

Graduate Theses, Dissertations, and Problem Reports

2005

## Analysis and synthesis of iris images

Sarvesh Makthal West Virginia University

Follow this and additional works at: https://researchrepository.wvu.edu/etd

#### **Recommended Citation**

Makthal, Sarvesh, "Analysis and synthesis of iris images" (2005). *Graduate Theses, Dissertations, and Problem Reports.* 1605.

https://researchrepository.wvu.edu/etd/1605

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

## ANALYSIS AND SYNTHESIS OF IRIS IMAGES

by

Sarvesh Makthal

Thesis submitted to the College of Engineering and Mineral Resources at West Virginia University in

partial fulfillment of the requirements for the degree of

#### MASTER OF SCIENCE

in

#### ELECTRICAL ENGINEERING

Approved by Dr. Arun Abraham Ross, Chair Dr. Lawrence Hornak Dr. Xin Li

Lane Department Of Computer Science and Electrical Engineering West Virginia University Morgantown, West Virginia 2005

Keywords: Iris Recognition, Markov Random Fields, Texture Synthesis, Synthetic Iris

© Copyright 2005 by Sarvesh Makthal

All Rights Reserved

## ANALYSIS AND SYNTHESIS OF IRIS IMAGES

## **Sarvesh Makthal**

#### ABSTRACT

Of all the physiological traits of the human body that help in personal identification, the iris is probably the most robust and accurate. Although numerous iris recognition algorithms have been proposed, the underlying processes that define the texture of irises have not been extensively studied. In this thesis, multiple pair-wise pixel interactions have been used to describe the textural content of the iris image thereby resulting in a Markov Random Field (MRF) model for the iris image. This information is expected to be useful for the development of user-specific models for iris images, i.e. the matcher could be tuned to accommodate the characteristics of each user's iris image in order to improve matching performance. We also use MRF modeling to construct synthetic irises based on iris primitive extracted from real iris images. The synthesis procedure is deterministic and avoids the sampling of a probability distribution making it computationally simple. We demonstrate that iris textures in general are significantly different from other irregular textural patterns. Clustering experiments indicate that the synthetic irises generated using the proposed technique are similar in textural content to real iris images.

## ACKNOWLEDGEMENTS

It has been my privilege to have Dr. Arun Ross as my advisor. His complete dedication towards research has motivated me to give my best time and again. I wish to thank him for the amount of time and effort he has put in to help me learn different concepts and his constant reminders of "Keep working hard" and "Any new developments?" have played a major role in making this thesis possible. I will always be deeply indebted to him for making my Master's experience a memorable one. I would also like to thank my committee members, Dr. Larry Hornak and Dr.Xin Li for all the help and support I have received from them.

I would like to take this opportunity to acknowledge some of my friends, whose company I will cherish for my life. Mahesh Subedar and Pavan Kumar Vedantam who have been on my side through all of my difficult phases in life, Rohan Nadgir, Rohin Govindarajan and Jidnya Shah who have been extremely helpful to me during my stay at the Biometric Systems Lab at WVU. I will always remember Chris Boyce, Phani Ivatury and Kiran Tadaka for the fun we had together.

My sincere acknowledgements to my parents and brother for their never ending love and support. It is to all these wonderful people that I dedicate this thesis to.

Last but not the least, I thank WVU for providing me with an opportunity for research. Go Mountaineers!

# Contents

Al	Abstract				
Ac	eknow	vledgem	ients		iii
Ta	ble of	f Conte	nts		iv
Li	st of l	Figures			vii
Li	st of [	<b>Fables</b>			ix
1	Intr	oductio	n		1
	1.1	Biome	etrics		1
	1.2	Iris as	a Biometric	• •	3
		1.2.1	Advantages	• •	5
		1.2.2	Disadvantages		6
	1.3	Overv	iew of Iris Recognition		7
		1.3.1	Appearance based methods		8
		1.3.2	Texture based methods		9

		1.3.3 Feature based methods	0
	1.4	Challenges in Iris Recognition	1
	1.5	Contributions of this thesis	2
	1.6	Organization	3
2	Ana	ysis of Iris Images 1	.4
	2.1	Introduction	4
	2.2	Texture Analysis	6
		2.2.1 Need for Texture Analysis	8
		2.2.2 Texture Analysis Methods	9
	2.3	Markov/Gibbs Random Field Model	:5
	2.4	Multiple Pairwise Pixel Interactions	:9
	2.5	Experiments	0
		2.5.1 Iris Image Localization	1
		2.5.2 Results	7
		2.5.3 Discussion	-2
3	Iris	mage Synthesis 4	4
	3.1	Introduction	4
	3.2	Image Texture Synthesis	0
		3.2.1 Single Resolution Algorithm	2
		3.2.2 Multiple Resolution Algorithm	6

		3.2.3	Generation from Multiple Primitives	57
		3.2.4	Acceleration of the Algorithm	60
	3.3	Result	s of Synthesis	61
		3.3.1	Generating Multiple Samples Per User Class	63
	3.4	Valida	ting Synthetic Irises	67
		3.4.1	Clustering of Iris Images	67
		3.4.2	Generation of Genuine/Impostor Distributions	69
	3.5	Discus	sion	70
4	Con	clusion	s and Future Work	74
	4.1	User S	pecific Models for Iris Recognition	75
	<ul> <li>4.2 Iris Image Indexing</li></ul>			75
				76
		4.3.1	Choice of Neighborhood	76
		4.3.2	Choice of Primitives	76

# **List of Figures**

1.1	Some popular biometric modalities used for authentication	2
1.2	Structure of the eye	4
1.3	Multiple instances of the same iris	5
1.4	Iris recognition systems in commercial deployment	6
1.5	Some common problems in iris image acquisition	7
2.1	First, second and third order neighborhoods defining a Markov random field	26
2.2	Examples of cliques that could be used to define a Markov/Gibbs random field	27
2.3	Pupil boundary	32
2.4	Limbic boundaries	33
2.5	Eyelid boundaries	34
2.6	Unwrapping procedure for iris image	35
2.7	Unwrapping an iris image	36
2.8	Enhancement of iris images	37
2.9	GLDH results for Brodatz album image D3	39
2.10	GLDH results for Brodatz album image D25	39

2.11	GLDH results for Brodatz album image D9	40
2.12	GLDH results for iris image of user 1	41
2.13	GLDH results for iris image of user 2	41
3.1	Detailed structure of iris texture	47
3.2	Segmented iris and its unwrapped version	49
3.3	Synthetic iris generation	51
3.4	Neighborhood of each pixel	53
3.5	Selection of primitives from a real iris image	55
3.6	Weights used for each of the primitives	58
3.7	Multiple primitives	59
3.8	The tree structure used for vector quantization	61
3.9	Synthesis results using a single primitive	62
3.10	Synthesis results using 2 primitives	63
3.11	Synthesis results using 3 primitives	64
3.12	Generating multiple samples per user class	66
3.13	Iris image subspace	68
3.14	Genuine/Impostor distributions	71
3.15	Failure in synthesis	72
3.16	Some examples of failures in the synthesis process	73

# **List of Tables**

1.1	Comparison of various iris recognition techniques.	10
3.1	Confusion matrix indicating iris and non-iris classification.	70

## Chapter 1

## Introduction

## **1.1 Biometrics**

Humans have traditionally relied on the use of face, voice or gait information to recognize others. For establishing the identity of people at public places like airports, we have always relied on documents such as passports or ID cards that carry information of the person possessing them. Such identification measures are highly susceptible to failure since these documents can be forged or tampered with easily. Hence it is necessary to use an authentication scheme that is based on the human body, and is unique to every individual.

*Biometrics* seek to link the identity of a person much more closely to the human body and some of its unique features. They are expected to perform better than traditional methods of identification because they depend on "who you are" and not "what you know" or "what you possess". Formally, biometrics refers to the automated or semi-automated use of physiological and/or behavioral traits associated with the human body to identify or verify an individual. Some of the popular biometrics that are used include fingerprints, face, iris, hand geometry, retina, signature, voice, gait, palmprints, etc [1]. Figure 1.1 illustrates some of the commonly used biometrics.



Figure 1.1: Some popular biometric modalities used for authentication.

An identification (or verification) system operating using biometrics first acquires the data, extracts relevant features and constructs a template from it, and then compares it with a database of such templates linked to various identities. In the 'verification' mode, the user claims an identity and the system compares the extracted features with the stored template of the asserted identity to determine if the claim is true or false. In the 'identification' mode, no identity is claimed and the extracted feature set is compared with the templates of all the users in the database in order to recognize the individual. In high security deployments (such as port of entry for a country), a system could be devised to work in the 'screening/watch-list' mode. This kind of operation simply determines if the individual is part of a watchlist.

Biometrics can be physiological or behaviorial. Physiological characteristics include fingerprints, face, retina, iris, etc. Behavioral characteristics include voice, signature, gait, etc. While fingerprint is the oldest and most widely used biometric, recent advances in iris recognition have made iris a very promising biometric trait. Of all the different unique characteristics of human beings, the usage of iris patterns has been found to be the most robust and accurate for the purposes of verification and identification of individuals [2].

## 1.2 Iris as a Biometric

The iris is the only internal organ of the body that is readily visible from the outside. Its purpose is to control the amount of light that enters the eye through the pupil, by using the dilator and sphincter muscles to control the pupil size. It is made up of an elastic fibrous tissue that gives it a very complicated texture pattern. This texture pattern has no links with the genetic structure of an individual and is generated by chaotic processes [3]. The human iris begins to form in the third month of gestation and the structure is complete by the eight month, even though the color and pigmentation continue to build through the first year of birth. After that, the structure of the iris remains stable throughout a person's life, except for direct physical damage or changes caused by eye surgery. The iris hence parallels the fingerprint in uniqueness but enjoys a further advantage that it is an internal organ and less susceptible to damages over a person's lifetime. The general structure of the eye and the position of the iris in the eye is shown in figure 1.2



Figure 1.2: Structure of eye<sup>1</sup>

Figure 1.3 shows multiple instances of the iris of the same person, taken over a period of time. Even though the eyelid occlusion and pupil dilation vary from image to image, the underlying iris texture remains the same.

Daugman has shown that iris patterns have about 250 degrees of freedom, i.e. the probability of two eyes having the same iris texture is about 1 in 7 billion [2]. Even the 2 irises of an individual are different thereby suggesting that iris textures are independent of the genetic constitution of an individual.

Iris recognition has been successfully deployed in many large scale and small scale applications. Figure 1.4 shows some of the applications where iris recognition has been

<sup>&</sup>lt;sup>1</sup>Available at : http://www.maculacenter.com/eyeanatomy.htm



Figure 1.3: Six different instances of the iris of a person showing the stability in iris texture pattern [4]

successfully put to use.

Even though it has many advantages and the fact that it is the most robust biometric available, the iris has some inherent drawbacks that cannot be denied. Some of the advantages and disadvantages of using iris as a biometric are discussed below.

### 1.2.1 Advantages

1. It is an internal organ of the body and is well protected. It is highly unlikely that the iris

patterns will be damaged due to normal day to day activities of a person [5]

<sup>&</sup>lt;sup>2</sup>Images © of Iridian Technologies Inc., USA



(a) Border control in UAE

(b) Expedited check-in at Narita Airport, Japan

(c) UN refugee control at the Pak-Afghan Border

Figure 1.4: Iris recognition systems in commercial deployment <sup>2</sup>

- 2. The iris patterns are highly random and it is assumed that no two individuals have the same iris pattern [2].
- There is no evidence of genetic penetrance (i.e., the heritability factor) in the structure of the iris patterns. Hence iris patterns of both eyes of the same person and those of identical twins are also different [3].

### 1.2.2 Disadvantages

- Iris image is difficult to acquire since the size of iris is very small, compared to say, faces.
   Capture of an iris image requires an elaborate set up with specialized cameras.
- 2. Since capturing the image is difficult, the enrollment process takes more time, which may not be acceptable to all the users. Also, the failure to enroll rate in iris is about 7% [6].

3. The iris could be obstructed by eyelashes, eyelids and reflections from the ambient and source light.

Figure 1.5 shows some common difficulties encountered in capturing good iris images.



Figure 1.5: Some common problems in iris image acquisition

## **1.3** Overview of Iris Recognition

The uniqueness of iris texture lies in the fact that the processes generating those textures are completely chaotic but stable. Hence in order to use the iris as a biometric, the feature extraction should be able to capture and encode this randomness present in the iris texture.

Based on an extensive literature survey, we classify iris recognition systems into three categories depending on the method by which the features from the texture are extracted for

matching purposes. These three categories are (a) appearance based, (b) texture based and, (c) feature based extraction. Appearance based techniques [7, 8] make use of classical statistical approaches like Principal component analysis (PCA) or Independent component analysis (ICA) to represent the iris images. Texture based techniques [2, 9, 10, 11, 12, 13] make use of filters for image processing and extract some properties of the filtered images to quantify the given iris image. Feature based techniques [14] make use of local features present in the image (blobs or patches) and record their locations and features to distinguish between different images.

A typical iris recognition system has three major components: (a) iris image localization, (b) feature extraction and, (c) the matching technique. Different iris recognition approaches have been summarized in table 1.1 under these criteria. We shall now present a detailed description of some of the methods from each category.

#### **1.3.1** Appearance based methods

Noh et.al have proposed an iris recognition system that makes use of Independent Component Analysis (ICA) to generate optimal basis vectors, which are then used to extract efficient feature vectors representing the iris signals [7]. Each of these vectors are then converted to an iris code by quantizing them to store them more efficiently. This method has shown performance similar to the gabor wavelet system proposed by Daugman [2], but also has some advantages over it. It uses a significantly smaller feature set and it is possible to extract the linear transform for the feature extraction from the iris image itself.

#### **1.3.2** Texture based methods

The first effective iris recognition system was proposed by Daugman [2] in 1993. It made use of a complete 2D Gabor transform [15] to generate a 256-byte binary 'iris code'. This 'iris code' is generated by taking the sign of the real and imaginary portions of the transformed image. A hamming distance measure is used to determine if the images are from the same person or not.

In another approach, Wildes [9] makes use of the Laplacian of the Gaussian filter to extract the features in the texture part. This is a relatively simple technique that relies on the edge features in the texture pattern. The matching is done based on computation of the normalized correlation between the acquired and database representations. Since the mean intensities are subtracted during the computation of normalized correlation, this technique is robust and can account for local variations in the image intensity.

More recently, Wang et al [10, 11] have proposed recognition techniques using multiple gabor filters. 8-directional gabor filters with multiple frequencies are used and the mean and variance values of these gabor filtered images are used as features for matching. Also another method exists that does not extract any explicit features, but makes use of the output of a correlation filter directly for aligning and then matching the iris images [12].

Boles et al proposed a technique based on wavelet transform zero crossings [13]. The dyadic wavelet transform is used to decompose the image at different resolutions. Since the higher levels are highly effected by noise, only a few lower resolutions are used. The zero-crossing signature of each image is generated by recording the zero crossings of these wavelet co-efficients. This technique offers more freedom in the method of image capture since it is translation and rotation

variant.

#### **1.3.3** Feature based methods

Wang et al have proposed a technique that makes use of the local variations in the iris texture along with wavelet transform [11]. The local variations are characterized by the appearance and disappearance of an important image structure. The two dimensional image is converted to a one dimensional signal and the dyadic wavelet transform is used to generate a position sequence that characterizes the local sharp variation points. These position sequences are then used for matching purposes.

Technique	Attribute			
	Localization	Feature Extraction	Matching Technique	
Daugman [93]	Integro-Differential Operator	2D Gabor Transform	Hamming Distance	
Wildes [97]	Circular Hough Transform	Laplacian of Gaussian Filters	Fisher Discriminant Analysis	
Noh et.al [03]	N/A	Multiscale ICA	Hamming Distance	
Wang et.al [03]	Hough Transform	Multiple Gabor Filters and Wavelet Transform	Weighted Euclidean Distance	
Boles et.al [98]	Edge and Contour Detection	Zero Crossings of Wavelet Transform	Global Energy of zero crossing signal	

Table 1.1: Comparison of various iris recognition techniques.

## **1.4** Challenges in Iris Recognition

Although there are iris recognition systems that can operate successfully at 0% false accept and false reject rates under ideal conditions [2, 14], there is ample scope for further research in this field. Some of the challenges that are yet to be overcome are enumerated in this section.

- Even after many systems being available for successful iris recognition, none of the present methods discuss the actual processes defining the iris structure. There is a need for more analysis of the iris texture as such. If the iris texture can be explicitly characterized, then a user specific matcher could be used for recognition, where in the matcher is tuned to a specific characteristic of each user.
- 2. Most of the current iris recognition techniques have never been tested on a large scale database. The largest publicly available database [4] has only 108 different users with 7 images per user. Due to the invasive nature of the image acquisition, creation of a large database has been a problem.
- 3. There have been no classification schemes for iris images like that are available for fingerprints. Though iris images appear chaotic and random, they do possess a global structure that can be perceived. Since all commercial systems can perform 1-to-N matches within a couple of seconds, such a scheme may not be of much commercial interest right now, but it is very important from an academic point of view. Such a technique will provide more insight into the structure of the iris texture and could provide more information about the individuality of iris images.

- 4. None of the literature in iris recognition today hints towards the development of an iris image quality measure. There is no set standard for the method of capture of iris images, resulting in a variety of iris images, differing from each other in many aspects. Of course, each algorithm has its own needs and specifications for the iris image capture, but the lack of a proper standard makes it difficult to compare various techniques on a common platform
- 5. All of the current literature focuses on operating with iris images which have been obtained by full co-operation from the user. Techniques that can make use of partial iris images or off-angle images need to be explored to reduce the stringent requirements of the iris image capture.

## **1.5** Contributions of this thesis

The work here tries to address some of the challenges discussed in the previous section. The major contributions of this thesis can be listed as follows.

- A new technique for analysis of iris images has been presented. It has been shown that this analysis has potential applications in developing a user-specific matcher for an iris recognition system. It is further discussed that the results of this analysis could be used for iris image indexing and efficient synthesis of iris images.
- 2. An attempt has been made to address the problem of non-availability of large iris databases for testing. A scheme is proposed to generate synthetic iris images that have iris like

properties and could be used for testing of algorithms. Markov Random field modeling is used to characterize and generate the synthetic iris images.

## 1.6 Organization

In chapter 2 we present the analysis of texture images in general and iris images in particular. We discuss how this analysis could play an important role in solving some of the unsolved problems in dealing with iris recognition. In chapter 3 we present a new technique for generating synthetic iris images based on texture synthesis. Summary and future work are described in chapter 4.

## Chapter 2

## **Analysis of Iris Images**

## 2.1 Introduction

The problem of iris matching involves determining the degree of correlation between any two iris images. A typical iris recognition algorithm has three major components: iris localization, feature extraction and feature matching. We have provided an overview of the iris recognition algorithms in Chapter 1. There are numerous algorithms that work efficiently to extract relevant features from the iris texture image and use them for identification and/or verification purposes. But none of the algorithms explicitly characterize the texture of the iris image. All the methods make use of 'feature extraction', which is nothing but representation of the image is some other form. We attempt to characterize the iris texture using a Markov Random Field (MRF) model that would aid in understanding the structural texture pattern of the iris. We propose a characterization which not only helps in classification of the iris images, but also helps in the understanding of the texture properties to a large extent.

Our technique is predominantly based on Gimel ' Farb's methodology [16] to describe natural textures with the help of a Markov Random Field that is generated based on the multiple pairwise pixel interactions. Traditional MRF/Gibbs modeling of images like the Gaussian Markov Random Field (GMRF) use predefined pairs of interacting neighbors and hence are not a good choice for non-uniform textures like iris images. In contrast, techniques based on multiple pairwise interactions make use of the relationship between two pixels that are close by as well as between pixels that are far away in the image lattice. The pairs that are used to characterize the texture and build the model are selected based on a potential strength assigned to them, which is calculated based on the Gray Level Difference Histogram (GLDH) of the image. Hence the pairs that characterize a particular image are found out from the image itself and are not pre-defined. Thus the model is developed based on individual training images and should result in a more accurate description of the texture structure.

Given a training image, the MRF model for the texture image is built in the following manner:(a) Estimation of initial parameters of the model from the training image and selecting the pairs of pixels that have maximum interactions, (b) using these pairs to refine the initial estimate by means of stochastic relaxation and (c) converging iteratively to a final solution for the model parameters based on the similarity of the GLDH of the generated and training images.

The uniqueness of the iris comes from the underlying texture patterns in it. Hence any form of feature extraction or analysis of the iris image will constitute a form of texture analysis. Image texture analysis is a well studied problem and the next section provides an introduction to texture analysis.

### 2.2 Texture Analysis

In real world, 'texture' is ubiquitous. Every perceivable object or scene has a texture associated with it and it is very easy for humans to perceive what texture is and the human mind is able to perceive and differentiate between two different types of textures with little difficulty. But in the field of machine learning and computer vision, making a computer learn and understand what is the texture of a particular segment of an image, is a challenging task. The definition of texture is difficult because of its abstract nature and the fact that practically every object seen exhibits some texture. Various definitions have been proposed but none of them provide a complete description of texture. This is because the definition is highly fluid and is not the same for all the classes of texture. Texture can be discriminated by the change in the intensity values due to the presence of solid objects present in the scene. Such a texture is mapped as intensity changes when being captured as an image. There are many instances in literature attempting to define texture, some of which are enumerated below.

 Chaudhuri, Sarkar and Kundu state that "Textured regions give different interpretations at different distances and at different degrees of visual attention. At standard distance with normal attention, it gives the notion of macro regularity that is characteristic of the particular texture. When viewed closely and attentively, homogeneous regions and edges are noticeable." [17]

- Sklansky states that "A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying, or approximately periodic". [18]
- 3. "Texture is defined for our purposes as an attribute of a field having no components that appear enumerable. The phase relations between the components are thus not apparent. Nor should the field contain an obvious gradient. The intent of this definition is to direct attention of the observer to the global properties of the display- i.e., its overall "coarseness", "bumpiness" or "fineness." Physically, nonenumerable (aperiodic) patterns are generated by a stochastic as opposed to deterministic processes. Perceptually, however, the set of all patterns without obvious enumerable components will include many deterministic (and even periodic) textures." [19]

These varied definitions of texture presented in literature illustrate the difficulty in putting forth a definition of texture that is complete in all respects. Thus there is no generally agreed upon definition of texture. But there are some general properties that one can assign to a texture:

- 1. Texture cannot be defined for a single point, it is a property that can be described only for areas.
- Since texture is visualized in the gray scale values of an image, any statistic that can describe the nature of the gray scale distribution of an image is capable of analyzing the texture.

3. Texture properties also vary with the resolution they are captured. At lower resolution, we can capture the global textural properties of an image and at higher resolutions of the same image; we could be able to gather information about the basic building blocks of the texture.

Other than these, there are other qualities that one can "feel" about texture like coarseness or fineness, contrast, density, regularity or irregularity, directionality etc.

#### 2.2.1 Need for Texture Analysis

Texture analysis in a given image may be needed for more than one good reason. We might need to find different regions in an image, which are separated by their distinctive texture. Such kind of texture segmentation is important in the analysis of aerial images and the images obtained from satellites. The segmentation can also be used to other applications like separation of text from the remaining part of an image. One may use texture analysis to determine if a particular texture, known apriori, is present or not in a given image. This is known as 'texture classification'. Such a classification has applications where there is a need to match two different textures and identify the degree of similarity, like biometrics. Texture identification is used primarily in biometric recognition or identification using iris. Another use of texture classification is in the medical image analysis, wherein the classifier will have to differentiate between normal tissue texture and abnormal tissue structure in the medical images.

Another application can be the extraction or construction of a three dimensional surface from variations in the textural properties in an image. Such an application helps us recreate the 3D surface, whose projection on the 2-D image is the given texture.

Since texture cannot be completely described in one particular definition, we also cannot find just one way that is able to completely characterize any given texture and extract its features successfully for all the different applications. Hence there have been different approaches to analyze texture by many people, depending on the kind of application and also depending on the type of texture being analyzed. The different ways in which texture has been analyzed can be divided into the following categories:

- 1. Statistical Methods
- 2. Geometrical or Structural Methods
- 3. Model based Methods
- 4. Signal Processing based Methods

#### 2.2.2 Texture Analysis Methods

1. Statistical Methods

Such methods take into account the statistical relationship that exists between the gray scale values of the pixels in the given image. Most of this work is based on the results of Julesz [20] describing the fact that humans use statistical features to discriminate between different textures. These could be first order or second order or higher order features. Since

many different features can be defined, but in the end most of them are inter related, we shall consider only a few important gray level features.

(a) Gray Level Co-Occurrence Matrices

The gray level co-occurrence probabilities for an image specify the conditional probability of finding two pixels with the specified gray level values at a specific distance 'a' in a specific direction 'd'. These probabilities, when found out all over the image, form the gray level co-occurrence matrix for the image, for any given 'a' and 'd'. There are no specific rules to determine values of the separation between the pixels and the direction of separation. They have to be decided empirically by visually looking at the texture and deciding upon the directionality and coarseness or fineness of the texture. The gray level co-occurrence pairs (GLCPs) are a kind of second order statistic on the gray scale levels in the image since they take into consideration pairs of pixels instead of one. For any particular window size, they represent the joint probability that two pixels separated by some distance in some particular direction, have specific gray level values. Hence one could construct a number of co-occurrence matrices for any given image and then calculate many different features from those matrices to analyze the texture.

The co-occurrence matrix is defined by Astrit et.al in [21] as follows:

Let I be an image whose pixel gray levels are in the range 0...255. Let  $\delta = (u, v)$  be an integer-valued displacement vector;  $\delta$  specifies the relative position of the pixel coordinates (x, y) and (x + u, y + v). A spatial co

occurrence matrix of an image is a matrix whose element (i, j) is the number of pairs of pixels in relative position such that the first pixel has gray level iand the second one has gray level j.

If  $\delta$  is very much smaller than the dimensions of the texture elements, then the pixels separated by displacement of  $\delta$  will have almost similar values. Thus, the co occurrence matrix will have larger values concentrated near its diagonal. But if the texture is fine compared to the size of  $\delta$ , then the matrix will be fairly uniform. Such features of the co-occurrence matrices have to be captured for effective texture analysis.

For any given co occurrence matrix, M, a number of statistical characteristics can be found out. One can make use of these features for analysis of texture. Some of these characteristics have been outlined from [22]:

- i. Energy  $-\sum_i \sum_j M_{\delta}^2(i,j)$
- ii. Entropy  $-\sum_i \sum_j M_{\delta}(i,j) \log M_{\delta}(i,j)$
- iii. Contrast  $-\sum_i \sum_j (i-j)^2 M_{\delta}(i,j)$
- iv. Homogeneity  $-\sum_{i}\sum_{j}\frac{M_{\delta}(i,j)}{1+|i-j|}$ v. Correlation  $-\frac{\sum_{i}\sum_{j}(i-\mu_{x})(j-\mu_{y})M_{\delta}(i,j)}{\sigma_{x}\sigma_{y}}$

These features can either be used individually or a combination of these could be used to analyze or classify a given texture.

One such use was by Davis [23], wherein he proposed a new tool called Polarograms for texture analysis. A Polarogram is a plot in polar coordinates of the texture statistics

as a function of orientation. A Polarogram can thus be described by constructing different co-occurrence matrices for fixed distances but variable orientations. Texture features can then be derived from the Polarogram by calculating the geometrical features of the Polarogram. Davis conducted two experiments on a database of 5 classes using 16 orientations to calculate the Polarogram for these textures. He used linear discriminant analysis to classify the textures and could obtain accuracy in the range of 75% to 90%. This method can be improved by finding the exact number of co occurrence matrices that need to be calculated for proper representation of the texture. Hence the basic problem of finding the right number of co occurrence matrices still remains open.

#### (b) Autocorrelation

Since texture is basically considered as a repetition of basic structural elements, the autocorrelation function of a texture image can provide some useful information about the texture properties. It can provide us with the amount of regularity and also information about the coarseness or fineness of the texture. The autocorrelation function of an image,  $\rho(x, y)$  is defined as:

$$\rho(x,y) = \frac{\sum_{u=0}^{N} \sum_{u=0}^{N} I(u,v)I(u+x,v+y)}{\sum_{u=0}^{N} \sum_{u=0}^{N} I^{2}(u,v)}.$$
(2.1)

In general, if the texture is coarse, then the auto correlation function will drop off slowly and if it is fine, then it will drop off sharply. For well defined textures, it will be periodic with regular peaks and valleys.

#### (c) SGLD, GLR and GLD Matrices

The Spatial Gray Level Dependence Matrix (SGLD), Spatial Gray Level Run Length Matrix (SGLRL) and the Gray Level Difference Matrix (GLDM) are three variants of the co-occurrence matrix and take into consideration the second order statistics of the texture image. Methods for texture classification using features extracted from these matrices were proposed by Mir, Hanmandlu and Tandon in [24]. The SGLD is nothing but the co-occurrence matrix itself; the SGLRL takes into consideration the run length, i.e. the maximum length of a connected set of pixels in a given direction that have the same gray scale value and the GLDM is similar to the co occurrence matrix except that it considers the difference between two gray scale levels instead of the *absolute* difference between two gray scale values.

#### 2. Geometrical Methods

Geometrical methods define the texture to be composed of structural elements or primitives. These primitives are usually determined as a result of geometric analysis of the image. Since these methods try to define the texture around a basic primitive, they are more effective in describing repetitive or structurally arranged texture patterns. After the structural elements are identified for an image, they could be analyzed by studying the statistical properties of those elements or by determining the placement rule of the elements in the texture. We will now consider some of the popular approaches based on the analysis of texture primitives.

#### (a) Voronoi Tesselation Features

Tuceryan and Jain used features of Voronoi tesselations for texture segmentation [25]. Voronoi tesselations have been used because they help characterize the local spatial relations in a texture pattern. These tesselations are constructed using texture tokens and the local spatial distributions of the tokens are reflected in the tesselations. Features of each token are computed from the Voronoi tesselation and they are used to evaluate if the texture is uniform or not. These features could be used for segmentation of texture images. Regions of large dissimilarities could constitute for a texture edge.

(b) Structural Methods

Structural methods assume that the texture is formed using only a few primitives that are placed repetitively in some geometrical or statistical pattern. These algorithms are most effective in dealing with extremely regular textures and fail when irregular or random textures are encountered. A number of authors have discussed methods to extract these texture primitives from the images [26], [27], [28]. The texture primitives are determined using multi-scale filtering or blob detection and then properties like intensity and shapes of the detected primitives are computed for texture segmentation.

#### 3. Model Based Methods

Model based methods characterize image to fit into a specific model that describes the inter relations between the different pixels in the image. This characterization can not only be used to describe the texture, but it can also act as a guide towards synthesizing similar texture images. Most algorithms model textures using Markov Random Fields (MRF). MRFs have been popular because of their ability to characterize a wide variety of textures and hence can be generalized easily. Another type of modeling is done using Fractals Mandelbrot [29]. The fractal dimension measures the roughness of a surface. But since most natural surfaces (and hence textures) are not deterministic and have certain stochastic properties attached to them, describing them with only fractal dimensions is difficult.

4. Signal Processing based Methods

In approaches using signal processing techniques for texture analysis, most methods make use of filtering techniques and then extract features from the filter response and use them for classification or segmentation. Both spatial and frequency domain filters can be used to extract features, but frequency domain filters perform better because of the reason that the human visual system performs frequency domain operations to analyze texture images. Extensive work has been done on using Gabor filters and wavelets for texture analysis with excellent results [2, 30, 31].

## 2.3 Markov/Gibbs Random Field Model

Markov Random fields have been popularly used in image modeling because of their ability to describe a wide range of different textures. Probabilistic image modeling treats the image as a random field and relates the pixels to a particular random field. If such a model accurately captures the inter pixel dependencies (which are described as probabilities), then multiple images

could be generated such that all the inter pixel relations in the generated image also follow the probabilities specified by the model.

A Markov Random Field (MRF) for any image can be broadly described as a probability distribution that governs the intensity value at each pixel in an image such that the value of a pixel depends only on the values of a specific "neighborhood" of the pixel and is independent of all other pixels in the image. The neighboring pixels are described by an  $n^{th}$  order neighborhood, where n is selected depending on the type of texture.

Let **S** be an image lattice of size  $m \times n = R$  pixels, then the  $n^{th}$  order neighborhood for each pixel *i* in **S** is defined as,

$$N_i = [A_i^n = All \ pixels \ within \ a \ radius \ `n' \ of \ a \ given \ pixel, \forall A_i^n = \mathbf{S}]$$
(2.2)

The neighborhoods up to third order are shown in the figure 2.1 :







Figure 2.1: First, second and third order neighborhoods defining a Markov random field

An MRF assigns probability values for each of the neighbors of a site and the intensity value at that site depends only on the values of its neighbors and not on any other pixels from the image lattice.
If **S** is a given image, then  $\mathbf{F} = f_1, f_2, \dots, f_R$  is a MRF on **S** with respect to the neighborhood system  $N_i$  such that it satisfies the following conditions :

$$Positivity: P(f_i) \ge 0$$

$$Markovianity: P(f_i \mid f_{S-\{i\}}) = P(f_i \mid f_{N_i})$$
(2.3)

where  $P(f_i)$  = Joint probability that the  $i^{th}$  pixel takes on one particular value given one configuration of neighboring pixels.

Another kind of random field modeling of images, that makes use of irregular neighborhoods and sometimes fewer pixels than a MRF is the Gibbs Random Field (GRF). The neighboring pixels for a GRF are described by what are known as "cliques" of pixels. A clique **C** can consist of just the pixel itself or 2 or more pixels. The spatial relationship between 2 or more pixels forming a particular clique can be defined in anyway that is convenient to efficiently characterize the given image. Some common examples of possible cliques are shown in figure 2.2:



Figure 2.2: Examples of cliques that could be used to define a Markov/Gibbs random field

A GRF is completely described by a probability distribution which is governed by the geometric structure and the intensity values of the pixels involved in the cliques.

The general form of a Gibbs Probability Distribution(GPD) is given as [32] :

Given an image **S**, with m x n = R pixels,  $\mathbf{F} = f_1, f_2, \dots, f_R$  is said to be a Gibbs random field if the probability values take the form :

$$P(f_i) = \frac{1}{Z} e^{-\frac{1}{T}U(f_i)}, \forall i \in R$$

$$(2.4)$$

Where Z is a normalizing constant and T is called the temperature and will be assumed unity here, and U(f) is the energy function.

The energy function is given as :

$$U(f_i) = \sum_{c \in C} V_c(f_i), \forall i \in R$$
(2.5)

where  $V_c$  is a function of the signal combinations  $[s_i : i \in C]$  in the clique **C** and is called the Gibbs potential function.

Further, an 'interaction structure' is defined in [33] as a particular geometrical pattern that is a clique **C**. The 'interaction strength' of a particular set of cliques **C** of pixels with the same geometrical relationship is specified by its potential function  $V_c$ .

Hence the GPD is now specified as :

$$P(f_i) = \frac{1}{Z} exp \sum_{c \in C} V_c(s_i : i \in C)$$

$$(2.6)$$

# 2.4 Multiple Pairwise Pixel Interactions

Traditional MRF models like the auto-binomial and the the Gauss-Markov model cannot take into account the full interaction properties in a given texture image primarily because they have predefined pixel pairs or 'cliques'. The auto-binomial model exploits only the first order neighborhood and the GPD is based on only four cliques. The Gauss-Markov model is more flexible than the auto-binomial model in the fact that it assumes all the pixels within a specified window are neighbors of each other and form a clique. But even this kind of a model is restricted in its use because very large windows cannot be used since they increase the computation cost.

Hence the Gibbs models should be generalized to include arbitrary cliques where in pixels placed at distances much larger than the maximum possible window size from earlier models can also be taken into account; if they contribute a significant amount of interaction to the texture structure. If such cliques are considered, then a pixel can be a part of many second order cliques and the whole 'interaction structure' i.e. the union of all such cliques that contribute significantly to the structure of the texture can be defined for each image separately. This is useful in modeling textures better than just randomly pre-selecting the cliques to describe the texture properties.

These 'clique families' are defined in [33] as : 'Each second order clique family

$$C_a = C_{\mu_a, v_a} = [(i, j) : i, j \in R; i - j = (\mu_a, v_a)$$
(2.7)

contains all the translation invariant pixel pairs (i, j) that have a fixed inter-pixel shift  $(\mu_a, v_a)$ . These inter-pixel shifts  $(\mu_a, v_a)$  can vary arbitrarily depending on the texture to be modeled and can thus be selected so as to represent the texture properties as accurately as possible. The model employed here assumes that the GLDH of the given image is a sufficient statistic to characterize the given image using an MRF. A GLDH is the second order histogram of an image. It gives the probability of two pixels in an image seperated by  $(\mu_a, v_a)$  having a gray-level difference of  $d \in [-q_{max}, \dots, 0, \dots, q_{max}]$ . Hence the GLDH obtained from the given image is used to assign the potential function values to each of the clique family a:

$$V_a(d = q - q_a) \propto H_a(d \mid g^o) \tag{2.8}$$

where  $q, q_a$  are gray values of pixels with a shift of  $d = [-q_{max}, \dots, 0, \dots, q_{max}]$  and  $H_a(d | g^o)$  is a component of the GLDH for a given training sample  $g^o$ . Hence by looking at the potential values, we can determine which gray level difference occurs more frequently in a given texture and thus only those cliques which have potential values greater than a specific value can be assumed to be true representative of the underlying texture. Thus by using this criterion for selecting the most active cliques, we can define the interaction structure for each image separately and thus model the texture in a more efficient manner.

# 2.5 Experiments

We performed experiments for MRF modeling on iris images from the CASIA iris database [4]. Before any image processing algorithm can be performed on these eye images, the iris portion must be localized and converted to the rectangular strip format. The iris image localization technique is described in detail next.

### 2.5.1 Iris Image Localization

The iris image localization is performed by detecting the image intensity changes using the integro-differential operator [2]. The primary steps in this image preprocessing can be defined as follows : (a) Define pupillary boundary, (b) Define limbic-iris boundary, (c) Define upper and lower eyelid boundary, and (d) Unwrap the circular iris portion to a rectangular image block.

1. Detecting Pupillary Boundary

The pupillary boundary is indicated in figure 2.3 the region that separates the iris from the pupil of the eye. This boundary is detected as an abrupt or sharp change in the gray level values when viewed along the circumference of a circle. This change will peak when the circle in consideration has its center located very close to the true pupil center and its radius is very close to the pupillary radius. In image processing, this can be accomplished by the use of an integro-differential operator. For an image I(x, y), the operator would search for the maximum in the derivative, with respect to changing radius r, of the contour integral of I(x, y) along a circle of radius r and center co-ordinates  $(x_0, y_0)$ . This can be expressed in the form of an equation as :

$$max_{(r,x_0,y_0)} \mid \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \mid$$
(2.9)

The operator behaves like a circular edge detector searching for a maximum contour integral derivative with increasing radius, and the location of the circles is determined by  $(x_0, y_0)$ .



Figure 2.3: (a) Original eye image and (b) Detected pupil boundary

### 2. Detecting Iris Limbic Boundary

After the pupillary boundary and the center of the pupil have been successfully determined, we proceed to detect the outer boundaries of the iris. This can be achieved in a similar fashion, only that the boundary is less prominent and has a lesser abrupt change than for the pupillary boundary. Instead of finding for an abrupt change in the circular fashion, we limit ourselves to arcs on the left and right hand sides of the image, since the iris is not visible as a full circle in most of the cases because of eyelid occlusion. Also, since the center of the iris coincides with or lies very close to the center of the pupil, we restrict the range of  $(x_0, y_0)$  to lie with a very small radius of the center of the pupil. This results in reduction of unnecessary computations. Figure 2.4 depicts the detected pupillary and limbic boundaries using this technique.



Figure 2.4: (a) Detected pupillary and limbic boundary

3. Detecting Eyelid Boundaries

Most iris recognition algorithms assume that the eyelid occlusion is very minor and do not take into account the portion of the iris near the eyelids. But the CASIA dataset iris images, which are used in the experiments here, is derived from people of Asian origin and the occlusion due to eyelids is very prominent in these images, compared to that in the images of people from Caucasian origins. Hence we have extended the iris image localization to even finding out the boundaries of the upper and lower eyelids for more accurate image representation. Again, we look for sharp changes in the gray scale intensities, but eyelids are better approximated by parabolic curves. Also, since the eyelids are bound to occur in the top and lower portions of the pupil center, we have narrowed our search to only those areas. We use the parametric definition of a parabola to construct parabolas of different sizes and shapes. The parametric equation of a parabola is given as :

$$x = at^2$$
  
$$y = 2at$$
 (2.10)

Hence we search for an abrupt change in the summation of gray scale values over the shape of a parabola, for various shapes and sizes of parabolas. This results in detection of the upper and lower eyelid boundaries in the eye image. Figure 2.5 shows the result of such a detection.



Figure 2.5: (a) Detected eyelid boundaries

### 4. Unwrapping

Daugman [2] represented the iris as a fixed rectangular block of image in a doubly dimensionless pseudo polar co-ordinate system. We have unwrapped the iris image in a similar fashion and the process of the unwrapping is illustrated in figure 2.6.



Figure 2.6: Unwrapping procedure for iris image

The iris portion of the eye image is mapped to a rectangular block of fixed size using the following procedure:

$$x(r,\theta) = (1-r)x_p(\theta) + rx_i(\theta)$$
  
$$y(r,\theta) = (1-r)y_p(\theta) + ry_i(\theta)$$
 (2.11)

where I(x, y) is the iris region in the eye image. (x, y) are the original cartesian coordinates and  $(r, \theta)$  are the corresponding polar co-ordinates. Figure 2.7 shows the result of unwrapping. Here the regions above the eyelid of the eye have been masked and are indicated in black. This region is not used for any analysis.



Figure 2.7: Unwrapping an iris image

This mapping takes into account various dilations and constrictions of the pupil and returns an image of the fixed size all the time. Thus the iris region is modeled as a flexible rubber sheet anchored at the iris boundary with the pupil center as the reference point.

5. Enhancement by Normalization

To account for the various changes in illumination and difference in contrast across various iris images, we perform normalization of the gray scale values for enhancing the quality of the image. We adopt the image normalization technique described by Hong et al in [34]. Let I denote the image and let M and V be the estimated mean and variance of image I respectively. Then, the normalized image, G can be obtained as :

$$G(x,y) = \begin{cases} M_0 + \sqrt{\frac{V_0(I(x,y) - M)^2}{V}} & \text{if } I(x,y) > M \\ M_0 - \sqrt{\frac{V_0(I(x,y) - M)^2}{V}} & \text{otherwise} \end{cases}$$
(2.12)

Where  $M_0$  and  $V_0$  are the desired mean and variance respectively. We have determined  $M_0$ and  $V_0$  empirically by taking the average of the mean and variances across all the images in the CASIA database. Some of the results of the enhancement using this technique are shown in figure 2.8.



Figure 2.8: Iris images (a)before enhancement (b) after enhancement

#### 2.5.2 Results

After the initial pre-processing is completed, the analysis is performed only on the output of the pre processing stage, i.e., the enhanced image. Also, to exclude any kind of eyelid occlusion, we consider only a  $25 \times 360$  portion of the unwrapped iris image for our analysis purposes. We have also used other kinds of images to compute the GLDH table from them and to study its properties. The different kinds of images used include (a) Regular (repetitive) images from the Brodatz album, (b) Irregular images from the Brodatz album and, (c) Iris images. We

analyze the importance of inter pixel relationships in these different kinds of images, based on the computation of their respective Gray level difference histograms (GLDHs).

For different kinds of textures, the relationship between pairs of pixels will be different. Since the GLDH captures the gray level differences between all the different possible pairs of pixels in an image (or cliques), it is fair to expect that the GLDH should carry most of the discriminatory information regarding the texture images being analyzed. The results presented here depict the maximum value of the gray level differences computed across the different cliques possible. Since we use a  $25 \times 360$  image for computation of the GLDH, the maximum possible pixel pairs are 9000. But since we consider two complementing pairs to be same (For e.g. (0, 1) and (0, -1)are treated as the same clique), the number of pairs is reduced to 4500. After the computation of the GLDH, we look at the maximum of the histogram across each of the 4500 cliques and select only the top 10% of the calculated GLDHs. This finally results in a GLDH table corresponding to the top 450 of the cliques in the image. In the graphical description, the X - axis corresponds to the clique number and the Y - axis corresponds to the maximum value of the GLDH for that particular clique. The values on the X - axis correspond to the cliques (0, 0) and (0, 1) and so on. Figures 2.9, 2.10 and, 2.11 shows some of the results obtained for the Brodatz images.



Figure 2.9: Brodatz album image D3 and its corresponding GLDH values



Figure 2.10: Brodatz album image D25 and its corresponding GLDH values



Figure 2.11: Brodatz album image D9 and its corresponding GLDH values

Figures 2.12 and 2.13 depict the GLDH results for iris images from 2 different users.



Figure 2.12: Iris image of user 1 and its corresponding GLDH values



Figure 2.13: Iris image of user 2 and its corresponding GLDH values

### 2.5.3 Discussion

In the earlier section, we presented some graphical results of the GLDH analysis performed on the different kinds of texture images. Some of the interesting observations of this analysis and their possible applications are discussed in this section. The primary observations were :

The primary goal of this exercise was to study the various pair wise pixel interactions in the texture image. Analysis of the GLDH values for each clique reveal that for textures having regularity and where the repetitive element is small, the pixels lying close to each other have more interaction than the ones that are lying far apart. This is illustrated in the GLDH values in figure 2.9. For images which did not have any regularity, there is almost always a significant relationship between pairs of pixels that lie close by and also for those pixels which are far apart. This fact is evident from figures 2.11, 2.12, and 2.13. Hence these results illustrate the fact that iris texture is not local, but has some global properties attached to it. Hence any random field modeling that expects to model an iris image effectively will have to take into consideration the relationship between all the pixels in the iris texture image.

One important observation is that across all the different texture images, the waveform that represents the maximum of the GLDH values for all the cliques in the image, is not similar, i.e., the fact that the GLDH values peak at different cliques for different kinds of texture images could be used as discriminatory information. Also, there is no indexing technique for iris image databases. This information could be used to develop an iris image classifier that could categorize the iris images into broad classes.

By examining the peak values in the GLDH, we can assign 'interaction strengths' to the

cliques. The cliques, ordered according to their interaction strengths, can form a unique identifier for each iris image. Thus, these prominent cliques could be stored for each user in an iris identification system and only these cliques can be used for identification of each user. These prominent cliques can be called as the 'interaction structure' for each user and thus a user-specific matcher can be devised that can be tuned for each user's specific characteristics.

Another application of this 'interaction structure' could be in the field of texture synthesis using Markov Random Fields (MRFs). Texture synthesis using MRFs requires probability values assigned to a specific neighborhood of each pixel to be generated. All the traditional texture synthesis methods right now use fixed size neighborhoods around the pixel to be generated. If the interaction structure is accurately determined through this analysis, then the neighbors of a pixel could be assigned probability values based on the values of their interaction strengths. This will result in much more efficient and accurate texture synthesis, since the MRF is being tailored to each image and not vice-versa.

# Chapter 3

# **Iris Image Synthesis**

# 3.1 Introduction

A number of iris recognition algorithms have been proposed in the literature over the past few years. Though most commercial systems claim a 0% false accept rate, only Daugman's method [2] has been tested on a large database (~ 350,000 images), which is proprietary . The Chinese Academy of Sciences (CAS) has made available a public database which is modest in size and comprises 756 images pertaining to 108 users [4]. With the introduction of new iris recognition algorithms (e.g., [9], [10], [11], [13], [14]), there is a pronounced need for a large public database to test and compare multiple techniques. The use of synthetic databases in biometrics is not new and has been previously studied in the context of fingerprint and face biometrics. Orlans et al. [35] have outlined the advantages of studying synthetic biometric capabilities which are enumerated below.

- 1. The development of parametric models that permit the *testing* of a biometric system under various conditions.
- 2. The mitigation of *privacy* concerns typically associated with real-world data.
- 3. The *efficiency* (with respect to cost and time) in assembling a large database representing a variety of intra-class and inter-class variations.
- 4. The design to statistical procedures to *predict* the performance of systems in large-scale deployments.

Also, an analysis-by-synthesis approach helps researchers gain insight into the individuality of biometric patterns.

In the fingerprint literature, a technique for generating synthetic fingerprints was proposed by Capelli et al. [36]. The resulting software known as SFINGE has been used to generate artificial databases that have been incorporated in multiple editions of the Fingerprint Verification Contest (FVC2000, FVC2002, FVC2004 [37]). The authors observe that the performance of competing algorithms on these synthetic databases is comparable with their performance on real datasets thereby suggesting the ability of SFINGE to realistically model intra-class and interclass dynamics. Similarly, the FaceGen software [38] along with the 3D Studio Max and Visage FaceTools has been used by Orlans et al. [39] to devise test protocols for face recognition.

There are, however, very few literature precedents discussing synthetic generation of iris images. Wang et al. [40] recently used the Principal Component Analysis (PCA) approach to generate synthetic irises. The eigen-coefficients obtained from a training set of real iris images

are modified in a controlled manner; the eigen basis is then used to generate synthetic iris images using these new coefficients. Proper modification of these coefficients results in new classes of iris texture. Lefohn et al. [41] created renditions of the human iris by stacking several transparent layers with each layer describing a certain iris component (such as stroma, collarette, sphincter, etc.).

In this thesis, we propose a new technique for generating synthetic iris using Markov Random Field (MRF) models. The patterns visible in any iris image can be categorized as a texture pattern. We describe a synthetic iris image generation scheme that is based on texture synthesis. Myriad texture synthesis algorithms are available in literature which describe the generation of large texture images given small samples as inputs or *primitives* [42, 43, 44, 45, 16, 32]. We synthesize iris texture using one particular class of fast texture synthesis algorithm proposed by Li-Yi Wei [42]. Multiple and distinct sources of texture are used as primitives for this algorithm. These primitive texture images are small portions of real iris textures that have been preselected to contribute in the synthesis procedure. Since the iris texture itself is random in nature, synthesis of an image with a combination of such primitives generates an image which also has a high degree of randomness.

The iris is a flattened circular region of the eye that is made of a fibrovascular tissue called stroma. The stroma connects the sphincter muscle, that contracts the pupil, with the dilator muscles, which open it. Due to the presence of stroma, the iris has a highly fibrous and textured appearance. The iris is usually pigmented, with the pigmentation being majorally from a substance called 'melanin'. The iris colors vary from brown to green or blue or hazel. The lighter

colored irises occur due to the lack of melanin pigmentation, which is a characteristic that mostly depends on the race of the person (Caucasian, African, Asian, etc.). The iris is a highly textured structure that exhibits specific characteristics in different regions.

Figure 3.1 shows some of the major regions of the iris texture. Based on the amount and



Figure 3.1: Detailed structure of iris texture

structure of the stromatic texture, the iris can be divided primarily into three major divisions :

(a) The inner pupillary region, where the fibrous texture is radially and very compactly arranged, (b) The central collarette portion where the fibrous texture begins to taper away. Also, this region is characterized by the presence of blob like structures called 'crypts', (c) The outermost portion of the iris is called the limbus and displays very little texture information. Only a few artifacts occur in this region due to the constant dilation and contraction of the pupil. These appearances are called as the contraction furrows.

We select the primitives used to generate synthetic iris images from the real iris images such that they reflect these different portions and characteristics of the iris texture. The primitives are selected from each of the above specified regions and they are used in the same order while generating the output image. During the generation process, the image is generated in distinct steps, for each of the primitive selected. For example, to generate the innermost portion of the iris, the primitives that would be used would have been selected only from the pupillary regions of a true iris image. After that we would proceed to generate the central portion of the iris, which would have information from primarily the collarette region of the iris, and finally the outermost region is generated from a third primitive which has been selected from the limbical region of a true iris image.

As explained in Chapter 1, most iris recognition systems perform segmentation of the image of the eye and extract the iris portion from the image to a rectangular portion. We proceed by generating the synthetic iris texture directly in this rectangular format. There are some inherent advantages in generating the images in this format.

- Observation has revealed that the local features in an iris are predominantly radial in nature. When a circular iris is mapped as a rectangular strip, these transform to vertical features. This process is illustrated in figure 3.2. Generation of images with vertical features is much simpler than trying to incorporate a process which would generate a circular image with radial features. Thus images can be generated in the form of a rectangular block and then remapped in a circular fashion to give the radial effect to the pattern.
- 2. If the image has to be generated in a circular fashion, it is inevitable that we generate a large image with dimensions equal to the radius of the circle and the circular pattern will then have to be selected from this image. But large images will require more time for generation. This complexity in terms of time required for generation can be reduced significantly if we



Figure 3.2: Segmented iris and its unwrapped version

simply generate an image in the form of the rectangular block, which will be of much smaller proportions.

Thus we first generate the iris images as a rectangular strip and then remap them onto an eye template. These mapped images are then subjected to perspective transformations [46] to simulate the effect of differences in camera focus and blur, thus creating multiple samples of the same image. The portion corresponding to the iris in these images is again unwrapped and these rectangular strips are used in the experiments. We perform two kinds of experiments on these images to validate the synthesis procedure.

1. The first of these experiments is aimed at showing that these images can be categorized as 'iris like' images. For that, we put together all the images from the CASIA iris database along with all the synthetic iris images and some of the standard stochastic textures from the Brodatz image album. Features characterizing the texture are computed using the cooccurrence matrix [22] and using these features, these images are classified by a k-means classifier into just 2 classes. It will be shown that all the iris images, original and synthetic, group together into one cluster while most of the Brodatz texture images are grouped into the second cluster.

2. In the second experiment, we try to measure the inter-class and intra-class distance between the generated iris images. For this purpose an iris matching algorithm is implemented that is based on extracting some features using the Haar Wavelet decomposition of the images and then performing a 1-to-n matching. The receiver operating characteristic(ROC) is also shown as a measure of the matching performance obtained using the synthetic images.

The whole procedure of synthesizing iris images can be divided broadly into the following sections: (a) Selection of primitives from real iris images, (b) Generation of a spatial weighting function to determine the influence of each of the primitives in the output synthesized image, (c) Texture synthesis in the output image using the primitives and their corresponding spatial weights, and finally (d) Validating the synthesized images by clustering and matching experiments. Figure 3.3 gives a snapshot of the synthesis procedure.

# 3.2 Image Texture Synthesis

The task of texture synthesis can be divided into 2 major divisions: (a) To model the texture ,i.e. to be able to characterize the stochastic and structural portions of the input sample and (b) To sample this model to produce new textures similar to the ones described by the model. The model is usually a probability distribution that describes the spatial relationship between groups



Figure 3.3: Synthetic iris generation

of pixels. Model sampling, in such a case, entails determining the pixel values for the new texture based on this probability distribution.

Our approach for iris texture generation is based on a fast texture synthesis technique [42] because of its high efficiency, low computational complexity and ease of implementation. The input to the basic algorithm consists of a small patch of an example texture patch (the primitive image) and a random noise image of size as desired as the output. The algorithm then iteratively modifies the random noise image to make it look similar to the primitive texture image. While a single primitive is sufficient for the generation algorithm, more than one primitives may also be used to generate outputs that look similar to a blend of the primitives. The use of multiple primitives may be necessary because (a) a typical iris exhibits rich texture in the immediate vicinity of the pupil which tapers away in intensity as one moves away from the pupil; and (ii)

the blending of multiple textures increases the complexity of the randomness of the synthesized image. In the following subsections we describe synthesis based on both single and multiple primitives.

The basic algorithm explained in [42] is primarily for images that can be categorized as a local and stationary process. That means, the any portion of the image looks similar when viewed within a small neighborhood of a pixel and that the value of that pixel is characterized by those surrounding pixels only. These are the basic characteristics of a Markov random field. This algorithm assumes these properties for the input texture patch and tries to incorporate these properties into the larger output sample. It generates the image pixel by pixel via a raster scan, while maintaining the local relations of the pixels as similar as possible to the primitive image's pixels. Since this process is completely deterministic and requires no sampling of a probability distribution, it is computationally inexpensive and can generate images very quickly. We first describe the single resolution algorithm which will then be followed by an extension of it to multiple resolutions to improve the efficiency. Also, this algorithm can be accelerated further by using techniques like tree-structured vector quantization.

## 3.2.1 Single Resolution Algorithm

Let the primitive sample be denoted as  $I_p$  and let  $I_s$  represent the initial random noise image. For each pixel r in  $I_s$ , let  $N_s(r)$  represents its neighborhood as shown in figure 3.4.

For our convenience we arrange the pixel values from the neighborhood in a vector form as



Figure 3.4: A figure illustrating the neighborhood of each pixel

shown in figure 3.4. Let  $N_p = \{N_p(r)\}$  be the set of *all* such neighborhoods in  $I_p$ . Each element of  $N_p$  is compared with the  $N_s(r)$  using the Euclidean distance metric. The purpose of this exercise is to determine the neighborhood in  $I_p$  that closely resembles the current neighborhood of r in  $I_s$ . The pixel value  $I_s(r)$  is then updated based on this comparison as

$$I_s(r) = I_p(t^*),$$
 (3.1)

where 
$$t^* = \arg \min_{t \in I_p} || N_s(r) - N_p(r) ||$$
 (3.2)

Here  $\| \cdot \|$  represents the  $L_2$  norm. This process is repeated iteratively in raster scan fashion for all the pixels. Thus the local relations between the pixels in the input texture are

incorporated into the noise image resulting in the generation of a similar looking texture. Note that more sophisticated comparison techniques other than the  $L_2$  norm may be used to compare the neighborhood intensities of the two images; however, experiments suggest that the  $L_2$  norm suffices for our purpose.

### 3.2.1.1 Choice of Neighborhood

Since the neighborhood effectively decides the quality of the synthesized texture, its size and shape play an important role. For any regular repetitive texture pattern, it should be equal to or greater than the largest regular structure present in it. But for the generation of iris images, which are non-homogenous, this constraint can be relaxed. One has to just make sure that the input texture block displays a good amount of 'random texture' elements. To facilitate this, images from the CASIA database [4] have been used to obtain the primitive patterns in our experiments. Since all regions of the iris do not have rich texture information, the dataset was carefully studied and only selective portions of each image were used as primitives by the synthesis algorithm. Figure 3.5 shows two  $30 \times 30$  blocks extracted from an arbitrarily chosen iris image; it also shows some of the sample primitives that have been selected for synthetic image generation. The neighborhoods were chosen based on the visual appearance of the iris texture and their size varied anywhere between  $9 \times 9$  to  $23 \times 23$  pixels in our experiments. It should be mentioned here that any other primitive demonstrating iris-like properties can be substituted for these primitives.

Also, the shape of the neighborhood is made 'causal', i.e. the neighborhood of a pixel will always have pixels that have already been generated in the order of the raster scan. This



Figure 3.5: (a) Selection of primitives from an actual iris image. The iris structure has been unwrapped and presented as a rectangular entity rather than a circular one. (b) Examples of sample primitives used in our experiments.

ensures that every output pixel is updated from only the pixels already generated and not purely from noise. This causal property of the neighborhoods accelerates the convergence speed of the algorithm. Only the first few pixels of the output image will use white noise as their neighborhoods in the first iteration, but subsequently, all pixels will use neighborhoods that have been assigned by the algorithm in the earlier pass.

#### 3.2.1.2 Edge Handling

The pixels near the image boundaries will not have the full portion of the specified neighborhood. This problem is circumvented by assuming that the image is toroidal in nature. Hence the pixel (x,y) can be mapped as  $(x \mod R, y \mod C)$  where R and C are the rows and columns of the image respectively. If the edge pixel neighborhoods are not handled properly, the output image will appear tiled and blocky. But it must be noted that due to the usage of causal neighborhoods, only toroidal mapping is not sufficient for smooth tiling of the output image. The multiple resolution algorithm to be introduced in the next subsection must be used in conjunction with toroidal mapping to overcome the problem of tiling.

### **3.2.2** Multiple Resolution Algorithm

As explained in Section 3.2.1.1, good quality texture synthesis requires the neighborhood to be at least as large as the largest regular structure present in the primitive image. For larger structural elements, the neighborhood size should be increased and increasing the neighborhood size would result in more computations and slower synthesis. A multi resolution approach can save computation because larger scale texture elements can be compactly represented by down sampled versions of the primitive. Thus smaller neighborhoods can still represent the large structural element.

The multi resolution approach is implemented by constructing Gaussian pyramids for both the primitive and output images. The output texture is first synthesized in the down sampled version and then this information is used to generate a larger image. Also, since the lower resolution image is already complete, a full neighborhood can be used instead of a causal one. The higher resolution image is constructed in a manner similar to the single resolution algorithm, with the only modification that the neighborhood now consists of pixels from the current resolution and also from those in the lower resolutions.

We have used a two level multi resolution synthesis in our experiments. The lower resolution image was obtained by down sampling the larger image by 2. Hence we even reduce the block size by 2 for the lower level. Thus if a  $5 \times 5$  block was used for the higher level, a  $3 \times 3$  block size was employed for the lower level.

#### **3.2.3** Generation from Multiple Primitives

In the previous sections, we described the generation of a texture based on a single primitive. That technique is valid for textures with repetitive patterns as the process would essentially generate a large texture that has portions of the small input texture repeated over the entire image. Iris texture, however, is not repetitive but exhibits significant randomness. To incorporate this randomness, we propose the use of more than one kind of iris texture as primitives for the synthesis process. Two or more primitives could be used to generate an image similar to a blend or mixture of the inputs. Such a scheme could be used to generate new textures that previously did not exist. Since iris texture itself is highly random in nature, the blend of multiple primitives will result in a very unique and random structure.

#### 3.2.3.1 Linear Weighting function

The algorithm takes a set of source iris textures  $I_{p_1}, I_{p_2}, \ldots, I_{p_k}$  as primitives with a weight image  $W_j$  associated with each primitive  $I_{p_j}$ . The weight image (whose size is the same as the desired synthetic image) specifies the importance of a primitive when determining the value of a pixel in the synthetic texture  $I_s$ . For example, if we use two primitives,  $I_{p_1}$  and  $I_{p_2}$ , and want the upper portion of the synthetic image,  $I_s$ , to resemble  $I_{p_1}$  and the lower portion to resemble  $I_{p_2}$ , then higher weights are assigned to the upper part of  $W_1$  and the lower part of  $W_2$ . The weight

images could be modified in more complex ways to achieve various types of synthetic textures. The simple case of weight assignment explained above is illustrated in figure 3.6.



Figure 3.6: Weights used for each of the primitives

To incorporate more randomness in the synthesis procedure, we treat the weight values as probability measures when updating pixel values in  $I_s$ . Thus, to update pixel r in  $I_s$  we first compare  $N_s(r)$  with all neighborhoods in the k primitives using the  $L_2$  norm. For each primitive,  $I_{p_j}$ , the best neighborhood,  $N_{p_j}(t_j)$  is reported. Then  $I_s(r)$  is assigned the value  $I_{p_j}(t_j)$  with a probability  $W_j(t_j)$ . This ensures that sufficient randomness is introduced in the synthetic image.

#### 3.2.3.2 Gaussian Weighting Function

More complicated ways of using multiple primitives could be devised. As explained in Chapter 1, the iris is divided into two prominent regions, the pupillary and the ciliary regions. The ciliary regions consists of closely arranged radial patterns that seem to fade away in the pupillary regions. The pupillary region is characterized by loosely formed radial patterns and also the presence of 'crypts', or large blob-like structures. Finally, the outer boundary is formed by the Limbus, which

joins the iris to the outer Sclera. We identify three 'key primitives' that describe the different formations found in the ciliary, pupillary and the limbus regions of the iris. By using a weighting function that accommodates these three primitives, iris images of enhanced visual quality are generated.

To accomplish this, the top one-third portion of the iris image is assigned to one primitive, and this primitive is placed in that region with a probability value if 1 (absolute certainty). In the remaining two-thirds region, we form gaussian peaks of the weights, which help in placing the crypts at arbitrary locations. The third primitive is placed in regions that are not occupied by these two primitives. Figure 3.7 shows an example of the three primitives that can be used in this scheme of weighting and the corresponding weights they occupy in the synthesized image.



Figure 3.7: Three primitives representing (a) Primitives, (b) Corresponding weights

## **3.2.4** Acceleration of the Algorithm

The texture synthesis procedure described above is totally deterministic and avoids the computational complexity arising from sampling a probability distribution. But this technique still requires a exhaustive search of the closest neighborhood for each pixel. This process can be accelerated by considering the neighborhood vectors of the primitive as points in a multidimensional space. Tree-structured vector quantization could then be used to efficiently search through them.

Tree-structured vector quantization is generally used for data compression. It first takes a set of vectors as training data and generates a tree-structured codebook. The training data here would be the 900 ( $30 \times 30 \ pixels$ ) neighborhood vectors of the primitive. These vectors are divided into two clusters using the k-means clustering technique. Each of these two groups can further be divided similarly by recursing the same algorithm on them. Hence, as shown in figure 3.8, a binary tree-like structure is developed. Let this tree structure be called as **T**. The depth of the tree can be decided empirically. The tree structure with clustering up to three levels is used our experiments.

To use this tree during synthesis, the closest point to the neighborhood vector  $N_s(r)$  is found out by doing the best traversal over the earlier generated tree structure **T**. If  $N_L$  is the number of pixels in the primitive, this kind of a tree structure has a computational complexity of  $O(log(N_L))$ as compared to  $O(N_L)$  for the exhaustive search. Thus the tree structured vector quantization greatly helps in accelerating the process of texture synthesis. The iris image synthesis algorithm was implemented in MATLAB on a Pentium IV 2.8 GHz machine with 2 GB RAM. While the



Figure 3.8: The tree structure used for vector quantization

average time of each single iteration to generate a  $30 \times 30$  image without any vector quantization was about 50 seconds, the time required for a similar operation using vector quantization was about 1.25 seconds.

# 3.3 Results of Synthesis

Images from the CASIA database have been used as the primitives for the experiments using this method. Since all the portions of the iris do not have rich texture information, the dataset was carefully eyeballed and only selective portions of each image were used as primitives to the synthesis algorithm. As explained in Section 3.1 the synthetic patterns are generated as rectangular entities and are subsequently mapped onto a circular structure to create iris-like images. If only one primitive is used for the synthesis procedure, the synthesized image, though iris like, has repetitions of the primitive in it. This fact is illustrated by the results obtained by the synthesis of images using only one primitive as shown in figure 3.9.



Figure 3.9: Synthesis results using a single primitive: (a) The primitive used, (b) Synthesized iris image

Hence 2 images of size  $30 \times 30$  pixels that have been extracted from the iris images are used as primitives to the synthesis algorithm. The CASIA dataset has 108 distinct iris images and we select two blocks from each image. Thus we have total of 216 separate blocks that can serve as primitives to our algorithm and the maximum number of possible classes that can be generated using this technique would be  $\binom{216}{2} = 23220$ . Figure 3.10 shows the results of synthesis using the multi resolution algorithm using 2 primitives and the linear weighting function (Figure 3.6).

By combining more than two texture primitives or by altering the weight images, this number can be exponentially increased. By using 3 primitives, we can generate  $\binom{216}{3} = 1656360$  distinct


Figure 3.10: Synthesis results using 2 primitives and the linear weighting scheme

iris images. Figure 3.12 shows the results of synthesis using the multi resolution, multiple primitive (3 in this case) algorithm with the gaussian weighting function for synthesis. The results depict the output after 10 iterations of the algorithm.

### 3.3.1 Generating Multiple Samples Per User Class

Even though iris patterns do not change over a period of time, multiple images of the same iris in real world scenarios, taken simultaneously or at different times, do not result in the same image

. Variations are caused due to rotation (cyclovergence or tilting of the head), translation and



Figure 3.11: Synthesis results using 3 primitives and the gaussian weighting scheme.

dilation (due to change in camera position, focus and zoom factors). These factors have to be added in a controlled manner to generate multiple images of same class. The synthetic image(rectangular) is rotated cyclically in order to incorporate the rotation parameter. Translation is included by shifting the circular iris along the x- and y-axis by a few pixels. To include the distortion due to camera motion, perspective transformations are used [46]. If  $x_u$  and  $y_u$  are the undistorted co-ordinates and  $x_d$  and  $y_d$  are the co-ordinates after distortion, then the lens distortion model can be written as :

$$x_u = x_d(1 + kr_d^2),$$
  
 $y_u = y_d(1 + kr_d^2),$  (3.3)

where  $r_d = \sqrt{x_d^2 + y_d^2}$  is the distorted radius. The inverse distortion model is given by :

$$r_u = r_d (1 + k r_d^2), (3.4)$$

where  $r_u = \sqrt{x_u^2 + y_u^2}$  is the undistorted radius.

Solving for  $x_d$  and  $y_d$  using equations 3.3 and 3.4 :

$$x_d = x_u \frac{r_d}{r_u},$$
  

$$y_d = y_u \frac{r_d}{r_u}$$
(3.5)

This model represents the simplest of the lens distortion cases, but higher distortion of cameras could also be added [46].

Figure 3.12 describes the procedure followed to generate multiple samples of an iris image after the rectangular block of image has been generated.



Figure 3.12: Generation of within class images: (a) Generated iris image, (b) Image wrapped to an eye template, (c)-(e) Same class images generated by multiple operations of perspective blurring and rotation

# 3.4 Validating Synthetic Irises

### 3.4.1 Clustering of Iris Images

Iris images have distinct texture properties when compared to other natural textures. Since images are high dimension data, it is difficult to visualize and compare different kinds of images easily. We hypothesize that in the space occupied by textured images, the iris texture resides in a compact, non linear manifold. Learning the structure of such manifolds is the problem of nonlinear dimensionality reduction. Such a solution will allow us to classify images into iris and non-iris textures. Our hypothesis is shown graphically in figure 3.13.

A simple form of clustering using the k-means clustering method is sufficient to show that the iris images are significantly distinct from other irregular texture images. We have studied the clustering property of real and synthetic iris images relative to the texture patterns in the Brodatz texture library. Since iris images are non-homogeneous and non-repetitive nature, we compare them with only the irregular textures present in the Brodatz library.

Some of the well known features for texture classification and analysis are the ones derived from the co-occurrence matrix of the images. The spatial gray level co-occurrence matrix of an image tries to capture the second order statistics of the image. The co-occurrence matrix  $C_d$  for a displacement vector  $\mathbf{d} = (dx, dy)$  can be defined as follows [22]: The entry (i, j) of  $C_d$  is the number of occurrences of the pair of gray levels *i* and *j* which are at a distance **d** apart. It can be stated mathematically as:



Figure 3.13: Figure depicting our hypothesis that iris images reside in an exclusive subspace within irregular textured images

$$P_d(i,j) = |((r,s),(t,v)) : I(r,s) = i, I(t,v) = j|$$
(3.6)

where  $(r, s), (t, v) \in NXN, (t, v) = (r + dx, s + dy).$ 

The co-occurrence matrices are used to compute various features like the energy, entropy, contrast and homogeneity. They are defined as follows [22]:

Energy = 
$$\sum_{i} \sum_{j} P_{d}^{2}(i, j)$$
  
Entropy =  $-\sum_{i} \sum_{j} P_{d}(i, j) log(P_{d}(i, j))$   
Contrast =  $\sum_{i} \sum_{j} (i - j)^{2} P_{d}^{2}(i, j)$   
Homogeneity =  $\sum_{i} \sum_{j} \frac{P_{d}(i, j)}{1 + |i - j|}$  (3.7)

There are no well defined methods for selecting the displacement vector **d**. Hence, based on experiments, we choose to select **d** in increments of 5 pixels along the x- and y-axis respectively. Each co-occurrence matrix is used to compute the four features as shown above in equation 3.7. The input image is of size  $(20 \times 360)$  and 111 co-occurrence matrices calculated. Thus a feature vector of length  $111 \times 4 = 444$  is derived for each image.

These features were computed for 100 images in the CASIA dataset, 100 images in the synthesized dataset and 20 irregular images in the Brodatz library. The k-means clustering technique is used to classify all these images into 2 categories. In this experiment, 97% of the real iris images and 100% of the synthetic iris images were classified together in one group, where as 75% of the Brodatz textures were categorized as non-iris images. The confusion matrix is presented in table 3.1

#### 3.4.2 Generation of Genuine/Impostor Distributions

Apart from being iris like, the synthetic iris images have to unique. Only then any iris recognition algorithm can make use of this database to prove itself. We tried using a well tested approach

True	Assigned Class	
Class	Iris	Non-Iris
Real Iris	97	3
Synthetic Iris	100	0
Brodatz	5	15

Table 3.1: Confusion matrix indicating iris and non-iris classification.

to extract features from the images and perform a 1 - n matching of the synthetic iris images. The Haar Wavelet decomposition is used to extract discriminatory features from the iris texture images.

The genuine and impostor match score distributions of the synthetic iris images (100 classes, 4 samples per class) by extracting features using the wavelet approach and comparing the features using the manhattan distance. Figure 3.14 compares these distributions with that associated with real iris images (100 classes, 4 samples per user).

## 3.5 Discussion

The algorithm presented in this chapter for generation of synthetic iris images is derived from the work of Wei [47]. It uses the same causal neighborhood and raster scan approach along with the deterministic sampling approach for texture synthesis. However, when used for synthesis with multiple primitives, instead of using a weighting function to blend the primitives, our algorithm



Figure 3.14: The genuine and impostor match score distributions for real and synthetic iris images.

uses the weights as probabilities. Thus whenever an output pixel is being generated, it is not the weighted sum of the primitive pixel values, but is always from only one primitive. This ensures that the iris-like properties of the primitive are maintained in the synthesized image.

The primitives used in the synthesis process are hand picked and are only small portions of the true iris images. They are selected based on the visual appearance of the iris structure. If the iris image can be indexed into various portions automatically, the selection of the primitives could be automated, resulting in a fully automatic synthesis procedure.

Though iris texture is considered to be extremely random, human vision can perceive a global texture to the patterns. Since fixed size neighborhoods are used for synthesis, they are not able to capture the global properties of the iris structure. Present synthesis techniques end up with iris images which are locally similar to the original primitives, but are lacking a global structure. We have made an effort to incorporate the global structure by selecting 3 primitives and designing a weight function that mimics the true structure of the human iris. Also, if the neighborhood size is chosen small relative to the texture artifacts present in the primitive image, then the resulting synthetic image lacks any similarity to the primitive images. An example of such a failure is shown in figure 3.15



Figure 3.15: Failure in synthesis: (a) Primitives used for synthesis (b) Synthesized Image

Some more examples of the failure of the synthesis procedure are shown in figure 3.16



Figure 3.16: Some examples of failures in the synthesis process

# Chapter 4

# **Conclusions and Future Work**

The iris as a biometric is assuming lot of importance. While there exist very efficient systems for iris recognition, the properties of iris texture and the underlying processes generating it have not been explored. The goal of this thesis was to provide an insight into these aspects. Markov Random Field methods have been used to describe the spatial relationships between the pixels of the iris texture image. While traditional Markov Random Fields model a given texture image based on a fixed neighborhood, we have used a flexible neighborhood that takes into account all the pixels present in the image. We then come up with a unique description for each texture image based on the relationships of the pixels in the image, without restricting the neighborhood to a fixed size. This results in the Markov Random Field being tailored to a specific image. Thus, we have a more robust and accurate description of the underlying texture present in the image. The proposed algorithm is easy to use as it only requires the difference of pixels values located at various distances in the image.

Another contribution made in this thesis is the generation of a synthetic iris database. Again, Markov Random Fields have inspired our approach to generate synthetic iris images via image texture synthesis. A fast texture synthesis approach has been adopted. Multiple samples of textures extracted from real iris images are used as inputs for the texture synthesis. Extremely realistic images with iris like properties were generated using this approach. We have presented multiple methods to generate different kinds of synthetic iris images. Robust testing has been performed to validate the generated images in more than one way.

We envision several possible directions of future work as described below.

## 4.1 User Specific Models for Iris Recognition

As explained in Chapter 2, we are able to define each individual user's specific characteristics and this information could be used to define a user-specific matcher. A user-specific matcher could be tuned to each enrolled user's characteristics. When a person approaches for verification as a particular identity, the system then invokes that particular identity's specific matcher to perform the verification process. This should result in a more robust biometric system since the matcher is specific to each user and makes use of the characteristics that are unique to the users.

## 4.2 Iris Image Indexing

There are no published techniques that address to the problem of iris image indexing. The userspecific characteristics can be used to detect, on a broader scale, the properties of various users in a iris recognition system for indexing the iris database.

## 4.3 Improvements in Synthesis Algorithm

### 4.3.1 Choice of Neighborhood

To maintain simplicity, we have restrained ourselves to using a fixed neighborhood in determining the inter-pixel relations during the synthesis of artificial iris images. This process could be further refined by using only the prominent cliques or the interaction structure (learnt by the multiple pair wise pixel analysis of the image) as the neighborhood for synthesis procedure. The results of synthesis can also serve as an assessment platform for the analysis. The better the learnt interaction structure, the better the synthesized result.

#### 4.3.2 Choice of Primitives

We have tested the synthesis algorithm using 1, 2 and 3 primitives as inputs. More number of primitives could be used to generate more realistic looking iris images. More complex weighting functions could be developed that help capture the true properties of the iris texture. Also, using more number of primitives will greatly increase the number of distinct classes (i.e., users) that can be generated.

# **Bibliography**

- [1] Anil K. Jain, Arun Ross, and Salil Prabhakar, "An introduction to biometric recognition," IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Image and Video-Based Biometrics, vol. 14, no. 1, pp. 4–20, Jan 2004.
- [2] John G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1160, Nov 1993.
- [3] Ales Muron and Jaroslav Pospisil, "The human iris structure and its usages," Acta Univ.
   Palacki. Olomuc. Fac. Rerum Nat. Phys., vol. 39, pp. 87–95, 2000.
- [4] The CASIA Iris Image Database, ," http://www.sinobiometrics.com.
- [5] Elaine R Berman, Biochemistry of the Eye, Springer, 1991.
- [6] Anil K. Jain, "Biometric recognition: How do I know who you are?," *Carnegie Mellon University Distinguished Lecture Series*, Nov 2004.

- [7] Kwanghyuk Bae, Seungin Noh, and Jaihei Kim, "Iris feature extraction using independent component analysis," Proc. 4<sup>th</sup> Intl. Conf. on Audio and Video Based Biometric Person Authentication (AVBPA 2003), pp. 838–844, 2003.
- [8] Vivekanand Dorairaj, Gamal Fahmy, and Natalia Schmid, "Performance evaluation of iris based recognition system implementing pca and ica techniques," *Proc. of SPIE Conf. on Biometric Technology for Human Identification*, March 2005.
- [9] Richard P. Wildes, "Iris recognition: An emerging biometric technology," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1348–1363, Sept 1997.
- [10] Li Ma, Yunhong Wang, and Tieniu Tan, "Iris recognition using circular symmetric filters,"
   *Proc. of* 16<sup>th</sup> Intl. Conf. on Pattern Recognition, vol. 2, pp. 805–808, 2000.
- [11] Li Ma, Yunhong Wang, and Tieniu Tan, "Personal identification based on iris texture analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1519–1533, Dec 2003.
- [12] Vijaya B.V.K. Kumar, C. Xie, and J. Thornton, "Iris verification using correlation filters," *Proc.* 4<sup>th</sup> Intl. Conf. on Audio and Video Based Biometric Person Authentication (AVBPA 2003), pp. 697–705, June 2003.
- [13] W.W. Boles and B. Boashash, "A human identification technique using images of the iris and wavelet transform," *IEEE Trans. on Signal Processing*, vol. 46, no. 4, pp. 1185–1188, Apr 1998.

- [14] Li Ma, Tieniu Tan, and Yunhong Wang, "Efficient iris recognition by characterizing key local variations," *IEEE Trans. on Image Processing*, vol. 13, no. 6, pp. 739–750, June 2004.
- [15] John G. Daugman, "Complete discrete 2-d gabor transforms by neural networks for image analysis and compression," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, July 1988.
- [16] Georgy L. Gimel'farb, "Texture modeling by multiple pairwise pixel interactions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 11, pp. 1110 1114, Nov 1996.
- [17] B.B. Chaudhuri, N. Sarkar, and P. Kundu, "Improved fractal geometry based texture segmentation technique," *IEE Proceedings part E*, vol. 140, pp. 233–241, 1993.
- [18] J. Sklansky, "Image segmentation and feature extraction," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-8, pp. 460–473, 1978.
- [19] W. Richards and A.Polit, "Texture matching," *Kybernetic*, vol. 16, pp. 155–162, 1977.
- [20] B. Julesz, "A theory of preattentive texture discrimination based on first-order statistics of textons," *Biological Cybernetics*, vol. 41, pp. 131–138, 1981.
- [21] R. Astrit, R. Azriel, and M. Farzin, "Extracting boundaries from images by comparing co occurrence matrices," *Proc. Of VIIth Digital Image Computing: Techniques and Applications*, Dec 2003.

- [22] L.F. Pau and P.S.P. Wang, *The Handbook of Pattern Recognition and Computer Vision*, chapter 2.1, pp. 207–248, World Scientific Publishing Co., 2<sup>nd</sup> edition, 1998.
- [23] L.S. Davis, "Polarograms : a new tool for image texture analysis," *Pattern Recognition*, vol. 13, no. 3, pp. 219–223, 1981.
- [24] A.H. Mir, M. Hanmandlu, and S.N. Tandon, "Texture analysis of ct images," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 6, pp. 781–786, Nov-Dec 1995.
- [25] Mihan Tuceryan and Anil Jain, "Texture segmentation using voronoi polygons," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 211–216, 1990.
- [26] H. Voorhees and T. Poggio, "Detecting textons and texture boundaries in natural images," *Proceedings of the First International Conference on Computer Vision*, pp. 250–258, 1987.
- [27] D. Blostein and N. Ahuja, "Shape from texture: Integrating texture-element extraction and surface estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1233–1251, 1989.
- [28] Tomita, Fumiaki, and S. Tsuji, *Computer Analysis of Visual Textures*, Kluwer Academic Publishers, Boston, 1990.
- [29] B.B. Mandelbrot, The Fractal Geometry of Nature, Freeman, San Fransico, 1983.
- [30] Anil K Jain and Farshid Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, December 1991.

- [31] M. Clark and A. C. Bovik, "Texture segmentation using gabor modulation/demodulation," *Pattern Recognition Letters*, vol. 6, pp. 261–267, September 1987.
- [32] Stan Z. Li, Markov Random Field Modeling in Image Analysis, Springer-Verlag, 2001.
- [33] Georgy L. Gimel'Farb, Image Textures and Gibbs Random Fields, Kluwer Academic, 1999.
- [34] Lin Hong, Yifei Wan, and Anil Jain, "Fingerprint image enhancement:algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, August 1998.
- [35] N.M. Orlans, D.J. Buettner, and J. Marques, "A survey of synthetic biometrics: Capabilities and benefits," *Intl. Conf. on Artificial Intelligence*, 2004 (To appear).
- [36] R. Cappelli, D. Maio, D. Maltoni, and A. Erol, "Synthetic fingerprint-image generation,"
   *Proc. of the* 15<sup>th</sup> Intl. Conf. on Pattern Recognition, vol. 3, pp. 471–474, 2000.
- [37] "University of bologna: Biometric systems laboratory," Available at: http://bias.csr.unibo.it/research/biolab/sfinge.html.
- [38] "The FACEGEN software," Available at: http://www.facegen.com/.
- [39] N.M. Orlans, Alan T. Piszcz, and Richard J. Chavez, "Parametrically controlled synthetic imagery experiment for face recognition testing," in *Proc. of the 2003 ACM SIGMM workshop on Biometrics methods and applications*, Berkley, CA, 2003, pp. 58–64.

- [40] Jiali Cui, Yunhong Wang, JunZhou Huang, Tieniu Tan, and Zhenan Sun, "An iris image synthesis method based on PCA and super-resolution," *Proceedings of the* 17<sup>th</sup> *International Conf. on Pattern Recognition*, vol. 4, pp. 471–474, Aug 2004.
- [41] Aaron Lefohn, Brian Budge, Peter Shirley, Richard Caruso, and Erik Reinhard, "An ocularist's approach to human iris synthesis," *IEEE Computer Graphics and Applications*, vol. 23, no. 6, pp. 70–75, Nov/Dec 2003.
- [42] Li-Yi Wei, Texture Synthesis by Fixed Neighborhood Searching, Ph.D. thesis, Stanford University, 2001.
- [43] Anil K. Jain and G.R. Cross, "Markov random field texture models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 25–39, 1983.
- [44] Rupert Paget, "Strong markov random field," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 408–413, March 2004.
- [45] Rupert Paget and I. Dennis Longstaff, "Texture synthesis via a noncausal nonparametric multiscale markov random field," *IEEE Trans. on Image Processing*, vol. 7, no. 6, pp. 925–931, June 1998.
- [46] Frederic Devernay and Olivier Faugeras, "Straight lines have to be straight," *Machine Vision and Applications*, vol. 13, no. 1, pp. 14–24, 2001.

[47] Li-Yi Wei and Mark Levoy, "Fast texture synthesis using tree-structured vector quantization," *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 479–488, 2000.