Figure 24 c.** Query and best results by local dominant colors matching





**Figure 25.** Comparative analysis of retrieval performances of global color pdf and Local color pdf

- Row one – Query and best matches – Global color histogram matching
- Row two – Query and best matches – One to One Local color pdf matching
- Row three– Query and best matches – One to Many Local color pdf matching
- Row Four – Query and best matches – Many to Many Local color pdf matching



**Figure 26.** Comparative analysis of retrieval performances of global color pdf and Local color pdf

Row one – Query and best matches – Global color histogram matching

Row two – Query and best matches – One to One Local color pdf matching

Row three– Query and best matches – One to Many Local color pdf matching

Row Four – Query and best matches – Many to Many Local color pdf matching



**Figure 27 a.** Query and results by global color histogram matching



**Figure 27 b.** Query and results by local color pdf (One to One matching)





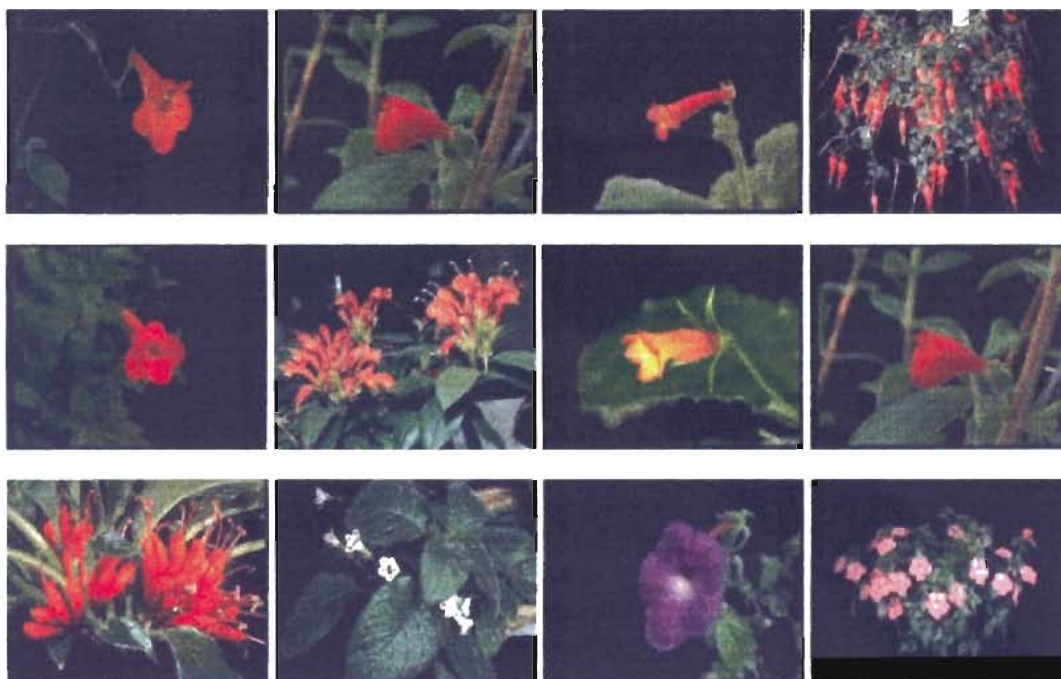
**Figure 27 c.** Query and results by Hierarchical matching  
 Stage one – Global dominant colors  
 Stage two – Local color pdf



**Figure 28 a.** Query and results by Global histogram matching



**Figure 28 b.** Query and results by Local color pdf (One to One)



**Figure 28 c.** Query and results by hierarchical matching  
 Stage one – Global dominant colors  
 Stage two – Local color pdf



## CHAPTER VI

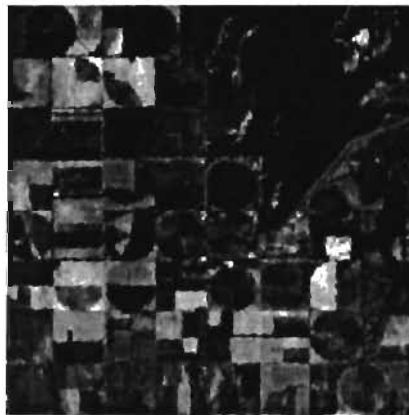
### APPLICATIONS

#### 6.1 Introduction

We have demonstrated a segmentation based CBR engine for general digital image libraries. This is a broad definition of the specific CBR applications. CBR applications can be varying in nature such as Internet applications, geographic information systems (GIS) and digital image libraries. In general the possible applications of CBR include digital libraries, medical images, GIS , and consumer applications. [22] categorizes these applications into three main models: 1. The Consumer Model 2. The Video-on –Demand model and 3. The Digital library model. [22] details the various specifics of these models and concludes that the digital library model holds the best prospect in terms of viability. Most of our intended applications chiefly fall under the realm of digital libraries. We have already demonstrated a CBR scheme for general digital imagery. In this chapter we outline an extension of our CBR scheme to remotely sensed imagery and elaborate on a novel application of CBR namely automated inspection.

## 6.2 CBR for Remotely sensed imagery

Due to increasing applications of remotely sensed imagery there are been an increase in the number of satellites, available bandwidth and remotely sensed image libraries. Remote sensing data are finding applications in many diverse areas including agriculture, meteorology, geology and urban planning. We use the LANDSAT TM imagery to demonstrate our CBR engine for remotely sensed imagery. Figure 29 shows a sample band 3 LANDSAT TM image. It can be seen that the figure is rich in texture and contains circularly shaped objects.



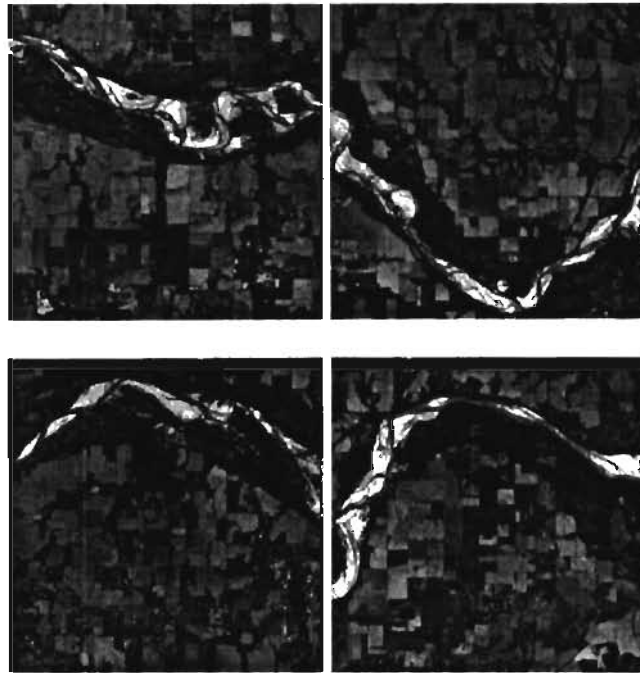
**Figure 29.** A sample LANDSA TM band 3 image

This circular shape is due to the center pivot irrigation system (CPIS), on the ground. Hence we are well-motivated in using shape and texture features for retrieval. Texture provides a rich description of the type of land (irrigated, unirrigated, urban). The shape feature helps in distinguishing between various objects like square fields, circular CPIS fields, lakes and river banks. Figure 30 shows some sample texture feature based retrieval results from a remotely sensed imagery database. These features could help determining areas of similar irrigation, vegetation patterns. In short the potential applications can be

classified under precision agriculture. One problem researchers have faced with remotely sensed imagery is retrieval of cloud free imagery for analysis. Such a tool would have applications in studying vegetation patterns and soil moisture indices. The most attractive part of the remote sensing application is the availability of means to verify results. The Oklahoma mesonet for example provides ground truth for soil moisture images. This ground truth of soil moisture can drive the segmentation process in terms of scale. Also this ground truth can be used in conjunction with the semantic segmentation to provide effective retrieval results. Such a CBR prototype can be used to investigate the effectiveness/validity of various soil moisture models developed. Thus potential applications of the remote sensing CBR engine include precision agriculture, hydrology, meteorology and retrieval of cloud-free imagery for image analysis. More details regarding this remote sensing application can be found in [48].



**Figure 30 a.** CPIS Query Image, Best matches – in decreasing order of similarity (left to right, top to bottom)



**Figure 30 b.** Riverbank query image, Best matches in decreasing order of similarity (left to right, top to bottom)

## **6.3 CBR for automated inspection of circuit boards**

### **6.3.1 Overview**

Traditional applications of CBR have focussed strictly on digital image libraries. We have investigated a real world application namely automated inspection. We provide a brief background of the preliminary work done on the inspection system and then elaborate on our CBR application for automated inspection.

Traditional testing of circuit boards has been via electrical tests. These tests are limited in nature given the varied number of defects known to occur on circuit boards i.e. open and short circuits. Hence at present technicians spend a considerable amount of time probing circuit boards that fail conventional electrical tests. It is our goal

to reduce this waste of technician effort. As part of our inspection system, the circuit boards are imaged by an infrared digital imaging system. The thermal signature from the operating chips is captured by an infrared video sequence. All circuit boards that fail electrical tests are designated faulty circuit boards (FCB's) while the others are designated known good boards (KGB's). Figure 31 highlights a FCB w.r.t. to a KGB. The proposed CBR system is intended to be an integral part of the overall inspection system. Presently data from the last frame of the video sequence is used for CBR, as the chips have reached thermal equilibrium. Given a query in the form of specific intensity features of a board image, the CBR system retrieves similar images from the board image database.

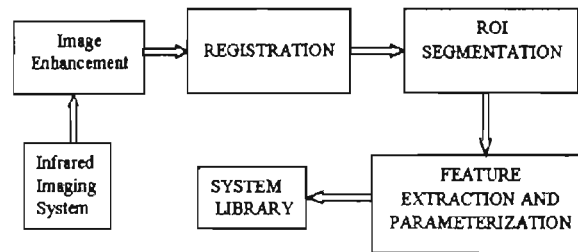


**Figure 31.** Highlighting the difference between a bad board and a good "reference" board

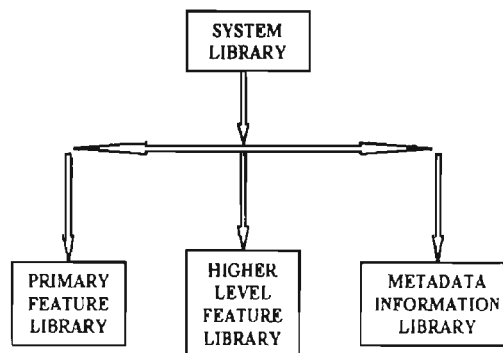
The ultimate goal of this CBR system is to retrieve similar FCB's so that they can be repaired simultaneously.

The proposed CBR system can be broken down into two parts – the actual search engine and the off-line processing that generates the system libraries. Figure 32 shows the outline of the CBR system. We first explain the off-line processing of the inspection system (previous work) and the consequent generation of the system library.

Then we elaborate on the methodology for formulation of higher level features to be used in CBR. Experimental results that demonstrate the successful application of CBR to manufacturing are given. Figure 33 shows the configuration of the system library.



**Figure 32.** Off-line processing

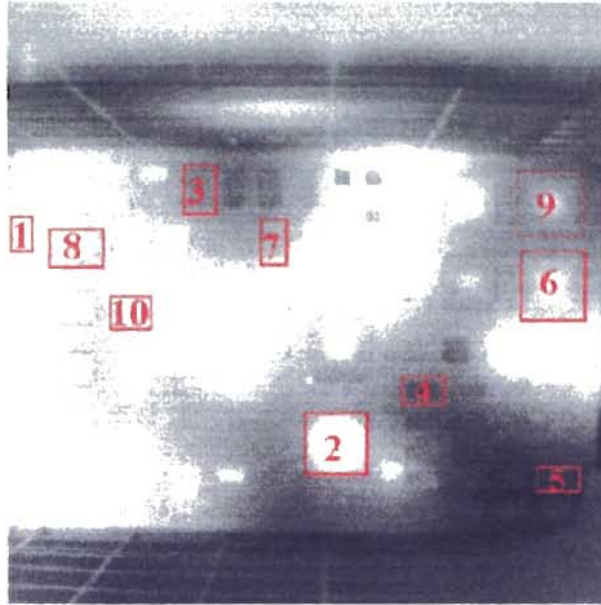


**Figure 33.** Configuration of system library

### 6.3.2 Preprocessing and primary feature extraction

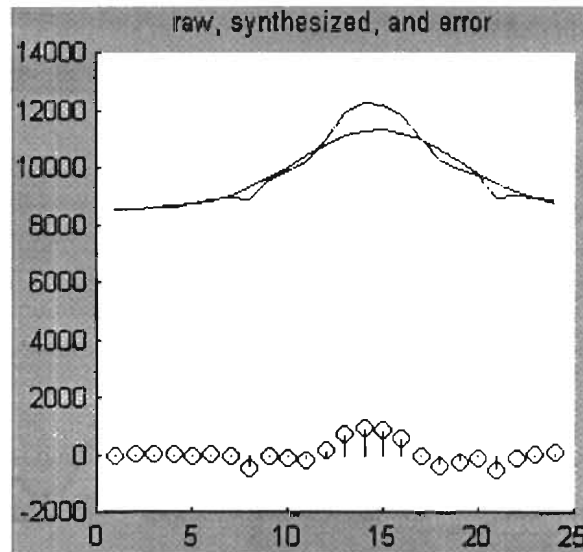
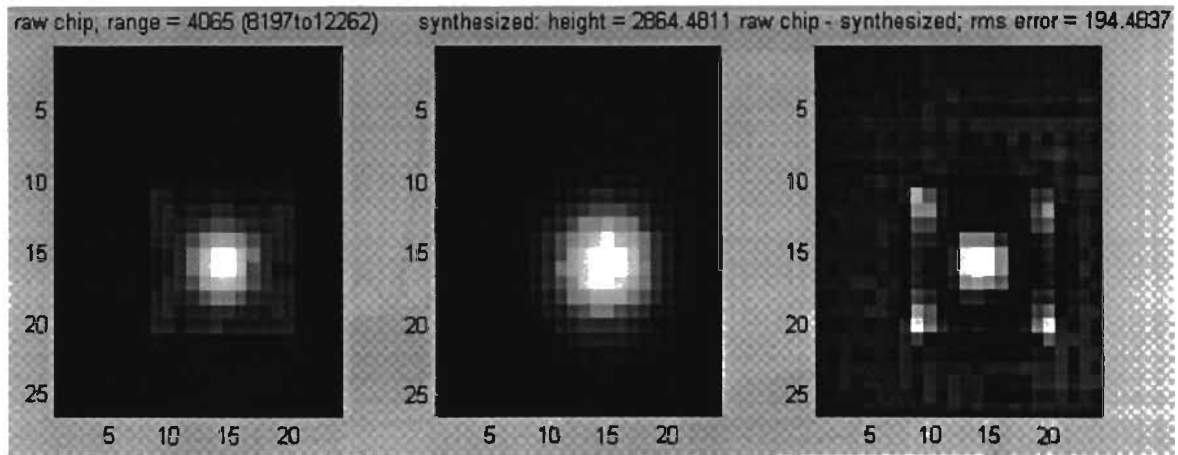
Each of the digital image sequences is first subjected to an image enhancement routine by which image contrast is improved. Image registration is performed to remove the effects of translation and rotation during the imaging process. Various regions of interest (ROI's) are identified by an image segmentation routine (see Figure 34). The ROI template delineates regions corresponding to important integrated

circuit packages on the board. Feature extraction through the video sequence is performed within image segments specified by the ROI template.



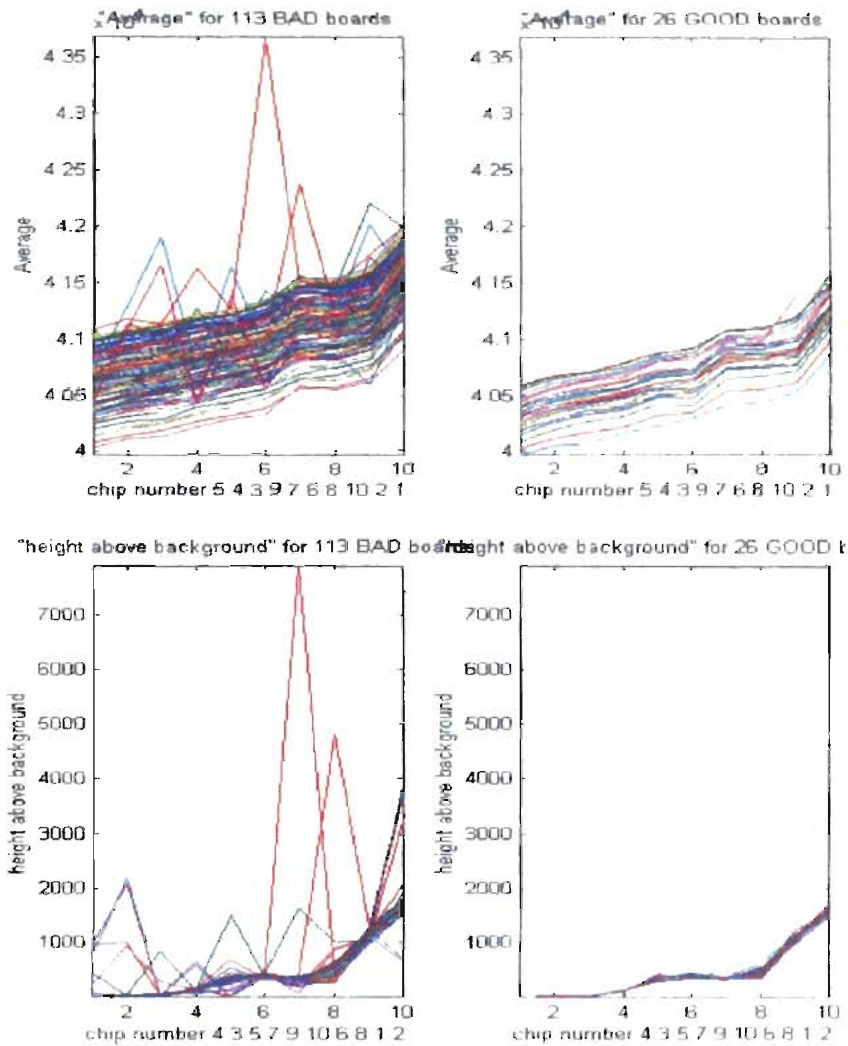
**Figure 34.** A segmentation template with ROI's (chips) numbered

Some of the primary features extracted are the average intensity of chip, the maximum intensity of chip, the minimum intensity of chip, image granulometry and the *chip isolation intensity*. The chip isolation intensity is the height of a Gaussian model fit to the 2-D intensity signature of a particular chip. Figure 35 illustrates this concept of the Gaussian chip model. This chip isolation intensity is used to eliminate the effects of background temperature on various primary features. Figure 36 illustrates the usage of the chip isolation intensity. More details regarding the elimination of background temperature can be found in [49]. In addition to these primary features, each circuit board in the sequence database library has some associated *metadata*. The purpose of using metadata in CBR is to facilitate rapid retrieval of possible matches. In this application the metadata are quite limited. We are given metadata that reveal the class of a board and its type (KGB/FCB).



**Figure 35.** Formulating a Gaussian model of intensity for a chip

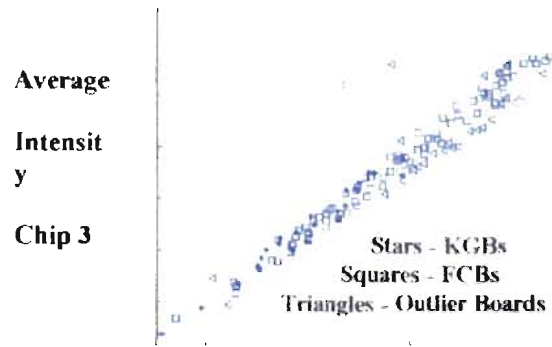




**Figure 36.** Eliminating the effects of background temperature via image statistics (chip model)

### 6.3.3 Formulation of Higher Level Features

The video feature extraction process generates an overabundance of data that necessitates reduction to higher level features. A pairwise feature analysis technique is used for formulation of higher level features from primary features. The pairwise relationship of a single primary feature such as average chip intensity can be visualized as in Figure 37.



**Average Intensity Chip 1**

**Figure 37.** -Pairwise feature scatterplot with linear regression  
Pairwise feature analysis for prototype modeling

The various points in the scatterplot correspond to various FCB's and KGB's. A line can be fit to the KGB's (prototype instances) to the feature pair coordinates via linear regression. The line is defined by

$$y = \beta_0 x + \beta_1 \quad (72)$$

where  $\beta_0$  corresponds to the slope and  $\beta_1$  corresponds to the y-intercept of the regression line. The residual  $\gamma$  of a particular feature pair vector coordinate  $(x_i, y_i)$  with respect to the linear regression result is given by

$$\gamma = | y_i - \beta_0 x_i - \beta_1 | \quad (72)$$

This residual is computed for all pairs of prototype features on the feature plot. The mean of all these residuals is referred to as  $\gamma_{KGB}$ , the standard deviation for the prototype set.

$\gamma_{FCB}$  is defined as the residual of a feature pair corresponding to a FCB  $i$ .

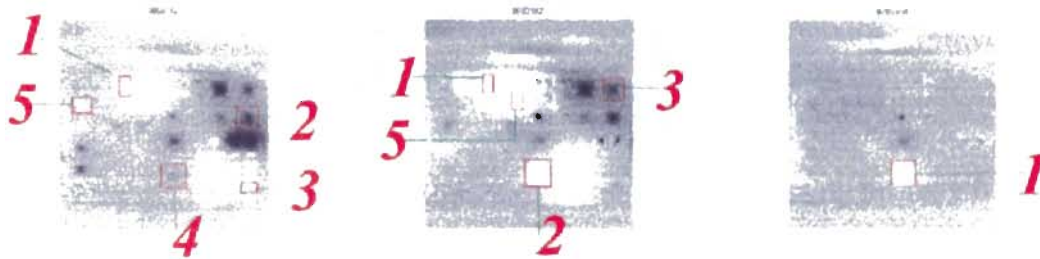
A tolerance factor  $\Gamma_p$  is computed from the standard deviation of the features pair values in the prototype set. If  $\gamma_{FCB} \leq \Gamma_p$  is not satisfied, the corresponding

feature pair on FCB  $i$  is considered to be an outlier. In other words, the point corresponds to an outlier board. An outlier board has an associated residue factor  $\varepsilon$  given by

$$\varepsilon = \left| \gamma_{iFCB} - \Gamma_p \right|. \quad (73)$$

This residue factor is a measure of the distance of the outlier board from the estimated regression line. Figure 37 shows instances of outlier boards.

A list of outlier boards is computed for all pairwise feature sets (each scatterplot). This list of outlier boards for all pairwise features is reduced to a list of outlier boards associated with each ROI. This list is computed using the aggregate residue factors corresponding to a particular board in various feature pairs. Hence these boards have an aggregate residue factor associated with each ROI. The aggregate residue factors are designated as higher level features. These results are used to prioritize the various ROI on a board in order of most likely cause of defect to least likely cause of defect and to perform similarity assessment in CBR. Figure 38 shows priority factors for various FCBs.

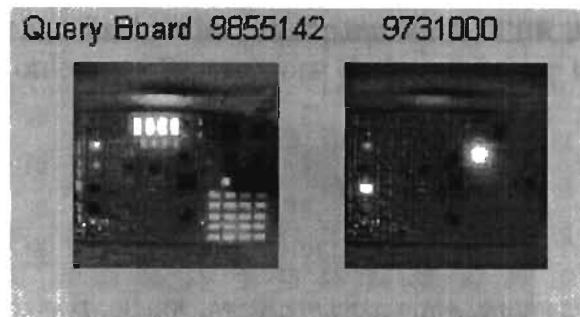


**Figure 38.** FCBs with priority factors for various chips (ROI).

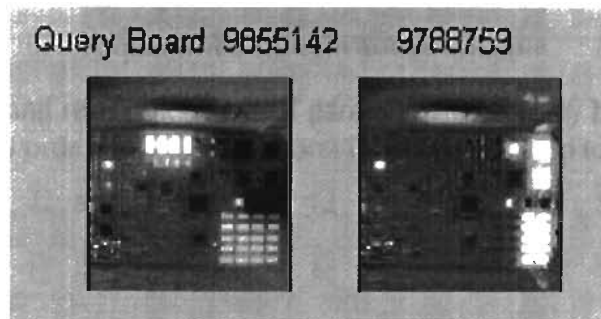
### 6.3.4 Query Formulation and CBR

In this application, we have tested two techniques for CBR. One approach uses primary features, another exploits higher level features. The primary feature used in the CBR examples in is the chip isolation intensity. The query can be formulated by

specifying the query FCB. The objective of CBR in this context is to find FCB's with a similar set of faulty chips. Based on the metadata information associated with the query board, the CBR engine examines only boards within the target class. The candidate boards are selected from the target database, and a similarity factor is computed by a least squares matching between higher level features generated by our pairwise methodology as compared to CBR using primary features. The matches are displayed in order of their similarity factors. Figure 39 shows the query FCB and the best match by CBR using primary features, and Figure 40 shows the query FCB and the best match for CBR using higher level features generated by our pairwise feature analysis. CBR by higher level features has shown the advantage of increased match relevance. For the circuit board application, this relevance is quantified in terms of overlapping faulty chips between the boards with respect to the query FCB.

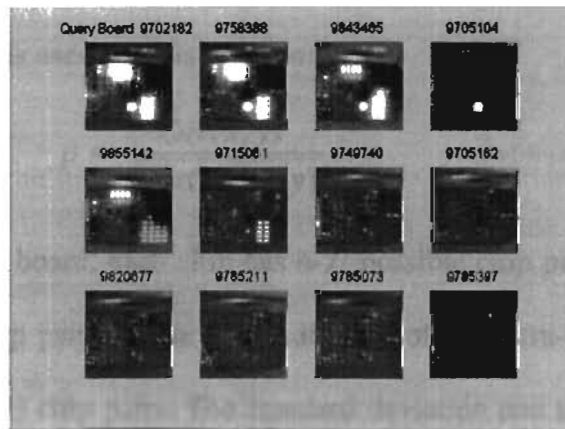


**Figure 39.** Query board and best match from CBR using primary features

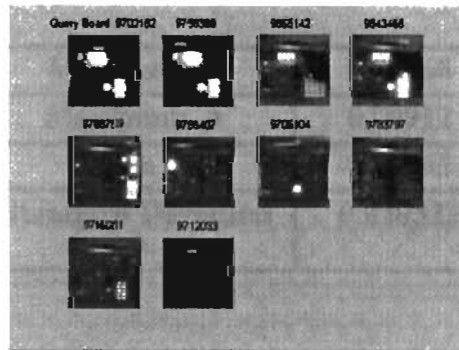


**Figure 40.** Query board and best match from CBR using higher level features

One of the most difficult problems in CBR is the designation of a cutoff parameter that limits the pool of possible matches. In this application, the cutoff parameter is specified automatically by the first retrieved KGB in the pool of possible matches. Figures 41 and 42 show query and results using the cutoff parameter for CBR using primary features and CBR using higher level features.



**Figure 41.** Query and results with cutoff parameter for CBR by primary features  
Matches ordering – Best to worst (Left to right, top to bottom)



**Figure 42.** Query and results with cutoff parameter for CBR by higher level features  
Matches ordering – Best to worst (Left to right, top to bottom)

### 6.3.5 Analysis

We now introduce two terms: the intra-board correlation factor and the inter-board correlation factor in the analysis of our CBR (for automated inspection) methodology and our results. The intra-board correlation is a correlation measure computed for average intensity data sets for various chip pairs. The Karl-Pearson correlation coefficient is used for this purpose:

$$\rho = \frac{Cov(x, y)}{\sqrt{Var(x)Var(y)}}. \quad (74)$$

Given  $n$  chips on each board, each chip has  $n-1$  possible chip pairs. On the whole there are  $n(n-1)$  possible chip pairs. Table 2 is indicative of the intra-board correlation factor behavior over all  $n(n-1)$  chip pairs. The standard deviation and the average of the intra-board correlation factor  $\rho$  from Table 2 clearly portray the linear behavior of the given data sets. This validates the application of linear regression to the given data sets.

<b>Minimum <math>\rho</math></b>	<b>0.92768</b>
<b>Maximum <math>\rho</math></b>	<b>0.99912</b>
<b>Mean <math>\rho</math></b>	<b>0.97765</b>
<b>Standard Deviation <math>\rho</math></b>	<b>0.020323</b>

**Table 2.** Intra-board correlation factor behavior for 90 possible chip combinations (*i.e.* 10 chips)

The inter-board correlation factor is computed by means of a proportionality factor. Given boards  $B_1$  and  $B_2$  with feature sets  $(f_1, \dots, f_n)$  and  $(f_1', \dots, f_n')$ . The

proportionality factors for  $B_1$  and  $B_2$  are  $(\frac{f_1}{f_1} \dots \dots \frac{f_n}{f_n})$ . Let the average proportionality

factor be  $\frac{f}{f}$ . Let us now study the inter-board correlation between the query board and

each of the matches. One would expect the correlation to be maximal for a perfect match

and then decrease as the order progresses from best match to worst. Thus, the standard

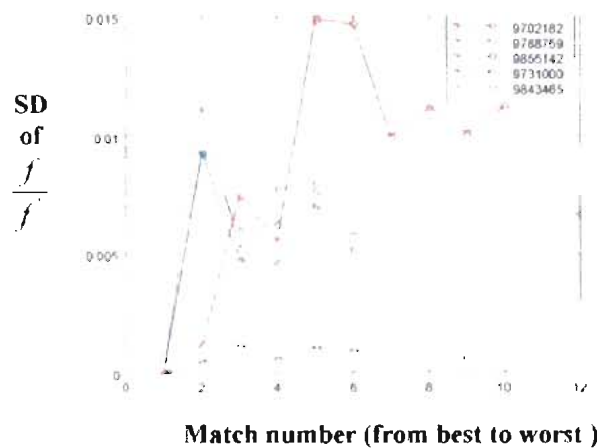
deviation of the associated proportionality factors should increase from the best to worst

matches. Figure 43 shows this relationship, which confirms the CBR results. The fact that

the behavior of the inter-board measure is not monotonic indicates that simple distance

measures or simple proportionality measures do not guarantee a proper relevance of

matches in terms of a decreasing order of similarity.



**Figure 43.** Standard deviation of inter-board correlation from best to worst matches for different query boards

We have developed a working prototype of the CBR engine described here. We have also developed tools to formulate queries for CBR by specific chip(s) on a specific board. Our experimental results show that the cutoff is reached earlier using CBR by higher level features generated by our pairwise methodology, as compared to CBR using primary

features. The pairwise feature analysis technique has proven to be an effective tool in the formulation of higher level features. This novel technique overcomes the shortcomings of conventional primary feature-based techniques by use of relative pairings of feature points rather than simple distance measures. The simple distance measures used with primary features are ineffective when imaging conditions are variable or when inter-chip interactions are significant. Currently, our focus is on automating the extraction of board features and consequent generation of the system library. The advantage of our methodology is associated with the pairwise technique used to generate a significantly simplified, yet comprehensive on-line database library, lending to the increased relevance of matches.



## **CHAPTER VII**

### **CONCLUSIONS**

#### **7.1 Summary**

Current searches in databases/libraries are based on text keyword descriptors called metadata. Recent advances in digital library technologies have advocated the concept of CBR. The emphasis has been on automatic extraction of semantic entity descriptors and using these descriptors to access the library. Text metadata perform well for text libraries consisting mainly of documents. Thus the need for CBR is more apparent in growing multimedia databases. Access to multimedia libraries is based on the semantic nature of the content. Our research has focussed specifically on digital image libraries. Text metadata are ill-suited for this purpose. Recent research has focussed on automated extraction of low level global image descriptors in hope of their contributing to a crude semantic description of the image. In cue with recent advances in human psychological understanding, we have advocated a segmentation based CBR. We have demonstrated a segmentation scheme for CBR. Rather than providing a complete partition of the image, we provide only segments of significance based on a unique LLGG metric. As part of an overall system we have used this segmentation to generate features for retrieval purposes.

We have also shown an extension of our CBR methodology to remotely sensed imagery. The potential applications of the remote sensing CBR engine include precision agriculture, hydrology, meteorology and retrieval of cloud-free imagery for image analysis. Apart from these rather general applications we have also elaborated on a CBR application profitable to industry namely automated inspection. CBR for automated inspection is guided by the simple philosophy that groups of FCBs can be repaired at the same time saving time and money.

In spite of advances made in CBR research some open research issues remain. The main amongst these is the combination of the diverse features involved in CBR for similarity analysis. It is very difficult to determine in an automated manner the object and associated feature of interest. In our CBR engine the user specifies such parameters. Perhaps the solution to the problem lies in understanding of how the human brain perceives similarity as a combination of diverse features. Nevertheless this true problem of similarity analysis will prove to be an exciting area of research in the years to come. Some other possibilities include investigation of ground truths for segmentation and CBR. Validation of segmentation and CBR results can be achieved by development of quantitative metrics for the same.

## BIBLIOGRAPHY

- [1]. S.T. Acton and D.P. Mukherjee, "Scale space classification using area morphology," *IEEE Transactions on Image Processing.*, May 2000.
- [2]. S.T. Acton, "Fast Algorithms for Area Morphology," submitted to *Digital Signal Processing: A Review Journal*.
- [3]. S.T. Acton, "Diffusion-based Edge Detectors," in *The Image and Video Processing Handbook*, Academic Press, to appear in 1999.
- [4]. S. T. Acton and J. Landis, "Multispectral anisotropic diffusion," *Int. Journal of Remote Sensing*, Vol. 18, pp. 2877-2886, 1997.
- [5]. J. Ahmed, J. Bosworth and S. Acton, "Image Segmentation techniques for Object-Based Coding", *Proc. Of the IEEE Southwest Symposium on Image Analysis and Interpretation*, Austin, TX, 2-4 April 2000.
- [6]. K. Arbter, "Affine-invariant Fourier descriptors,' *From Pixels to Features*, J.C. Simon, ed., Elsevier Science Publishers, pp. 153-164, 1989.
- [7]. J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R.C. Jain and C. Shu., "Virage image search engine: an open framework for image management. *The symposium on Electronic Imaging: Science and Technology – Storage & Retrieval for Image and Video Databases IV*, volume 2670, pages 76 – 87. IS&T/SPIE, 1996.
- [8]. J. A. Bangham, P.D. Ling, and R. Harvey, "Scale-space from nonlinear filters," *IEEE trans. PAMI*, vol. 18, no. 5, pp. 520-527. 1996.

- [9]. Banham, M.R.; Katsaggelos, A.K. Digital image restoration IEEE Signal Processing Magazine Volume: 14 2 , March 1997 , Page(s): 24 –41
- [10]. M. J. Basstiaans, “Gabor’s signal expansion and degrees of freedom of a signal”, Opt. Acta, vol. 29, no. 9, pp. 1223-1229, 1982.
- [11]. S.O. Belkasim, M. Shridhar, and M. Ahmadi, “Pattern recognition with moment invariants: A comparative study and new results,” Pattern Recognition, vol. 24, pp. 1117-1138, 1991.
- [12]. S. Belongie, C. Carson, H. Greenspan and J. Malik, “Color- and Texture-Based Image Segmentation Using EM and Its Application to Content-Based Image Retrieval,” *Proc. of ICCV*, Bombay, Jan 1998, pp. 675-682.
- [13]. Irving Biederman, “Recognition-by-Components: A Theory of Human Image Understanding,” *Psychological Review*, Vol. 94, No. 2, pp. 117-147, 1987
- [14]. C. Blakemore and F. W. Campbell, “On existence of neurones in the human visual system selectively sensitive to the orientation and size of the retinal images”, J. Physiology, London, vol. 203, pp. 237-260, 1969.
- [15]. Joseph H. Bosworth, “Scale-Space and Morphology in Image Processing”, Technical Report, The Oklahoma Imaging Laboratory.
- [16]. A.C. Bovik, M. Clark, and W.S. Geisler, “Multichannel texture Analysis Using Localized Spatial Filters,” IEEE Trans. PAMI, vol. 12, no. 1, pp. 55-73, Jan. 1990.
- [17]. P. Brodatz, Textures: A Photographic Album for Artists and Designers. New York: Dover, 1966.
- [18]. T. Caelli and P. Bevan, “Probing the spatial frequency spectrum for orientation sensitivity in stochastic textures”, Vision Research, vol. 23, pp. 39-45, 1983.

- [19]. F. W. Campbell and J. G. Robson, "Applications of Fourier analysis of the visibility of gratings", *J. Physiology*, London, vol. 197, pp. 551-556, 1968.
- [20]. Darrell T., and Pentland A. P., "Robust Estimation of a Multi-Layer Motion Representation", in *Proceedings IEEE Workshop on Visual Motion*, pp. 173-177, 1991.
- [21]. J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters", *J. Opt. Soc. America*, vol. 2, no. 7, pp. 1160-1169, July 1985.
- [22]. E. J. Delp, "Video and Image Databases: Who Cares?," *Proceedings of the SPIE/IS&T Conference on Storage and Retrieval for Image and Video Databases VII*, January 23-29, 1999, San Jose, California, pp. 274-277
- [23]. Y. Deng, C. Kenney, M.S. Moore, and B.S. Manjunath, "Peer group filtering and perceptual color image quantization," *Proc. of IEEE Intl. Symposium on Circuits and Systems (ISCAS)*, vol. 4, p. 21-24, 1999.
- [24]. S. DiZenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics, and Image Processing*, Vol. 33, pp. 116-125, 1986.
- [25]. S.R. Dubois and F.H. Glanz, "An autoregressive model approach to two-dimensional shape classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 55-66, 1986.
- [26]. Editors – H. G. Feichtinger, T. Strohmer, "Gabor Analysis and Algorithms – Theory and Applications", Birkhauser, 1998.
- [27]. D. Gabor, "Theory of communication", *J. Inst. Elect. Eng. London*, vol. 93, no. III, pp. 429-457, 1946.

- [28]. Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing," Addison-Wesley publishing company, USA, 1992.
- [29]. N. Graham and J. Nachmias, "Detection of grating patterns containing two spatial frequencies: A comparison of single-channel and multiple-channels models", *Vision Research*, vol. 11, pp. 251-259, March 1971.
- [30]. R. M. Haralick, "Statistical and structural approaches to texture", *Proc. IEEE*, vol. 67, no. 5, pp. 786-804, May 1979.
- [31]. J.P. Havlicek, A.C. Bovik, and D. Chen, "AM-FM Image Modeling and Gabor Analysis," in *Visual Information Representation, Communication, and Image Processing*, C.W. Chen and Y. Zhang, ed., *Optical Engineering Series* by Marcel Dekker, Inc., New York, 1999, pp. 343-385.
- [32]. J.P. Havlicek, "The evolution of modern texture processing," *Elektrik, Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 5, no. 1, special issue on image processing, pp. 1-28, 1997.
- [33]. H. Kauppinen, T. Seppänen and M. Pietikäinen, "An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 2, pp. 201-207, 1995.
- [34]. W. Y. Ma and B. S. Manjunath, "NeTra: A Toolbox for Navigating Large Image Databases," *Proc. of the IEEE Int. Conf. on Image Processing*, Santa Barbara, CA, pp. 568-571, Oct. 26-29, 1997.
- [35]. S. Marcelja, "Mathematical description of the responses of simple cortical cells", *J. Opt. Soc. America.*, vol. 70, no. 11, pp. 1297-1300, November 1982.

- [36]. D. Marr and E. Hildreth, "Theory of edge detection," in Proc. Of Royal Society, (Sec B, 207), pp. 187-217, 1980
- [37]. S. Masnou; J-M. Morel, "Level lines based disocclusion", *Proc. of the IEEE Int. Conf. on Image Processing*, Chicago, IL, pp. 259-263, Oct. 4-7,1998.
- [38]. J.-M. Morel and S. Solimini, *Variational Methods in Image segmentation* (Birkhauser, Boston, 1995).
- [39]. D.P. Mukherjee and S.T. Acton, "Document page segmentation using multiscale clustering," Proc. IEEE Int. Conf. on Image Processing, Kobe, Japan, Oct. 25-29, 1999.
- [40]. W. Niblack et al., "The QBIC project: querying images by content using color, texture, and shape," *SPIE Conf. On Storage and Retrieval for Image and Video Databases*, vol. 1908, San Jose, CA, pp. 173-187, February 1993.
- [41]. K.-R. Park and C.-N. Lee, "Scale-space using mathematical morphology." *IEEE Trans. PAMI*, vol. 8, no. 11, pp.1121-1126, 1996.
- [42]. A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: tools for content based manipulation of image databases," *SPIE Conf. On Storage and Retrieval for Image and Video Databases - II*, No. 2185, San Jose, CA, pp. 34-47, February 1994.
- [43]. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. on Pattern Anal. Mach. Intell.* PAMI-12, 629-639 (1990).
- [44]. E. Persoon and K. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst., Man, and Cybern.*, vol. 7, pp. 170-179, 1977.

- [45]. K. Pope and S.T. Acton, Modified Mean Curvature Motion for Multispectral Anisotropic Diffusion. Proc. of the IEEE Southwest Symposium on Image Analysis and Interpretation, Tucson, April 6-7, 1998.
- [46]. R.J. Prokop and A.P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," *Graphical Models and Image Processing*, vol. 54, pp. 438-460, 1992.
- [47]. B. Raghunathan and S.T. Acton, "Area Morphological segmentation for Content Based Retrieval," Accepted for publication at *Proc. of the IEEE Int. Conf. on Image Processing*, Vancouver, Canada, September 10 - 13, 2000.
- [48]. B. Raghunathan and S.T. Acton, "Content Based Retrieval for Remotely Sensed Imagery," *Proc. of the Southwest Symposium on Image Analysis and Interpretation*, Austin, USA, Apr. 2-4, 2000.
- [49]. B. Raghunathan and S.T. Acton, "A Content Based Retrieval Engine for Circuit Board Inspection ," *Proc. of the IEEE Int. Conf. on Image Processing*, Kobe, Japan, Oct. 25-29, 1999.
- [50]. P. Salembier and J. Serra, "Flat zones filtering, connected operators, and filters by reconstruction," *IEEE Trans. Image Processing*, vol. 4, no. 8, pp. 1153-1160, August 1995.
- [51]. G. Sapiro and D. L. Ringach, "Anisotropic diffusion of multivalued images with applications to color filtering," *IEEE Transactions on Image Processing*, Vol. 5, pp. 1582-1586, 1996.
- [52]. C. Andrew Segall and Scott T. Acton, Morphological Anisotropic Diffusion. IEEE International Conference on Image Processing, October 26-29, 1997.



- [53]. I.Sekita, T. Kurita, and N.Otsu, "Complex autoregressive model for shape recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 489-496, 1992.
- [54]. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.
- [55]. J.R. Smith and S.F. Chang, "VisualSEEk: a fully automated content-based image query system," *ACM Multimedia '96*, Boston, MA, Nov. 20, 1996.
- [56]. M. Swain and D. Ballard, "Color indexing," *Int. J. Comp. Vision.*, 7(1):11-32, 1991.
- [57]. L. Vincent, "Morphological gray scale reconstruction in image analysis: applications and efficient algorithms," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 176-201, April 1993.
- [58]. J.Y.A. Wang and Edward H. Adelson, "Layered Representation for Motion Analysis," *IEEE CVPR '93*.
- [59]. A. Witkin, "Scale-space filtering," in *Proceedings of the International Joint Conf. Artificial intelligence*, (IJCAI, Inc. Karlsruhe, Germany, 1983), pp. 1019-1021.
- [60]. Y. Y. Zeevi and M. Porat, "Combined frequency-position scheme of image representation in vision", *J. Opt. Soc. America*, A vol. 1, no. 12, p. 1248, December 1984.

## VITA

Badrinarayan Raghunathan

Candidate for the Degree of

Master of Science

Thesis: IMAGE SEGMENTATION AND CONTENT BASED IMAGE  
RETRIEVAL

Major Field: Electrical Engineering

Biographical:

Education: Received Bachelor of Engineering degree in Electrical Engineering from University of Madras, Tamil Nadu, India in May 1998. Completed requirements for the Master of Science degree at Oklahoma State University in July, 2000.

Professional Experience: Graduate Research Assistant, Oklahoma Imaging Laboratory, School of Electrical and Computer Engineering, Oklahoma State University, September 1998 to July 2000. Intern for ADRIN, Department of Space, Government of India, August 1997 to March 1998.

Professional Activities: Institute of Electrical and Electronics Engineers (IEEE), Reviewer – The IEEE Transactions on Image Processing.