

2012

Iris Indexing and Ear Classification

Ravindra B. Gadde
West Virginia University

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

Recommended Citation

Gadde, Ravindra B., "Iris Indexing and Ear Classification" (2012). *Graduate Theses, Dissertations, and Problem Reports*. 4855.
<https://researchrepository.wvu.edu/etd/4855>

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.

Iris Indexing and Ear Classification

by

Ravindra B Gadde

Thesis submitted to the
Benjamin M. Statler 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

Donald Adjero, Ph.D.
Xin Li, Ph.D.
Arun A. Ross, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2012

Keywords: Biometrics, Classification, Indexing, Ear, Iris, Burrows-wheeler Transform

Copyright 2012 Ravindra B Gadde

Abstract

Iris Indexing and Ear Classification

by

Ravindra B Gadde

Master of Science in Electrical Engineering

West Virginia University

Arun A. Ross, Ph.D., Chair

To identify an individual using a biometric system, the input biometric data has to be typically compared against that of each and every identity in the existing database during the matching stage. The response time of the system increases with the increase in number of individuals (i.e., database size), which is not acceptable in real time monitoring or when working on large scale data. This thesis addresses the problem of reducing the number of database candidates to be considered during matching in the context of iris and ear recognition. In the case of iris, an indexing mechanism based on Burrows Wheeler Transform (BWT) is proposed. Experiments on the CASIA version 3 iris database show a significant reduction in both search time and search space, suggesting the potential of this scheme for indexing iris databases. The ear classification scheme proposed in the thesis is based on parameterizing the shape of the ear and assigning it to one of four classes: round, rectangle, oval and triangle. Experiments on the MAGNA database suggest the potential of this scheme for classifying ear databases.

Acknowledgements

It's my pleasure to thank all the people involved in my thesis.

I am indebted to my advisor Dr. Arun Ross, whose invaluable assistance, support and guidance were the roots for my Master's thesis work at WVU. Dr. Ross has been the source of inspiration for my work. My skills have increased multifold during my association with him.

I am very much grateful to my committee members Dr. Donald Adjero, Dr. Xin Li for their support during the start of my Masters, encouragement, technical discussions and feedback. I am also thankful to Dr. Natalia Schmid for her valuable suggestions on Biometrics during my first semester at WVU.

I would like to thank Siemens Corporate Research in Princeton for giving me an opportunity to get exposed to Industrial Research, through an internship experience.

I dedicate this thesis to my parents and sister. I am very thankful to my family for all the support they gave me through my childhood.

I would take this opportunity to thank my fellow researchers at (Integrated Pattern Recognition and Biometrics) iPRoBE Lab and at WVU for making my stay, memorable. Special thanks to my friends Srinitya Musunuru, Sriram Thotakura, Nikhil Yalamanchili, Sivakanth Mikkilineni, Praneeth Polavarapu, Sriram Sankar, Sodith Gandavarapu, Manoj Dobbala, Nagendra Vemulapalli, Raghu kiran Yalamanchili, Maurya Srungarapu, Anveeksh Koneru, Srikanth Parupati, Raghunandan Pasula, Raghav Jillela, Savan Suri and Asem Othman for all their support and encouragement during my stay away from home.

Last but not the least I thank lord almighty for giving me this opportunity and blessing me with such wonderful people in my life.

Contents

Acknowledgements	iii
List of Figures	vi
List of Tables	viii
1 Introduction	1
1.1 Biometrics	1
1.2 Biometric Systems	2
1.3 Iris as a biometric	3
1.4 Ear as a biometric	6
1.5 Problem statement	9
1.6 Thesis structure	9
2 Indexing iris images	10
2.1 Introduction	10
2.2 The BWT and Iris Indexing	11
2.2.1 BWT Context Partitions	12
2.3 Proposed Approach	13
2.3.1 Pre-Processing	15
2.3.2 Pattern Finding	16
2.3.3 Indexing	18
2.4 Results	19
2.4.1 Experiments on IrisCodes	21
2.5 Conclusion	24
3 Classification of ear images	25
3.1 Introduction	25
3.2 History of ear classification	26
3.2.1 Limitations of Iannarelli system	29
3.3 Proposed approach	29
3.3.1 Overview of ear classification system	30
3.4 Ear detection	31
3.4.1 AdaBoost	32
3.4.2 Cascade of classifiers	34

3.4.3	Forward Feature Selection	35
3.5	Ear segmentation	37
3.5.1	Boosted Edge Learning	38
3.5.2	Contour extraction and segmentation	40
3.6	Estimation of shape of the ear using an ovoid model	42
3.6.1	Ovoid model	42
3.6.2	Parameter estimation of the ovoid	44
3.6.3	Gradient Descent Algorithm	45
3.7	Classification	47
3.8	Experimental Evaluation	48
3.8.1	Databases Used	48
3.8.2	Ear detection	51
3.8.3	Ear segmentation	56
3.8.4	Classification results	58
3.9	Conclusion and future work	62
4	Conclusion and Future Work	64
4.1	Iris Indexing	64
4.2	Ear Classification	65
A	Sample ear images	66
A.1	Classification based on quality	66
A.2	Classification based on shape	67
	References	70

List of Figures

1.1	Fingerprint scanner at the entrance of Universal studios (Image taken from entertainment planet)	2
1.2	Block diagrams of enrollment, verification and identifications modes of a biometric system	3
1.3	Cross section of iris (adapted from 2005 Mission for Vision)	4
1.4	Block diagram of iris recognition system	5
1.5	The six auricular hillocks in a six week old human embryo (taken from [1])	6
1.6	Ear Terminology	7
1.7	Ear biometric system	8
2.1		14
2.2	Positions of pattern P_1 on the normalized binary image.	18
2.3	The proposed indexing scheme. The vertical regions correspond to the segments numbered 1 to K.	19
3.1	A typical classification scheme.	26
3.2	Ear identification card (front view) used by Iannarelli (Image taken from [2]).	26
3.3	Level-1 classification.	27
3.4	Ear classification scheme based on shape used by Iannarelli (Image taken from [2])	28
3.5	Classification based on position (a)normal set, (b)high set, (c)low set) used by Iannarelli	28
3.6	Block diagram of the classification scheme.	30
3.7	Figure showing simple Haar features. A)Two rectangular B)Three rectangular C)Four rectangular features	33
3.8	Figure showing cascade of classifiers.	35
3.9	Comparison of Naive Adaboost feature selection with Forward Feature selection ((b) taken from [3]).	36
3.10	Figure showing the working of PBT.	39
3.11	Post processing steps used for segmenting the ear.	41
3.12	Effect of γ and a/b on ovoid.	43
3.13	Figure showing segmented ear and it's contour estimated using Alvarez <i>et al.</i> model [4].	45

3.14	Figure showing the efficiency of the modified gradient descent algorithm. (a) Gradient descent algorithm converged at local minimum. (b) Parameters at global minimum achieved by modified gradient descent algorithm.	48
3.15	Figure showing the proposed ear classification scheme.	49
3.16	Sample images of a single user in WVU database.	50
3.17	Sample images in UND database.	51
3.18	Sample images of a single user in Magna database.	52
3.19	Figure showing images with detected ears.	53
3.20	Figure showing images with false rejects.	53
3.21	Figure showing images with false detections.	54
3.22	Images showing detection performance at various scales	54
3.23	Images showing detection performance in bottom-up scaling approach	56
3.24	Training images for BEL (ear image along with the labeled helix edge)	57
3.25	Figure showing different types of segmentation outputs.	58
3.26	Figure showing cases where gradient error occurred.	59
3.27	Figure showing examples of cases where a human failed to assign the correct class.	61
3.28	Figure showing errors of individual modules.	63
A.1	Sample images showing images with no ear occlusions.	66
A.2	Sample images showing images with partial ear occlusions.	67
A.3	Sample images showing images with fully occluded ears.	67
A.4	Sample images showing round shape ears.	68
A.5	Sample images showing rectangle shape ears.	68
A.6	Sample images showing oval shape ears.	69
A.7	Sample images showing triangle shape ears.	69

List of Tables

2.1	BWT forward transformation for $T = mississippi$. M' is the matrix of cyclic rotations. M is the matrix after sorting. * shows the row in M corresponding to the original sequence.	13
2.2	Performance of the proposed indexing scheme for a 4-bit pattern (P_1)	20
2.3	Performance of the proposed indexing scheme for a 4-bit pattern (P_2)	21
2.4	Performance of the proposed indexing scheme for an 8-bit pattern (P_1)	21
2.5	Performance of the proposed indexing scheme for an 8-bit pattern (P_2)	22
2.6	Noisy test data: Performance of the proposed indexing scheme for a 4-bit pattern (P_1)	22
2.7	Noisy test data: Performance of the proposed indexing scheme for a 4-bit pattern (P_2)	23
2.8	Noisy test data: Performance of the proposed indexing scheme for an 8-bit pattern (P_1)	23
2.9	Noisy test data: Performance of the proposed indexing scheme for an 8-bit pattern (P_2)	23
2.10	Performance of the proposed indexing scheme for a 4-bit pattern (P_1) on IrisCodes	24
3.1	Number of ear detects and rejects at each scale	55
3.2	Ground truth established by manual classification of ear images	59
3.3	Confusion matrix for perfectly segmented images	60
3.4	Confusion matrix for perfect + over segmented images	60
3.5	Ground truth after correcting for human errors.	62
3.6	Confusion matrix for perfectly segmented images after correcting human errors . .	62
3.7	Confusion matrix for perfect and over segmented images combined after correcting for human errors.	62

Chapter 1

Introduction

1.1 Biometrics

Human recognition systems have gained great importance recently in a wide range of applications like access, control, criminal investigation and border security [5]. Traditionally these applications are based on passwords or simple authentication tokens such as ID cards which are susceptible to security threats especially when the passwords are weak (easily predictable), shared with unauthorized users or when ID cards are lost [5].

Given the disadvantages associated with these systems, biometrics has become an area of great research interest. The word biometrics originating from the Greek words [6] “bios” meaning life and “metrics” meaning measure refers to recognition of individuals based on their physical or behavioral traits such as fingerprints, iris, face, ear and voice. Recognition systems based on biometrics have an edge over other traditional systems as they are based on ‘who you are’ rather than ‘what you know or have’ (that includes passwords or I.D cards) [7].

Not all physical or biological traits may be used as biometric traits [5]. An ideal biometric trait should have the following characteristics:

- **Universality:** for a trait to be considered as a biometric trait, all the individuals must possess this trait.
- **Distinctiveness:** the trait should have sufficient variation across individuals.
- **Permanence:** the trait should be relatively invariant (with respect to the matching criterion)

over a period of time.

- **Collectability:** the trait should be compatible enough to be quantitatively collected from different individuals.

Some of the commonly used biometric traits that more or less satisfy some of these characteristics include fingerprint, hand geometry, face, iris, voice, etc. Each biometrics has its own strengths and limitations, and the choice of a biometric trait depends on the application [5]. Also, sometimes a combination of these traits can be used to get a better performance [8].

1.2 Biometric Systems

Having identified the biometric trait for a specific application, a biometric based identification system is then developed and usually operates in two distinct modes, enrollment and authentication (verification/identification) modes [9]. In the enrollment mode, the biometrics traits are captured from the subjects and relevant information is extracted from these traits by the feature extractor and stored in a database. Each database entry is labeled with the identity of an individual. In identification mode, the system extracts the features from the raw data of the subject and compares it with the features in the database to determine the identity of the subject, while in verification the features from the raw measurements of the subject are compared with the enrolled features of the claimed identity.



Figure 1.1: Fingerprint scanner at the entrance of Universal studios (Image taken from entertainment planet)

As a general example, in entertainment parks like universal studios a single ticket could be used as a multiple entry pass for a year. In such cases there is a possibility that the tickets could be misused by sharing among the individuals . Universal studios thus started adapting finger-print based biometric identification system. During the first visit, the biometric system works in the enrollment mode, where fingerprint is collected from the user and stored it with an associated ID (usually ticket). During the subsequent visits, the system operates in the authentication mode and the access is provided based on the confidence score. The architecture of a simple biometric system is indicated in Figure 1.2.

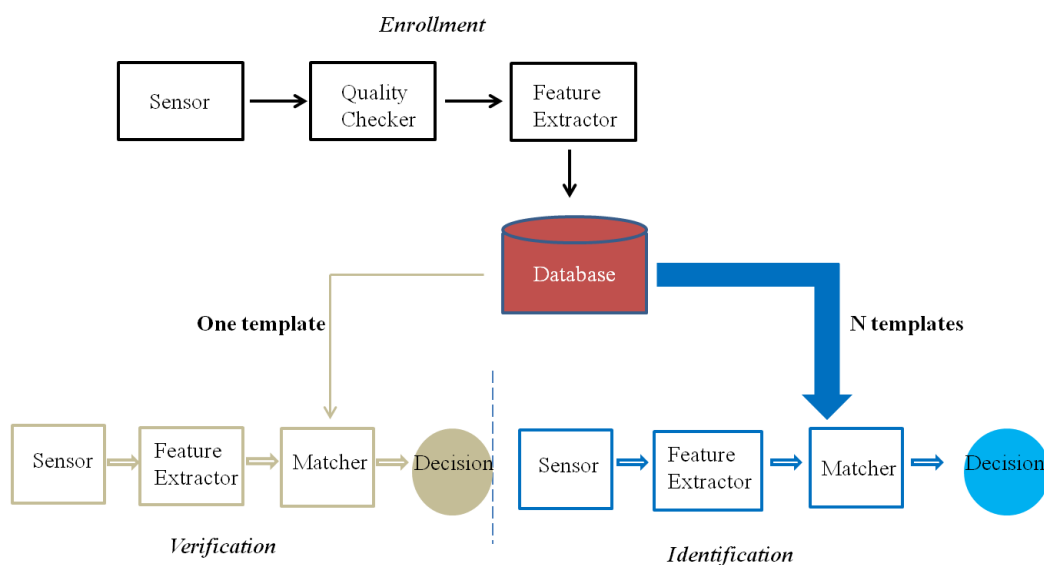


Figure 1.2: Block diagrams of enrollment, verification and identifications modes of a biometric system

1.3 Iris as a biometric

Iris is a thin circular diaphragm, between the cornea and and the lens of the human eye surrounding the pupil[10]. Human iris is usually expected to form during the third through the eight month of gestation, with the pigmentation to continue for a year after the birth of the baby. An iris as reported in literature usually has four layers [11]:

1. Posterior pigment epithelium

2. Dilator pupillae
3. Stroma
4. Anterior border layer(visible layer)

The posterior pigment epithelium, being the closest layer to the pupil controls the amount of light leaking through the iris. The iris controls the size of the pupil through the sphincter and dilator muscles. Iris acts like a knob to control the pupil size. The pupil is contracted during bright light by sphincter and enlarged during dark light by dilator muscles respectively. Literature reports indicate that irides usually have an average diameter of around 12mm, with the size of the pupil varying anywhere between 10 to 80 % of the diameter of the iris enlarging the size of pupil [12]. The stroma is a layer of loosely connected tissue containing collagen, melanocytes, mast cells and macrophages. The anterior border layer is a dense collection of fibroblasts and melanocytes. It is deficient in some areas, causing crypts and heavily pigmented in other areas causing nevi [11]. The anterior surface of the iris is further divided into the ciliary zone and the pupillary zone by a thickened region called the collarette which forms the visible iris pattern. The color of the iris is usually determined by each of these layers together with a combination of scattering effects and pigmentation [11]. Some other factors that could play a role in determining the iris pattern include, crypts, pigmented areas, moles, freckles etc.

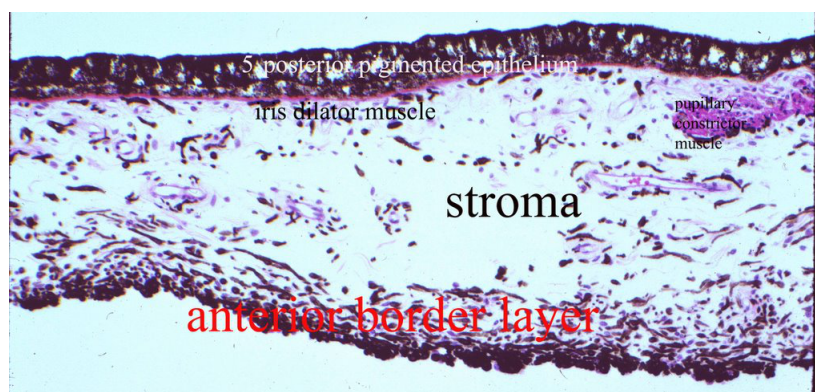


Figure 1.3: Cross section of iris (adapted from 2005 Mission for Vision)

The rich structure, apparent stability and individuality of the iris structure account for its use as a biometric. Some other factors, that contribute for iris to be a useful biometric include:

- The iris is not easily affected by environmental changes.
- People with identical faces; even genetically identical twins have been observed to have different irides.
- Daugman's algorithm based commercially deployed iris recognition systems have exhibited a false non-match rate of less than 1% [13].
- Having known that the iris lies within the eye, it is relatively difficult to alter the structure of the iris.

Iris recognition systems are built to take the advantage of its rich texture for identification, and the goal of such a system is to extract, represent and compare the iris texture. In an image processing, pattern recognition point of view these steps can be defined as segmentation, feature extraction and matching and are the building blocks of an iris recognition system. A typical iris recognition system is as indicated in Figure 1.4.

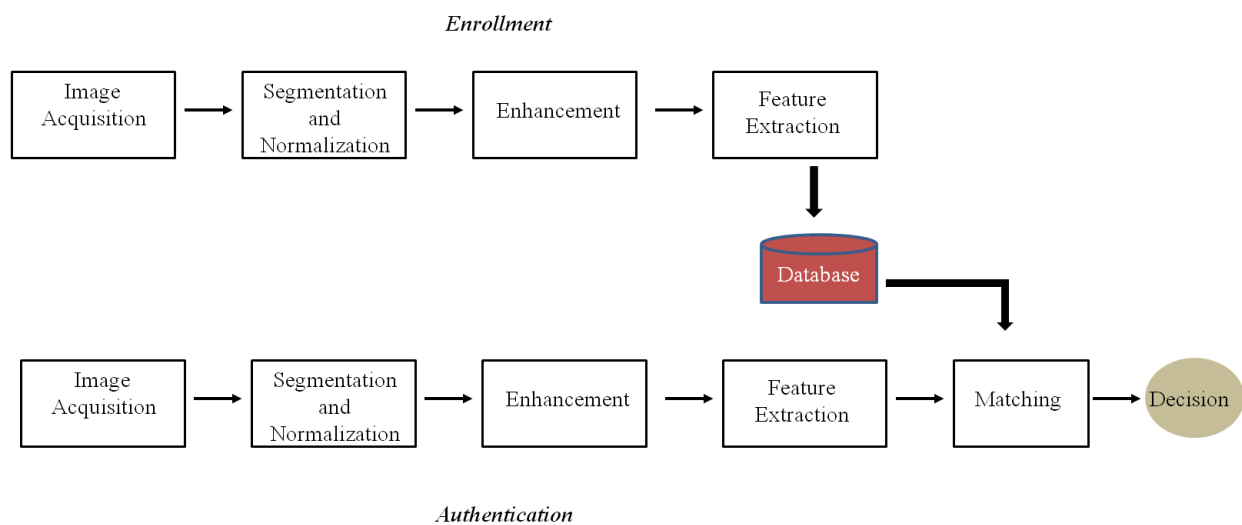


Figure 1.4: Block diagram of iris recognition system

Images are acquired using near-infrared cameras in the 700-900nm wavelength. Use of NIR wavelengths over visible wavelengths helps in illuminating the rich and complex features even for dark-colored irides. The common stages during enrollment and authentication are image acquisition, segmentation, normalization, enhancement and feature extraction. During enrollment, the

features extracted are stored in the database, while during authentication they are used to match against the stored templates and a decision is made accordingly.

The review of iris recognition techniques in literature suggests three major approaches to iris segmentation and three categories of iris encoding techniques. The three mostly used segmentation techniques are 1) Integro-differential operator [14] 2) Hough transform [10] 3) Active shape models [15]. The three basic categories for encoding the iris are [16] a) appearance based [17][18] b) filter based [14][19] and c) feature based [20][21]. The features thus extracted are some times encoded as a binary image (IrisCode) which is compared against another using hamming distance, euclidean distance etc.

Besides having tremendous advantages over other biometrics, there are some shortcomings like sensor cost, need for user cooperation, need for high quality images, susceptibility to spoofing, etc.

1.4 Ear as a biometric

The ear first appears during the fifth week of the embryonic life as in the shape of six individual hillocks [22]. These hillocks progress and combine to form the structure of the ear. The hillocks for a six week old embryo can be seen in Figure 1.5. Starting from the year 1882 [22], there have been many theories, some them proposed by Davis [23], Wood [24], Streeter [22], etc., as how these six hillocks progress and combine to form the ear. According to these theories, the uniqueness of the ear across individuals is due to the difference in growth and alignment of these hillocks [25].

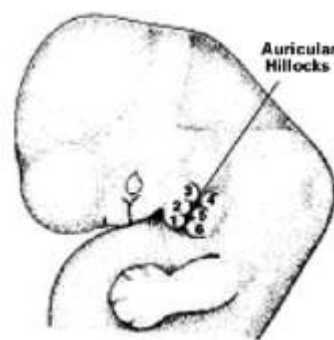


Figure 1.5: The six auricular hillocks in a six week old human embryo (taken from [1])

The commonly used terminology for the external ear can be seen in Figure 1.6. The most

prominent rim on the ear is the helix. Parallel to it is another curved rim named antihelix. The antihelix divides into two crura at the top, between which a triangular depression is formed, known as triangular fossa. The space between the helix and antihelix is the scapha. The concha is a deep cavity, which is surrounded by antihelix. The concha extends into the lower part as the incisura. The incisura is surrounded by two bumps namely tragus and antitragus. As mentioned, due to difference in growth and alignment of the hillocks, the structure of the ear can be used as a biometric.

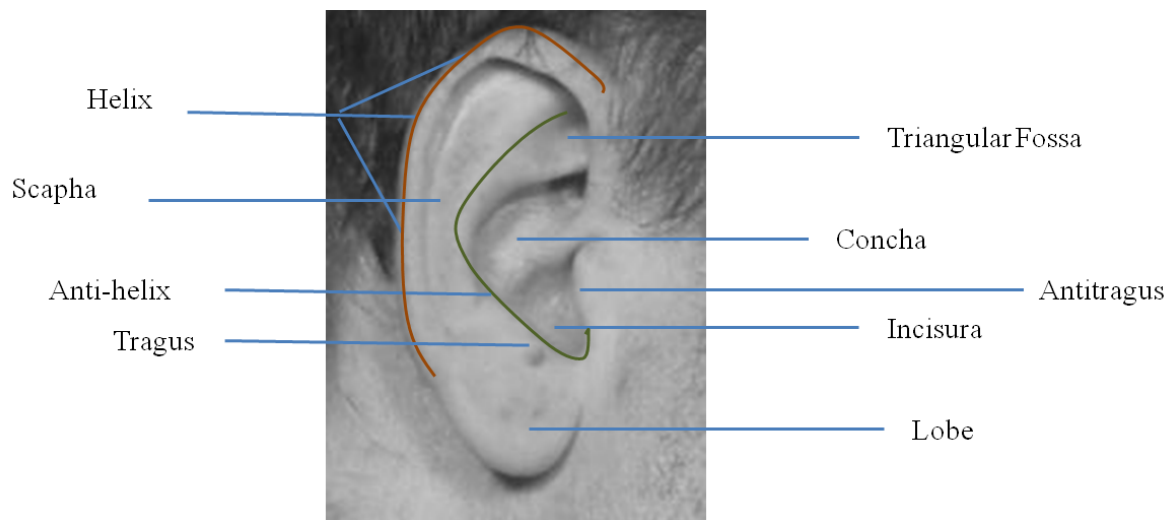


Figure 1.6: Ear Terminology

Till this point of time, there is no clear evidence that ears are unique, although no evidence to the contrary has been presented. Alfred Iannarelli [2], a criminologist, photographed and examined over 10,000 ears and demonstrated that they could be used for accurate human recognition. Ear, as a biometric was first used by Alphonse Bertillon [26]. He used ear height, shape and degree of adherence of the lobe and prominence and inclination of antitragus as some of the anthropometric measurement's to identify a person. However, it was Alfred Iannarelli [2], who developed a practical recognition system based only on ear. Iannarelli's classification scheme was based on gender, race, anthropometric measurements, shape, size, and position of the ear for recognition. After several years, in 1998, Burge and Burger automated the ear recognition system using Voronoi diagrams [27]. A typical 2D ear recognition system can be seen in Figure 1.7.

The common stages during both enrollment and authentication are image acquisition, detec-

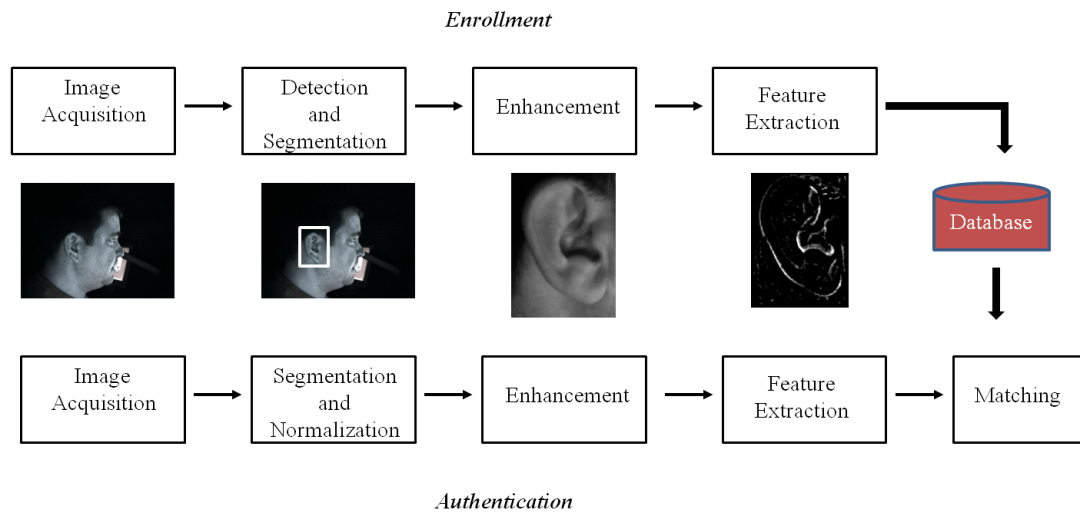


Figure 1.7: Ear biometric system

tion, segmentation, enhancement and feature extraction. The review of various feature extraction methods in the literature suggests two major approaches [28]: geometrical approach and global approach. Geometrical approach consists of methods based on vornoi diagrams [27], geometrical parameters method [29], etc., and global approach consists of methods based on force field transformation [30], PCA [31], etc. Apart from using only 2D images, a multimodal technique based on 2D and 3D ears have been researched [32] [33]. Also, based on video frames, a 3D ear model was built using structure from motion(SFM) and shape from shading(SFS) techniques [34].

Even though there is no commercial ear recognition system, the ear has a good scope of becoming a reliable biometric because of it's advantages like:

- Except the part of lobule, other structures in the external ear do not change much during human life.
- Change in facial expressions, occlusions due to beard, and the presence of glasses does not easily affect the performance of the ear recognition system.
- Twins who have identical faces, have different ears.
- Ear can be imaged from a distance without human cooperation.

- Ear can be captured easily using an ordinary camera or it can be extracted from existing face profile images, which makes the system cost low.

Besides these advantages, it does have some challenging problems like occlusion due to hair, variations due to pose , lighting conditions, etc.

1.5 Problem statement

In most biometric identification systems, the input biometric data has to be compared against that of every identity in the database in order to determine the identity of the input. A major problem with this approach is the impact on response time which can increase significantly with the size of the database. In certain applications such as real time monitoring, this delay may not be acceptable. For rapid identity retrieval, the database can be divided into subsets using two methods. The first method is classification, where the entire database is divided into a small number of classes based on features extracted from the biometric trait. During authentication, the query is assigned with one (or more) of these classes and the matching is done against only these subset of images. Note that the classes need not be mutually exclusive. The second method is indexing, where every entry in the database is assigned an index code based on which a small subset of images in the database with similar index values as the query image are retrieved and matched. The contribution of this thesis is in designing novel indexing technique for iris and classification technique for ear.

1.6 Thesis structure

In chapter 2, indexing iris images using burrow wheeler transform is discussed and experimental on the CASIA version 3 iris database are reported. In chapter 3, classification of ear images is discussed and experimental results on MAGNA database are reported. Finally, in chapter 4 contributions of this thesis are discussed along with the future work.

Chapter 2

Indexing iris images

2.1 Introduction

Biometric identification is the process of associating an identity to the input biometric data by comparing it against the enrolled identities in a database [5]. In many systems, this comparison is undertaken in an exhaustive manner, i.e., the input data of an individual is compared against *all* enrolled data in order to determine the identity of the individual. The false positive identification rate as well as the matching time can increase with the size of database [35]. So, a procedure that reduces the search space is necessary in order to increase the efficiency of the system.

In this chapter, we concern ourselves with the problem of iris recognition. Due to its rich texture and the observed statistical independence of the iris codes of different eyes (the iris code is the feature template extracted from the iris), the iris is considered to be a very reliable biometric modality for identification [14]. Thus, iris identification has been shown to be very effective in large databases. However, the processing and matching of non-ideal irides can be computationally expensive and, therefore, methods to speed up the identification process are necessary [36]. The search space of an identification operation can be reduced by rapidly choosing a subset of iris images from the database before matching. This can be accomplished using two methods: classification and indexing. In a classification scheme, the database is partitioned into a certain number of classes based on the physical and structural properties of the iris. During identification, the input image is assigned to one of the classes and is subsequently compared against those irides in the database which are in the same class. Existing classification schemes are mainly based on color

[37] or texture [38, 39, 40] details of the iris. The main limitation of the classification approach is the potentially uneven distribution of the identities across the classes and the effect of noise in determining the correct class.

In the indexing approach, each iris is assigned with an index value. However, the index values of two iris images of the same individual may not be the same because the process of data acquisition and processing is subject to noise. Therefore, indexing systems retrieve those identities whose indices are similar to the index value of the input data. The input image is matched only against the retrieved identities thereby reducing the identification time and, potentially, the identification error rate. Indexing of iris images is often based on iris codes [41, 42] or key features such as SIFT [43]. Hao et al. [41] proposed a fast search algorithm for a large fuzzy database that stores iris codes. Mehrotra et al. [43] used SIFT key points to obtain indices of the hash table during indexing and retrieved a list of iris images from the hash table by casting votes. Work on indexing of iris images by Mukherjee et al. [42] used both iris codes and features extracted from the iris texture and demonstrated that about 80% of the input images could be correctly identified with a low penetration rate of 8%.

In this work, we present a new method for indexing iris images based on the context clustering property of the Burrows-Wheeler Transform (BWT). The next section provides a brief description of the BWT. Section 2.3 then describes the general procedure for indexing iris images. Results are presented in 2.4. Section 2.5 concludes the chapter.

2.2 The BWT and Iris Indexing

The Burrows-Wheeler Transform (BWT) [44] performs a permutation of the characters in a sequence, such that characters in lexically similar contexts are in proximity. Given an input text string $T = t_1, t_2, \dots, t_m$ over some finite alphabet $\Sigma = \alpha_1, \alpha_2, \dots, \alpha_{|\Sigma|}$, the forward BWT is done in three steps:

1. Form m permutations of T by cyclic rotations of the characters in T . The permutations form a $m \times u$ matrix M' , with each row in M' representing one permutation of T ;
2. Sort the rows of M' lexicographically to form another matrix M . M includes T as one of its

rows;

3. Record L , the last column of the sorted permutation matrix M , and id , the row number for the row in M that corresponds to the original sequence T .

As an example, suppose $T=\text{mississippi}$. Let F and L denote the array of first and last characters in M respectively (see Table. 2.1). Then, $F=\text{iiiimppsss}$ and $L=\text{pssmipissii}$. The output of the transformation will be the pair: $(L, id)=(\text{pssmipissii}, 5)$. Generally, the effect is that the contexts that are similar in the input sequence T are brought closer together in the output sequence L . (Notice how all the rows that started with the same context, i.e., same starting symbols, are grouped together in L , e.g. the rows that start with **issi**, **ssi**, etc.). This similarity in nearby contexts is the key to improved compression performance in BWT-based compression systems [44, 45]. The BWT is reversible. It is quite striking that given only the (L, id) pair, the original sequence can be recovered exactly. Adjero et al. presented a detailed treatment of the BWT in the recent book [45], including its performance, connection with suffix trees and suffix arrays, and various applications. In this work, our focus is on studying the potential of the sorted contexts of the BWT in indexing iris images.

2.2.1 BWT Context Partitions

With the suffix sorting stage of the BWT, suffixes that are similar in the original sequence will be placed together in the sorted matrix of rotations of the BWT. Thus the BWT output symbols in the same region of the output array L are likely to have similar following suffixes, that is, similar forward contexts in the original sequence. Therefore, the output stream can be partitioned into different segments based on the similarity in the symbol contexts. For example, consider the sequence $T=\text{mississippi}$ used in Table 2.1, with $L=\text{pssmipissii}$. For order-1 context partitions, all the symbols in the L array with the same starting symbol in the corresponding rows in the F array will be in the same context. Thus, the first four symbols (p,s,s,m) in L will be in the same context partition - the i -partition. This context clustering (or partitioning) property of the BWT can be exploited this in indexing iris images.

Consider the iris image after preprocessing to remove the effect of, say, illumination and noise in the image. Suppose we can locate patches in the iris image that are similar in their textural

Table 2.1: BWT forward transformation for $T = mississippi$. M' is the matrix of cyclic rotations. M is the matrix after sorting. * shows the row in M corresponding to the original sequence.

M'	M	(F)	(L)
mississippi	imississipp	i	p
ississippim	ippimississ	i	s
ssissippimi	issippimiss	i	s
sissippimis	ississippim	i	m
issippimiss	mississippi	m	i*
ssippimissi	pimississip	p	p
sippimissis	ppimississi	p	i
ippimississ	sippimissis	s	s
ppimississi	sissippimis	s	s
pimississip	ssippimissi	s	i
imississipp	ssissippimi	s	i

characteristics and that occur a certain number of times in the iris [42, 46]. We speculate that the geometric relation between the positions of occurrence of the patches in the iris image along with the specific pattern within these patches can form the basis for constructing an effective index for the iris images. The key challenge is to develop a fast method for determining such repeated patches and for locating the positions of occurrence of such patches on the iris image. This can be viewed as a pattern matching problem, which is a major application of the BWT [45]. In this work, we use the BWT as the basis for identifying the repeated patches and their positions of occurrence in the iris image.

2.3 Proposed Approach

The iris exhibits a very rich texture due to the numerous anatomical entities on its stroma. The texture in an iris is believed to be relatively stable (a claim that has been challenged in the recent literature) and have been observed to vary in its intricate detail from one eye to another [47]. This facilitates the design of an identification system based on this texture. Apart from recognition, the

iris texture can also be used for indexing. In this section, we design an indexing scheme based on the distribution of the patterns formed because of the texture.

A classical iris recognition system has an image acquisition module, an iris localization or segmentation module, a normalization module, an encoding module and a matching module. The acquisition module obtains an image of the iris in the near-infrared (NIR) spectrum. During segmentation, the iris is localized and isolated from the sclera, pupil, eyelids and eyelashes. In the normalization module, the segmented iris is unwrapped and converted from the cartesian domain into a doubly dimensionless (polar-like) domain resulting in a rectangular entity that is subjected to subsequent processing. Normalization is performed using Daugman's rubber sheet model [47]. Fig. 2.1(a) shows an example of a normalized iris image.



(a)



(b)



(c)

Figure 2.1: (a) Example of a normalized iris image. (b) Normalized iris image after filtering. (c) Normalized binary iris image

In the proposed algorithm, the normalized gray scale iris image is first converted to a binary

image. Next, a certain horizontal n -bit pattern is chosen and the locations of these patterns are found in the iris image. Then, the normalized iris image is divided into vertical segments and, based on the maximum occurrence of the n -bit pattern among these segments, the iris image is assigned an index value based on the segment number.

2.3.1 Pre-Processing

In cases where thresholded images are used, changes in illumination can have a significant impact on the results. Thus, a pre-processing technique is used to mitigate this problem.

Illumination variation

The iris patterns are relatively sensitive to illumination variations, especially when dealing with binary images. To overcome this limitation, we adapt the color ratio model [48] to generate illumination-invariant versions of the iris images.

Assume that the illumination variation in a scene is such that the illumination is locally constant for a given spatial position in the image. Then, the average intensity value over a small region should not change much within the region. Thus, the ratio of a pixel value in a region to the mean intensity value over the region or window (for instance, using a small window of size W) should be close to one.

Let $h(x, y)$ be a window of size $W \times W$ around the pixel $I(x, y)$. The color ratio¹ is defined as follows:

$$R(x, y) = \frac{I(x, y)}{\mu(h(x, y))}, \quad (2.1)$$

where $\mu(h(x, y))$ is the mean pixel intensity in the window.

However, when there is a definite boundary (or edge variations), the ratio will move away from 1. Thus, essentially, the ratio encodes the boundary variations (or edge variations) in the image. Because such boundaries are invariant to illumination changes, we expect a robust result against potential illumination variations in the iris image. Fig. 2.1(b) shows the normalized iris image of Fig. 2.1(a) after filtering using the color ratio model.

¹Although the term “color” is used for historical purposes, the measure in this work is used on grayscale images

Conversion to binary image

To convert the gray scale image to binary image the cumulative distribution function of the normalized iris image is calculated. Based on the cumulative distribution function, a threshold is determined and values of pixels greater than this threshold are considered as ‘1’ and those less than this threshold are considered as ‘0’.

Consider a discrete gray scale image $J(x)$ and let t_i be the number of occurrences of gray level i . The probability of occurrence of a pixel of level i in the image is given by

$$p_x(i) = p(J(x) = i) = \frac{t_i}{t} \quad 0 \leq i < L, \quad (2.2)$$

where L is the total number of gray levels in the image and t is the total number of pixels in the image. Here, $p_x(i)$ represents the normalized histogram of pixel value i .

The cumulative distribution function corresponding to p_x is given by

$$F_x(i) = \sum_{j=0}^i p_x(j). \quad (2.3)$$

Now consider, the value of i where $F_x(i)$ is $\frac{1}{2}$. The values of pixels above i are considered as ‘0’ and below i are considered as ‘1’. The result of this procedure on the example iris image of Fig. 2.1(b) is shown in Fig. 2.1(c).

2.3.2 Pattern Finding

After obtaining the binary image, we identify a horizontal n -bit binary pattern which is consistent among all the images. This can be done by choosing the pattern with the least value of coefficient of variation. The coefficient of variation (CV) for a particular n -bit pattern is simply the ratio of the mean and standard deviation of the number of occurrences of that pattern across all the training images. The coefficient of variation for each of the 2^n patterns has to be calculated using the same training images and then a pattern (say P_i) with the least CV has to be chosen. Thus, the key problem is to count the number of occurrence of each n -bit horizontal pattern in the normalized binary image and to locate the positions of each occurrence. This is essentially a problem of pattern matching, which is performed using the BWT.

First, the 2D binary image is converted into a 1D sequence by simply concatenating the rows, from the first row to the last. Each row is terminated using a special character, such that the search for patterns will not report patterns that span two different rows. Then, the 1D sequences from the individual iris images in the database are further concatenated to form one long sequence, T . A special end of file marker is used to indicate the end of each iris image. The BWT is then applied to T , the database sequence, producing the L array, and an index id as the output. Pattern matching is then performed on the database via the BWT output pair (L, id) . Notice that since the size of each iris image is known, and the rows are of fixed length, a position in T uniquely determines the corresponding image and the (x, y) position in that image.

Consider an n -bit binary pattern, P . To search for P in T (via L), we use an auxiliary mapping array, W defined as follows:

$$\forall i : 1 \leq i \leq n, T[i] = L[W^i[id]]$$

where $W^1[x] = x$, $W^{i+1}[x] = W[W^i[x]]$, and id is the position in F of the first character of the text.

W provides a one-to-one mapping between the BWT F and L arrays, viz., $F[i] = L[W[i]]$. Using the earlier example with mississippi, we will have $W = [5, 7, 10, 11, 4, 1, 6, 2, 3, 8, 9]$. Construction of W is performed in linear time with respect to the length of T .

The sorted list of suffixes of T are available from M , the sorted matrix of rotations. If a pattern appears in the database T , it must appear at the start of one or more rows in M . Given that the rows in M are sorted, all occurrences of such a pattern will be located next to each other in M . In practice, we access M via F , the array of first characters. F in turn is accessed from L (the BWT output) via the mapping array W . Then, the search is performed using a modified binary search algorithm [49, 45], using L , W , and id . The procedure below shows the comparison function used by the modified binary search algorithm, where i denotes the row in the rotation matrix M being compared to the pattern. Let s be the string representation of the row. The function will return 0 if P is a prefix of s , < 0 if $P < s$, and > 0 if $P > s$

Using the above, each valid n -bit pattern is searched in T (the database). For each pattern, the occurrence counts and the locations of the pattern in T , where each occurred are reported. Using

Algorithm 1. String comparison function for Binary Search

```

BINARY-SEARCH-STRCMP( $P, W, L, i$ )
1  $m \leftarrow \text{LENGTH}(P); j \leftarrow 1; i \leftarrow W[i];$ 
2 while  $m > 0$  and  $L[i] = P[j]$  do
3    $i \leftarrow W[i]; m \leftarrow m - 1; j \leftarrow j + 1;$ 
4 end while
5 if  $m = 0$  then return 0;
6 else return  $P[j] - L[i];$ 
7 end if

```

the least coefficient of variation in their occurrence counts over all the images in the database, the stable patterns are determined. In this work, two patterns with the least values for CV are selected as the basis of our index and denoted P_1 and P_2 . Fig. 2.2 shows the positions of occurrences of pattern P_1 in the sample binary image used in the running example.

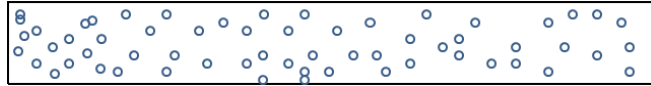


Figure 2.2: Positions of pattern P_1 on the normalized binary image.

2.3.3 Indexing

The indexing scheme has two stages: i) Training and ii) Testing.

i) Training is an offline process, in which every training image, x , in the database is assigned a particular index I_x .

ii) Testing is an online process, in which the probe image, y , is assigned a particular index I_y . The probe image is then matched only with those set of images in the database with I_y as their index value, along with those with indices similar to I_y .

In this work, the indexing scheme is based on the *count* of occurrence of the n -bit binary patterns (say P_1) in the normalized binary iris image (x). Thus, during training, all the positions where P_1 occurred in image x are identified. The image is divided into K segments along the

vertical (i.e., radial) direction. The iris image is then assigned an index with segment number I_x , where the segment I_x has the maximum number of occurrences for pattern P_1 . The division of the iris image into segments in the radial direction compensates for the affine transformations due to changes in viewing angle. During testing, a procedure similar to that used for training is followed. However, in this case instead of assigning a single index corresponding to a single segment number I_x , the image is assigned four indices which correspond to the top 4 segments with maximum counts of occurrence of pattern P_1 . Now, the test image (probe) is compared only against those images in the database that have these 4 indices as their index, starting with the first index. Fig. 2.3 shows the basic indexing scheme described above.

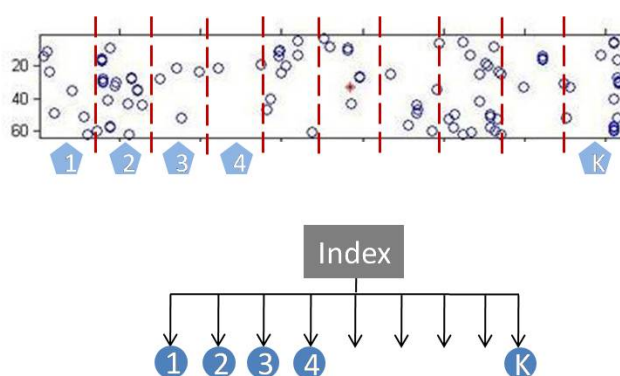


Figure 2.3: The proposed indexing scheme. The vertical regions correspond to the segments numbered 1 to K.

2.4 Results

The CASIA version 3 iris image database was used in the experimental evaluation of the iris indexing technique proposed here. The images in this database were segmented and normalized using the algorithms proposed by Shah and Ross [50]. In this work, the left normalized iris images are used which contain images of resolution 360 x 64 pixels pertaining to 249 subjects. In some cases, the normalized images of a subject were not successfully generated by the algorithm. Only those users with at least two normalized sample images per eye were considered in this work (a total of 189 users). The images for each subject were equally divided into training and test sets if the number of images for that subject was even. The training set had an extra image from a subject

if the number of images for that subject was odd.

Experiments are conducted using 4-bit and 8-bit binary patterns. The two patterns P_1 , P_2 with the least values of coefficient of variation are chosen (P_1 has the overall least value) as per our discussion in section 2.3.2. We studied the performance of various window sizes in the color-ratio model, viz., $W=3, 5, 7$ and for different number of segments, viz., $K= 4, 6, 8, 10, 12$. The hit rate (R_h) and penetration rate (R_p) for different combinations of W and K were noted to study the performance of the technique. In order to find the optimal trade-off between hit rate(R_h) and penetration rate (R_p), we defined a new variable γ , which combines both the hit rate and penetration rate:

$$\gamma = \sqrt{(R_h) * (1 - R_p)}. \quad (2.4)$$

Experiment 1: The performance of the indexing scheme for 4-bit patterns P_1 and P_2 can be seen in tables 2.2 and 2.3, respectively. For pattern P_1 , the best value of $\gamma = 90.90\%$ occurred at a hit rate of 99.83% and a penetration rate of 17.23% with $W=3$ and $K=10$. For pattern P_2 , the best value of $\gamma = 84.52\%$ occurred at a hit rate of 99.50% and a penetration rate of 28.19% with $W=3$ and $K=8$.

Table 2.2: Performance of the proposed indexing scheme for a 4-bit pattern (P_1)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
$K=4$	100	23.51	100	26.24	100	48.96
$K=6$	100	21.77	100	36.44	100	57.82
$K=8$	99.83	21.83	100	39.18	100	59.56
$K=10$	99.83	17.23	100	26.80	100	33.62
$K=12$	99.17	18.61	100.0	28.52	100.0	42.16

Experiment 2: Similar experiments were conducted for 8-bit patterns with varying W and K . The performance of the indexing scheme with varying window sizes W and for different values of K for patterns P_1 and P_2 can be seen in tables 2.4 and 2.5, respectively. For pattern P_1 , the best value of $\gamma = 89.30\%$ occurred at a hit rate of 98.5% and a penetration rate of 19.03% with $W=3$ and $K=10$. For pattern P_2 , the best value of $\gamma = 89.09\%$ occurred at a hit rate of 96.68% and a

Table 2.3: Performance of the proposed indexing scheme for a 4-bit pattern (P_2)

	$W=3$		$W=5$		$W=7$	
	R_h	R_p	R_h	R_p	R_h	R_p
$K=4$	100	29.38	100	39.58	100	48.44
$K=6$	99.50	28.19	100	48.39	100	51.50
$K=8$	100	32.30	100	55.26	100	55.84
$K=10$	99.66	39.55	100	56.84	100	59.26
$K=12$	99.83	38.82	100.0	58.44	100.0	59.31

penetration rate of 17.90% with $W=3$ and $K=10$.

Experiment 3: In order to test the robustness of the algorithm, noise was added to the binary images by randomly inverting 50% of the bits for the images in the test set and repeating the aforementioned experiments. The performance of the indexing scheme with varying window sizes W and number of segments K for patterns P_1 and P_2 are shown in tables 2.6,2.7,2.8 and 2.9.

Table 2.4: Performance of the proposed indexing scheme for an 8-bit pattern (P_1)

	$W=3$		$W=5$		$W=7$	
	R_h	R_p	R_h	R_p	R_h	R_p
$K=4$	100	23.26	100	23.96	100	25.43
$K=6$	99.66	20.99	99.50	24.77	99.83	25.65
$K=8$	99	19.54	99.33	23.34	97.68	24.05
$K=10$	98.50	19.03	96.19	21.88	95.19	22.89
$K=12$	96.52	19.66	95.69	20.04	93.54	23.62

From the above experiments, we observe that the hit rates and penetration rates using 4-bit and 8-bit patterns are affected when the binary images are very noisy. Currently, we are looking at ways to impart noise resilience to the index patterns.

2.4.1 Experiments on IrisCodes

Similar experiments were conducted for 4-bit pattern with varying W on IrisCodes. Before discussing the performance of the proposed indexing scheme on IrisCodes, a brief summary on how the IrisCodes are generated is discussed below.

Table 2.5: Performance of the proposed indexing scheme for an 8-bit pattern (P_2)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
$K=4$	100	24.28	100	57.73	100	59.65
$K=6$	99	22.49	100	47.16	100	55.95
$K=8$	98.01	21.17	99.83	40	100	48.54
$K=10$	96.68	17.90	100	38.79	100	45.28
$K=12$	100	18.52	100	31.36	100	33.84

Table 2.6: Noisy test data: Performance of the proposed indexing scheme for a 4-bit pattern (P_1)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
$K=4$	100	32.01	100	43.08	100	42.56
$K=6$	99	27.13	88.24	38.19	85.76	41.40
$K=8$	97.51	29.20	75.99	36.32	77.64	34.03
$K=10$	89.23	26.66	65.39	30.41	69.70	29.71
$K=12$	88.74	25.29	61.09	27.87	58.60	29.13

IrisCode Generation:

To match two normalized irides, it is desired to obtain numerical features from each normalized iris image to facilitate the comparison. The process of extracting numerical feature set from a normalized iris image is known as iris encoding. The encoding mechanism employed in our experiments uses quadrature 2D Gabor wavelets to extract the local phasor information of the iris structure. This is performed by convolving 2D Gabor filter with an normalized iris image. Let the 2D Gabor wavelet over image domain (x, y) be given as

$$h_{\{Re, Im\}} = sgn_{\{Re, Im\}} \int_{\rho} \int_{\phi} I(\rho, \phi) e^{-i\omega(\theta_0 - \phi)} \cdot e^{-(r_0 - \rho)^2 / \alpha^2} e^{-(\theta_0 - \phi)^2 / \beta^2} \rho d\rho d\phi \quad (2.5)$$

The result obtained through convolution with Gabor wavelet is complex, which consists of both real and imaginary parts. These components were separated and phase quantization was performed. Each phasor response is then encoded using two bits of information based on the quadrant of the complex plane in which it lies. The resulting 2 bit code is referred to as IrisCode.

Table 2.7: Noisy test data: Performance of the proposed indexing scheme for a 4-bit pattern (P_2)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
$K=4$	100	34.99	100	44.24	100	55.23
$K=6$	92.54	28.51	99	48.36	99.66	55.33
$K=8$	77.15	23.51	96.19	51.92	99.50	55.25
$K=10$	70.19	19.37	97.68	54.4	99.17	56.36
$K=12$	61.09	16.48	97.68	56.45	99.17	56.54

Table 2.8: Noisy test data: Performance of the proposed indexing scheme for an 8-bit pattern (P_1)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
$K=4$	100	27.59	100	23.28	100	24.93
$K=6$	98.34	24.16	99.33	22.38	100	22.62
$K=8$	96.19	21.28	98.67	21.20	98.67	19.40
$K=10$	92.21	20.70	96.68	19.09	97.35	17.35
$K=12$	85.43	19.80	93.70	17.11	96.02	16.81

Table 2.9: Noisy test data: Performance of the proposed indexing scheme for an 8-bit pattern (P_2)

	W=3		W=5		W=7	
	R_h	R_p	R_h	R_p	R_h	R_p
$K=4$	100	27.56	100	35.02	100	41.36
$K=6$	98.50	20.12	93.70	30.25	90.23	35.26
$K=8$	96.68	21.13	85.59	28.24	79.47	32.29
$K=10$	95.19	18.42	84.27	23.78	71.35	27.81
$K=12$	93.04	16.26	79.13	22.63	70.69	24.23

Two such IrisCodes are compared using hamming distance which is an important factor in iris matching. Hamming distance computes the number of corresponding bits that are different across the iris codes. It is expected that the hamming distance between two iris codes of same iris will be less than the hamming distance between two iris codes of different irides. It is empirically proven that Hamming Distance of 0.3 (70% overlap) is sufficient to claim the identity of an individual.

Performance of proposed indexing scheme on IrisCodes:

The IrisCodes are generated from normalized images. The size of each of the IrisCode is 720×64 pixels, as each pixel is represented by two bits. The performance of the proposed indexing scheme with different values of K for pattern P_1 can be seen in table 2.10.

Table 2.10: Performance of the proposed indexing scheme for a 4-bit pattern (P_1) on IrisCodes

	R_h	R_p
$K=8$	99.66	75.36
$K=12$	99.17	64.99
$K=16$	98.01	51.08
$K=20$	98.01	38.28
$K=24$	96.52	37.34

For pattern P_1 , the best value of $\gamma = 77.6\%$ occurred at a hit rate of 98.01% and a penetration rate of 38.28% with $K=20$. This suggests that the proposed indexing scheme performs better on binary pattern than IrisCodes.

2.5 Conclusion

In this chapter the problem of iris indexing had been addressed. The proposed algorithm is a fast searching technique that uses Burrows-Wheeler Transform to perform iris indexing. Compared to the previous work by Mukherjee et al. [42] (with hit rate = 80% and penetration rate = 8%, with $\gamma=85.79\%$), this method showed significant improvement in performance with a 99.83% hit rate at a 17.23% penetration rate, with $\gamma=90.90\%$. Further work is required to reduce the penetration rate, for instance, by using an alternate searching mechanism which combines different index patterns (P_1, P_2) of the same length. Future work would involve designing appropriate data structures to organize the indices generated for the iris images. This can potentially lead to an efficient retrieval strategy for very large databases.

Chapter 3

Classification of ear images

3.1 Introduction

As stated earlier, the problem of automatic identification involves comparing a query image, q , with the gallery entries, $P = \{p_1, p_2, p_3, \dots, p_n\}$, in order to determine its identity. As the number of entries n in the database increases the number of comparisons increase and this has a significant impact on the system efficiency. A classification scheme partitions the given database into multiple classes resulting in a corresponding reduction in the number of comparisons in order to boost real-time performance. For example, suppose that there are N classes, $6N$ number of enrolled images and a classification scheme which divides these images equally among these N classes. Whenever a query image comes in, the same classification scheme assigns a class to the query image and the query is compared only against those images in the database that have the same class. In this example, any query is likely to be searched only among 6 images instead of $6N$ images, thus reducing the time complexity. An illustration of this classification scheme is shown in Figure 3.1.

Ear is an emerging biometric which has rich and stable structure and which can potentially be implemented reliably and cost efficiently. Currently, there are no commercially available ear recognition systems. However, there is a lot of potential for using ear in conjunction with the face for authentication or classification purposes. Ear in classification mode can be used to reduce the search space. Apart, in the classification mode, the ear can also be used as a soft biometric to classify side profile facial images. Furthermore, an ear classification system with classes defined by visual words can be even used in criminology for human identification.

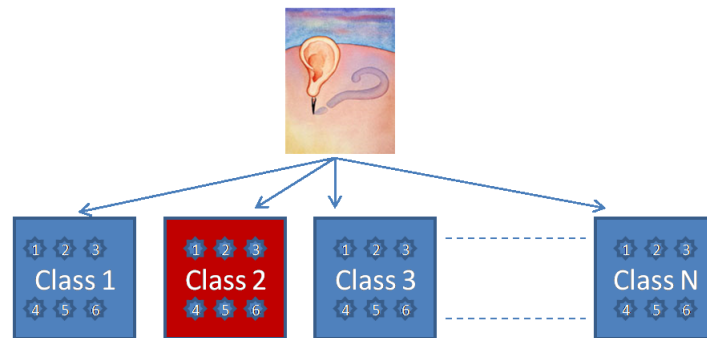


Figure 3.1: A typical classification scheme.

3.2 History of ear classification

Iannarelli was the pioneer in developing a human ear identification system. He developed this system as an alternative to fingerprint identification [2]. The ear identification system was primarily designed to identify a the newborn infant and to identify a child or an adult who cannot be identified by other means. The photograph of the ear was taken at the time of the birth and then after nine months. The photograph along with certain information were used to fill a standard ear identification card of 8"x8" as shown in Figure 3.2. Some of the information on the card was used to file the card systematically for quick access during identification.

STANDARD EAR IDENTIFICATION CARD	
IDENTIFICATION NUMBER: 100456	Sex Classification: M-2-2-7-2-P-7 W-2-2-5-2-O-9
Name: John Able Doe	Age: 43
Color: White	Height: 5-10
Weight: 150	Build: Slender
Place of Birth: Oakland, Alameda, Calif.	Date of Birth: Jan. 22, 1921
Date of Photograph: March 16, 1959	Department: Oakland, Alameda, California
Signature of Official: [Signature]	
GENERAL FORM	PECULIARITIES
1. Oval <input type="checkbox"/>	1. Wink of helix thin <input type="checkbox"/>
2. Round <input type="checkbox"/>	2. Rim of helix thick <input type="checkbox"/>
3. Rectangular <input type="checkbox"/>	3. Prominent helix line <input type="checkbox"/>
4. Triangular <input type="checkbox"/>	4. Cart before ear <input type="checkbox"/>
	5. Asymptotic helix line <input type="checkbox"/>
	6. Cart after ear <input type="checkbox"/>
	7. Cart upper ear <input type="checkbox"/>
SIZE	8. Flat helix rim <input type="checkbox"/>
1. Very small <input type="checkbox"/>	9. Cart above ear <input type="checkbox"/>
2. Small <input type="checkbox"/>	10. Deformed ear <input type="checkbox"/>
3. Normal <input checked="" type="checkbox"/>	11. Deformed ear <input type="checkbox"/>
4. Large <input type="checkbox"/>	12. Flag ears <input type="checkbox"/>
5. Very large <input type="checkbox"/>	13. Lobe marked in <input type="checkbox"/>
	14. Lobe pointed out <input type="checkbox"/>
	15. Lobeless ear <input type="checkbox"/>
POSITION	16. Square lobe <input type="checkbox"/>
1. Normal set <input checked="" type="checkbox"/>	17. Round lobe <input type="checkbox"/>
2. High set <input type="checkbox"/>	18. Pointed lobe <input type="checkbox"/>
3. Low set <input type="checkbox"/>	19. Flattened lobe <input type="checkbox"/>
4. Skewed set <input type="checkbox"/>	20. Twisted lobe <input checked="" type="checkbox"/>
5. Close set <input type="checkbox"/>	21. Descending lobe <input type="checkbox"/>
6. Deep set <input type="checkbox"/>	22. Prominent tragus <input type="checkbox"/>
7. Protruding <input type="checkbox"/>	23. Foreshort tragus <input type="checkbox"/>
PRINCIPAL PARTS	24. Missing antihelix <input type="checkbox"/>
1. I. Helix <input type="checkbox"/>	25. Prominent antihelix <input type="checkbox"/>
2. II. Helix <input type="checkbox"/>	26. Fract. helix <input type="checkbox"/>
3. III. Helix <input type="checkbox"/>	27. Symmetrical enlargement <input type="checkbox"/>
4. Isthmus <input type="checkbox"/>	28. Symmetrical indentation <input type="checkbox"/>
5. Concha <input type="checkbox"/>	29. Curved ear projection <input type="checkbox"/>
6. Antihelix <input type="checkbox"/>	30. Accessible notch <input type="checkbox"/>
7. Cartil. cochlear <input type="checkbox"/>	31. Accessible notch <input checked="" type="checkbox"/>
8. Incusca Intelligens <input type="checkbox"/>	32. Accessible notch <input checked="" type="checkbox"/>
	33. Accessible notch <input checked="" type="checkbox"/>
	34. Accessible notch <input checked="" type="checkbox"/>
	35. Accessible notch <input checked="" type="checkbox"/>

Figure 3.2: Ear identification card (front view) used by Iannarelli (Image taken from [2]).

The card consists of eight sections, which were filled with applicable information. The sections of the card are:

1. The identification number, ear photograph and personal history section.
2. First and second ear classification section
3. History of individual
4. Front and right profile images
5. Remarks
6. Criminal history
7. Finger print classification
8. Rolled right thumb impression

The ear identification card contained two levels of classification . The first level of classification was based on sex, race and anthropometric measurements of the ear (see Figure 3.3). Based on sex, one of the two classes, M for Male or F for Female, were assigned. Based on race, one of the four classes W for White, B for Black, A for Asian, O for Other were assigned. Later, the anthropometric measurements between certain points on human ear were taken manually.

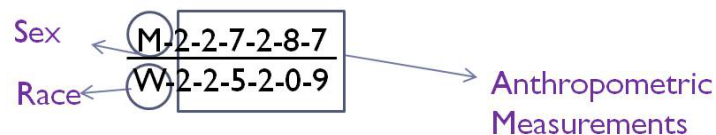


Figure 3.3: Level-1 classification.

The second level of classification was based on shape, size, position and peculiarities of the ear. Based on shape, ear was classified into one of round, oval, triangular or rectangular classes as shown in Figure 3.4; Based on size, ear was classified as being small, large or medium.

Based on position, ear was classified into one of three categories, normal, high or low set. If the position of the eyebrow and position of the helix are at the same level, the ear was classified as

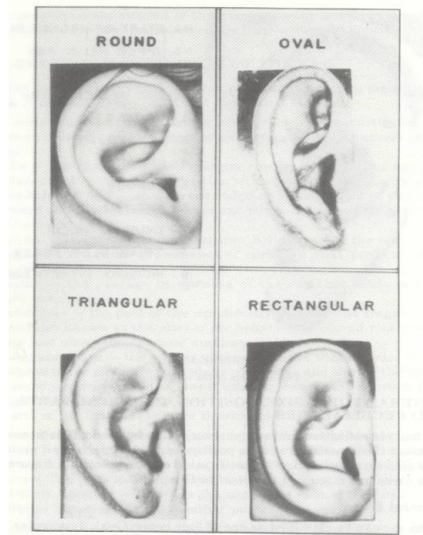


Figure 3.4: Ear classification scheme based on shape used by Iannarelli (Image taken from [2])

normal set. If the upper helix rim is above the eyebrow level, it was classified as high set. If the upper helix rim is below the eyebrow level, it was classified as low set as shown in Figure 3.5.

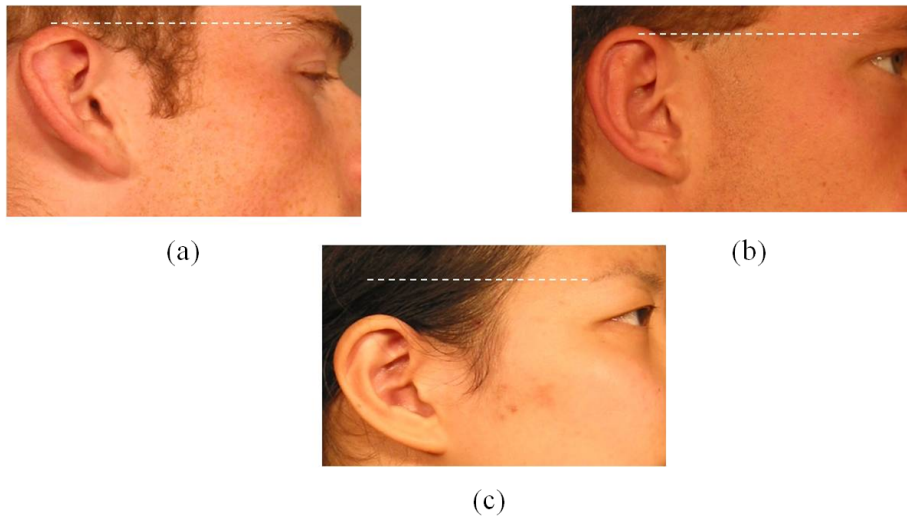


Figure 3.5: Classification based on position (a)normal set, (b)high set, (c)low set) used by Iannarelli

Apart from the above classification methods, the ear was further characterized using freckles, moles, scars and various distinctive distortions on the ear and its principal components. The first and second levels of classification were used for filing the card effectively for a quick access while

the anthropometric measurements along with the first, second level classifications and peculiarities were used for identification purposes.

3.2.1 Limitations of Iannarelli system

In Iannarelli's identification system the photographs were collected in a constrained way from a constant distance. So, every image was registered in terms of scale and orientation. Added to this is the fact that, the first and second levels of classification were based on human observations. However, in automated systems the ear images may not be collected in a constrained manner and one needs to automate the process of classification without any human intervention. To the best of our knowledge, to date there is no such method which automates the classification scheme. In this chapter, a new automatic classification scheme based on shape is proposed and evaluated.

3.3 Proposed approach

The structure of ear is not quite random as iris, but it has a definite shape and structure just like the face does. Helix and lobule define the shape of the ear while antihelix, triangular fossa, scapha, concha, incisura, tragus and antitragus define its structure. Even though shape and structure can be used for designing a classification scheme, the shape-based classification scheme is expected to be more efficient than structure based scheme because of the following reasons.

- The perceived structure of the ear is easily affected by variations in pose. The effect is maximized, when the structure is completely occluded. Whereas in the case of shape, given two different profile images of a person (e.g., different poses), the shape of one profile can be used to mathematically model the shape of other profile image.
- The shape of the ear can be visually estimated by a human, even from a certain distance. This enables us to use the shape-based classification scheme even in criminology.

Considering the advantages of shape-based classification scheme over structure-based scheme, the focus of this chapter is mainly on automating shape-based classification similar to the one proposed by Iannarelli. In the remaining sections, an automated classification scheme is proposed and evaluated.

3.3.1 Overview of ear classification system

The block diagram of a typical shape-based automatic ear classification system is shown in Figure 3.6. It consists of five components:

1. Image acquisition: As the ear is part of a face, no separate system is required to acquire ear images. The side profiles of the face images are sufficient for building a classification system.
2. Ear detection: After acquiring the profile images, ear detection algorithm is used to locate the ear.
3. Ear segmentation/localization: After detection, ear is segmented from other parts such as hair, skin, ear rings, etc.
4. Shape parameter estimation: The segmented ear is modeled using parameters which defines it's shape.
5. Classification: Using the shape parameters, ear is assigned to one of the pre-defined classes.

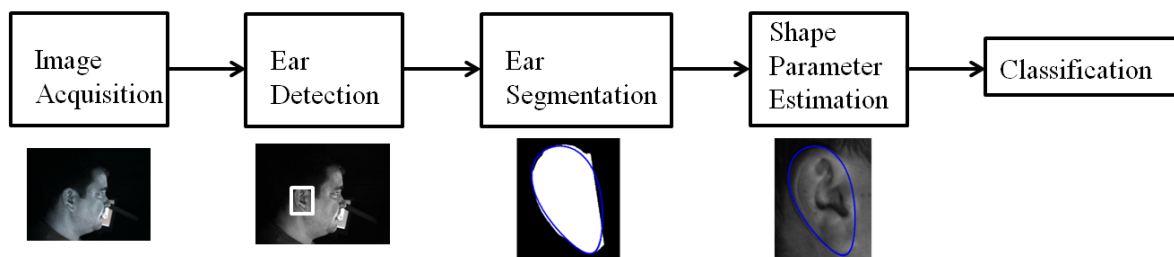


Figure 3.6: Block diagram of the classification scheme.

Generally, ear detection and segmentation are combined and treated as a single component. But, in this work, the two components are separated in order to prune the search space of the ear in the image. The entire system is operated in two different modes (i) Enrollment mode, (ii) Identification mode. During enrollment, the ear is assigned to a particular class and stored. During identification, the input ear is assigned to a particular class and compared against the images in the corresponding class. The important aspect to be considered while designing each component of an ear classification system is that, it should be invariant to rotation, translation, and scale.

3.4 Ear detection

The first step in an ear classification system is ear detection. A typical ear detection algorithm locates the position of the ear in an image and represents it as a rectangular box. Based on the level of human intervention, ear detection methods in 2D images can be classified into two categories: (i) semi automatic (ii) automatic. Semi-automated methods require user to manually anotate landmarks within an image. Use of semi-automated methods is likely to result in better performance but they may not be useful if the goal is to build an automated system. However, Yan and Bowyer [51] and Alvarez *et al.*

[4] employed semi-automatic detection methods in their recognition systems. Their main aim was to evaluate the performance of ear identification based on a set of defined features rather than ear detection.

The first completely automated system was developed by Burge and Burger [52] using deformable contours. The existing techniques for ear detection can be broadly categorized into 4 types: contour-based, feature-based, template-based and learning-based. Techniques proposed by Burge and Burger [52], Ansari and Gupta [53], Attarchi *et al.* [54], etc. extract contour of the ear for ear detection/localization, which can be considered as contour-based ear detection techniques. Most of the contour-based techniques work well on images that have good lighting condition, but they may fail when encountering poor lighting conditions.

Some techniques use features like skin color [55], features from force field transformation [30], features from ray transformation [56], etc. to detect ear. These techniques rely on specific image features to detect ear and can be considered as feature-based ear detection techniques.

In template-based techniques, a pre-defined template is used to find a match in the given image. Sana *et al.* [57], Abdel-Mottaleb and Zhou [58], Prakash *et al.* [59], Yuizono *et al.* [60], etc. considered template-based approach to segment the ear. In real time operations, their predefined template may not handle variations in pose, size and shape of the ear itself.

Learning-based techniques use training data to learn a strong classifier which can distinguish ear images from other images. Islam *et al.* [61] and Abaza *et al.* [62] used boosting techniques to train a strong classifier with Haar-like features. Learning-based schemes are computationally expensive during training but efficient during testing.

Each method listed above has a drawback either in the form of computational cost or in terms of performance due to variations that occur when operated in an unconstrained environment. When designing a real time ear detection algorithm, one needs to consider all variations including pose, lighting and occlusion. This can possibly be achieved by adapting learning-based schemes. Learning-based schemes has the ability to handle noise and objects with large variations. The training data contains both positive (object) and negative (non-object) samples which helps in training measurement models that are robust to background noise.

Considering the advantages of learning-based schemes, ear detection is performed using learning-based scheme. The scheme proposed by Islam *et al.* [61] uses the classical Viola-Jones AdaBoost algorithm [63], which may take quite a long time to train the classifier for ear detection. In order to avoid long training times, Abaza *et al.* [62] adapted the Forward Feature Selection (FFS) algorithm proposed by Wu *et al.* [3], which reduces the training times significantly. Ear detection in our experiments is achieved by training the classifier using FFS, similar to Abaza *et al.* [62].

3.4.1 AdaBoost

Conventional methods uses features like shape, size, structure or keypoints to detect an object within the image. Instead of explicitly extracting features, Viola and Jones [63] used a set of pre-defined features to define an object. The features used were based on simple Haar like filters which can easily be computed using integral image. Figure 3.7 shows the type of rectangular filters used for detection. The value of a two-rectangle feature is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent.

A three-rectangle filter computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a four-rectangle filter computes the difference between diagonal pairs of rectangles. For a given 24x24 window, around 45,396 features can be calculated using two, three and four rectangular filters. Because of the high computational complexity, not all the features are used to represent an object. An object can be represented with only a small number of features which are combined to an efficient classifier. In order to pick the best features to form a strong classifier, AdaBoost is used. Adaboost performs both selection of features (i.e., the

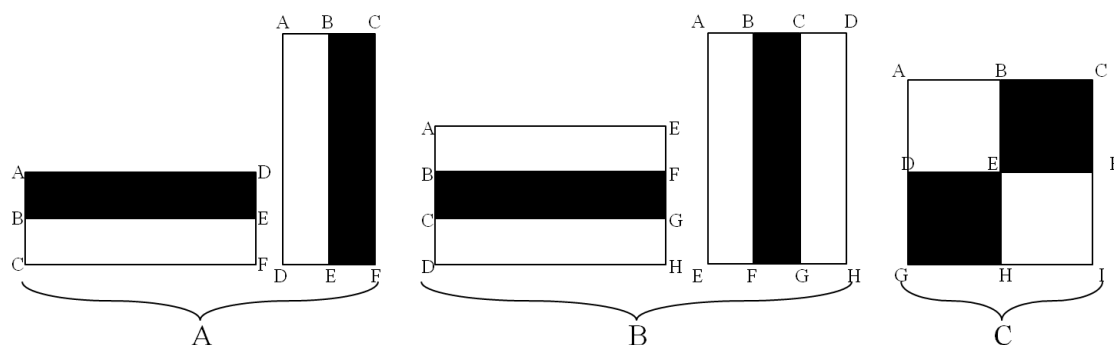


Figure 3.7: Figure showing simple Haar features. A)Two rectangular B)Three rectangular C)Four rectangular features .

filters) and training of classifier simultaneously. For a given set of N training samples (x^i, y^i) containing both positive and negative samples and simple rectangular filters h_j , a final classifier H is built using AdaBoost algorithm which is summarized in Algorithm 2. For every iteration 't', among all the available simple features, a feature (which acts as a weak classifier) with the smallest weighted error with respect to D_t is chosen. After choosing the weak classifier, D_t is updated. This procedure is repeated T number of times and the final classifier is formed.

The final strong classifier is a combination of T weak classifiers which distinguishes positive samples (ear) from negative samples (non-ear). Testing is done by scanning this final classifier across the image at multiple scales and locations. Scaling is achieved by scaling the filters, rather than scaling the image (Scaling of features by a factor of 1.25 times is equivalent to scaling down of image by a factor of 0.8). The detector is scanned across each location by shifting the window by a displacement δ . At the end, all the detected regions are post-processed in order to remove repetitive detections.

In real time, not all the combination of T classifiers are required to reject a negative sample. Sometimes, even a combination of few simple features are enough for this purpose. By taking advantage of this, Viola-Jones proposed a cascade approach which significantly reduces the computation time and increases the efficiency of the classifier during testing.

Algorithm 2. AdaBoost

Input: A set of training samples (x^i, y^i) , $x^i \in \mathcal{X}$, $y^i \in \{-1, 1\}$, $i = 1, \dots, N$ and h_j , $j = 1, \dots, M$

Output: A strong classifier $H(x)$, which is a combination of simple classifiers h_j

- 1 Initialize weights ($D_1^i = 1/N$) for each training sample. $D_1^i = 1/N, \forall i$;
- 2 **for** $t = 1$ to T **do**
- 3 Find a classifier h_t with the lowest error, ε_t ;
- 4 Compute voting strength α_t of h_t using ε_t ;
- 5 Update weights of the samples, D_t^i , which are calculated from ε_t ;
- 6 Normalize weights;
- 7 **end for**

Output:

$$H(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T \alpha_t h_t(x) \geq 1/2 \sum_{t=1}^T \alpha_t; \\ 0 & \text{otherwise} \end{cases}$$

3.4.2 Cascade of classifiers

In the cascade approach, a series of boosted classifiers, similar to a degenerate decision tree, were constructed which accepts almost all positives and rejects many negatives. In a cascade, the classifier at each stage is trained by AdaBoost in such a way that simple classifiers, with less number of features, are used in the early stages to reject the majority of sub-windows before subjecting them to complex classifiers having more number of features. Rejecting most of the negative samples in the early stages reduces the number of computations and increases the performance of the system. The thresholds are set such that each stage yields higher detection rates and, hence, considerable false positive rates (i.e., a non-ear entity is incorrectly classified as an ear). Generally, the threshold in each stage is adjusted to detect 100% of the objects with a false positive rate of 50%. This will ensure that all the positive samples are detected, while rejecting most of the negative samples in initial stages. A typical cascade configuration can be seen in Figure 3.8.

The cascade approach helps in reducing the time during testing. However, the time taken to train the strong classifier H , is still quite long. A typical implementation of Viola-Jones AdaBoost algorithm may take a few days to completely build a cascade of strong classifiers. In order to avoid

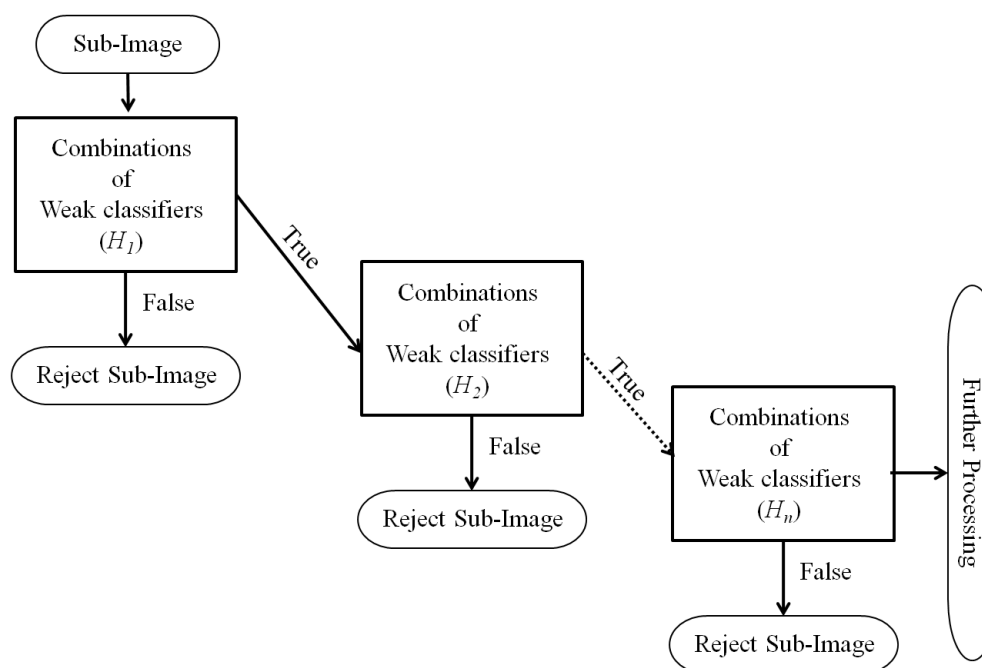


Figure 3.8: Figure showing cascade of classifiers.

the long training times, Wu *et al.* [3] proposed a Forward Feature selection (FFS) algorithm.

3.4.3 Forward Feature Selection

In the Viola-Jones approach, AdaBoost picks the rectangular filters with the smallest weighted error with respect to the weight distribution D_t . In every iteration t the boosting process adds a rectangular filter to the final classifier. After T iterations, the final classifier is formed from the combination of selected rectangular filters. In this process, D_t is updated at every iteration, which in turn requires that the weak classifiers are re-trained at every round, as indicated in Figure 3.9(a). For an object detection application to be robust, the number of training samples and the number of features to be used are in the order of thousands. Re-training of a large number of rectangular features takes more time and this is the most time consuming component in the algorithm. In order to avoid re-training, Wu *et al.* [3] proposed a greedy feature selection method based on Forward Feature Selection (FFS). The key intuition in FFS is precomputing, where the results of previously trained weak classifiers are stored and re-used. This restricts the training of the weak classifiers to only once. This intuition can be clearly seen in Figure 3.9. The time consuming component, which

is the training of all weak classifiers, is moved out of the loop in FFS.

For a given N training samples (x^i, y^i) containing both positive and negative samples, and simple rectangular filters h_j , a final classifier H is built using Forward Feature Selection algorithm which is summarized in Algorithm 3. In FFS, there is no distribution D_t maintained for each training sample and each sample is treated equally in every iteration. Instead, FFS maintains a storage table V which stores classification results of each weak classifier (Line 2 of Algorithm 3). The storage table V is updated only once and contains binary values (0 for correct classification and 1 for miss classification). Using V , v in Line 6 is updated after every iteration. v , which contains value ranging from 0 to t , acts similar to D_t in the AdaBoost and helps in determining the best weak classifier h_k in Line 5. FFS acts as a faster version of AdaBoost when values in V are initialized with original response of weak classifiers $h_j(x)$ rather than the binary values.

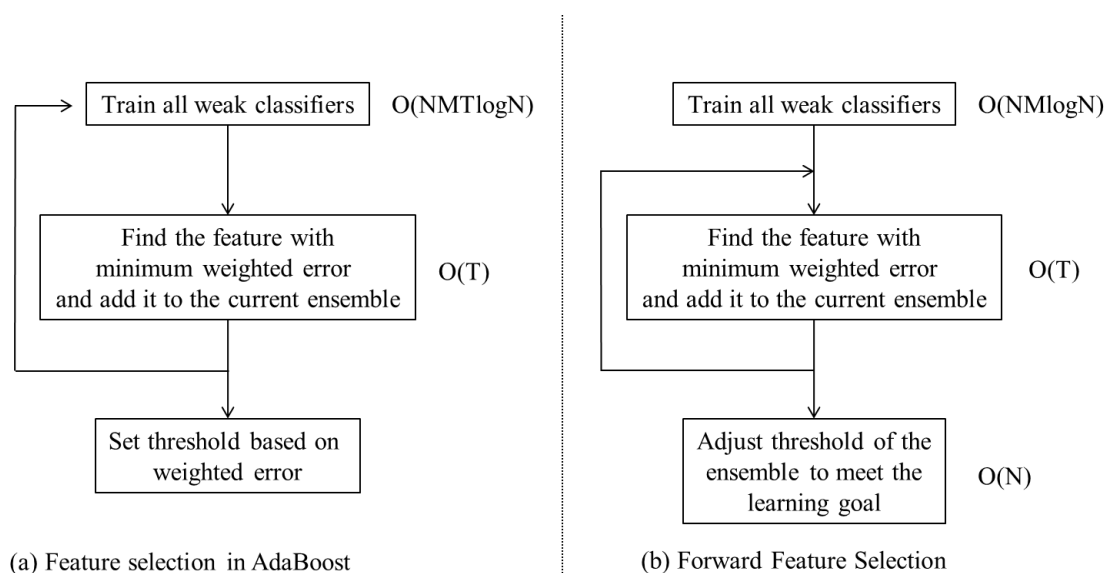


Figure 3.9: Comparison of Naive Adaboost feature selection with Forward Feature selection ((b) taken from [3]).

After training, the final strong classifier is tested on input images similar to what is done in AdaBoost. Use of FFS decreases the training time from days to hours without compromising accuracy.

Algorithm 3. Forward Feature Selection

Input: A set of training samples (x^i, y^i) , $x^i \in \mathcal{X}$, $y^i \in \{-1, 1\}$, $i = 1, \dots, N$ and h_j , $j = 1, \dots, M$

Output: A strong classifier $H(x)$, which is a combination of simple classifiers h_j

- 1 $\forall h_i(x)$, find a threshold which has least error on the training dataset
- 2 Form a table V_{ji} such that $V_{ji} = h_j(x_i)$, $1 \leq i \leq N$, $1 \leq j \leq M$
- 3 Initialize $S \leftarrow \phi$, $v \leftarrow 0_{1 \times N}$ where S holds the best weak classifier at each iteration and v holds the weights of all samples
- 4 **for** $t = 1$ to T **do**
- 5 Find a classifier h_k which when added to S gives least error ε_k
- 6 Add h_k to S i.e., $S = S \cup h_k$ and update v using V_{ji} i.e., $v = v + V_k$, where V_k is k th row of V ;
- 7 **end for**

Output: $H(x) = \text{sgn}(\sum_{h \in S} h(x) - \theta)$. Adjust θ to meet the learning goals

3.5 Ear segmentation

After finding the ear location in an image, the next task is to segment the ears from other regions in the detected window. Along with the ear, a typical detected window contains hair, skin, occlusions due to ear rings, etc. An ear segmentation algorithm should separate ear region from the other unnecessary regions in the detected window. This can be done by extracting the edges of the ear and identifying the outer edges which enclose the ear. The outer edges of the ear correspond to the edges of helix and lobule. So in order to segment the ear, it is sufficient to have helix and lobule edges (together termed as helix edge from now on).

The canny edge detector is perhaps the most widely used edge detector in the literature. It is based mainly on the local gradient and has a tunable scale parameter. Further, if the goal is to detect object boundaries in order to perform object segmentation or detection, then all other detected edges except those pertaining to object boundaries should be considered to be false detects. In case of ear segmentation, except for the helix edges all other edges are irrelevant. Using a canny-based edge detector or other edge detectors may lead to detection of unnecessary edges like hair, inner structure of the ear from the ear image. Also, when working on operational data which have lighting variations, these algorithms may fail to detect edges properly.

To overcome these problems, a learning-based edge detection algorithm is used. In this work, a supervised learning algorithm referred as Boosted Edge Learning (BEL) proposed by Dollar *et al.* [64] has been used. Similar to the AdaBoost, it is a learning based algorithm which in fact enables us to train a classifier for a specific task. The output of BEL is a probability image with pixel values suggesting the chance of it being the desired edge.

3.5.1 Boosted Edge Learning

Boosted Edge Learning uses an extended version of Probabilistic Boosting Tree (PBT) [65]. Inputs to BEL are original images and images labeled with desired edges to be detected. Using the label information, BEL creates training samples (both positive and negative) of window width ‘W’ and also calculates features like Haar responses, gradients, Difference of Gaussian (DoG), Difference of offset Gaussian (DooG) at multi scales for these samples. PBT takes these training samples and features as input and builds a tree with classifiers as nodes. PBT is similar to AdaBoost as far as a cascade framework is concerned except that at each node instead of rejecting the negative samples completely it assigns a probability value to each sample. Also, PBT works more effectively with large data and more number of features than AdaBoost.

Training PBT is similar to training a decision tree, except for the difference that at each node a boosted classifier is used to split the training data as shown in Figure 3.10. In the figure, positive samples are represented with blue dots and negative samples with red dots. The first level of the tree divides the whole set into two parts. The right part of the tree mostly contains positive samples and the left part consists of a combination of both. Instead of rejecting the negative samples (i.e., the left part) at each node, the tree expands on the nodes where positive and negative samples are mixed. Algorithm 4 discusses the steps involved in training a PBT with N training samples (x^i, y^i) and M simple features h_j where each sample is assigned an equal weight. The algorithm recursively learns a tree. At each node in the tree, a strong classifier is learned using boosting algorithms like AdaBoost. The training samples are divided into two new sets S_R and S_L . Based on the probability values $q(+1 | x_i)$ and $q(-1 | x_i)$, the sample is assigned to either S_R or S_L . The weights of the samples are modified such that the misclassified samples (i.e., positives in S_L and negatives in S_R) get high values. The left sub tree and right sub tree are again trained recursively

until the tree reaches a maximum depth (or) until a certain condition is satisfied (line 3 in Algorithm 4).

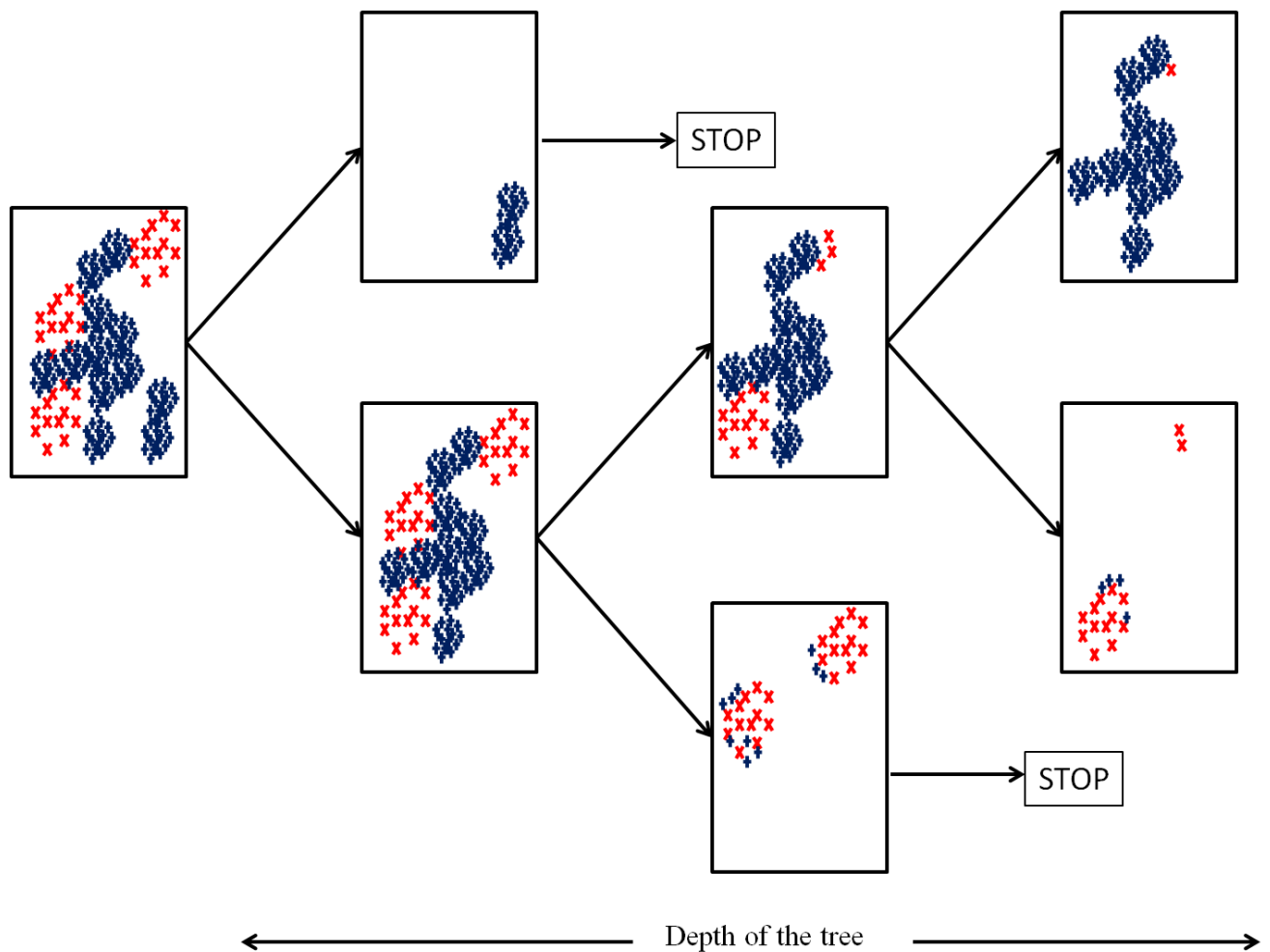


Figure 3.10: Figure showing the working of PBT.

During testing, BEL converts the input image into overlapped sub-windows 'W'. For each sub-window, PBT calculates the posterior probability $\tilde{p}(y | x)$ recursively and the sub-window is assigned with that value. Because of its advantage to train a tree for a specific task, BEL is used to train the classifier for detecting helix edges of the ear.

Algorithm 4. Probabilistic boosting tree

Input: A set of training samples (x^i, y^i) , $x^i \in \mathcal{X}$, $y^i \in \{-1, 1\}$, $i = 1, \dots, N$ and h_j , $j = 1, \dots, M$

Output: A tree with strong classifier at each node

- 1 Initialize, $w_i = 1/N$, $\forall i$
- 2 Compute the empirical distribution at a node, $\hat{q}(y) = \sum_i w_i \delta(y_i = y)$
- 3 **if** $\theta \leq \hat{q}(y) \leq 1 - \theta$
- 4 Train a strong classifier at that node
- 5 Split data into two sets S_R and S_L based on $q(+1 | x_i)$ (probability of x_i being positive) and $q(-1 | x_i)$ (probability of x_i being negative)
- 6 Modify weights w_i to $q(+1 | x_i) * w_i$ in S_R and $q(-1 | x_i) * w_i$ in S_L
- 7 Train S_R and S_L recursively by calling step 2
- 8 **else**
- 7 Stop

Output : Trained tree with strong classifiers

Testing

- 1 **if** present node has no children
- 2 $\tilde{p}(y | x) = \hat{q}(y)$
- 3 Stop
- 4 **else**
- 5 $\tilde{p}(y | x) = q(+1 | x)\tilde{p}_R(y | x) + q(-1 | x)\tilde{p}_L(y | x)$
 To calculate p_R and p_L go to step 1 recursively

3.5.2 Contour extraction and segmentation

After training the BEL with helix edge labels, the obtained detector is used to find the helix edges. During testing, BEL outputs a probability image and pixels with high values are considered to be helix edges. After obtaining the probability image, the output is smoothed with Gaussian filter which helps in connecting the helix edges seamlessly. Next, a hard threshold is applied to the smoothed image and the longest connecting path is chosen. The longest connecting path is considered as an estimate of helix edge.

The estimated helix edge is used to segment the ear after following some post processing steps, which are listed below:

1. The estimated helix edge represents the entire helix and, therefore, is not a thinned contour. So in order to obtain the contour, the estimated helix edge is thinned using morphological operations proposed in Gonzales and Woods [66].
2. The thinned image may contain some parasitic edges (noise). In order to remove these edges, the image is converted to a graph with pixels denoting edges and pixel values denoting vertices.
3. Sometimes due to false edges the contour contains some spurs. Such spurs and parasitic edges are removed by considering the longest connected path in the graph.
4. The ends of the longest connected path are considered to be the ends of the helix and the shortest connected path between these two ends is calculated, and this is considered to be the helix contour.
5. Finally, the shortest connected path is smoothed and the ends of the shortest path are connected by a straight line. The area enclosed by the shortest path is considered to be the ear.

The post processing steps can be visually seen in the Figure 3.11.

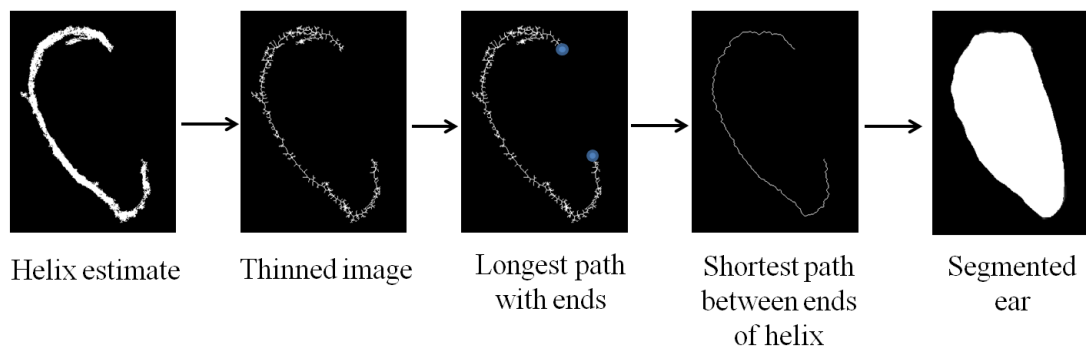


Figure 3.11: Post processing steps used for segmenting the ear.

The contour of the segmented ear looks more like an ovoid and the parameters for this ovoid are estimated using the technique proposed by Alvarez *et al.* [4].

3.6 Estimation of shape of the ear using an ovoid model

3.6.1 Ovoid model

Generally the shape of an ear can be defined as being an oval or an ovoid. In geometry, an oval or ovoid is any curve resembling an egg. Strictly speaking, ovoid is a special case of ellipse and every ovoid is an ellipse. Alvarez *et al.* [4] were the first, to design an ovoid model to represent ears. In Alvarez *et al.*'s model, ovoid was obtained by deforming the ellipse in the vertical direction. The parametric model for an ellipse with a as major axis and b as minor axis is given by

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_0 + acost \\ y_0 + bsint \end{pmatrix} \quad (3.1)$$

By adding an ellipse deformation factor γ in the vertical direction, an ovoid is formed. The parametric equation for this ovoid is given by

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_0 + acost \\ y_0 + b(1 + \gamma cost)sint \end{pmatrix} \quad (3.2)$$

The parameter γ induces distortion along the vertical axis of the ellipse, thereby helping the model to better estimate the complex contour of the ear. The effect of γ and a/b on the model is depicted in Figure 3.12. One can observe that as the value of γ increases the elongation of the ovoid increases, and the parameter a/b controls the size of the ovoid.

In order to incorporate rotation into the model, a parameter α is introduced into equation 3.2 and the final equations are denoted as:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} x_0 + a\cos\alpha cost + \sin\alpha b(1 + \gamma cost)sint \\ y_0 - a\sin\alpha cost + \cos\alpha b(1 + \gamma cost)sint \end{pmatrix} \quad (3.3)$$

The parametric model thus obtained has six parameters in it and is advantageous over other models for the following reasons:

- It has just one more additional parameter than ellipse to estimate.
- It has been demonstrated to fit most of the human ears quite well [4].
- The parameters (a/b and γ) can be used as scale, translation and rotation invariant parameters.

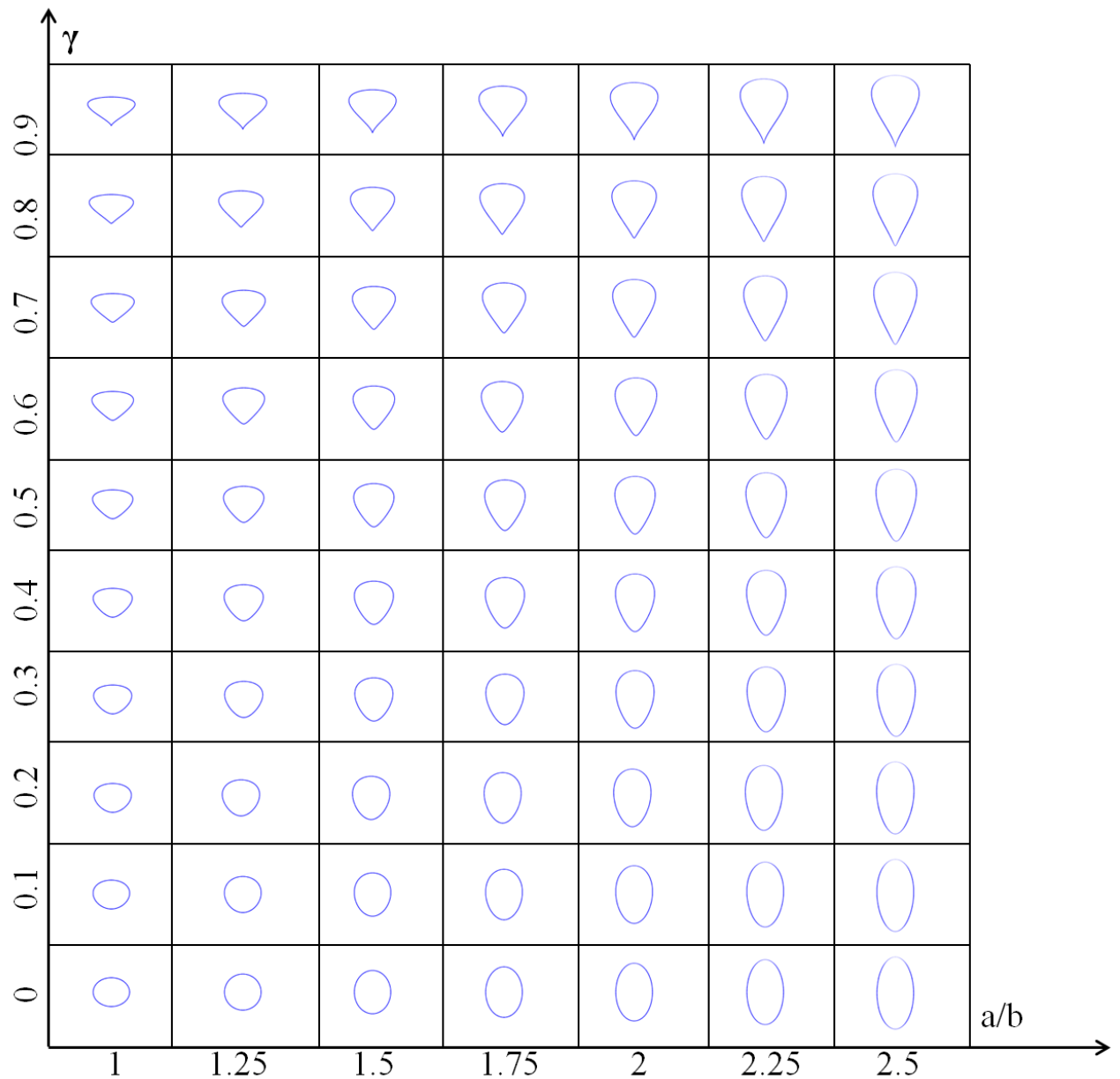


Figure 3.12: Effect of γ and a/b on ovoid.

- Finally, based on the parameters (a, b, γ and α) two ears can be registered for further comparison.

Considering these advantages, the parameters of this model are estimated using the properties of the segmented ear region.

3.6.2 Parameter estimation of the ovoid

The final model in equation (3.3) has six parameters ($a, b, \gamma, \alpha, x_0, y_0$) and these parameters can be estimated using the properties of the segmented ear. Alvarez *et al.* [4] estimated the initial parameters of an ovoid by following these sequence of steps :

1. Initially, first order moments m_{00} (area), m_{10} (x -coordinate of the centroid), m_{01} (y -coordinate of the centroid) of the segmented ear are calculated.
2. Using first order moments, second order moments denoted by m_{20}, m_{02}, m_{11} are calculated. A matrix M_2 is formed using these second order moments .

$$M_2 = \begin{pmatrix} m_{20} & m_{11} \\ m_{11} & m_{02} \end{pmatrix} \quad (3.4)$$

3. Next, eigenvector $v_{max} = (x_{max}, y_{max})$ corresponding to the maximum eigen value of matrix M_2 is calculated. α is then calculated as

$$\tan(\alpha) = y_{max}/x_{max} \quad (3.5)$$

4. All the contour points of the segmented ear are projected on to a straight line that passes through the centroid in the direction of v_{max} . On this straight line, the maximum and minimum projected points (p_{max} and p_{min}) from the centroid are calculated. Using p_{max} and p_{min} the center and semiaxis of the deformed ellipse are found using

$$(x_0, y_0) = (p_{max} + p_{min})/2 \quad (3.6)$$

$$a = ||p_{max} - p_{min}||/2 \quad (3.7)$$

5. The remaining two parameters b and γ are calculated as

$$b = m_{00}/(\pi a) \quad (3.8)$$

$$\gamma = 3 * (2A - \pi ab)/(4ab) \quad (3.9)$$

Here A is the area of the semiplane generated by the straight line that passes through the center and with direction v_{max} as normal vector.

The parameters obtained using this procedure may not fit the contour of the ear precisely. This can be seen in the Figure 3.13, where the estimated contour is represented in blue. So a gradient descent algorithm was used to improve these parameters. In order to increase the accuracy of the estimated parameters, a modified gradient descent algorithm is proposed and adapted.

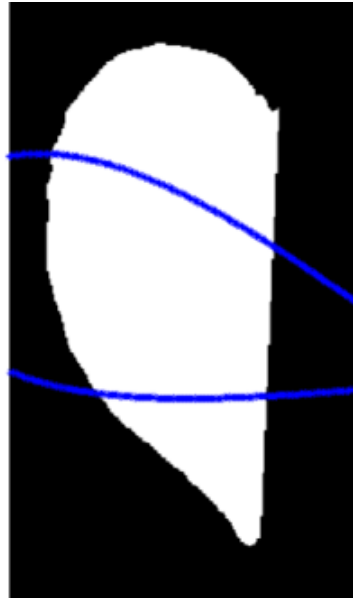


Figure 3.13: Figure showing segmented ear and its contour estimated using Alvarez *et al.* model [4].

3.6.3 Gradient Descent Algorithm

Gradient decent is a general machine learning algorithm that can be used to minimize any arbitrary function. For instance, consider an arbitrary function denoted by $J(\theta_0, \theta_1)$ where θ_0, θ_1 denotes the parameters of the function. The goal is to minimize this function which can be defined as $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$. The following steps can be followed to achieve the goal:

1. Initialize θ_0, θ_1 to arbitrary values.
2. Update or change θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until local minimum is reached.

The procedure followed by the gradient descent algorithm to achieve the desired goals as outlined above, is briefly explained by the following steps.

1. Initialize θ_0, θ_1 with some arbitrary values (say $\theta_0 = 0, \theta_1 = 0$)
2. Repeat until convergence

$$\left\{ \begin{array}{l} \theta_j := \theta_j - \beta \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \text{ (for } j=0 \text{ and } j=1) \end{array} \right\}$$

Here β denotes the learning rate of the gradient decent algorithm, which controls the step size adopted to update θ_j . $\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ denotes the gradient (equivalent to slope of the tangent in one dimension).

Learning rate β is a critical parameter in the gradient descent algorithm. If β is too small, the algorithm can be slow and if it is too large, the solution can overshoot the minimum and thus may fail to converge (or may even diverge in certain cases). The above algorithm can be generalized to minimize a function with more than two parameters, for instance $J(\theta_0, \theta_1, \theta_2, \dots, \theta_n)$.

An ovoid represented with six parameters $a, b, \gamma, \alpha, x_0$ and y_0 can be minimized using the gradient descent algorithm. The function to be minimized is the distance between the estimated ovoid and the contour points of the segmented ear, which is formulated as $J(a, b, \gamma, \alpha, x_0, y_0) = \sum_{i=0}^N \text{dist}(\text{Ovoid}, (x^i, y^i))$, where N denotes the number of contour points. Initial parameters $a, b, \gamma, \alpha, x_0$ and y_0 are obtained by following the steps described in Section 3.6.2. Later, gradient descent algorithm is applied on the initial parameters to obtain the parameters at the local minimum. One of the problems with the gradient descent algorithm is that in some cases the parameters get stuck in a local minima and the estimated parameters are far from the global minimum. To avoid this, a small value, which is proportional to the difference of estimated parameters to the predicted parameters is added to the parameters and the gradient descent algorithm is resumed with the modified parameters as the initial parameters. This step helps in perturbing the parameters stuck at a local minimum and pushes them towards the global minimum. This technique does not unduly add overall consumption time since the initial parameters estimated using Alvarez *et al.*'s equations are already close to the solution. Also, adding a small noise to the parameters that are already at the global minimum will have little impact on the estimated final parameters. The modified gradient descent algorithm is shown in Algorithm 5.

Algorithm 5. Modified Gradient descent algorithm

Input: Initial parameters $a_0, b_0, \gamma_0, \alpha_0, x_{00}, y_{00}$

Output: Final parameters $a_i, b_i, \gamma_i, \alpha_i, x_{0i}, y_{0i}$

- 1 Initialize $a_0, b_0, \gamma_0, \alpha_0, x_{00}, y_{00}, i = 0$
- 2 Update $a_i, b_i, \gamma_i, \alpha_i, x_{0i}, y_{0i}$ to reduce $J(a_i, b_i, \gamma_i, \alpha_i, x_{0i}, y_{0i})$ until local minimum is reached
- 3 **for** $i = 1$ to N
- 4 **if** $\gamma_{i-1} > 1$
- 5 $\gamma_i = 0.99$
- 6 Go to step 2
- 7 **else if** $a_{i-1} > \text{height of the detected ear region}$
- 8 $a_i = 2 * a_{i-1} - \text{height of the detected ear region}$
- 9 Go to step 2
- 10 **else if** $b_{i-1} > \text{width of the detected ear region}$
- 11 $b_i = 2 * b_{i-1} - \text{width of the detected ear region}$
- 12 Go to step 2
- 13 **else**
- 14 Stop

Figure 3.14(a) shows the output of the gradient descent algorithm on a segmented ear, where parameters are initially stuck at a local minimum. Figure 3.14(b) shows the output of the modified gradient descent algorithm. The modified gradient descent algorithm outperforms the traditional gradient descent algorithm in this case.

3.7 Classification

As discussed in Section 3.6.1, among the six ovoid parameters, a/b and γ are the scale, translation and rotation invariant parameters which can be used for designing a classification scheme. From Figure 3.12, one can observe that for lower values of γ (≤ 0.2), where the elongation is less, the shape tends to be more of an a rectangle (for higher values of a/b) or circular (for lower values of a/b). On the other hand, for mid values of γ ($0.3 \leq \gamma \leq 0.6$) the ovoid tends to be more

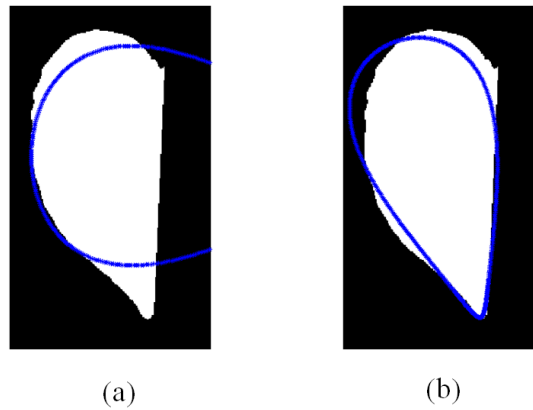


Figure 3.14: Figure showing the efficiency of the modified gradient descent algorithm. (a) Gradient descent algorithm converged at local minimum. (b) Parameters at global minimum achieved by modified gradient descent algorithm.

like oval and for higher values of γ ($\gamma \geq 0.7$) the shape looks like a triangle. By observing these values *viz.*, a/b , γ , an ear classification scheme was designed as depicted in the flowchart shown in Figure 3.15.

3.8 Experimental Evaluation

This section presents the details regarding the database used, experiments performed and results observed.

3.8.1 Databases Used

WVU Database:

The WVU (West Virginia University) ear database consists of 460 video sequences of 402 different subjects. Each video begins at the left profile (0 degrees) of the subject, and takes about 2 minutes to end at the right profile (180 degrees) [62]. Of the 402 subjects, this database contains 42 subjects with earrings, 38 subjects with partially occluded ears, and 2 subjects with fully occluded ears. From these, a total of 120 subjects with 10 subjects having earrings, 10 subjects having partially occluded ears, 10 subjects with eyeglasses and 90 subjects with no constraints were selected for training purposes. To capture the pose variations, 10 images which vary from left side profile

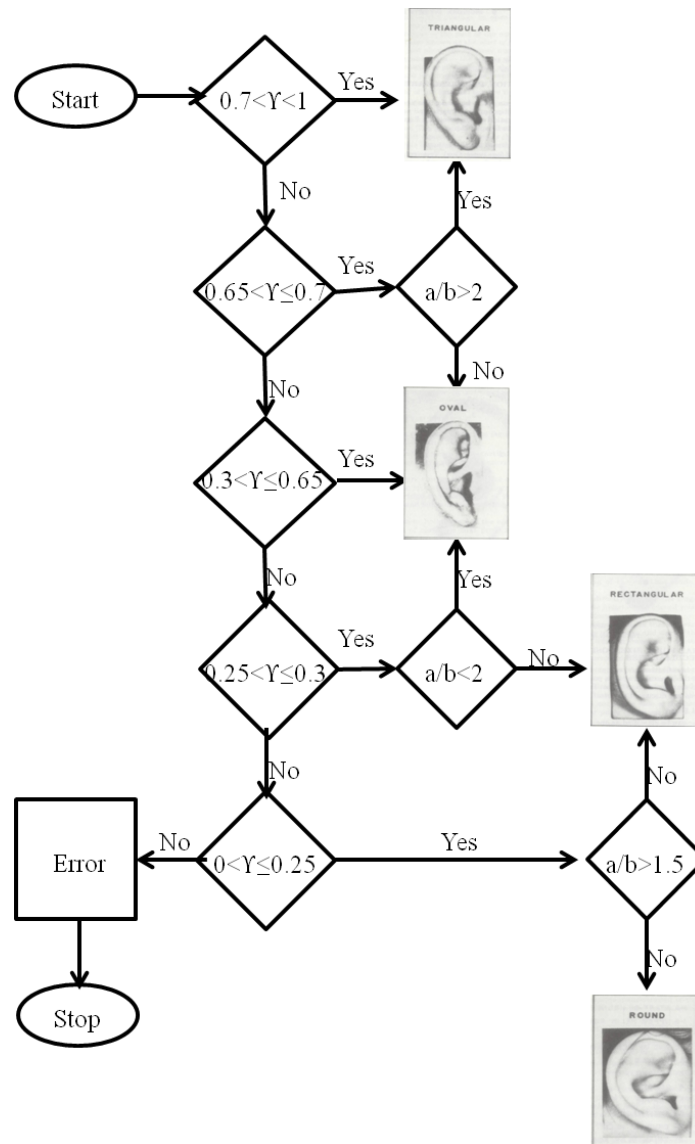


Figure 3.15: Figure showing the proposed ear classification scheme.

to front profile with incremental deviation were extracted from the video sequence. The extracted images are of resolution 480×640 . Figure 3.16 shows a few samples from the WVU database.

UND ear database:

The UND (University of Notre Dame) databases contain several collections for various modalities. Among them, collection E, collection F, collection G and collection J2 are the available databases for the ear modality. Collection E contains right profile images of 114 distinct subjects



Figure 3.16: Sample images of a single user in WVU database.

with different illuminations. Hence, this database is used for training purposes. The size of each image is 1200×1600 , 24-bit RGB. At least three images of each user were taken under different levels of illumination. Figure 3.17 shows sample images in the collection E data set.

MAGNA database:

The MAGNA database contains 3D and 2D facial images. These images are captured under low lighting conditions with few illumination variations. A total of eight cameras (Geometrix device) were used to capture five profile images from left to right, and the remaining three were used for obtaining stereo views. Stereo views and the profile images are combined together to create a 3D representation from the eight 2D images. Along with this, a Cyberware 3D scanner was used to obtain an accurate representation of the x, y, and z coordinate geometry and a 24-bit RGB color image texture map for each subject. A total of 3,115 subjects with both 3D and 2D images are present in the database. Due to the large size of the database, the experiments were



Figure 3.17: Sample images in UND database.

performed only on the left side profile images. The resolution of each image is 1200×1600 pixels. Of the 3,115 left images in this database, 164 images had fully occluded ears, 986 had partially occluded ears and 1963 images had no occlusion. Figure 3.18 shows sample images from this dataset.

3.8.2 Ear detection

Training:

The FFS technique discussed in section 3.4.3 is used for detecting ears. In order to train the classifier, 1200 images from WVU database and 300 images from UND database are taken and the ear region is manually segmented from each image. A total of 1500 ear segments are obtained from the two databases. Each ear segment is converted to gray scale and bilateral mirror images are obtained and finally a positive sample database with 3000 images is formed. To further increase the size of the database, these 3000 images are modified by minor translation, rotation and



Figure 3.18: Sample images of a single user in Magna database.

contrast stretching and the modified images are added to the positive database. A total of 10,000 ear samples from the positive database are used for training. For the negative samples, non-ear images are considered. Finally, a set of 10,000 ear examples and 10,000 initial non-ear examples of size 24×18 are considered for training. Two, three and four rectangular features are used for training. 20,000 features sampled uniformly from the entire set of rectangular features are used. A maximum of 50 nodes are allowed in the cascade configuration. A target of 99% detection rate and 50% positive rate is set for each node. After training, the final detector consists of 19 nodes which include a total of 2332 features.

Testing (top-down process):

In order to detect an ear in an input image, the image is scanned using the classifier generated by the cascaded Adaboost system. The input image is divided into sub-images of size 24×18 overlapped with a shift of one pixel and scanned. Further, the features are scaled up by 1.25 times (which is same as down sampling the image by 0.8) and scanned again. At the end, all the detected regions are post-processed in order to remove repeated detections. This procedure can be viewed as a top-down process, as each image is scaled down sequentially by 0.8 and re-scanned.

Results (top-down process):

The classifier is tested on the MAGNA database. Of the 3115 images, 1963 images with no ear occlusions were considered for testing. To increase the efficiency of the system, the image is

scaled down to 120X160 pixels and the detection algorithm is applied in the testing mode. If there is no ear detected in the image, the image is scaled up 5 times and the process is repeated. This method reduces the time complexity, as well as increases the efficiency. To measure the efficiency of the detection algorithm, two rates were used.

I) False Detection rate : It is the ratio of the number of non-ears detected to the total number of images.

II) False Reject Rate : It is the ratio of the number of non-detected ears to the total number of images.

On the MAGNA database, the system yields a False Detection Rate of 8.5%(168 of 1963) and a False Reject Rate of 9.2%. So the total error of the system is 17.7%. Figure 3.19 shows correctly detected images, while Figure 3.20 shows images for which the ear detector failed to detect ears. Figure 3.21 shows images where non-ear objects are falsely detected.



Figure 3.19: Figure showing images with detected ears.



Figure 3.20: Figure showing images with false rejects.

Testing(bottom-up process):

The top-down process when applied on images with high resolutions may lead to more number of false detections at the end. Also, it has been observed that on MAGNA database, detection rate at high resolution is less than that at low resolution. For example, in Figure 3.22, at 1200x1900 image resolution (i.e., at 1:1 scale) along with the true ear a false ear is also detected but as the



Figure 3.21: Figure showing images with false detections.

image is scaled down to 1:10, only the ear region is detected. In order to take advantage of this, a bottom-up scaling procedure is proposed.



Figure 3.22: Images showing detection performance at various scales

In bottom-up scaling procedure, initially the input image is re-scaled to the lowest resolution and the image is scanned by the classifier. During scanning the image is further re-scaled by a factor 0.8 and scanned. If there are no detected regions at this scale, the image is again re-scaled to a resolution slightly higher than the previous resolution and the image is re-scanned by the classifier. This procedure is repeated until a single detected region is found in the image (or) for a certain number of iterations. There are some advantages in following such a bottom-up scaling procedure:

- As the classifier is scanned on a relatively lower number of combinations of sub-images, the number of false detects decreases
- Because of fewer number of sub-images, the total time taken to detect an ear in an image reduces significantly.

Results(bottom-up process): This technique is tested on MAGNA database. Of the 3115 images, 1963 images with no occlusions were considered for testing. As we have prior knowledge that the database contains only one ear region in each image, the classifier is modified to detect only one patch per image (i.e., patch with highest area). To increase the efficiency of the system, the image is first scaled down to 108×144 (9:100 of the original image) pixels and the detection algorithm is applied in the testing mode. If there is no ear detected at this scale, the image is scaled up to 10:100 and the process is repeated. This method is sequentially repeated with scales 11:100, 12:100, 15:100, 18:100, 20:100 and 22:100 of the original size until an object (region) is found.

For 1963 images in MAGNA database with no occlusion, at a scale of 9:100, 1302 images contain one detected region each. Of these 1302 regions, 20 are False Detects and the remaining 1282 are ear regions. The failed images, which are considered as False Rejects at this particular scale, are scaled up to 10:100 of the original size and tested with the same classifier and number of False Detects and False Rejects are noted. This process is repeated for different scales and the number of False Rejects and False Detects at each scale can be seen in Table 3.1

Table 3.1: Number of ear detects and rejects at each scale

Total Images	Scale	Number of detected regions	False Detects	Remaining images (False Rejects)
1963	9:100	1302	20	661
661	10:100	158	4	503
503	11:100	70	1	433
433	12:100	46	1	387
387	15:100	31	4	356
356	18:100	19	1	337
337	20:100	21	5	316
316	22:100	14	4	302

This procedure can be repeated with more number of scales, but doing so will increase the detection time. Comparing this results with the previous one, where detection is performed at only two scales i.e., at scale 10:100 and 50:100, the false detection rate is decreased from 8.5 % (168 of

1963) to 2 % (40 of 1963) and false reject rate increased from 9.2 % (182 of 1963) to 15 % (302 of 1963). Though the total error of the detection system remains nearly same (17 %) the present approach is more efficient because of less false detection rate. Figure 3.23, shows that at scales 9:100, 11:100 and 12:100 no region is detected. But, at scale 15:100 the ear is detected.

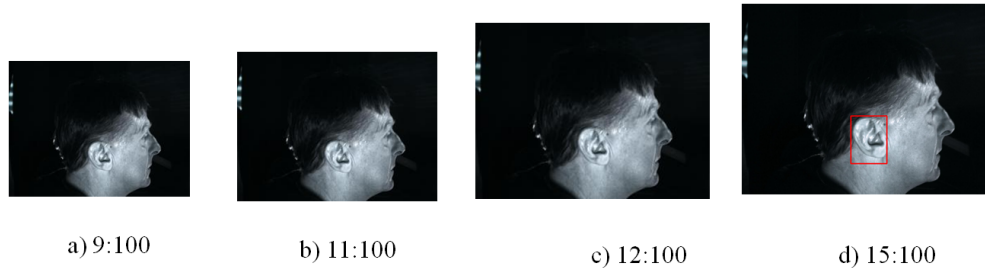


Figure 3.23: Images showing detection performance in bottom-up scaling approach

3.8.3 Ear segmentation

Training:

The entire ear structure is surrounded by the helix and in order to segment the ear, the correct estimate of helix edges are enough. As discussed in section 3.5.1, one of the strengths of the BEL algorithm is that it can learn an edge detector tuned for a specific domain. Ear images along with helix labels are used to train the BEL classifier. For training the classifier, 100 images from the MAGNA database are used. The helix edges are manually labeled by first applying Difference of Gaussians (DoG) filter and by removing unnecessary edges as shown in Figure 3.24. At each point of the label, a patch size of 'W' is considered by the BEL algorithm and used for training. In order to capture the structure of the ear around the helix a patch size of $W=30$ is chosen. As the maximum allowed depth for a tree increases, the positive samples will be well separated from the negative samples. At the same time, as depth increases the speed of the final detector decreases. So, an optimum value of depth 20 is set. Also, the number of weak classifiers in each AdaBoost node is set to 240.

Testing:



Figure 3.24: Training images for BEL (ear image along with the labeled helix edge)

In order to detect the helix edges in an ear, the trained classifier is used to scan each and every pixel by considering a window of size 30×30 around the pixel. The pixel is assigned a value in the range of [0-255] based on the probability of it being the edge pixel on the helix. At the end, the values of the pixels which belong to the helix edges are assigned with high value. A hard threshold is used to remove the spurious edges, and the longest connected path is chosen as helix.

After extracting helix edges, the contour of the helix is extracted by following the steps in Section 3.5.2 and the segmented ear is obtained.

Results:

The segmentation algorithm is tested on MAGNA database. Of the 3115 images, 1963 images with no ear occlusions are considered for testing. In order to reduce the search space for helix edges, ears are automatically detected first and then helix edges are extracted. To calculate the error for this algorithm independently (i.e., without relying on the output of the ear detection module preceding it), the ears which are not detected properly are manually cropped. To measure the efficiency of the algorithm, three segmentation measures were used.

1. The segmentation is said to be perfect segmentation, if the segmented region contains only the ear.
2. The segmentation is said to be an over segmentation, if the segmented region contains some false regions along with the ear. These kind of segmentations occur because of false detects of the helix edges by the BEL algorithm.
3. The segmentation is said to be an under segmentation, if the segmented part does not represent the ear at all. These kind of segmentations occur because of false rejects of the helix edges by BEL algorithm or during the post processing step used for segmentation.

Figure 3.25 shows examples for each of these outputs. On the MAGNA database, the segmentation algorithm yields a perfect segmentation rate of 66.33% (1302 of 1963), an over segmentation of 12.38% (243 of 1963) and an under segmentation of 21.29% (418 of 1963). Most of the segmentation errors are due to the presence of complex helix edges, variations in lighting conditions and minor occlusions due to hair.

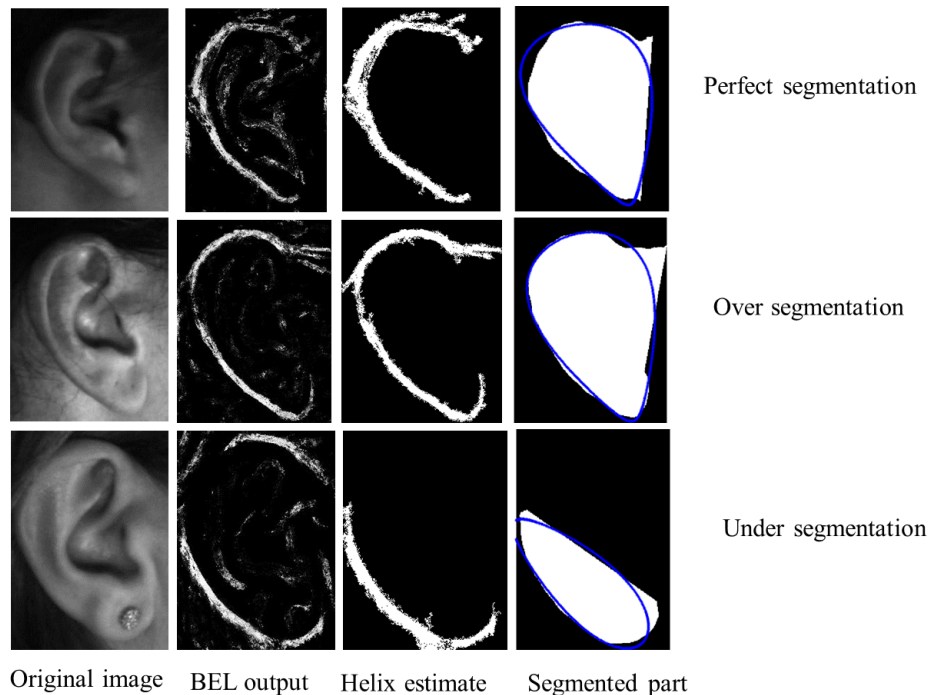


Figure 3.25: Figure showing different types of segmentation outputs.

3.8.4 Classification results

After segmenting the ear, the modified gradient descent algorithm (discussed in section 3.6.3) is used to estimate the parameters $(a/b, \gamma)$. The error occurred due to incorrect estimation of parameters by modified gradient descent algorithm is termed as gradient descent error. This error is measured manually by visually observing the output of each image. The modified gradient descent algorithm is applied on perfect and over segmented ears and an error of 1% (15 out of 1546) is reported. Examples for cases where gradient descent failed can be seen in Figure 3.26.

Using parameters a/b and γ , ears are classified in to one of four classes (i) Round, (ii) Rectan-

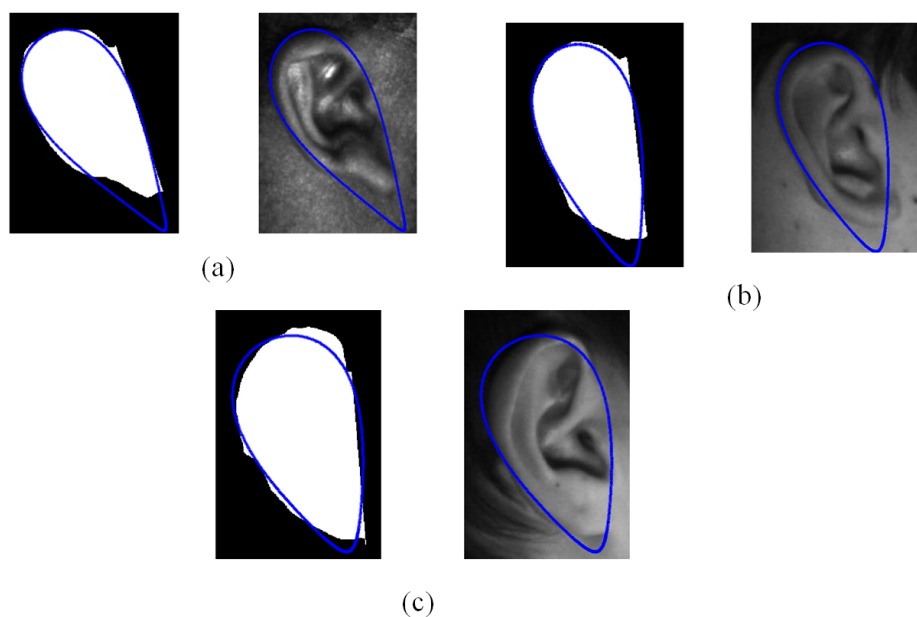


Figure 3.26: Figure showing cases where gradient error occurred.

gle, (iii) Oval, (iv) Triangle, based on the flowchart described in Figure 3.15. These automatically classified ears are compared with the ground truths which have been manually established. Here, in order to study the performance of the classification component independently, only perfect and over segmented ears are used. Table 3.2 shows the number of ears in each class, whose ground truth is established manually. Ground truth is formed based on the opinion of two human operators. The opinions are combined using a voting scheme. Conflicts are once again manually corrected.

Table 3.2: Ground truth established by manual classification of ear images

Ground Truth	Perfectly segmented images	Perfect and Over segmented images
Round	27	33
Rectangle	112	144
Oval	877	1042
Triangle	286	326

Table 3.3 shows the confusion matrix of the automated classification scheme on perfectly seg-

mented images. A classification rate of 91.5% is achieved for this experiment.

Table 3.3: Confusion matrix for perfectly segmented images

		Predicted class				Total
		Round	Rectangle	Oval	Triangle	
Ground truth	Round	18	0	9	0	27
	Rectangle	0	72	40	0	112
	Oval	7	9	837	24	877
	Triangle	0	0	21	265	286
Total		25	81	907	289	1302

Table 3.4 shows the results of the classification scheme on both perfect and over segmented images. A classification rate of 83.5% is achieved for this experiment.

Table 3.4: Confusion matrix for perfect + over segmented images

		Predicted class				Total
		Round	Rectangle	Oval	Triangle	
Ground truth	Round	20	0	12	1	33
	Rectangle	0	74	57	13	144
	Oval	11	16	893	122	1042
	Triangle	0	0	23	303	326
Total		31	90	985	439	1545

Manual classification is sometimes prone to biasing and errors. In some cases, the automated classification output can often supply new perspectives and potential corrections to the manually established ground truth. In this sense, the comparison of automated classification with manual classification can produce important lessons regarding not only how humans perceive the shape of the ear, but also for revising some of the human labels. To understand potential human classification errors, those images with no error in the segmentation, and whose estimated γ values are not in the range of 0.2 to 0.3 (boundary for rectangle and oval) and 0.6 to 0.7 (boundary for oval and triangle), are considered. The automated classification scheme for such ears is supposed to predict the class correctly (i.e., with high confidence). An ear is said to be misclassified by the human, if the class label assigned by the human does not coincide with the class assigned by the automated

classification scheme with high confidence. Such an error is termed as human classification error. For example, consider the case in Figure 3.27(a). Because of the complexity of the lobule, it is highly probable that a human may be confused when annotating the class and would end up labeling it as an oval. However, the automated scheme would label this as a triangle with high confidence. Same is the case with Figure 3.27(b) and (c), where a human labeled the ears as oval while the automated scheme labeled them as rectangle with high confidence. 101 such images are corrected (ground truth correction) and the performance is re-evaluated.

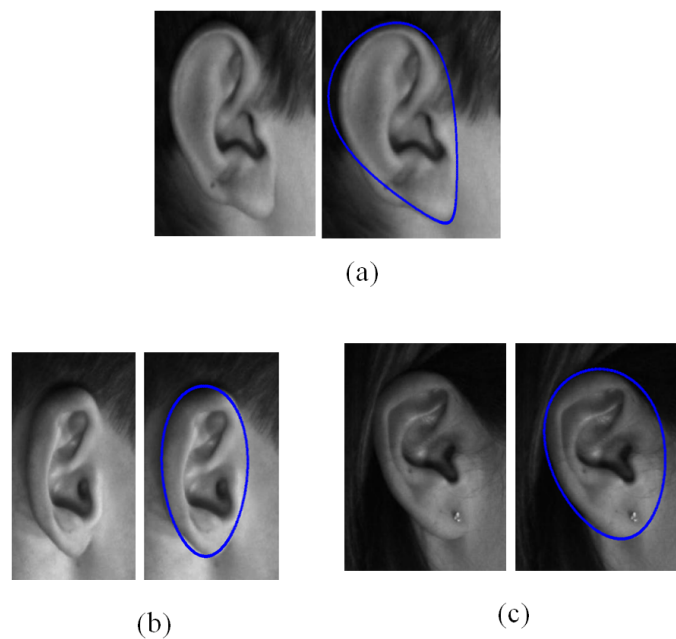


Figure 3.27: Figure showing examples of cases where a human failed to assign the correct class.

Table 3.5 shows the number of ears in each class after correcting for human errors.

Table 3.6 shows the results of the classification scheme on perfectly segmented images after correcting for human errors. A classification rate of 97.9% is achieved for this experiment.

Table 3.7 shows the results of the classification scheme on perfect and over segmented images after correcting for human errors. A classification rate of 88.9% is achieved for this experiment.

Table 3.5: Ground truth after correcting for human errors.

Ground Truth	Perfectly segmented images	Perfect and Over segmented images
Round	26	32
Rectangle	91	123
Oval	909	1073
Triangle	276	317

Table 3.6: Confusion matrix for perfectly segmented images after correcting human errors

		Predicted class				Total
		Round	Rectangle	Oval	Triangle	
Ground truth	Round	24	0	2	0	26
	Rectangle	0	80	10	1	91
	Oval	1	1	895	12	909
	Triangle	0	0	0	276	276
Total		25	81	907	289	1302

Table 3.7: Confusion matrix for perfect and over segmented images combined after correcting for human errors.

		Predicted class				Total
		Round	Rectangle	Oval	Triangle	
Ground truth	Round	26	0	5	1	32
	Rectangle	0	82	27	14	123
	Oval	5	8	951	109	1073
	Triangle	0	0	2	315	317
Total		31	90	985	439	1545

3.9 Conclusion and future work

An ear classification scheme based on shape has been proposed and evaluated. The ability of learning-based algorithms to learn under a wide variety of conditions has been explored for ear de-

tection and segmentation. However, it should be noted that the BEL based segmentation algorithm performs poorly. Considering only perfectly segmented images as being successful outputs, the error for the segmentation algorithm is 23%. Also, when considering the classification result on perfectly segmented images, the error is 2%. Figure 3.28 shows the error rates for each component in the classification system. As the ear classification system is a sequential system, the error at one stage will be propagated to the subsequent stages. Because of the proposed bottom-up ear detection approach, although the total error remains constant, the False Detection Rate is reduced. Using such methods helps in reducing the propagated error, since the number of false detects is reduced.

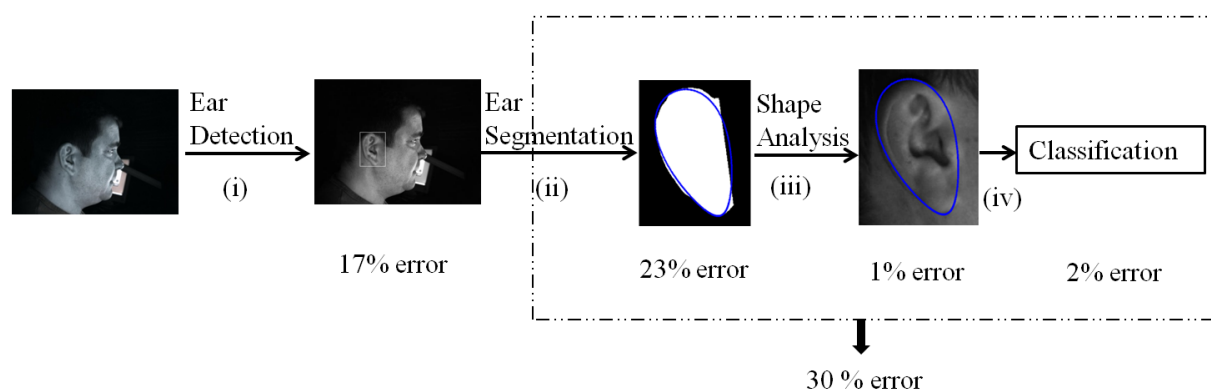


Figure 3.28: Figure showing errors of individual modules.

As discussed in the previous section, the classification rate for perfect and over segmented images is 88.93%. When considering under segmented images as being failed outputs, the classification rate is 70%. This implies that if ear segmentation, shape analysis and classification are considered together as a single unit, the total error would be 30%. Thus, future work will include reducing the error in each stage independently. Since this is the first time automated ear classification is being addressed, this work attempts to highlight a new direction of research that would be helpful for classifying side profile face images in the future.

Chapter 4

Conclusion and Future Work

In this thesis, an indexing scheme for iris and a classification scheme for ear has been proposed and evaluated. The iris indexing scheme is based on the Burrows Wheelers Transform (BWT) and the ear classification scheme is based on parameterizing the different shapes of the ear, *viz.*, rectangular, oval, round and triangular.

4.1 Iris Indexing

The proposed iris indexing technique was tested on CASIA version 3.0 database. Results indicate that the proposed scheme had a significant improvement (with $\gamma = 90.90\%$) over the previously reported work by Mukherjee et.al [42] (with $\gamma = 85.79\%$). Further work is required to reduce the penetration rate, for instance, by using an alternate searching mechanism which combines different index patterns (P1, P2) of the same length.

Further investigation has to be carried out to design appropriate data structures to organize the indices generated for the iris images. This could result in a potentially efficient retrieval strategy for very large databases. It should be noted that, in this work, only binary patterns of the iris image were considered for designing an indexing scheme. Further progress could be carried out to design a scheme which could also use non-binary patterns.

4.2 Ear Classification

The proposed ear classification scheme was tested on the MAGNA database. A classification rate of 98% was successfully achieved on perfectly segmented images using the parameters a/b and γ . These parameters are believed to represent many of the most commonly represented ear shapes like rectangular, oval, round and triangle. Ears with more complex shapes, however, require the inclusion of a shape representation that could accommodate more parameters.

The classification scheme designed in this thesis uses only the shape of the ear. An alternate approach could be to use the structure of the ear. Future work could involve designing a classification scheme using structure of the ear. Further, the shape and structure of the ear when used together could result in improved performance. It should also be noted that, in the present work, experiments are only conducted on profile images. The performance of the shape-based classification scheme on the off-axis facial images could be studied and a new system that can accommodate pose variations could be designed.

The sequential design of the system as discussed in section 3.9, results in propagation of error from one stage to another resulting in a gradual decrease in the system performance. Future work, involving design of automated metrics that could reject erroneous samples in the initial stages itself, could be a good addition to this work. It has been observed that, compared to the other stages in the classification system, the errors in the detection stage and segmentation stage are relatively high. Alternate detection and segmentation techniques could be researched to enhance the performance of the designed ear classification system.

Appendix A

Sample ear images

A.1 Classification based on quality

As discussed in the section 3.8.1, the left profile images in MAGNA database are divided in to three types : Ears with no occlusion, Ears with partial occlusions and Ears with full occlusions. Few examples of each category are shown in Figures A.1-A.3



Figure A.1: Sample images showing images with no ear occlusions.



Figure A.2: Sample images showing images with partial ear occlusions.



Figure A.3: Sample images showing images with fully occluded ears.

A.2 Classification based on shape

As discussed in section 3.8.4, ears in the MAGNA database can be divided in to four types: Round, Rectangle, Oval, Triangle. Ground truth for MAGNA database is established by a human and Figures A.4-A.7 show sample images for each category.



Figure A.4: Sample images showing round shape ears.



Figure A.5: Sample images showing rectangle shape ears.



Figure A.6: Sample images showing oval shape ears.



Figure A.7: Sample images showing triangle shape ears.

References

- [1] T.W. Sadler and J. Langman, *Langman's essential medical embryology*, vol. 1, Lippincott Williams & Wilkins, 2004.
- [2] A.V. Iannarelli, *Ear identification*, Paramount Publishing Company, 1989.
- [3] J. Wu, S.C. Brubaker, M.D. Mullin, and J.M. Rehg, "Fast asymmetric learning for cascade face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 369–382, 2008.
- [4] L. Alvarez, E. González, and L. Mazonza, "Fitting ear contour using an ovoid model," in *International Carnahan Conference on Security Technology*, 2005, pp. 145–148.
- [5] A.K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.
- [6] M. Faundez-Zanuy, "Biometric security technology," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 21, no. 6, pp. 15–26, 2006.
- [7] A.A. Ross, *Information fusion in fingerprint authentication*, Ph.D. thesis, Michigan State University, 2003.
- [8] A. Ross and A.K. Jain, "Multimodal biometrics: An overview," in *Proceedings of 12th European Signal Processing Conference*, 2004, pp. 1221–1224.
- [9] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*, Springer-Verlag New York Inc, 2009.
- [10] R.P. Wildes, "Iris recognition: an emerging biometric technology," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1348–1363, 1997.
- [11] A. Lefohn, B. Budge, P. Shirley, R. Caruso, and E. Reinhard, "An ophthalmologist's approach to human iris synthesis," *Computer Graphics and Applications, IEEE*, vol. 23, no. 6, pp. 70–75, 2003.
- [12] J.V. Forrester, A. Dick, P. Mcmenamin, and W. Lee, "Anatomy of the eye and orbit," *The Eye: Basic Sciences in Practice*, vol. 2, pp. 66–90, 2002.
- [13] J. Daugman, "Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1927–1935, 2006.

- [14] J.G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1161, 1993.
- [15] A. Abhyankar and S. Schuckers, "Active shape models for effective iris segmentation," in *SPIE*, 2006, vol. 6202, pp. 6202:H1–H10.
- [16] S. Shah, "Enhanced iris recognition: Algorithms for segmentation, matching and synthesis," M.S. thesis, West Virginia University, 2008.
- [17] K. Bae, S. Noh, and J. Kim, "Iris feature extraction using independent component analysis," in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2003, pp. 1059–1060.
- [18] V. Dorairaj, N.A. Schmid, and G. Fahmy, "Performance evaluation of iris based recognition system implementing PCA and ICA encoding techniques," in *SPIE*, 2005, vol. 5779, pp. 51–58.
- [19] WW Boles and B. Boashash, "A human identification technique using images of the iris and wavelet transform," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1185–1188, 1998.
- [20] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Efficient iris recognition by characterizing key local variations," *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 739–750, 2004.
- [21] Z. Sun, T. Tan, and X. Qiu, "Graph matching iris image blocks with local binary pattern," *Advances in Biometrics*, pp. 366–372, 2005.
- [22] G.L. Streeter, "Development of the auricle in the human embryo," *Contrib. Embryol*, vol. 69, pp. 111, 1922.
- [23] J. Davis, "Surgical embryology," *Davis J, editor. Aesthetic and Reconstructive Otoplasty. 1st ed. New York, NY: Springer-Verlag*, pp. 93–125, 1987.
- [24] F. Wood-Jones, I. Wen, et al., "The development of the external ear," *Journal of anatomy*, vol. 68, pp. 525, 1934.
- [25] B. Arbab-Zavar and M.S. Nixon, "On guided model-based analysis for ear biometrics," *Computer Vision and Image Understanding*, vol. 115, no. 4, pp. 487–502, 2011.
- [26] A. Bertillon, *La photographie judiciaire: avec un appendice sur la classification et l'identification anthropométriques*, Gauthier-Villars, 1890.
- [27] M. Burge and W. Burger, "Using ear biometrics for passive identification," in *Proceedings of the IFIP TC11 14th international conference on information security, SEC*, 1998, vol. 98, pp. 139–148.
- [28] M. Choras, "Ear biometrics," in *Encyclopedia of Biometrics*, pp. 233–240. Springer, 2009.
- [29] M. Choras, "Ear biometrics based on geometrical feature extraction," *Electronic Letters on Computer Vision and Image Analysis (Journal ELCVIA)*, vol. 5, no. 3, pp. 84–95, 2005.

- [30] D.J. Hurley, M.S. Nixon, and J.N. Carter, “Force field feature extraction for ear biometrics,” *Computer Vision and Image Understanding*, vol. 98, no. 3, pp. 491–512, 2005.
- [31] B. Victor, K. Bowyer, and S. Sarkar, “An evaluation of face and ear biometrics,” in *16th International Conference on Pattern Recognition*, 2002, vol. 1, pp. 429–432.
- [32] H. Chen and B. Bhanu, “Human ear recognition in 3d,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 718–737, 2007.
- [33] P. Yan and K.W. Bowyer, “Biometric recognition using 3d ear shape,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1297–1308, 2007.
- [34] S. Cadavid and M. Abdel-Mottaleb, “Human identification based on 3d ear models,” in *First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, 2007, pp. 1–6.
- [35] A. Mhatre, S. Palla, S. Chikkerur, and V. Govindaraju, “Efficient search and retrieval in biometric databases,” in *SPIE Defence and Security Symposium*, 2005, vol. 5779, pp. 265–273.
- [36] A. Ross, “Iris Recognition: The Path Forward,” *IEEE Computer*, vol. 43, no. 2, pp. 30–35, 2010.
- [37] X. Qiu, Z. Sun, and T. Tan, “Global texture analysis of iris images for ethnic classification,” *Advances in Biometrics*, vol. 3832, pp. 411–418, 2005.
- [38] L. Yu, D. Zhang, K. Wang, and W. Yang, “Coarse iris classification using box-counting to estimate fractal dimensions,” *Pattern Recognition*, vol. 38, no. 11, pp. 1791–1798, 2005.
- [39] X. Qiu, Z. Sun, and T. Tan, “Coarse iris classification by learned visual dictionary,” *Advances in Biometrics*, vol. 4642/2007, pp. 770–779, 2007.
- [40] A. Ross and M. SamSunder, “Block Based Texture Analysis for Iris Classification and Matching,” in *IEEE Computer Society Workshop on Biometrics at the Computer Vision and Pattern Recognition (CVPR) Conference*, June 2010.
- [41] F. Hao, J. Daugman, and P. Zielinski, “A fast search algorithm for a large fuzzy database,” *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 203–212, 2008.
- [42] R. Mukherjee and A. Ross, “Indexing iris images,” in *19th International Conference on Pattern Recognition*, 2008, pp. 1–4.
- [43] H. Mehrotra, B. Majhi, and P. Gupta, “Robust iris indexing scheme using geometric hashing of SIFT keypoints,” *Journal of Network and Computer Applications*, vol. 33, no. 3, pp. 300–313, May 2010.
- [44] M. Burrows and D.J. Wheeler, *A block-sorting lossless data compression algorithm*, Technical Report: Digital Equipment Corporation, Palo Alto, CA, 1994.

- [45] D. Adjeroh, T. Bell, and A. Mukherjee, *The Burrows-Wheeler Transform: Data Compression, Suffix Arrays, and Pattern Matching*, Springer-Verlag New York Inc, 2008.
- [46] S. Makthal and A. Ross, "Synthesis of Iris Images using Markov Random Fields," in *Proc. of 13th European Signal Processing Conference (EUSIPCO)*, 2005.
- [47] J. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, 2004.
- [48] D. A. Adjeroh and M. C. Lee, "On ratio-based color indexing," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 36–48, 2001.
- [49] A. Firth, T. Bell, A. Mukherjee, and D. Adjeroh, "A comparison of BWT approaches to string pattern matching," *Software: Practice and Experience*, vol. 35, no. 13, pp. 1217–1258, 2005.
- [50] S. Shah and A. Ross, "Iris Segmentation Using Geodesic Active Contours," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 824–836, December 2009.
- [51] P. Yan and K.W. Bowyer, "Empirical evaluation of advanced ear biometrics," in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, 2005, pp. 41–41.
- [52] M. Burge and W. Burger, "Ear biometrics in computer vision," in *15th International Conference on Pattern Recognition*, 2000, vol. 2, pp. 822–826.
- [53] S. Ansari and P. Gupta, "Localization of ear using outer helix curve of the ear," in *International Conference on Computing: Theory and Applications*, 2007, pp. 688–692.
- [54] S. Attarchi, K. Faez, and A. Rafiei, "A new segmentation approach for ear recognition," in *Advanced Concepts for Intelligent Vision Systems*. Springer, 2008, pp. 1030–1037.
- [55] H. Liu and D. Liu, "Ear segmentation using histogram based k-means clustering and hough transformation under CVL dataset," in *Proceedings of SPIE*, 2009, vol. 7495, pp. 74952N–74952N–7.
- [56] A. Cummings, M. Nixon, and J. Carter, "A novel ray analogy for enrolment of ear biometrics," in *IEEE Conference on Biometrics: Theory, Applications Systems (BTAS)*, 2010.
- [57] A. Sanaa, P. Gupta, and R. Purkait, "Ear biometrics: A new approach," in *International Conference on Advances in Pattern Recognition(ICAPR)*, 2007, pp. 46–50.
- [58] M. Abdel-Mottaleb and J. Zhou, "Human ear recognition from face profile images," *Advances in Biometrics*, pp. 786–792, 2005.
- [59] S. Prakash, U. Jayaraman, and P. Gupta, "A skin-color and template based technique for automatic ear detection," in *Seventh International Conference on Advances in Pattern Recognition*, 2009, pp. 213–216.

- [60] T. Yuizono, Y. Wang, K. Satoh, and S. Nakayama, “Study on individual recognition for ear images by using genetic local search,” in *Proceedings of the 2002 Congress on Evolutionary Computation*, 2002, vol. 1, pp. 237–242.
- [61] S.M.S. Islam, R. Davies, M. Bennamoun, and A.S. Mian, “Efficient detection and recognition of 3d ears,” *International Journal of Computer Vision*, pp. 1–22, 2011.
- [62] A. Abaza, *High performance image processing techniques in automated identification systems*, Ph.D. thesis, West Virginia University, 2009.
- [63] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. I–511.
- [64] P. Dollar, Z. Tu, and S. Belongie, “Supervised learning of edges and object boundaries,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 1964–1971.
- [65] Z. Tu, “Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering,” in *Tenth IEEE International Conference on Computer Vision*, 2005, vol. 2, pp. 1589–1596.
- [66] R.C. González and R.E. Woods, *Digital Image Processing*, Pearson/Prentice Hall, 2008.