BIOMET: A MULTIMODAL BIOMETRIC AUTHENTICATION SYSTEM FOR PERSON IDENTIFICATION & VERIFICATION

A Thesis submitted to

SAURASHTRA UNIVERSITY

For the award of the degree of

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE

In the Faculty of Science

Submitted By

HIREN D. JOSHI

Department of Computer Science

Rollwala Computer Centre

Gujarat University, Ahmedabad

(Ph.D. Reg. No.: 3363 & Date: 02/03/2006)

Under the Guidance of

DR. N. N. JANI

Ex. Prof. & Head, Dept. of Computer Science,

Saurashtra University

(Guide Recog. No.: 744 & Date: 13/06/1999)

DIRECTOR – MCA PROGRAMME, SKPIMCS

DEAN – FACULTY OF COMPUTER & IT

KADI SARVA VISHWAVIDYALAYA, GANDHINAGAR

JANUARY 2009

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER – 1

Biometric Authentication System

1.1 Authentication

In our daily life, we often check if a particular person genuinely is who he or she purports to be. For example

- Alice checks the authenticity of Bob when she meets him on the street or speaks to him on the phone.
- Alice proves her own genuineness with her Personal Identity Number (PIN) at a cash machine or with her passport at international checkpoints.
- When Alice receives a letter, she checks the genuineness of the sender by looking at the signature.

The process of checking genuineness (authenticity) is known as authentication. Authentication is the proof of genuineness**.**

Authentication can be established in different ways. Let's take a look at procedure that can be used for authentication outside internet. We assume that Bob wants to check Alice's authenticity. He has three basic ways of doing this:

1.

Bob checks whether Alice knows a certain fact (something you know). Examples are:

–Passwords

–Secret numbers

–Secret keys or personal information

This is called authentication by knowledge

2.
Bob checks whether Alice is in possession of any object that is difficult to forge (something you have). For example

–Passport

This is called authentication by possession

3.
Bob checks an unmistakable, difficult to fake, personal characteristic of Alice (something you are). Examples

–Facial Image

–Fingerprints

This is called authentication by personal characteristics

In short, one authenticates oneself through something one knows, something one has, or what one is.

Authentication helps establish trust by identifying who a particular user is. Authentication ensures that the claimant is really what he/she claims to be.

There are many ways to authenticate a user. Traditionally, user ids and passwords have been used. But there are many security concerns in this mechanism. Password can travel in clear text or can be stored in clear text on the server, both of which are dangerous propositions. Modern password -based authentication techniques use alternatives as encrypting passwords, or using something derived from the password to protect them.

Authentication tokens add randomness to the password-based mechanism, and make it far more secure. This mechanism requires the user to possess the tokens. Authentication tokens are quite popular in application that demand high security.

Modern life today is littered with passwords: they stand in front of everything from children's personal computers to extensive business and financial resources. In theory, a single person memorizes a password, it's hard to guess, it's never written down, and it's never shared. In practice, however, people constantly violate these expectations. Passwords are often written down, shared with other people, or chosen from among a small number of easy to guess words. There is an inevitable tug of war between choosing a password that's easy to remember and one that's hard to guess. Some systems try to force people to choose hard to guess passwords, and many people respond by keeping written lists of their hard to guess passwords. Of course, once this list is copied or stolen, the passwords provide no protection at all.

Although passwords are both widely used and easily compromised, they illustrate the fundamental mechanism of automated authentication: the user must provide some information or input that cannot be provided by someone else. Consider what happens if an authorized user named Cathy tries to log in to a server, such as an e-mail server. The server takes information Cathy provides and compares it with her previously stored information. If the comparison is satisfactory, the server acknowledges Cathy's identity. If a different person, Henry, for example, tries to impersonate Cathy, he should not be able to provide the same information, so the comparison should fail. We summarize these features as follows:

(1) Cathy provides an authenticator
A data item that cannot be provided by anyone else.

(2) The server contains a verifier
A data item that can verify the correctness of the authenticator.

(3) The server uses a verification procedure
An algorithm that compares an authenticator with a verifier.

(4) There is generally a base secret
A data item in Cathy's possession that produces the authenticator.

As we will see shortly, an authentication system's features take different forms according to the authentication factors involved. We examine this with examples in which Cathy tries to log in to her mail server while Henry tries to masquerade as Cathy. Different authentication factors provide subtly different types of information about a person's identify. In some cases, this simply affects the confidence we have in the results, while in other cases it enables other uses of the authentication.

Password and PINs

The simplest implementations of passwords and personal identification numbers yield the simplest of all authentication mechanisms. Cathy’s memorized password serves as the authenticator, verifier, and base secret. The verification procedure simply performs a character string comparison of the authenticator and verifier. In practice, password based systems incorporate various cryptographic techniques to resist attacks, notably password hashing.

Passwords work reliably only as long as they are not guessed or otherwise disclosed to potential adversaries through accident, subversion, or intentional sharing. If Cathy chooses her favorite color as a password, an acquaintance might guess it and try to log on as her. Since she chose a common word as a password, it’s also possible that Henry or some other attacker might use a “dictionary attack” to discover her password in a file of hashed passwords. If Cathy logs in to her mail server across the Internet, Henry might be able to intercept her password while in transit, and then use it himself.

Cards and Tokens

Physical authentication devices, such as smart cards and password tokens, were developed to eliminate certain weaknesses associated with passwords. A major benefit of cards and tokens is that they can’t be shared with the same freedom as sharing passwords. If Cathy shares her token with someone else, the other person can log in, but Cathy cannot.

In general, these devices store a large base secret. Since the token carries the secret, Cathy doesn't need to memorize it: she simply has to carry the token and have it available when she logs in. The devices usually contain a special procedure the uses the base secret to generate a hard to predict value for the authenticator. When Cathy needs to log in, her device generates the correct authenticator. Then she either types it in instead of a password, or she relies on a special authentication client to transmit the authenticator to the mail server.

Subverting the System

We use authentication systems because people occasionally try to misrepresent their identities. The previous section talked about Henry, who tried on occasion to assume Cathy's identity. Henry may be pursuing particular outcomes when he tries to subvert the authentication system; the next subsection characterizes those outcomes as risks. Henry might take a small number of general approaches to subvert the authentication system; the subsequent two subsections characterize those approaches as attacks. The final subsection reviews defenses used to resist these attacks.

Risks

The following risks represent different objectives an attacker like Henry might have when trying to subvert an authentication system. The attacker usually has a grander goal in mind, such as the embezzlement of a certain amount of money or the capture of certain goods or services. But for the authentication system itself, the attacker's goal is usually limited to one of the three described next: masquerade, multiple identities, or identify theft.

Masquerade

This is the classic risk to an authentication system. If Henry's goal is masquerade, he's simply trying to convince the system that he is in fact someone else, perhaps Cathy, since the system already knows how to recognize her. Henry proceeds by trying to trick the system into accepting him as being the other person.

Multiple Identities

Some systems, particularly those that dispense a government's social services program, are obligated to provide service to qualifying individuals within their jurisdiction. These individuals generally show up in person and request services. For many reasons, however, some people have found it profitable to register two or more times for the same benefits. For example, Henry might try to register himself twice or more so that he can collect multiple entitlement payment, or perhaps he can sell the registration to someone else, who, for whatever reason, may not qualify for the social services. Driver's license systems are similarly undermined if fraudulent identities are allowed to enter the system.

Identity Theft

This is the extreme case of authentication risks when an attacker establishes new accounts that are attributed to a particular victim but authenticated by the attacker. In a simple masquerade, the attacker may assume the victim’s identity temporarily in the context of systems the victim already uses. In an identity theft, the attacker collects personal identification information for a victim and uses it to assume the victim’s identity in a broad range of transactions. In a typical fraud, Henry opens credit accounts in Cathy’s name, although it’s also common for the criminal to loot existing accounts.

Trial and Error Attacks

When Henry goes after an authentication system, the first thing he considers is whether trial and error attempts are likely to succeed. Every authentication system is subject to some type of trial and error attack. The classic attack on passwords is an interactive attack, in which the attacker simply types one possible password after another, until either the list of possible passwords, or the attacker, is exhausted. Most systems resist such attacks by keeping track of the number of unsuccessful authentication attempts and then sounding an alarm when such things occur.

Password Guessing

With the introduction of password hashing and other techniques for obscuring a password cryptographically, a different technique emerged: the offline attack. These attacks take a copy of a cryptographically protected password and use a computer to try to “crack” it. An offline attack may succeed in two cases: when cracking small passwords and when using a dictionary attack. If people use small passwords or easily memorized common English terms, the offline attack can exhaustively check every possible password by comparing its hashed equivalent against the hashed or otherwise encrypted password being cracked. In a dictionary attack, the exhaustive search is against words in a list that are presumed to be likely choices for passwords. In fact, dictionary attacks are fast enough the dictionary can contain lots of unlikely words as well. In studies performed on hashed password files, dictionaries of English words have been successfully used in dictionary attacks to crack between 24.2 percent and 35 percent of the files passwords.

Why Use Biometrics?

(1) Convenient authentication: The convenience of quick and easy authentication makes for a smoother system of identity assurance than using keys, cards, tokens, or PINs. With biometric technology, there is nothing to lose or forget since the characteristics or traits of the person serve as the identifiers. Many of these "individual" identifiers remain relatively unchanged and are enduring over time. In addition, biometric technologies also provide greater convenience for the information technology and support organizations that manage user authentication. For example, biometrics helps to eliminate the need to replace badges or reset PINs.

(2) Increase need for strong authentication: Passwords and PINs can be stolen easily. Biometrics should reduce

the risk of compromise the likelihood that an adversary can present a suitable identifier and gain unauthorized access. With today’s intense focus on greater security for logical and physical access, biometrics offers an attractive method for guarding against stolen or lost identifiers, such as cards or passwords.

(3) Decreased costs: Over the years improvement in hardware and software technologies has brought down the costs of biometric authentication to be affordable at the commercial market level. In addition, advancements in computing power, networking, and database systems have allowed biometric systems to become easier to use over wide geographical and networked areas. Management systems have been developed to administer a cluster of devices.

(4) Increased government and industry adoption: Today numerous public and private organizations are using biometrics. As an outgrowth of the September 11, 2001, terrorist attacks; an increased awareness of physical security and public safety has also helped make biometrics attractive. Manufacturers are increasingly looking to provide biometrics with computer equipment and products. Many companies offer biometric authentication options and include biometric sensors and matching capabilities as part of their products. For example, there are instances of fingerprint sensors built right into keyboards, mice, and laptops, and second

generation sensors are becoming much more “plug and play.”

An Introduction to Biometric Authentication Systems

1.2 Introduction

“Biometric technologies” are automated methods of verifying or recognizing the identity of a living person based on a physiological or behavioral characteristic

There are two key words in this definition: “automated” and “person”. The word “automated” differentiates biometrics from the larger field of human identification science. Biometric authentication techniques are done completely by machine, generally (but not always) a digital computer. Forensic laboratory techniques, such as latent fingerprint, DNA, hair and fiber analysis, are not considered part of this field. Although automated identification techniques can be used on animals, fruits and vegetables, manufactured goods and the deceased, the subjects of biometric authentication are living humans. For this reason, the field should perhaps be more accurately called “anthropometrics authentication”.

The second key word is “person”. Statistical techniques, particularly using fingerprint patterns, have been used to differentiate or connect groups of people or to probabilistically link persons to groups, but biometrics is interested only in recognizing people as individuals. All of the measures used contain both physiological and behavioral components, both of which can vary widely or be quite similar across a population of individuals. No technology is purely one or the other, although some measures seem to be more behaviorally influenced and some more physiologically influenced. The behavioral component of all biometric measures introduces a "human factors" or "psychological" aspect to biometric authentication as well.

In practice, we often abbreviate the term "biometric authentication" as "biometrics", although the latter term has been historically used to mean the branch of biology that deals with its data statistically and by quantitative analysis.

So "biometrics", in this context, is the use of computers to recognize people, despite all of the across-individual similarities and within-individual variations. Determining "true" identity is beyond the scope of any biometric technology. Rather, biometric technology can only link a person to a biometric pattern and any identity data (common name) and personal attributes (age, gender, profession, residence, nationality) presented at the time of enrollment in the system. Biometric systems inherently require no identity data, thus allowing anonymous recognition.

Ultimately, the performance of a biometric authentication system, and its suitability for any particular task, will depend upon the interaction of individuals with the automated mechanism. It is this interaction of technology with human physiology and psychology that makes "biometrics" such a fascinating subject.

1.3 History of Biometrics

References to biometrics, as a concept, date back over a thousand years. In East Asia, potters placed their fingerprints on their wares as an early form of brand identity. In Egypt's Nile Valley, traders were formally identified based on physical characteristics such as height, eye color, and complexion. This information helped identify trusted traders whom merchants had successfully transacted business with in the past.

In the nineteenth century, law enforcement professionals and researchers, spurred by the need to identify recidivist criminals, tried to find better ways to identify people. In France, Alphonse Betrillon developed anthropometrics, or a method of taking multiple physical measurements of the human body as well as noting peculiar characteristics of a person. In the United Kingdom, attention focused on fingerprints, thanks, in part, to work done by police officials in British India. As explained in future chapters, fingerprints came to be the recognized dependable identifiers for law enforcement purposes.

Interestingly enough biometric technology, in the sense of automated methods of human recognition, first appeared as an application for physical access control. This evolution did not track the growth of e-commerce but created more efficient and reliable authentication for physical access. Biometrics as a commercial, modern technology has been around since the early 1970's, when the first commercially available device was brought to market. One of the first commercial applications was used in 1972 when a Wall Street company, Shearson Hamil, installed Idetimat, a finger measurement device that served as a time keeping and monitoring application. Since this 1972 deployment, biometrics has improved tremendously in ease of use and diversity of applications. The advancement of biometrics has been driven by the increased computing power at lower costs, better algorithms, and cheaper storage mechanisms available today.

Primitive biometrics such as height, special body marks had been in use to identify people since the time of the ancient Egyptians. Fingerprints have been used for many years by police departments for criminal identification around the world. With the current ever improving biometrics technology has opened a window of possibilities. Today, biometric technology is not only being used for access to high security areas, but also for network security.

The scientific literature on quantitative measurement of humans for the purpose of identification dates back to the 1870s and the measurement system of Alphonse Bertillon. Bertillon’s system of body measurements, including such measures as skull diameter and arm and foot length, was used in the USA to identify prisoners until the 1920s. Henry Faulds, William Herschel and Sir Francis Galton proposed quantitative identification through fingerprint and facial measurements in the 1880s.The development of digital signal processing techniques in the 1960s led immediately to work in automating human identification. Speaker and fingerprint recognition systems were among the first to be explored. The potential for application of this technology to high-security access control, personal locks and financial transactions was recognized in the early 1960s. The 1970s saw development and deployment of hand geometry systems, the start of large-scale testing and increasing interest in government use of these “automated personal identification” technologies. Retinal and signature verification systems came in the 1980s, followed by face systems. Iris recognition systems were developed in the 1990s.

1.4 The Biometric Characteristics

Examples of physiological and behavioral characteristics currently used for automatic identification include fingerprints, voice, iris, retina, hand, face, handwriting, keystroke, and finger shape. But this is only a partial list as new measures (such as gait, ear shape, head resonance, optical skin reflectance and body odor) are being developed all of the time. Because of the broad range of characteristics used, the imaging requirements for the technology vary greatly. Systems might measure a single one-dimensional signal (voice); several simultaneous one-dimensional signals (handwriting); a single two-dimensional image (fingerprint); multiple two dimensional measures (hand geometry); a time series of two-dimensional images (face and iris); or a three-dimensional image (some facial recognition systems).

Which biometric characteristic is best? The ideal biometric characteristic has five qualities: robustness, uniqueness, universality, accessibility and acceptability.

1. Robustness measures how well a biometric unchanging on an individual over time.
2. Uniqueness is how well the biometric separates one individual from another.
3. Universality describes how common a biometric is found in each individual.
4. Accessibility explains how easy it is to acquire a biometric for measurement.

5. Acceptability means that people do not object to having this measurement taken from them.

Quantitative measures of these five qualities have been developed. Robustness is measured by the "false non-match rate" (also known as "Type I error"), the probability that a submitted sample will not match the enrollment image. Uniqueness is measured by the "false match rate" (also known as "Type II error") – the probability that a submitted sample will match the enrollment image of another user. Universality is measured by the "failure to enroll" rate, the probability that a user will not be able to supply a readable measure to the system upon enrollment. Accessibility can be quantified by the "throughput rate" of the system, the number of individuals that can be processed in a unit time, such as a minute or an hour. Acceptability is measured by polling the device users. The first four qualities are inversely related to their above measures, a higher "false non-match rate", for instance, indicating a lower level of robustness.

Having identified the required qualities and measures for each quality, it would seem a straightforward problem to simply run some experiments, determine the measures, and set a weighting value for the importance of each, thereby determining the "best" biometric characteristic. Unfortunately, for all biometric characteristics, all of the desired qualities have been found to be highly dependent on the specifics of the application, the population (both their physiological and psychological states), and the hardware/software system used. We cannot predict performance metrics for one application from tests on another. Further, the five metrics, which are correlated in a highly complex way, can be manipulated to some extent by administration policy.

System administrators might ultimately be concerned with: (1) the "false rejection rate", which is the probability that a true user identity claim will be falsely rejected, thus causing inconvenience; (2) the "false acceptance rate", which is the probability that a false identity claim will be accepted, thus allowing fraud; (3) the system throughput rate, measuring the number of users that can be processed in a time period; (4) the user acceptance of the system, which may be highly dependent upon the way the system is "packaged" and marketed; and (5) the ultimate total cost savings realized from implementing the system. These latter, more practical, measures depend upon the basic system qualities in highly complex and competitive ways that are not at all well understood, and can be controlled only to a limited extent through administrative decisions. Predicting the "false acceptance" and "false rejection" rates, and system throughput, user acceptance and cost savings for operational systems from test data, is a surprisingly difficult task.

For the users, the questions are simple: "Is this system easier, faster, friendlier and more convenient than the alternatives?" These issues, too, are highly application-, technology- and marketing-specific.

Consequently, it is impossible to state that a single biometric characteristic is "best" for all applications, populations, technologies and administration policies. Yet some biometric characteristics are clearly more appropriate than others for any particular application. System administrators wishing to employ biometric authentication need to articulate clearly the specifics of their application. In the following sections, we look more carefully at the distinctions between applications.

1.5 The Biometric Applications

The operational goals of biometric applications are just as variable as the technologies: some systems search for known individuals; some search for unknown individuals; some verify a claimed identity; some verify an unclaimed identity; and some verify that the individual has no identity in the system at all. Some systems search one or multiple submitted samples against a large database of millions of previously stored "templates" – the biometric data given at the time of enrollment. Some systems search one or multiple samples against a database of a few "models" – mathematical representations of the signal generation process created at the time of enrollment. Some systems compare submitted samples against models of both the claimed identity and impostor identities. Some systems search one or multiple samples against only one "template" or "model".

And the application environments can vary greatly – outdoors or indoors, supervised or unsupervised, with people trained or not trained in the use of the acquisition device.

To make sense out of all of the technologies, application goals and environments, we need a systematic method of approach – classification of uses and applications.

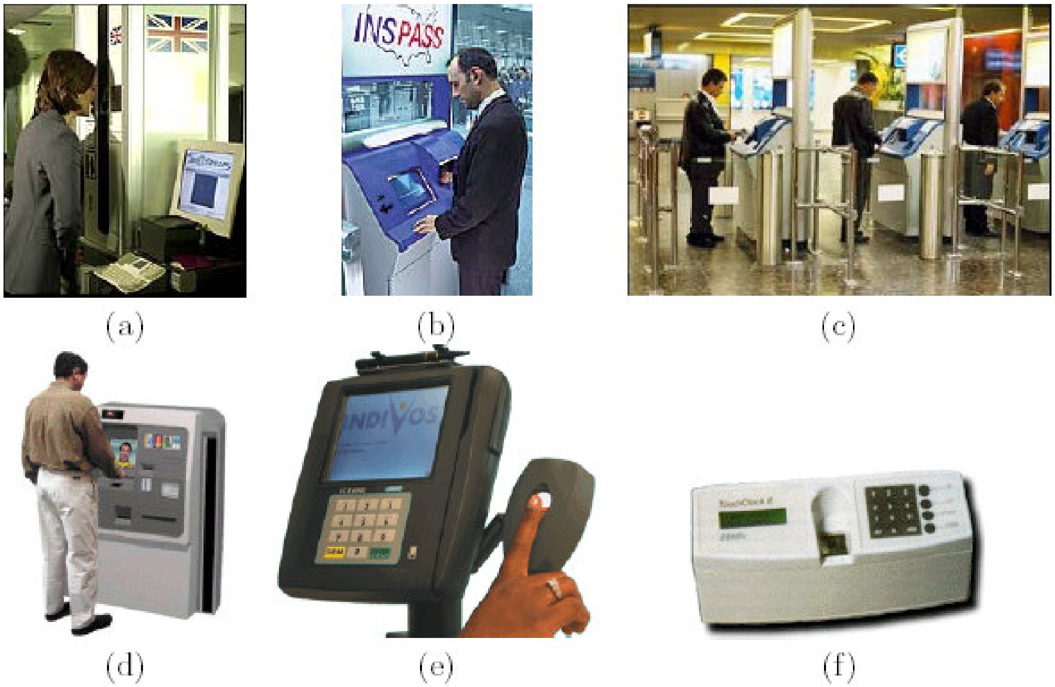


Figure 1.1: Biometric systems in civilian applications.

(a) A border passage system using iris recognition at London's Heathrow airport (news.bbc.co.uk).

(b) The INS Passenger Accelerated Service System (INSPASS) at JFK international airport (New York) uses hand geometry to authenticate travelers and significantly reduce their immigration inspection processing time (www.panynj.gov).

(c) Ben Gurion airport in Tel Aviv (Israel) uses Express Card entry kiosks fitted with hand geometry systems for security and immigration (www.airportnet.org).

(d) The FacePass system from Viisage is used in point-of-sale verification applications like ATMs, therefore, obviating the need for PINs (www.viisage.com).

(e) Indivos' “Pay by Touch" service uses fingerprints to help customers’ speed up payments in restaurants and cafeterias. When an enrolled customer places her finger on the sensor, the system retrieves her financial account and updates it (www.kioskbusiness.com).

(f) The Identix TouchClock fingerprint system is used in time and attendance applications (www.cardsolutions.com).

1.6 Verification and Identification

The most fundamental distinction in biometrics is between verification and identification. Nearly all aspects of biometrics – performance, benefits and risks of development, privacy impact and cost – differ when moving between these two types of systems.

Verification systems answer the question, "Am I who I claim to be?" by requiring that a user claim an identity in order for a biometric comparison to be performed. After a user claims an identity, he or she provides biometric data, which is then compared against his or her enrolled biometric data. Depending on the type of biometric system, the identity that a user claims might be a Windows username, a given name, or an ID number; the answer returned by the system is match or no match. Verification systems can contain dozens, thousands, or millions of biometric records, but are always predicated on a user's biometric data being matched against only his or her own enrolled biometric data. Verification is often referred to as 1:1 (one-to-one). The process of providing a username and biometric data is referred to as authentication.

Identification systems answer the question, "Who am I?" and do not require that a user claim an identity before biometric comparisons take place. The user provides his or her biometric data, which is compared to data from a number of users in order to find a match. The answer returned by the system is an identity such as a name or ID number. Identification systems can contain dozens, thousands or millions of biometric records. Identification is often referred to as 1:N (one-to-N or one-to-many), because a person’s biometric information is compared against multiple (N) records.

1.7 Logical Versus Physical Access

Once a biometric system has determined or verified an identity, what happens? The answer depends on the purpose for which the system is deployed. Biometric systems, and in many ways the entire biometric industry, can be segmented according to the purposes for which verification and identification are being performed. The two primary users for a biometric system are physical access and logical access.

Physical access systems monitor, restrict, or grant movement of a person or object into or out of a specific area. Most physical access implementations involve entry into a room or building: bank vaults, server rooms' control towers, or any location to which access is restricted. Time and attendance are a common physical access application, combining access to a location with an audit of when the authentication occurred. Physical access can also entail accessing equipment or material, such as opening a safe or starting an automobile, although most of the applications are still speculation. When used in physical access systems biometrics replace or complement keys, access cards, PIN cords, and security guards.

Logical access systems monitor, restrict, or grant access to data or information. Logging into a PC, accessing data stored on a network, accessing an account, or authenticating a transaction are examples of logical access. Biometrics replaces or complements password, PINs, and tokens in logical access systems. The core biometric functionally- acquiring and comparing biometric data- is often identical in physical and logical access systems. The same finger-scan algorithm and reader, for example, can be used for both desktop and doorway applications. What changes between the two is the external system into which the biometric functionality is integrated into a larger system be it a door control system, for example, or an operating system. The biometric match affects a result such as at the opening of a door or access to an operating system.

Because of the value of information stored on corporate networks and the transaction value of business-to business (B2B) and business-to consumer (B2C) e-commerce. The number of times an individual needs to provide authentication to a PC in a given day might be 20 or 30, while the instances of physical access authentication are less frequent and generally entail less value. The value of information and other intangible assets continually the potential value of biometric authentication as a logical access solution. However, biometric have proven very valuable in both types of applications.

Not every system fits neatly into the physical/ logical classification. Some identification systems, especially large-scale systems, are difficult to classify because the result of a match may be to investigate further- there is no resultant access to data or a physical object, but does so by allowing a user logical access to his or her data. Even allowing for difficult-to-classify applications, the differences between logical and physical access systems are generally very pronounced: the distinction between the two is a valuable tool in understanding biometrics. Key criteria such as accuracy, response time, fallback procedures, privacy requirements, cost, and complexity of integration vary substantially when moving from logical to physical access.

1.8 A Classification of Uses

A biometric system can be designed to test one of only two possible hypotheses: (1) that the submitted samples are from an individual known to the system; or (2) that the submitted samples are from an individual not known to the system. Applications to test the first hypothesis are called “positive identification” systems (verifying a positive claim of enrollment), while applications testing the latter are “negative identification” systems (verifying a claim of no enrollment). All biometric systems are of one type or the other. This is the most important distinction between systems, and controls potential architectures, vulnerabilities and system error rates.

Positive and negative identification are duals of each other. Positive identification systems generally serve to prevent multiple users of a single identity, while negative identification systems serve to prevent multiple identities of a single user. In positive identification systems, enrolled template or model storage can be centralized or decentralized in manner, including placement on optically read, magnetic stripe or smart cards. Negative identification systems demand centralized storage. Positive identification systems reject a user’s claim to identity if no match between submitted samples and enrolled templates is found. Negative identification systems reject a user’s claim to no identity if a match is found. Regardless of type of system, false rejections are a nuisance to users and false acceptances allow fraud.

An example of a positive identification system is the use of biometrics for employee access control at San Francisco International Airport.

Hand geometry has been used since the early 1990s to control access by employees to secured airport areas. There are currently 180 readers used by about 18,000 enrolled users. Employees activate the system by swiping a magnetic stripe identity card through a reader. The purpose of the system is to limit use of the identification card to the enrolled owner, thereby prohibiting use of the card by multiple users. Although the 9-byte template could be stored on the magnetic stripe, in this case it is stored centrally to allow updating upon successful use. The stored hand shape template indexed to the card is transmitted from the central server to the access control device. The user then places the right hand in the hand geometry reader, making the implicit claim, "I am the user who is enrolled to use this card". If the submitted hand sample is found to be "close enough" to the stored template, the user's claim is accepted.

Santa Clara County, located in California near the San Francisco International Airport requires the fingerprints of both left and right index fingers from all applicants for social service benefits. Citizens are only eligible for benefits under a single identity and must attest upon enrollment that they are not already enrolled in the system. Consequently, this biometric system is for "negative identification". When an applicant applies for benefits, he or she places the index fingers on an electronic scanner with the implicit claim, "I am not known to this system". The submitted fingerprints are searched against the entire centralized database of enrolled persons – although to facilitate the search, the prints in the database might be partitioned by gender. If no match is found, the claim of non-identity in the system is accepted.

Use of biometrics in positive identification systems can be voluntary because alternative methods for verifying a claimed identity exist. Those electing not to use biometrics can have their identity verified in other ways, such as by presentation of a passport or driver's license. Use of biometrics in negative identification systems must be mandatory for all users because no alternative methods exist for verifying a claim of no known identity.

Those wishing to avoid a positive identification system need to create a false match by impersonating an enrolled user. The possibility of biometric mimicry and forgery has been recognized since the 1970s. Those wishing to avoid a negative identification system need to submit altered samples not matching a previous enrollment. Table 1.1 summarizes these differences.

Historically, a distinction has been made between systems that verify a claimed identity and those that identify users without a claim of identity, perhaps returning a result that no identity was found. Some systems compare a single input sample to a single stored template or model to produce a verification or compare a single input sample to many stored templates to produce an identification. Identification systems are said to compare

Positive	**Negative**
To prove I am someone known to the system	To prove I am not someone known to the System
To prevent multiple users of a single Identity	To prevent multiple identities of a single user
Comparison of submitted sample to single claimed template – "one-to-one" under the most common system design	Comparison of submitted sample to all enrolled templates – "One-to-many"
A "false match" leads to "false acceptance"	A "false match" or a "failure to acquire" leads to a "false rejection"
A "false non-match" or a "failure to acquire" leads to a "false rejection"	A "false non-match" leads to a "false acceptance"
Alternative identification methods exist	No alternative methods exist
Can be voluntary	Must be mandatory for all
Spoofed by submitting someone else's biometric measures	Spoofed by submitting no or altered Measures

Table 1.1: Identification: positive and negative.

Samples from one person to templates from many persons, with verification being the degenerate case of "many" equal to one. In the mid-1990s, several companies began to promote "PIN-less verification" systems, in which verification was accomplished without a claim to identity. The "verification/identification" dichotomy has been further clouded by the development of surveillance and modern “few-to-many” access control systems, which cannot be consistently classified as either “verification” or “identification”. The uses and search strategies of biometric systems have expanded to the point where these distinctions of “verification/identification” and “one-to-one/one-to-many” are no longer fully informative.

Ultimately, a biometric system can only link a submitted sample to an enrolled template or model: that record created upon first use of the system by a person. That enrollment template/model need not be connected with any identifying information, such as a name or registration number. In fact, biometric measures and the enrollment templates/models derived from them contain no information about name, age, nationality, race or gender. Consequently, use of a biometric system without linkages of stored data to common identifiers allows for anonymous authentication. If system administrators have a need to connect the stored biometric data to other information, such as a name, that must be done by the presentation and human certification of trusted identifying credentials at the time of enrollment. Subsequent identification by the biometric system is no more reliable than this source documentation. But once that link has been made, subsequent identifications can be made without reference to the original source documents.

1.9 A Classification of Application Environments

In the early 1990s, as we gained experience with the use of biometric devices, it became apparent that variations in the application environment had a significant impact on the way the devices performed. In fact, accurate characterization of the operational environment is primary in selecting the best biometric technology and in predicting the system's operational characteristics. In this section, we will present a method for analyzing a proposed operational environment by differentiating applications based on partitioning into six categories beyond the "positive" and "negative" applications already discussed.

1.9.1 Overt Versus Covert

The first partition is "overt/covert". If the user is aware that a biometric identifier is being measured, the use is overt. If unaware, the use is covert. Almost all conceivable access control and non-forensic applications are overt. Forensic applications can be covert.

1.9.2 Habituated Versus Non-Habituated

The second partition, "habituated/non-habituated", applies to the intended users of the application. Users presenting a biometric trait on a daily basis can be considered habituated after a short period of time. Users who have not presented the trait recently can be considered "non-habituated". A more precise definition will be possible after we have better information relating system performance to frequency of use for a wide population over a wide field of devices. If all the intended users are "habituated", the application is considered a "habituated" application. If all the intended users are “non-habituated”, the application is considered “non-habituated”. In general, all applications will be “non-habituated” during the first week of operation, and can have a mixture of habituated and non-habituated users at any time there after. Access control to a secure work area is generally “habituated”. Access control to a sporting event is generally “non-habituated”.

1.9.3 Attended Versus Non-Attended

A third partition is “attended/unattended”, and refers to whether the use of the biometric device during operation will be observed and guided by system management. Non-cooperative applications will generally require supervised operation, while cooperative operation may or may not. Nearly all systems supervise the enrollment process, although some do not.

1.9.4 Standard Versus Non-Standard Environment

A fourth partition is “standard/non-standard operating environment”. If the application will take place indoors at standard temperature (20 °C), pressure (1 atm), and other environmental conditions, particularly where lighting conditions can be controlled, it is considered a “standard environment” application. Outdoor systems, and perhaps some unusual indoor systems, are considered “non-standard environment” applications.

1.9.5 Public Versus Private

A fifth partition is “public/private”. Will the users of the system be customers of the system management (public) or employees (private)? Clearly, attitudes toward usage of the devices, which will directly affect performance, vary depending upon the relationship between the end-users and system management.

1.9.6 Open Versus Closed

A sixth partition is "open/closed". Will the system be required, now or in the future, to exchange data with other biometric systems run by other management? For instance, some US state social services agencies want to be able to exchange biometric information with other states. If a system is to be open, data collection, compression and format standards are required. A closed system can operate perfectly well on completely proprietary formats.

This list is open, meaning that additional partitions might also be appropriate. We could also argue that not all possible partition permutations are equally likely or even permissible.

1.9.7 Examples of the Classification of Applications

Every application can be classified according to the above partitions. For instance, the positive biometric identification of users of the Immigration and Naturalization Service's Passenger Accelerated Service System (INSPASS) currently in place at Kennedy, Newark, Los Angeles, Miami, Detroit, Washington Dulles, Vancouver and Toronto airports for rapidly admitting frequent travelers into the USA, can be classified as a cooperative, overt, non-attended, non-habituated, standard environment, public, closed application. The system is cooperative because those wishing to defeat the system will attempt to be identified as someone already holding a pass. It will be overt because all will be aware that they are required to give a biometric measure as a condition of enrollment into this system. It will be non-attended and in a standard environment because collection of the biometric will occur near the passport inspection counter inside the airports, but not under the direct observation of an INS employee. It will be non-habituated because most international travelers use the system less than once per month. The system is public because enrollment is open to any frequent traveler into the USA. It is closed because INSPASS does not exchange biometric information with any other system.

The negative identification systems for preventing multiple identities of social service recipients can be classified as non-cooperative, overt, attended, non-habituated, open, standard environment systems.

Clearly, the latter application is more difficult than the former. Therefore we cannot directly compare hand geometry and facial recognition technologies based on the error rates across these very different applications.

1.10 A System Model

Although these devices rely on widely different technologies, much can be said about them in general. Figure 1.1 shows a generic biometric authentication system divided into five subsystems: data collection, transmission, signal processing, decision and data storage. We will consider these subsystems one at a time.

1.10.1 Data Collection

Biometric systems begin with the measurement of a behavioral/physiological characteristic. Key to all systems is the underlying assumption that the measured biometric characteristic is both distinctive between individuals and repeatable over time for the same individual. The problems in measuring and controlling these variations begin in the data collection subsystem.

The user's characteristic must be presented to a sensor. The presentation of any biometric characteristic to the sensor introduces a behavioral (and, consequently, psychological) component to every biometric method. This behavioral component may vary widely between users, between applications, and between the test laboratory and the operational environment.

The output of the sensor, which is the input data upon which the system is built, is the convolution of: (1) the biometric measure; (2) the way the measure is presented; and (3) the technical characteristics of the sensor. Both the repeatability and the distinctiveness of the measurement are negatively impacted by changes in any of these factors. If a system is to be open, the presentation and sensor characteristics must be standardized to ensure that biometric characteristics collected with one system will match those collected on the same individual by another system. If a system is to be used in an overt, non-cooperative application, the user must not be able to willfully change the biometric or its presentation sufficiently to avoid being matched to previous records.

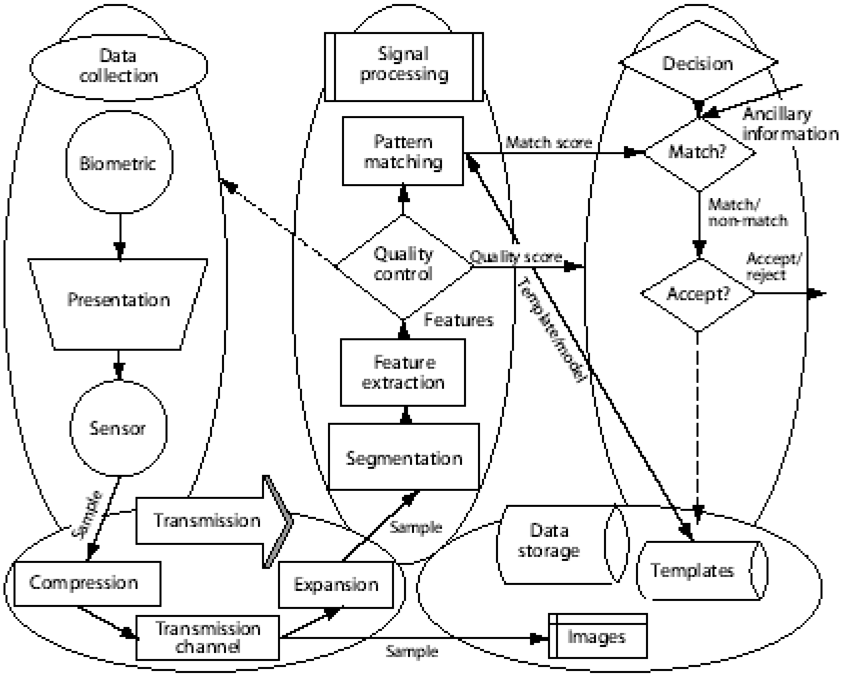


Figure 1.2: A generic biometric system.

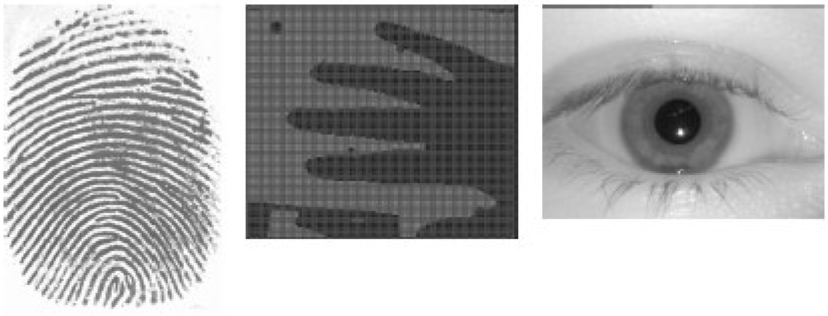


Figure 1.3: Fingerprint, hand and iris system input images.

Figure 1.3 shows input images from fingerprint, hand geometry and iris recognition systems.

1.10.2 Transmission

Some, but not all, biometric systems collect data at one location but store and/or process it at another. Such systems require data transmission. If a great amount of data is involved, compression may be required before transmission or storage to conserve bandwidth and storage space. Figure 1.1 shows compression and transmission occurring before the signal processing and image storage. In such cases, the transmitted or stored compressed data must be expanded before further use. The process of compression and expansion generally causes quality loss in the restored signal, with loss increasing with increasing compression ratio. The compression technique used will depend upon the biometric signal. An interesting area of research is in finding, for a given biometric technique, compression methods with minimum impact on the signal-processing subsystem.

If a system is to be open, compression and transmission protocols must be standardized so that every user of the data can reconstruct the original signal. Standards currently exist for the compression of fingerprints (Wavelet Scalar Quantization), facial images (JPEG), and voice data (Code Excited Linear Prediction).

1.10.3 Signal Processing

Having acquired and possibly transmitted a biometric characteristic, we must prepare it for matching with other like measures. Figure 1.1 divides the signal-processing subsystem into four tasks: segmentation, feature extraction, quality control, and pattern matching.

Segmentation is the process of finding the biometric pattern within the transmitted signal. For example, a facial recognition system must first find the boundaries of the face or faces in the transmitted image. A speaker verification system must find the speech activity within a signal that may contain periods of non-speech sounds. Once the raw biometric pattern of interest has been found and extracted from larger signal, the pattern is sent to the feature extraction process.

Feature extraction is fascinating. The raw biometric pattern, even after segmentation from the larger signal, contains non-repeatable distortions caused by the presentation, sensor and transmission processes of the system. These non-controllable distortions and any non-distinctive or redundant elements must be removed from the biometric pattern, while at the same time preserving those qualities that are both distinctive and repeatable. These qualities expressed in mathematical form are called "features". In a text-independent speaker recognition system, for instance, we may want to find the features, such as the mathematical frequency relationships in the vowels, that depend only upon the speaker and not upon the words being spoken, the health status of the speaker, or the speed, volume and pitch of the speech. There are as many wonderfully creative mathematical approaches to feature extraction as there are scientists and engineers in the biometrics industry. You can understand why such algorithms are always considered proprietary. Consequently, in an open system, the “open” stops here.

In general, feature extraction is a form of non-reversible compression, meaning that the original biometric image cannot be reconstructed from the extracted features. In some systems, transmission occurs after feature extraction to reduce the requirement for bandwidth.

After feature extraction, or maybe even before, we will want to check to see if the signal received from the data collection subsystem is of good quality. If the features “don’t make sense” or are insufficient in someway, we can conclude quickly that the received signal was defective and request a new sample from the data collection subsystem while the user is still at the sensor. The development of this “quality control” process has greatly improved the performance of biometric systems in the last few short years. On the other hand, some people seem never to be able to present an acceptable signal to the system. If a negative decision by the quality control module cannot be overridden, a “failure to enroll” error results.

The feature “sample”, now of very small size compared to the original signal, will be sent to the pattern matching process for comparison with one or more previously identified and stored feature templates or models. We use the term “template” to indicate stored features. The features in the template are of the same type as those of a sample. For instance, if the sample features are a “vector” in the mathematical sense, then the stored template will also be a “vector”. The term “model” is used to indicate the construction of a more complex mathematical representation capable of generating features characteristic of a particular user. Models and features will be of different mathematical types and structures. Models are used in some speaker and facial recognition systems. Templates are used in fingerprint, iris, and hand geometry recognition systems.

The term “enrollment” refers to the placing of a template or model into the database for the very first time. Once in the database and associated with an identity by external information (provided by the enrollee or others), the enrollment biometric data is referred to as the template or model for the individual to which it refers.

The purpose of the pattern matching process is to compare a presented feature sample to the stored data, and to send to the decision subsystem a quantitative measure of the comparison. An exception is enrollment in systems allowing multiple enrollments. In this application, the pattern matching process can be skipped. In the cooperative case where the user has claimed an identity or where there is but a single record in the current database (which might be a magnetic stripe card), the pattern matching process might only make a comparison against a single stored template. In all other cases, such as large-scale identification, the pattern matching process compares the present sample to multiple templates or models from the database one at a time, as instructed by the decision subsystem, sending on a quantitative “distance” measure for each comparison. In place of a distance measure, some systems use similarity measures, such as maximum likelihood values.

The signal processing subsystem is designed with the goal of yielding small distances between enrolled models/templates and later samples from the same individual and large distances between enrolled models/templates and samples of different individuals. Even for models and samples from the same individual, however, distances will rarely, if ever, be zero, as there will always be some non-repeatable biometric-, presentation-, sensor- or transmission-related variation remaining after processing.

1.10.4 Storage

The remaining subsystem to be considered is that of storage. There will be one or more forms of storage used, depending upon the biometric system. Templates or models from enrolled users will be stored in a database for comparison by the pattern matcher to incoming feature samples. For systems only performing "one-to-one" matching, the database may be distributed on smart cards, optically read cards or magnetic stripe cards carried by each enrolled user. Depending upon system policy, no central database need exist, although in this application a centralized database can be used to detect counterfeit cards or to reissue lost cards without re-collecting the biometric pattern.

The database will be centralized if the system performs one-to-*N* matching with *N* greater than one, as in the case of identification or "PIN less verification" systems. As *N* gets very large, system speed requirements dictate that the database be partitioned into smaller subsets such that any feature sample need only be matched to the templates or models stored in one partition, or indexed by using an appropriate data structure which allows the templates to be visited in an advantageous order during the retrieval. These strategies have the effect of increasing system speed and decreasing false matches, at the expense of increasing the false non match rate owing to partitioning errors. This means that system error rates do not remain constant with increasing database size and identification systems do not scale linearly. Consequently, database partitioning/indexing strategies represent a complex policy decision.

If it may be necessary to reconstruct the biometric patterns from stored data, raw (although possibly compressed) data storage will be required. The biometric pattern is generally not reconstructable from the stored templates or models, although some methods do allow a coarse reconstruction of patterns from templates. Further, the templates themselves are created using the proprietary feature extraction algorithms of the system vendor. The storage of raw data allows changes in the system or system vendor to be made without the need to re-collect data from all enrolled users.

1.10.5 Decision

The decision subsystem implements system policy by directing the database search, determines “matches” or “non-matches” based on the distance or similarity measures received from the pattern matcher, and ultimately makes an “accept/reject” decision based on the system policy. Such a decision policy could be to reject the identity claim (either positive or negative) of any user whose pattern could not be acquired. For an acquired pattern, the policy might declare a match for any distance lower than a fixed threshold and "accept" a user identity claim on the basis of this single match, or the policy could be to declare a match for any distance lower than a user-dependent, time-variant, or environmentally linked threshold and require matches from multiple measures for an "accept" decision. The policy could be to give all users, good guys and bad guys alike, three tries to return a low distance measure and be "accepted" as matching a claimed template. Or, in the absence of a claimed template, the system policy could be to direct the search of all, or only a portion, of the database and return a single match or multiple "candidate" matches. The decision policy employed is a management decision that is specific to the operational and security requirements of the system. In general, lowering the number of false non-matches can be traded against raising the number of false matches. The optimal system policy in this regard depends both upon the statistical characteristics of the comparison distances coming from the pattern matcher, the relative penalties for false match and false non-match within the system, and the a priori (guessed in advance) probabilities that a user is, in fact, an impostor. In any case, in the testing of biometric devices, it is necessary to decouple the performance of the signal processing subsystem from the policies implemented by the decision subsystem.

1.11 Biometrics and Privacy

Whenever biometric identification is discussed, people always want to know about the implications for personal privacy. If a biometric system is used, will the government, or some other group, be able to get personal information about the users? Biometric measures themselves contain no personal information. Hand shape, fingerprints or eye scans do not reveal name, age, race, gender, and health or immigration status. Although voice patterns can give a good estimation of gender, no other biometric identification technology currently used reveals anything about the person being measured. More common identification methods, such as a driver's license, reveal name, address, age, gender, vision impairment, height and even weight! Driver's licenses, however, may be easier to steal or counterfeit than biometric measures.

Biometric measures can be used in place of a name, Social Security number or other form of identification to secure anonymous transactions. Walt Disney World sells season passes to buyers anonymously, and then uses finger geometry to verify that the passes are not being transferred. Use of iris or fingerprint recognition for anonymous health care screening has also been proposed. A patient would use an anonymous biometric measure, not a name or Social Security number, when registering at a clinic. All records held at the clinic for that patient would be identified, linked and retrieved only by the measure. No one at the clinic, not even the doctors, would know the patient's "real" (publicly recognized) identity.

The real fear is that biometric measures will link people to personal data, or allow movements to be tracked. After all, credit card and phone records can be used in court to establish a person's activities and movements. There are several important points to be made on this issue.

Phone books are public databases linking people to their phone number. These databases are even accessible on the Internet. Because phone numbers are unique to phone lines, "reverse" phone books also exist, allowing a name to be determined from a phone number. Even if a number is unlisted, all information on calls made from that number may be available to law enforcement agencies through the subpoena process. There are no public databases, however, containing biometric identifiers, and there are only a few limited-access government databases. Five US states have electronic fingerprint records of social service recipients (Arizona, California, Connecticut, New York and Texas); six states (California, Colorado, Georgia, Hawaii, Oklahoma and Texas) maintain electronic fingerprints of all licensed drivers; nearly all states maintain copies of driver's license and social service recipient photos; the FBI and state governments maintain fingerprint databases on convicted felons and sex offenders; and the federal government maintains hand geometry records on those who have voluntarily requested border crossing cards. General access to this data is limited to the agencies that collected it, but like credit card and phone "toll records", this information can be released or searched by law enforcement groups acting under court order.

Unlike phone books, however, databases of biometric measures cannot generally be reversed to reveal names from measures because biometric measures, although distinctive, are not unique. Fingerprint, retinal and iris databases may be exceptions, allowing reversal if the biometric data was carefully collected. But general biometric measures do not serve as useful pointers to other types of data. Unique identifiers such as Social Security and credit card numbers always do the linking of records. Biometric measures are not generally useful in this regard, even if databases linking information to measures were to exist. For these reasons, biometric measures are not useful for tracking the movements of people, as is already possible using telephone and credit card numbers.

Databases of biometric images, and the numerical models or templates derived from them, are often encrypted with the intention of inhibiting their compromise in bulk. But compromise of individual measures cannot always be prevented by protecting databases and transmission channels because biometric measures, although privately owned, are sometimes publicly observable (e.g. a photo of a person's face can be taken with a camera or downloaded from a web page). In general, biometric measures are not secret, even if it might be quite complicated to acquire usable copies (e.g. a retinal map) without the cooperation of the owner. When used for security, biometric characteristics are more like public keys than private keys. Unlike public keys, however, biometric measures cannot be revoked if stolen or mimicked. The industry is currently working on methods for "live-ness testing" and revocation, hoping to improve these problems. Table 1.2 summarizes the privacy issues raised by the use of biometrics.

1.	Unlike more common forms of identification, biometric measures contain no personal information and are more difficult to forge or steal.
2.	Biometric measures can be used in place of a name or Social Security number to secure anonymous transactions.
3.	Some biometric measures (face images, voice signals and "latent" fingerprints left on surfaces) can be taken without a person's knowledge, but cannot be linked to an identity without a pre-existing invertible database.
4.	A Social Security or credit card number, and sometimes even a legal name, can identify a person in a large population. This capability has not been demonstrated using any single biometric measure.
5.	Like telephone and credit card information, biometric databases can be searched outside of their intended purpose by court order.
6.	Unlike credit card, telephone or Social Security numbers, biometric characteristics change from one measurement to the next.
7.	Searching for personal data based on biometric measures is not as reliable or efficient as using better identifiers, like legal name or Social Security number.
8.	Biometric measures are not always secret, but are sometimes publicly observable and cannot be revoked if compromised.

Table 1.2: Biometrics and privacy.

1.12 Statement of Problem

Biometrics has been adopted in a variety of large-scale identification application - ranging from border control to voter ID issuance. While the technology is conceptually adept, in reality there are numerous challenges associated with enrolling large populations using just single (unimodal) biometrics. These challenges can be overcome by deploying multimodal biometrics systems.

Unimodal biometric systems have to contend with a variety of problems such as noisy data, intra-class variations, restricted degrees of freedom, non-universality, spoof attacks, and unacceptable error rates. Some of these limitations can be addressed by deploying multimodal biometric systems that integrate the evidence presented by multiple sources of information.

The shortcomings of unimodal biometrics can be compensated by multimodal biometric system.

- The usage of certain biometrics makes it susceptible to noisy or bad data, such as inability of a scanner to read dirty fingerprints clearly. This can lead to inaccurate matching, as bad data may lead to a false rejection.
- Unimodal biometrics is also prone to inter-class similarities within large population groups. In case of identical twins, a facial recognition camera may not be able to distinguish between the two.
- Some biometric technologies are incompatible with a certain subset of the population. Elderly people and young children

may have difficulty enrolling in a fingerprinting system, due to their faded prints or underdeveloped fingerprint ridges

- Finally, unimodal biometrics are vulnerable to spoofing, where the data can be imitated or forged.

1.13 Objectives of the Research Studies

The main aim of this work is to investigate the effectiveness of fusion techniques for multimodal biometrics, with the following objectives:

- A review of the existing approaches.
- Investigations into effective fusion methods for selected types of biometrics (fingerprint and face).
- Propose a multimodal biometric authentication system model, which improve the identification and verification of a person using fingerprint and face recognition.
- Compare the False Rejection Rate with False Acceptance Rate at constant threshold value.

1.14 Limitations of the Study

- The research has performed in the normal laboratory/office environment.
- The research has taken fingerprint and face recognition as biometric trait for multimodal biometric.

1.15 Thesis Organization

The thesis is organized into five chapters. An overview of these chapters is presented below.

- Chapter 1 introduces the topic of biometric authentication system and gives the objectives of this Ph.D. thesis.
- Chapter 2 describes fingerprint recognition approaches and system design. This chapter also explains fingerprint image preprocessing, minutia extraction, minutia post processing, and minutia match.
- Chapter 3 explains face recognition system with face recognition processing. This chapter also explain technical issues related to face recognition and explain different face recognition technologies.
- Chapter 4 describes multimodal biometric system. It further elaborate fingerprint and face recognition for multiomdal.
- Chapter 5 explains a proposed multimodal biometric system using fingerprint and face recognition. It describes the hardware and software used, show the test result the research has taken.
- Finally the thesis shows conclusion and future scope of multimodal biometric system.

References:

- Biometrics Identity Verification in a Network world, Wiley Publications
- Biometrics: Personal Identification in Networked Society, Kluwer Academic Press
- Biometric Systems: Technology, Design and Performance Evaluation, Springer
- www.aamva.org
- www.csr.unibo.it
- www.cesg.gov.uk
- www.dodcounterdrug.com
- www.dss.state.ct.us

CHAPTTER - 2

FINGER PRINT RECOGNITION

2.1 Introduction
2.2 System Design
2.3 Fingerprint Image Preprocessing
2.4 Minutia Extraction
2.5 Minutia Post-processing
2.6 Minutia Match
2.7 Fingerprint Experimentation Evaluation

2. 1 Introduction

2.1.1 What is a Fingerprint?

A fingerprint is the feature pattern of one finger (Figure 2.1). It is believed with strong evidences that each fingerprint is unique. Each person has his own fingerprints with the permanent uniqueness. So fingerprints have being used for identification and forensic investigation for a long time.

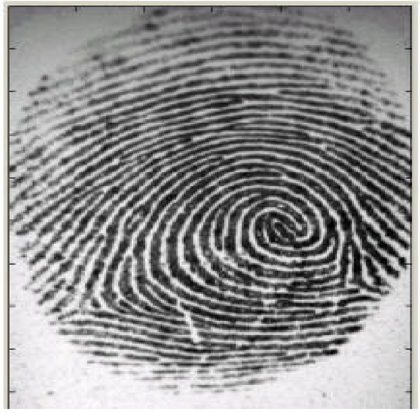


Figure 2.1: A fingerprint image acquired by an Optical Sensor.

A fingerprint is composed of many ridges and valleys. These ridges and valleys present good similarities in each small local window, like parallelism and average width.

However, shown by intensive research on fingerprint recognition, fingerprints are not distinguished by their ridges and valleys, but by Minutia, which are some abnormal points on the ridges (Figure 2.2). Among the variety of minutia types reported in literatures, two are mostly significant and in heavy usage: one is called termination, which is the immediate ending of a ridge; the other is called bifurcation, which is the point on the ridge from which two branches derive.

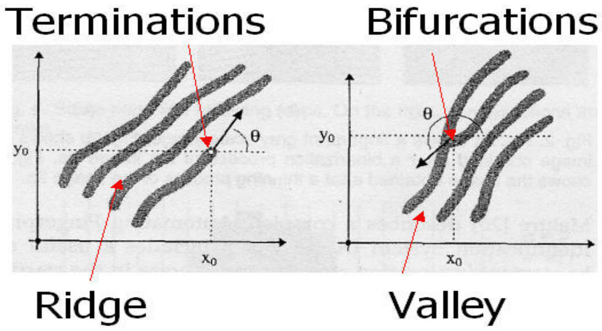


Figure 2.2: Minutia. (Valley is also referred as Furrow, Termination is also called Ending, and Bifurcation is also called Branch)

2.1.2 What is Fingerprint Recognition?

The fingerprint recognition problem can be grouped into two sub-domains: one is fingerprint verification and the other is fingerprint identification (Figure 2.3). In addition, different from the manual approach for fingerprint recognition by experts, the fingerprint recognition here is referred as AFRS (Automatic Fingerprint Recognition System), which is program-based.

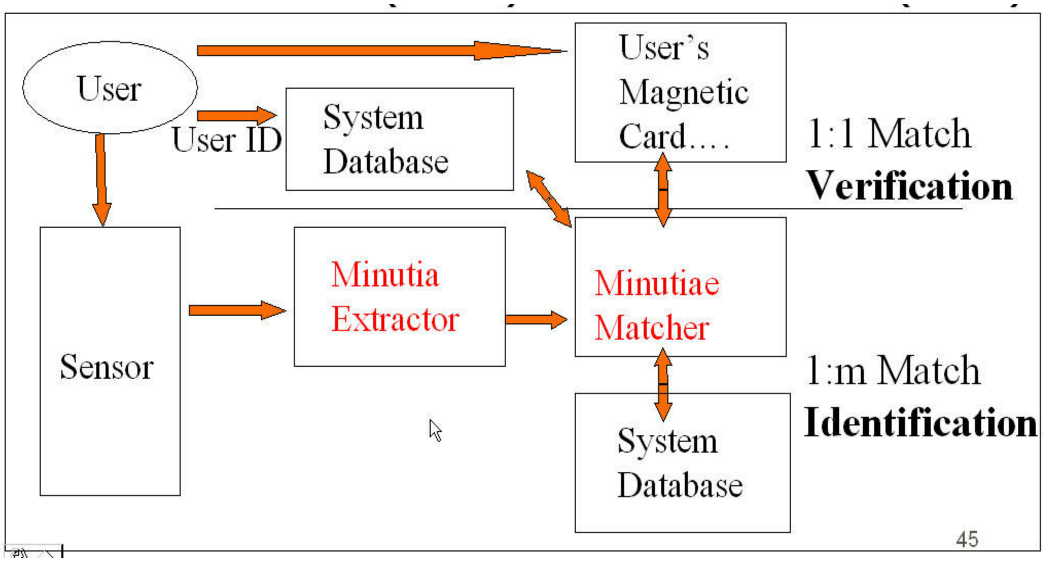


Figure 2.3: Verification vs. Identification

Fingerprint verification is to verify the authenticity of one person by his fingerprint. The user provides his fingerprint together with his identity information like his ID number. The fingerprint verification system retrieves the fingerprint template according to the ID number and matches the template with the real-time acquired fingerprint from the user. Usually it is the underlying design principle of AFAS (Automatic Fingerprint Authentication System).

Fingerprint identification is to specify one person’s identity by his fingerprint(s). Without knowledge of the person’s identity, the fingerprint identification system tries to match his fingerprint(s) with those in the whole fingerprint database. It is especially useful for criminal investigation cases. And it is the design principle of AFIS (Automatic Fingerprint Identification System).

However, all fingerprint recognition problems, either verification or identification, are ultimately based on a well-defined representation of a fingerprint. As long as the representation of fingerprints remains the uniqueness and keeps simple, the fingerprint matching, either for the 1-to-1 verification case or 1-to-m identification case, is straightforward and easy.

2.1.3 Two approaches for Fingerprint recognition

Two representation forms for fingerprints separate the two approaches for fingerprint recognition.

The first approach, which is minutia-based, represents the fingerprint by its local features, like terminations and bifurcations. This approach has been intensively studied, also is the backbone of the current available fingerprint recognition products.

The second approach, which uses image-based methods, tries to do matching based on the global features of a whole fingerprint image. It is an advanced and newly emerging method for fingerprint recognition. And it is useful to solve some intractable problems of the first approach.

2.2 System Design

2.2.1 System Level Design

A fingerprint recognition system constitutes of fingerprint acquiring device, minutia extractor and minutia matcher [Figure 2.4].

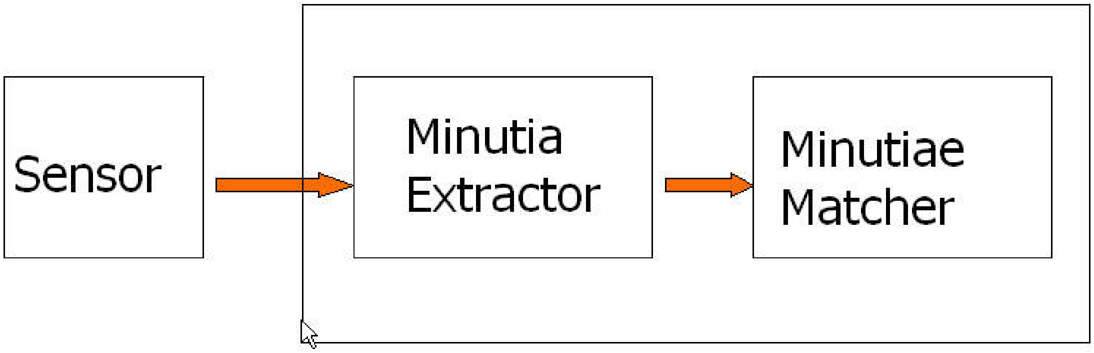


Figure 2.4: Simplified Fingerprint Recognition System

For fingerprint acquisition, optical or semi-conduct sensors are widely used. They have high efficiency and acceptable accuracy except for some cases that the user's finger is too dirty or dry. However, the testing database for this project is from the available fingerprints provided by FVC2002 (Fingerprint Verification Competition 2002). So no acquisition stage is implemented.

The minutia extractor and minutia matcher modules are explained in detail in the next part for algorithm design and other subsequent sections.

2.2.2 Algorithm Level Design

To implement a minutia extractor, a three-stage approach is widely used by researchers. They are preprocessing, minutia extraction and post-processing stage [Figure 2.5].

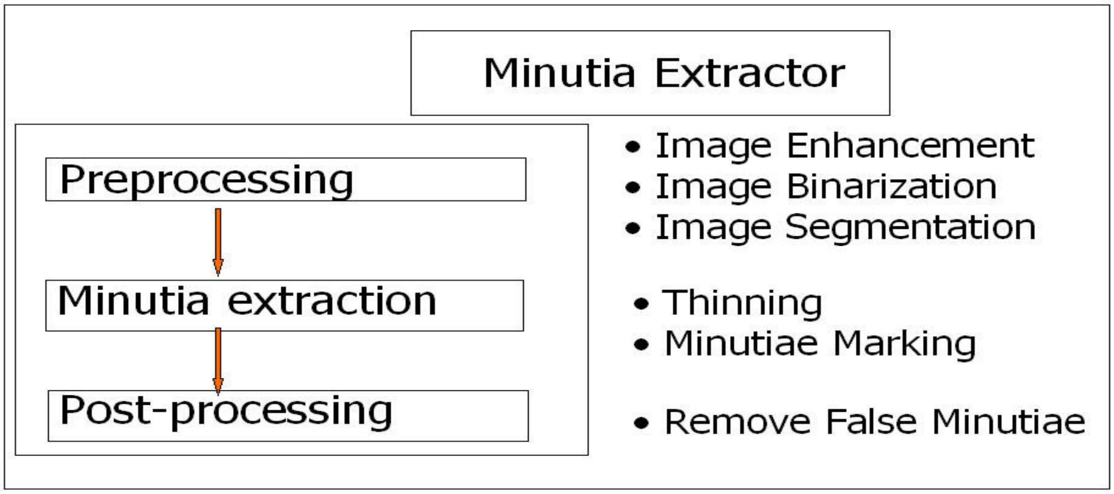
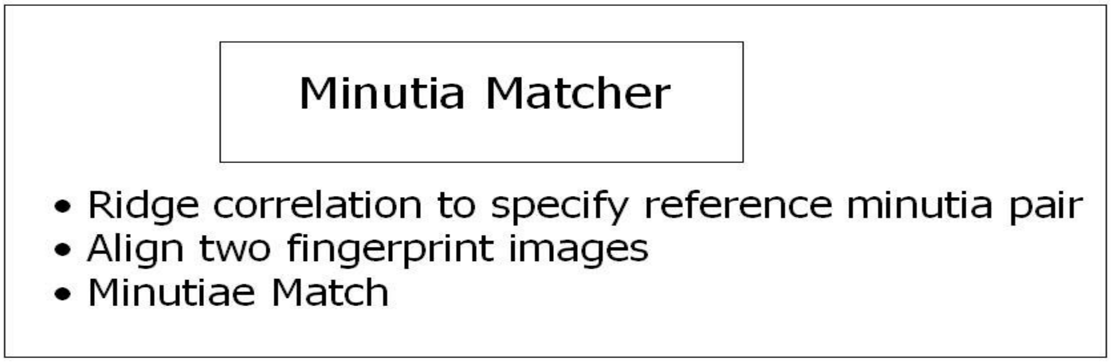


Figure 2.5: Minutia Extractor

For the fingerprint image preprocessing stage, I use Histogram Equalization and Fourier Transform to do image enhancement. And then the fingerprint image is binarized using the locally adaptive threshold method. The image segmentation task is fulfilled by a three-step approach: block direction estimation, segmentation by direction intensity and Region of Interest extraction by Morphological operations. Other researchers develop most methods used in the preprocessing stage but they form a brand new combination in this project through trial and error. Also the morphological operations for extraction ROI are introduced to fingerprint image segmentation.

For minutia extraction stage, three thinning algorithms are tested and the Morphological thinning operation is finally bid out with high efficiency and pretty good thinning quality. The minutia marking is a simple task as most literatures reported but one special case is found during this implementation and an additional check mechanism is enforced to avoid such kind of oversight.

For the post-processing stage, a more rigorous algorithm is developed to remove false minutia based on. Also a novel representation for bifurcations is proposed to unify terminations and bifurcations.



Minutia Matcher

- Ridge correlation to specify reference minutia pair
- Align two fingerprint images
- Minutiae Match

Figure 2.6: Minutia Matcher

The minutia matcher chooses any two minutias as a reference minutia pair and then match their associated ridges first. If the ridges match well, two fingerprint images are aligned and matching is conducted for all remaining minutia [Figure 2.6].

2. 3 Fingerprint Image Preprocessing

2.3.1 Fingerprint Image Enhancement

Fingerprint Image enhancement is to make the image clearer for easy further operations. Since the fingerprint images acquired from sensors or other medias are not assured with perfect quality, those enhancement methods, for increasing the contrast between ridges and furrows and for connecting the false broken points of ridges due to insufficient amount of ink, are very useful for keep a higher accuracy to fingerprint recognition.

Two Methods are adopted in this fingerprint recognition system: the first one is Histogram Equalization; the next one is Fourier Transform.

2.3.1.1 Histogram Equalization:

Histogram equalization is to expand the pixel value distribution of an image so as to increase the perceptional information. The original histogram of a fingerprint image has the bimodal type [Figure 2.7], the histogram after the histogram equalization occupies all the range from 0 to 255 and the visualization effect is enhanced [Figure 2.8].

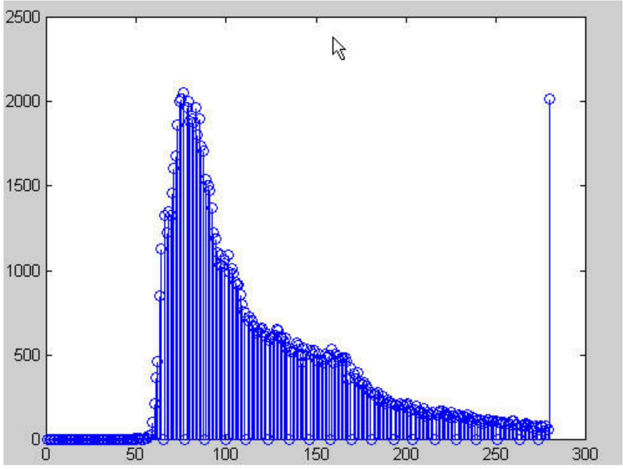


Figure 2.7: the Original histogram of a fingerprint image

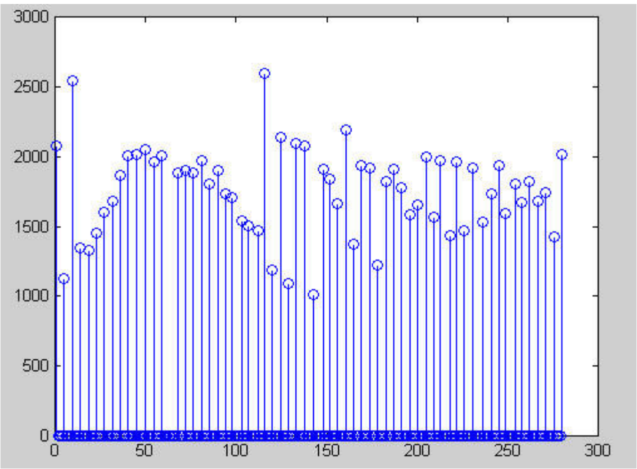


Figure 2.8 Histogram after the Histogram Equalization

The right side of the following figure [Figure 2.9] is the output after the histogram equalization.

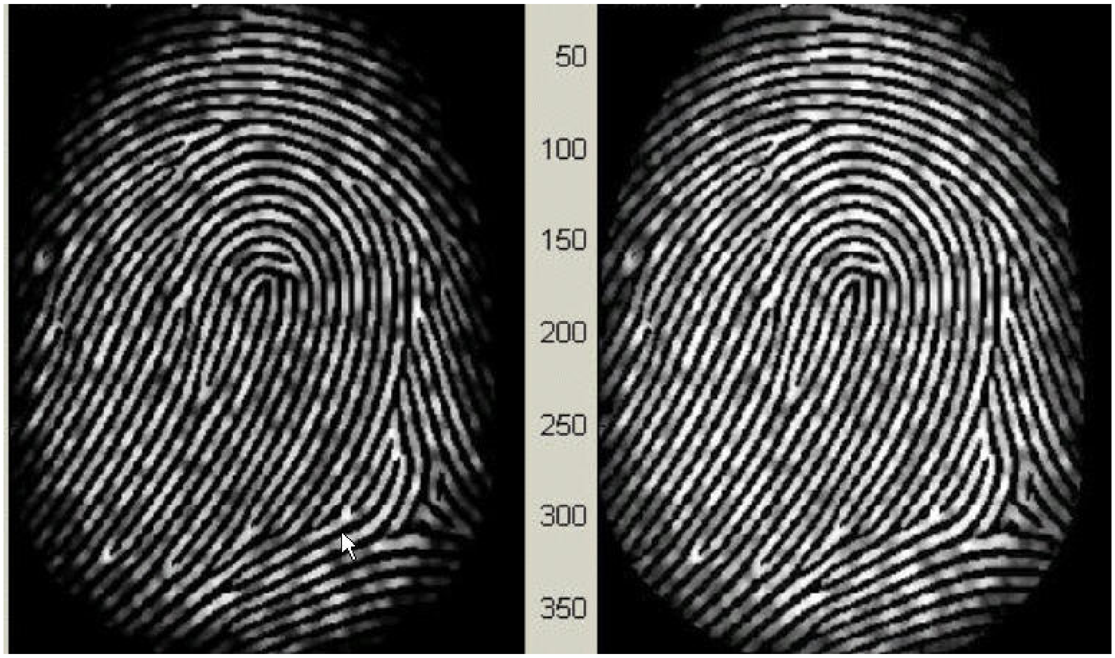


Figure 2.9: Histogram Enhancement.

Original Image (Left). Enhanced image (Right)

2.3.1.2 Fingerprint Enhancement by Fourier Transform

We divide the image into small processing blocks (32 by 32 pixels) and perform the Fourier transform according to:

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\times\exp\left\{-j2\pi\times\left(\frac{ux}{M}+\frac{vy}{N}\right)\right\} \qquad (1)$$

for u = 0, 1, 2, ..., 31 and v = 0, 1, 2, ..., 31.

In order to enhance a specific block by its dominant frequencies, we multiply the FFT of the block by its magnitude a set of times. Where the magnitude of the original FFT = abs(F(u,v)) = |F(u,v)|.

Get the enhanced block according to

$$g(x,y)=F^{-1}\left\{F(u,v)\times\left|F(u,v)\right|^{K}\right\} \qquad (2) ,$$

where F^{-1}(F(u,v)) is done by:

$$f(x,y)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}F(u,v)\times\exp\left\{j2\pi\times\left(\frac{ux}{M}+\frac{vy}{N}\right)\right\} \qquad (3)$$

for x = 0, 1, 2, ..., 31 and y = 0, 1, 2, ..., 31.

The k in formula (2) is an experimentally determined constant, which we choose k=0.45 to calculate. While having a higher "k" improves the appearance of the ridges, filling up small holes in ridges, having too high a "k" can result in false joining of ridges. Thus a termination might become a bifurcation. Figure 2.10 presents the image after FFT enhancement.

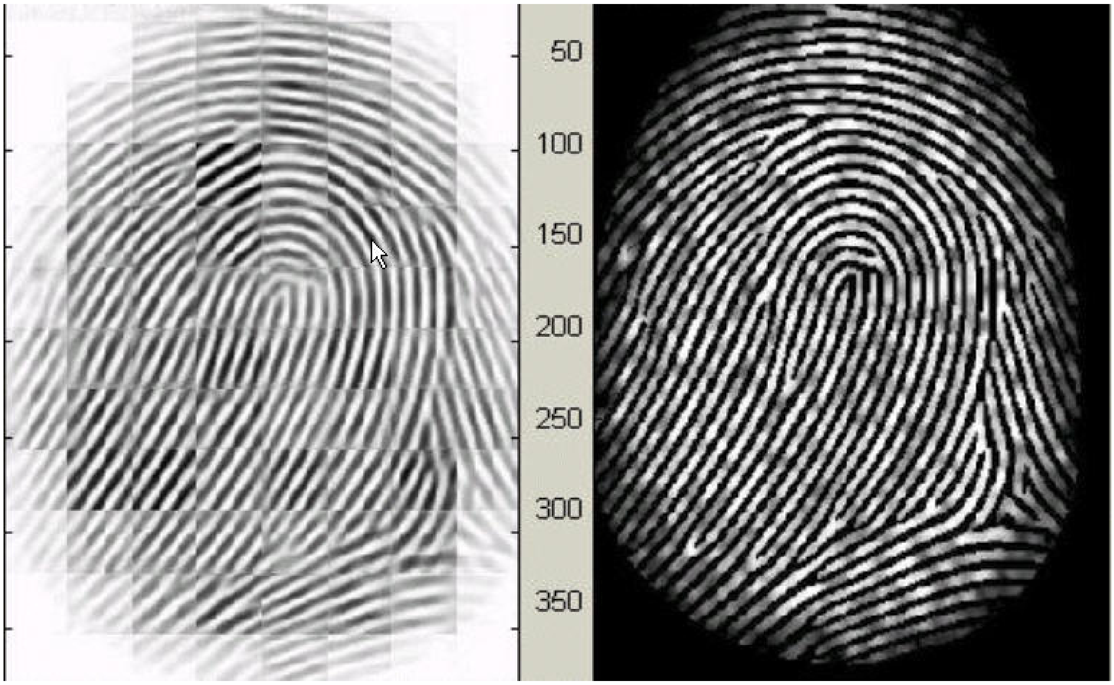


Figure 2.10: Histogram Enhancement.

Original Image (Left). Enhanced image (Right)

The enhanced image after FFT has the improvements to connect some falsely broken points on ridges and to remove some spurious connections between ridges. The shown image at the left side of figure 2.10 is also processed with histogram equalization after the FFT transform. The side effect of each block is obvious but it has no harm to the further operations because I find the image after consecutive binarization operation is pretty good as long as the side effect is not too severe.

2.3.2 Fingerprint Image Binarization

Fingerprint Image Binarization is to transform the 8-bit Gray fingerprint image to a 1-bit image with 0-value for ridges and 1-value for furrows. After the operation, ridges in the fingerprint are highlighted with black color while furrows are white.

A locally adaptive binarization method is performed to binarize the fingerprint image. Such a named method comes from the mechanism of transforming a pixel value to 1 if the value is larger than the mean intensity value of the current block (16x16) to which the pixel belongs [Figure 2.11].

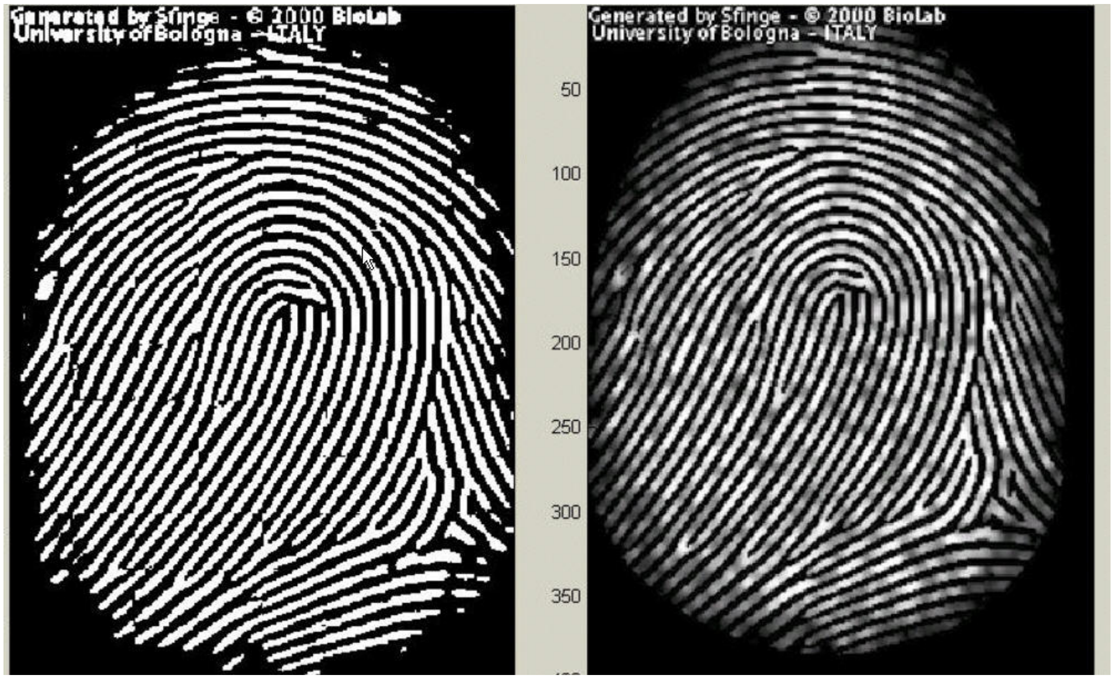

Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY
Generated by Sfinge - © 2000 BioLab
University of Bologna - ITALY
50
100
150
200
250
300
350

Figure 2.11: the Fingerprint image after adaptive binarization

Binarized image(left), Enhanced gray image(right)

2.3.3 Fingerprint Image Segmentation

In general, only a Region of Interest (ROI) is useful to be recognized for each fingerprint image. The image area without effective ridges and furrows is first discarded since it only holds background information. Then the bound of the remaining effective area is sketched out since the minutia in the bound region are confusing with those spurious minutia that are generated when the ridges are out of the sensor.

To extract the ROI, a two-step method is used. The first step is block direction estimation and direction variety check, while the second is intrigued from some Morphological methods.

1. Block direction estimation

1.1 Estimate the block direction for each block of the fingerprint image with WxW in size(W is 16 pixels by default). The algorithm is:

I. Calculate the gradient values along x-direction (g_x) and y-direction (g_y) for each pixel of the block. Two Sobel filters are used to fulfill the task.

II. For each block, use Following formula to get the Least Square approximation of the block direction.

$tg2ß = 2 \sum \sum (g_x * g_y) / \sum \sum (g_x^2 - g_y^2)$ for all the pixels in each block.

The formula is easy to understand by regarding gradient values along x-direction and y-direction as cosine value and sine value. So the tangent value of the block direction is estimated nearly the same as the way illustrated by the following formula.

$$tg2\theta = 2\sin\theta \cos\theta / (\cos^2\theta - \sin^2\theta)$$

1.2 After finished with the estimation of each block direction, those blocks without significant information on ridges and furrows are discarded based on the following formulas:

$$E = \{2 \sum \sum (g_x * g_y) + \sum \sum (g_x^2 - g_y^2)\} / W*W*\sum \sum (g_x^2 + g_y^2)$$

For each block, if its certainty level E is below a threshold, then the block is regarded as a background block.

The direction map is shown in the following diagram. We assume there is only one fingerprint in each image.

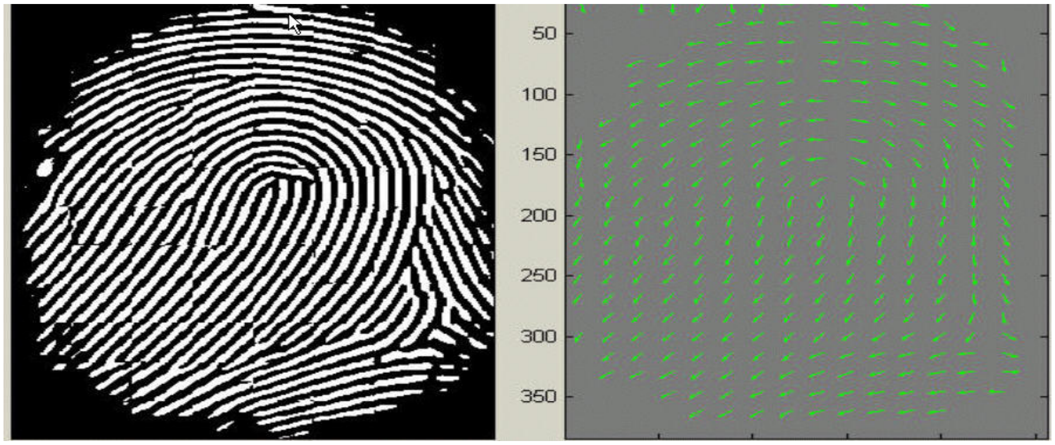


Figure 2.12: Direction map.

Binarized fingerprint (left), Direction map (right)

2. ROI extraction by Morphological operations

Two Morphological operations called ‘OPEN’ and ‘CLOSE’ are adopted. The ‘OPEN’ operation can expand images and remove peaks introduced by background noise [Figure 2.14]. The ‘CLOSE’ operation can shrink images and eliminate small cavities [Figure 2.15].

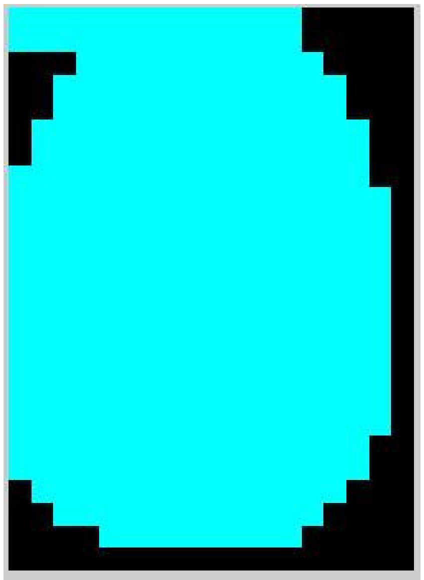


Figure 2.13: Original Image Area

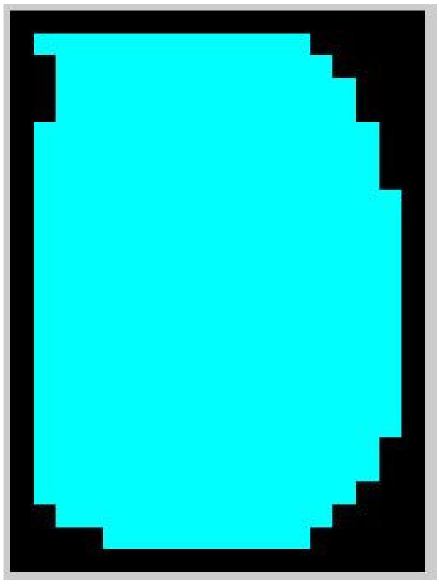


Figure 2.15: After CLOSE operation

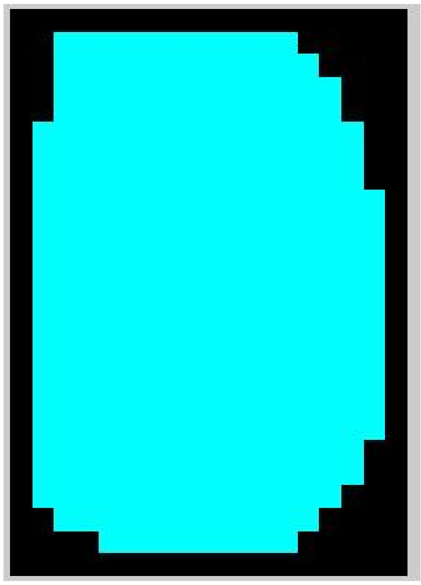


Figure 2.14: After OPEN operation

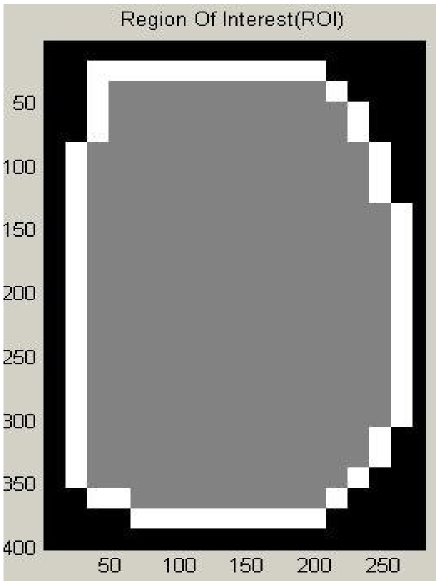


Figure 2.16: ROI + Bound

Figure 2.16 shows the interest fingerprint image area and its bound. The bound is the subtraction of the closed area from the opened area. Then the algorithm throws away those leftmost, rightmost, uppermost and bottommost blocks out of the bound so as to get the tightly bounded region just containing the bound and inner area.

2.4 Minutia Extraction

2.4.1 Fingerprint Ridge Thinning

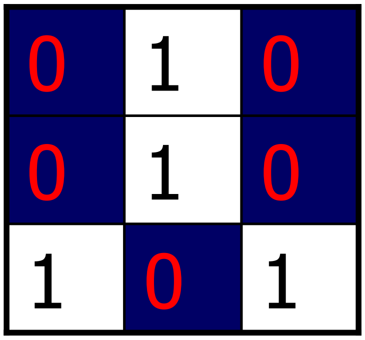
Ridge Thinning is to eliminate the redundant pixels of ridges till the ridges are just one pixel wide uses an iterative, parallel thinning algorithm. In each scan of the full fingerprint image, the algorithm marks down redundant pixels in each small image window (3x3). And finally removes all those marked pixels after several scans. In this testing, such an iterative, parallel thinning algorithm has bad efficiency although it can get an ideal thinned ridge map after enough scans. Uses a one-in-all method to extract thinned ridges from gray-level fingerprint images directly. Their method traces along the ridges having maximum gray intensity value. However, binarization is implicitly enforced since only pixels with maximum gray intensity value are remained. Also in this testing, the advancement of each trace step still has large computation complexity although it does not require the movement of pixel by pixel as in other thinning algorithms. Thus the third method is bid out which uses the built-in Morphological thinning function in MATLAB.

The thinned ridge map is then filtered by other three Morphological operations to remove some H breaks, isolated points and spikes.

2.4.2 Minutia Marking

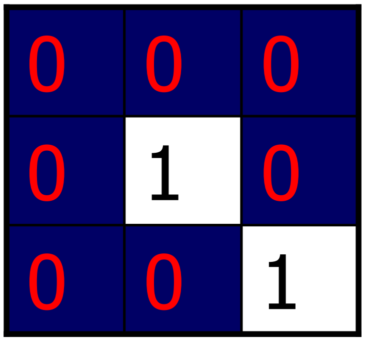
After the fingerprint ridge thinning, marking minutia points is relatively easy. But it is still not a trivial task as most literatures declared because at least one special case evokes this caution during the minutia marking stage.

In general, for each 3x3 window, if the central pixel is 1 and has exactly 3 one-value neighbors, then the central pixel is a ridge branch [Figure 2.17]. If the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending [Figure 2.18].



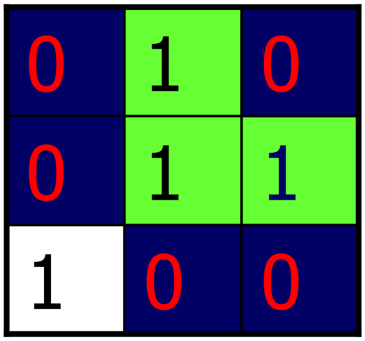
0	1	0
0	1	0
1	0	1

Figure 2.17:Bifurcation



0	0	0
0	1	0
0	0	1

Figure 2.18: Termination



0	1	0
0	1	1
1	0	0

Figure 2.19: Triple counting branch

Figure 2.19 illustrates a special case that a genuine branch is triple counted. Suppose both the uppermost pixel with value 1 and the rightmost pixel with value 1 have another neighbor outside the 3x3 window, so the two pixels will be marked as branches too. But actually only one branch is located in the small region. So a check routine requiring that none of the neighbors of a branch are branches is added.

Also the average inter-ridge width D is estimated at this stage. The average inter-ridge width refers to the average distance between two neighboring ridges. The way to approximate the D value is simple. Scan a row of the thinned ridge image and sum up all pixels in the row whose value is one. Then divide the row length with the above summation to get an inter-ridge width. For more accuracy, such kind of row scan is performed upon several other rows and column scans are also conducted, finally all the inter-ridge widths are averaged to get the D.

Together with the minutia marking, all thinned ridges in the fingerprint image are labeled with a unique ID for further operation. The labeling operation is realized by using the Morphological operation: BWLABEL.

2.5 Minutia Post-processing

2.5.1 False Minutia Removal

The preprocessing stage does not totally heal the fingerprint image. For example, false ridge breaks due to insufficient amount of ink and ridge cross-connections due to over inking are not totally eliminated. Actually all the earlier stages themselves occasionally introduce some artifacts, which later lead to spurious minutia. This false minutia will significantly affect the accuracy of matching if they are simply regarded as genuine minutia. So some mechanisms of removing false minutia are essential to keep the fingerprint verification system effective.

Seven types of false minutia are specified in following diagrams:

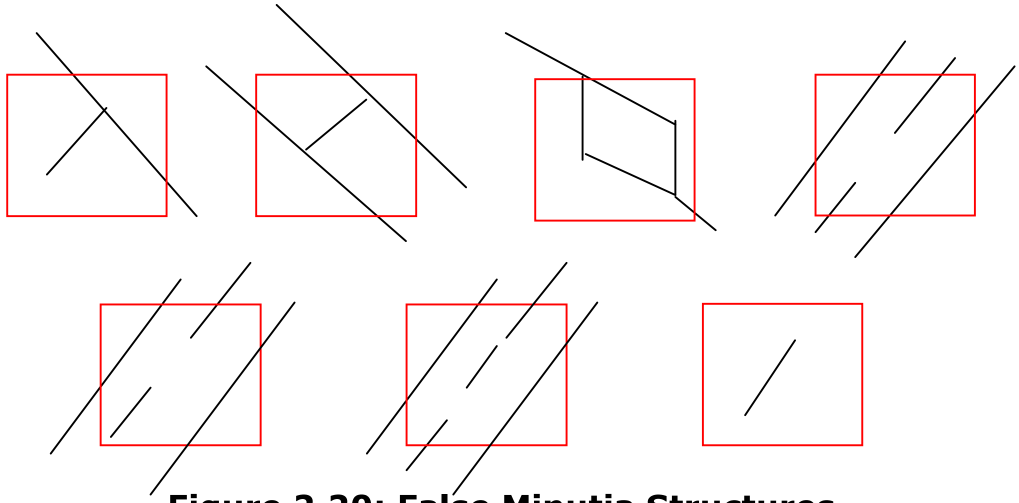


Figure 2.20: False Minutia Structures

m1 is a spike piercing into a valley. In the m2 case a spike falsely connects two ridges. m3 has two near bifurcations located in the same ridge. The two ridge broken points in the m4 case have nearly the same orientation and a short distance. m5 is alike the m4 case with the exception that one part of the broken ridge is so short that another termination is generated. m6 extends the m4 case but with the extra property that a third ridge is found in the middle of the two parts of the broken ridge. m7 has only one short ridge found in the threshold window.

only handles the case m1, m4,m5 and m6 and have not false minutia removal by simply assuming the image quality is fairly good has not a systematic healing method to remove those spurious minutia although it lists all types of false minutia shown in Figure 2.20 except the m3 case.

These procedures in removing false minutia are:

1. If the distance between one bifurcation and one termination is less than D and the two minutia are in the same ridge(m1 case) . Remove both of them. Where D is the average inter-ridge width representing the average distance between two parallel neighboring ridges.
2. If the distance between two bifurcations is less than D and they are in the same ridge, remove the two bifurcations. (m2, m3 cases).
3. If two terminations are within a distance D and their directions are coincident with a small angle variation. And they suffice the condition that no any other termination is located between the two

terminations. Then the two terminations are regarded as false minutia derived from a broken ridge and are removed. (case m4,m5, m6).

4. If two terminations are located in a short ridge with length less than D, remove the two terminations (m7).

this proposed procedures in removing false minutia have two advantages. One is that the ridge ID is used to distinguish minutia and the seven types of false minutia are strictly defined comparing with those loosely defined by other methods. The second advantage is that the order of removal procedures is well considered to reduce the computation complexity. It surpasses the way adopted by that does not utilize the relations among the false minutia types. For example, the procedure3 solves the m4, m5 and m6 cases in a single check routine. And after procedure 3, the number of false minutia satisfying the m7 case is significantly reduced.

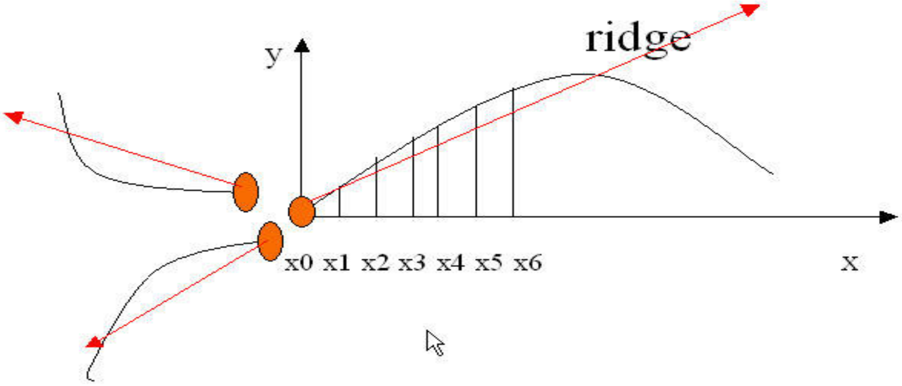
2.5.2 Unify terminations and bifurcations

Since various data acquisition conditions such as impression pressure can easily change one type of minutia into the other, most researchers adopt the unification representation for both termination and bifurcation. So each minutia is completely characterized by the following parameters at last: 1) x-coordinate, 2) y-coordinate, and 3) orientation.

The orientation calculation for a bifurcation needs to be specially considered. All three ridges deriving from the bifurcation point have their own direction, represents the bifurcation orientation using a technique proposed in [Figure 2.2] simply chooses the minimum angle among the three anticlockwise orientations starting from the x-axis. Both methods cast the other two directions away, so some information loses. Here I propose a novel representation to break a bifurcation into three terminations. The three new terminations are the three neighbor pixels of the bifurcation and each of the three ridges connected to the bifurcation before is now associated with a termination respectively [Figure 2.21].

0	0	1
1	1	0
0	0	1

Figure 2.21: A bifurcation to three terminations
Three neighbors become terminations (Top)
Each termination has their own orientation (Bottom)



And the orientation of each termination (tx,ty) is estimated by following method Track a ridge segment whose starting point is the termination and length is D. Sum up all x-coordinates of points in the ridge segment. Divide above summation with D to get sx. Then get sy using the same way.

Get the direction from: atan((sy-ty)/(sx-tx)).

2.6 Minutia Match

Given two set of minutia of two fingerprint images, the minutia match algorithm determines whether the two minutia sets are from the same finger or not.

An alignment-based match algorithm partially derived from is used in this project. It includes two consecutive stages: one is alignment stage and the second is match stage.

1. Alignment stage. Given two fingerprint images to be matched, choose any one minutia from each image, calculate the similarity of the two ridges associated with the two referenced minutia points. If the similarity is larger than a threshold, transform each set of minutia to a new coordination system whose origin is at the referenced point and whose x-axis is coincident with the direction of the referenced point.
2. Match stage: After we get two set of transformed minutia points, we use the elastic match algorithm to count the matched minutia pairs by assuming two minutia having nearly the same position and direction are identical.

2.6.1 Alignment Stage

1. The ridge associated with each minutia is represented as a series of x-coordinates (x_1, $x_2 \ldots x_n$) of the points on the ridge. A point is sampled per ridge length L starting from the minutia point, where the L is the average inter-ridge length. And n is set to 10 unless the total ridge length is less than 10*L.

So the similarity of correlating the two ridges is derived from:

$$S = \sum^{m}_{i=0} x_i X_i / [\sum^{m}_{i=0} x_i^2 X_i^2]\text{^}0.5,$$

where ($x_{i\sim} x_n$) and ($X_{i\sim} X_N$) are the set of minutia for each fingerprint image respectively. And m is minimal one of the n and N value. If the similarity score is larger than 0.8, then go to step 2, otherwise continue to match the next pair of ridges.

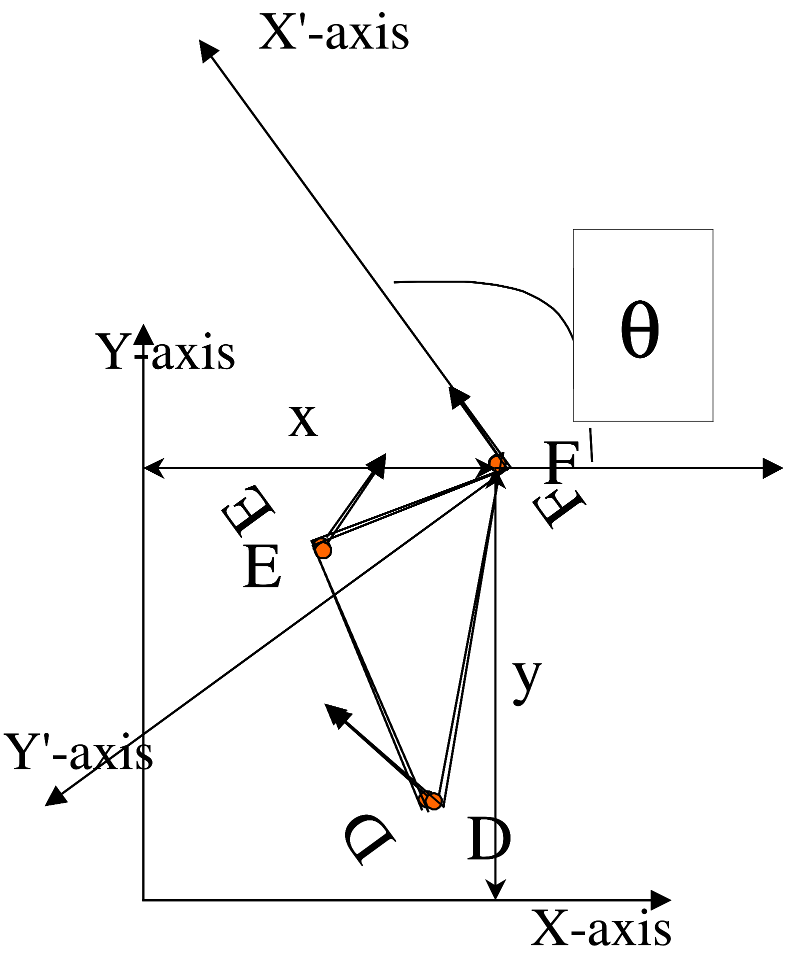
2. For each fingerprint, translate and rotate all other minutia with respect to the reference minutia according to the following formula:

$$\begin{pmatrix} xi\_new \\ yi\_new \\ \theta i\_new \end{pmatrix} = TM * \begin{bmatrix} (xi - x) \\ (yi - y) \\ (\theta i - \theta) \end{bmatrix},$$

where (x,y,θ) is the parameters of the reference minutia, and TM is

$$\text{TM} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The following diagram illustrate the effect of translation and rotation:



The new coordinate system is originated at minutia F and the new x-axis is coincident with the direction of minutia F. No scaling effect is taken into account by assuming two fingerprints from the same finger have nearly the same size.

This method to align two fingerprints is almost the same with the one used by but is different at step 2. Lin's method uses the rotation angle calculated from all the sparsely sampled ridge points. This method use the rotation angle calculated earlier by densely tracing a short ridge start from the minutia with length D. Since I have already got the minutia direction at the minutia extraction stage, obviously this method reduces the redundant calculation but still holds the accuracy.

Also Lin's way to do transformation is to directly align one fingerprint image to another according to the discrepancy of the reference minutia pair. But it still requires a transform to the polar coordinate system for each image at the next minutia match stage. This approach is to transform each according to its own reference minutia and then do match in a unified x-y coordinate. Therefore, less computation workload is achieved through this method.

2.6.2 Match Stage

The matching algorithm for the aligned minutia patterns needs to be elastic since the strict match requiring that all parameters (x, y, θ) are the same for two identical minutia is impossible due to the slight deformations and inexact quantizations of minutia.

This approach to elastically match minutia is achieved by placing a bounding box around each template minutia. If the minutia to be matched is within the rectangle box and the direction discrepancy between them is very small, then the two minutias are regarded as a matched minutia pair. Each minutia in the template image either has no matched minutia or has only one corresponding minutia.

The final match ratio for two fingerprints is the number of total matched pair over the number of minutia of the template fingerprint. The score is 100*ratio and ranges from 0 to 100. If the score is larger than a pre-specified threshold, the two fingerprints are from the same finger.

However, the elastic match algorithm has large computation complexity and is vulnerable to spurious minutia.

2.7 Fingerprint Experimentation Evaluation

2.7.1 Evaluation Indexes for Fingerprint Recognition

Two indexes are well accepted to determine the performance of a fingerprint recognition system: one is FRR (false rejection rate) and the other is FAR (false acceptance rate). For an image database, each sample is matched against the remaining samples of the same finger to compute the False Rejection Rate. If the matching *g* against *h* is performed, the symmetric one (i.e., *h* against *g*) is not executed to avoid correlation. All the scores for such matches are composed into a series of Correct Score. Also the first sample of each finger in the database is matched against the first sample of the remaining fingers to compute the False Acceptance Rate. If the matching *g* against *h* is performed, the symmetric one (i.e., *h* against *g*) is not executed to avoid correlation. All the scores from such matches are composed into a series of Incorrect Score.

2.7.2 Fingerprint Experimentation Analysis

A fingerprint database from the FVC2000 (Fingerprint Verification Competition 2000) is used to test the experiment performance. This program tests all the images without any fine-tuning for the database. The experiments show this program can differentiate imposturous minutia pairs from genuine minutia pairs in a certain confidence level. Furthermore, good experiment designs can surely improve the accuracy as declared by. Further studies on good designs of training and testing are expected to improve the result.

Here is the diagram for Correct Score and Incorrect Score distribution:

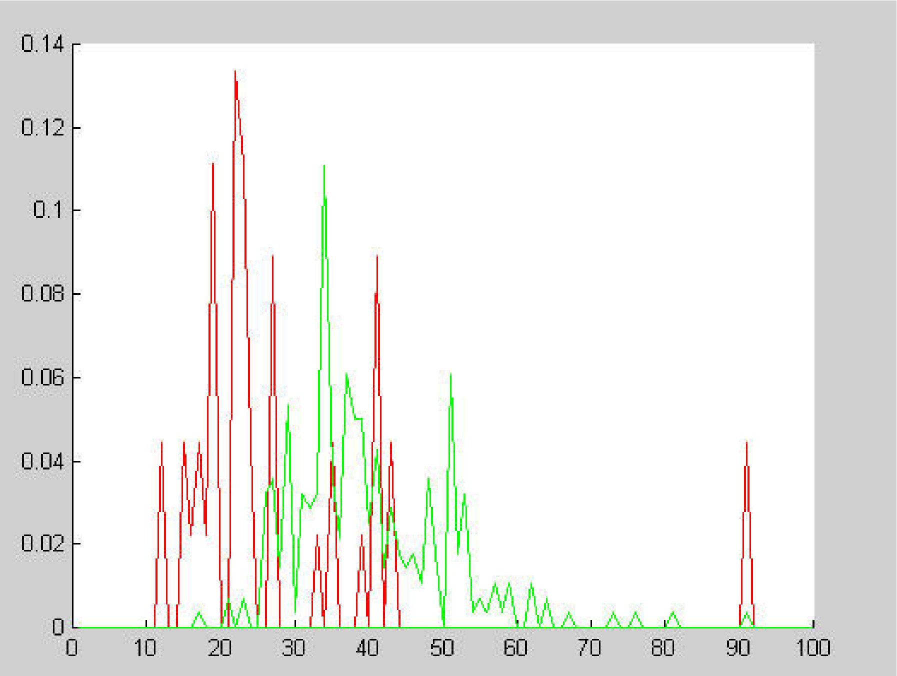


Figure 2.22: Distribution of Correct Scores and Incorrect Scores

Red line: Incorrect Score
Green line: Correct Scores

It can be seen from the above figure that there exist two partially overlapped distributions. The Red curve whose peaks are mainly located at the left part means the average incorrect match score is 25. The green curve whose peaks are mainly located on the right side of red curve means the average correct match score is 35. This indicates the algorithm is capable of differentiate fingerprints at a good correct rate by setting an appropriate threshold value.

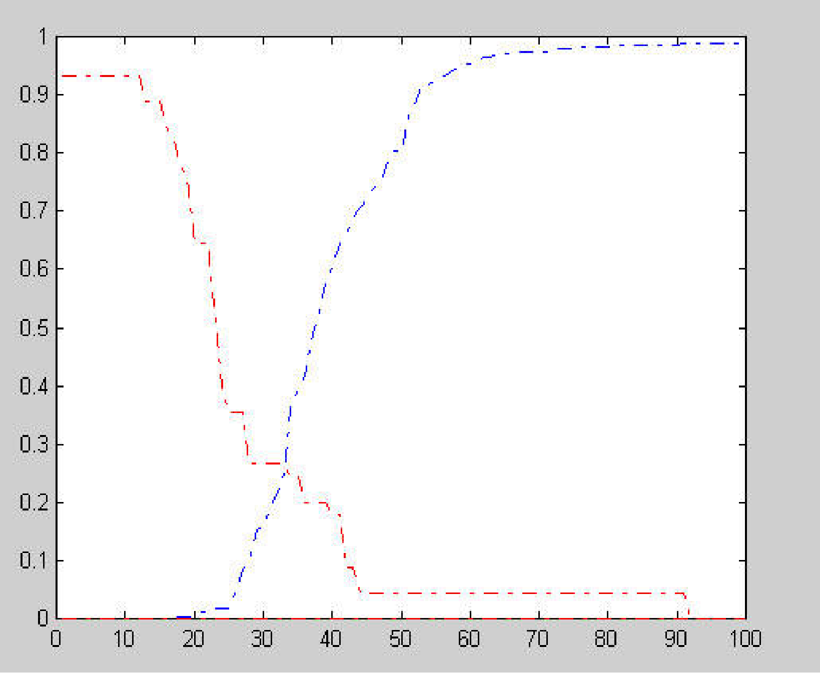


Figure 2.23: FAR and FRR curve
Blue dot line: FRR curve
Red dot line: FAR curve

The above diagram shows the FRR and FAR curves. At the equal error rate 25%, the separating score 33 will falsely reject 25% genuine minutia pairs and falsely accept 25% imposturous minutia pairs and has 75% verification rate.

The high incorrect acceptance and false rejection are due to some fingerprint images with bad quality and the vulnerable minutia match algorithm.

This project has combined many methods to build a minutia extractor and a minutia matcher. The combination of multiple methods comes from a wide investigation into research papers. Also some novel changes like segmentation using Morphological operations, minutia marking with special considering the triple branch counting, minutia unification by decomposing a branch into three terminations, and matching in the unified x-y coordinate system after a two-step transformation are used in this project.

Also a program coding with MATLAB going through all the stages of the fingerprint recognition is built. It is helpful to understand the procedures of fingerprint recognition. And demonstrate the key issues of fingerprint recognition.

References:

- Automatic Personal Identification Using Fingerprints , Wiley
- Biometrics Identity Verification in a Network world,
  Wiley Publications
- Biometric Systems: Technology, Design and Performance Evaluation, Springer
- www.biometrics.cse.msu.edu
- www.acs.com.hk
- www.findbiometrics.com
- www.biometricgroup.com
- www.precisebiometrics.com
- www.bioapi.org

Chapter - 3

Face Recognition System

3.1 Introduction

3.2 Face Recognition Processing

3.3 Analysis in Face Subspaces

3.4 Technical Challenges

3.5 Technical Solutions

3.6 Current Technology Maturity

3.7 Face Recognition Technologies

3.1 Introduction

Face recognition is a task that humans perform routinely and effortlessly in their daily lives. Wide availability of powerful and low-cost desktop and embedded computing systems has created an enormous interest in automatic processing of digital images and videos in a number of applications, including biometric authentication, surveillance, human-computer interaction, and multimedia management. Research and development in automatic face recognition follows naturally.

Research in face recognition is motivated not only by the fundamental challenges this recognition problem poses but also by numerous practical applications where human identification is needed. Face recognition, as one of the primary biometric technologies, became more and more important owing to rapid advances in technologies such as digital cameras, the Internet and mobile devices, and increased demands on security. Face recognition has several advantages over other biometric technologies: It is natural, nonintrusive, and easy to use. Among the six biometric attributes considered by Hietmeyer, facial features scored the highest compatibility in a Machine Readable Travel Documents (MRTD) system based on a number of evaluation factors, such as enrollment, renewal, machine requirements, and public perception, shown in Figure 3.1

A face recognition system is expected to identify faces present in images and videos automatically. It can operate in either or both of two modes: (1) face verification (or authentication), and (2) face identification (or recognition). Face verification involves a one-to-one match that compares a query face image against a template face image whose identity is being claimed. Face identification involves a one-to-many match that compares a query face image against all the template images in the database to determine the identity of the query face. Another face recognition scenario involves a watch-list check, where a query face is matched to a list of suspects (one-to-few matches).

The performance of face recognition systems has improved significantly since the first automatic face recognition system was developed by Kanade. Furthermore, face detection, facial feature extraction, and recognition can now be performed in "real time" for images captured under favorable (i.e., constrained) situations.

Although progress in face recognition has been encouraging, the task has also turned out to be a difficult endeavor, especially for unconstrained tasks where viewpoint, illumination, expression, occlusion, accessories, and so on vary considerably. In the following sections, we give a brief review on technical advances and analyze technical challenges.

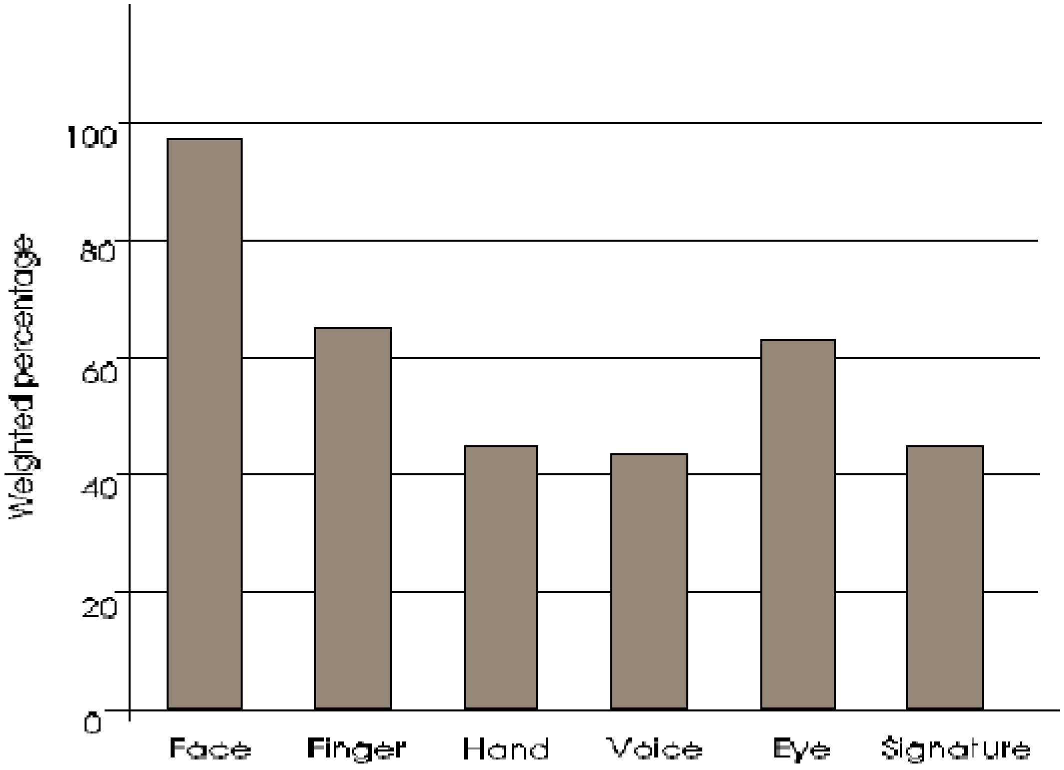


Figure 3.1: A scenario of using biometric MRTD systems for passport control (top), and a comparison of various biometric features based on MRTD compatibility (bottom).

3.2 Face Recognition Processing

Face recognition is a visual pattern recognition problem. There, a face as a three-dimensional object subject to varying illumination, pose, expression and so on is to be identified based on its two-dimensional image (three-dimensional images e.g., obtained from laser may also be used). A face recognition system generally consists of four modules as depicted in Figure 3.2: detection, alignment, feature extraction, and matching, where localization and normalization (face detection and alignment) are processing steps before face recognition (facial feature extraction and matching) is performed.

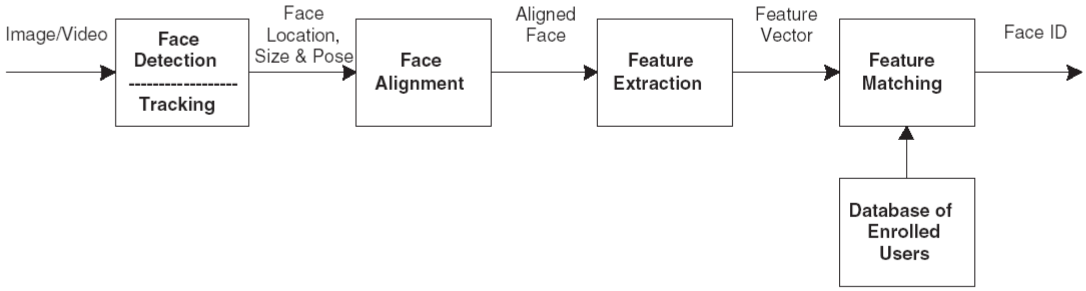


Figure 3.2: Face recognition processing flow.

Face detection segments the face areas from the background. In the case of video, the detected faces may need to be tracked using a face-tracking component. Face alignment is aimed at achieving more accurate localization and at normalizing faces thereby whereas face detection provides coarse estimates of the location and scale of each detected face. Facial components, such as eyes, nose, and mouth and facial outline, are located; based on the location points, the input face image is normalized with respect to geometrical properties, such as size and pose, using geometrical transforms or morphing. The face is usually further normalized with respect to photometrical properties such illumination and gray scale.

After a face is normalized geometrically and photometrically, feature extraction is performed to provide effective information that is useful for distinguishing between faces of different persons and stable with respect to the geometrical and photometrical variations. For *face matching*, the extracted feature vector of the input face is matched against those of enrolled faces in the database; it outputs the identity of the face when a match is found with sufficient confidence or indicates an unknown face otherwise.

Face recognition results depend highly on features that are extracted to represent the face pattern and classification methods used to distinguish between faces whereas face localization and normalization are the basis for extracting effective features. These problems may be analyzed from the viewpoint of face subspaces or manifolds, as follows.

3.3 Analysis in Face Subspaces

Subspace analysis techniques for face recognition are based on the fact that a class of patterns of interest, such as the face, resides in a subspace of the input image space. For example, a small image of 64 × 64 has 4096 pixels can express a large number of pattern classes, such as trees, houses and faces. However, among the $256^{4096} > 10^{9864}$ possible "configurations," only a few correspond to faces. Therefore, the original image representation is highly redundant, and the dimensionality of this representation could be greatly reduced when only the face pattern are of interest.

With the eigenface or principal component analysis (PCA) approach, a small number (e.g., 40 or lower) of eigenfaces are derived from a set of training face images by using the Karhunen-Loeve transform or PCA. A face image is efficiently represented as a feature vector (i.e., a vector of weights) of low dimensionality. The features in such subspace provide more salient and richer information for recognition than the raw image. The use of subspace modeling techniques has significantly advanced face recognition technology.

The manifold or distribution of all faces accounts for variation in face appearance whereas the non-face manifold accounts for everything else. If we look into these manifolds in the image space, we find them highly nonlinear and non-convex. Figure 3.3(a) illustrates face versus non-face manifolds and (b) illustrates the manifolds of two individuals in the entire face manifold. Face detection can be considered as a task of distinguishing between the face and non-face manifolds in the image (sub window) space and face recognition between those of individuals in the face manifold.

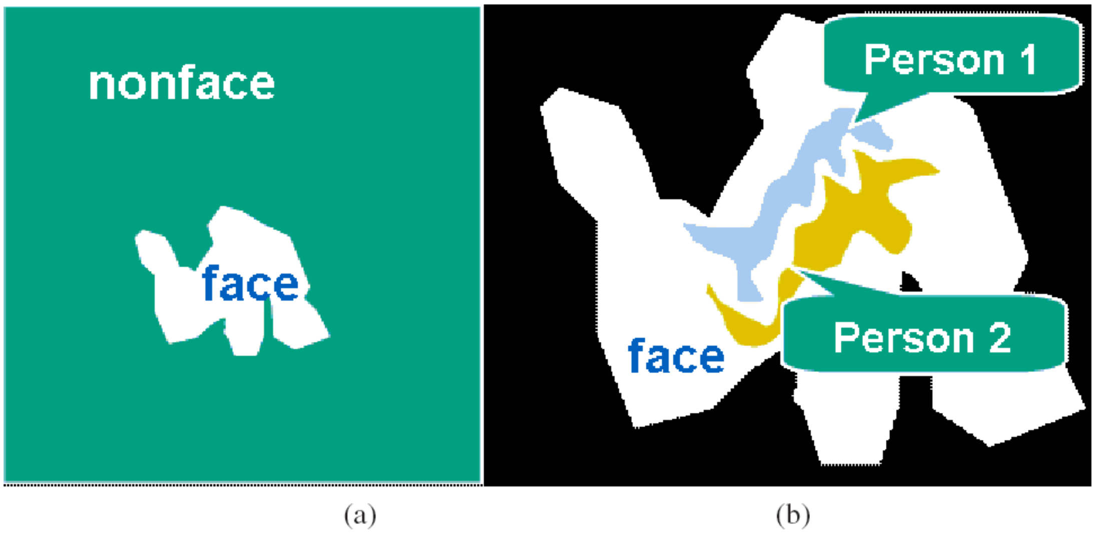


Figure 3.3: (a) Face versus non-face manifolds. (b) Face manifolds of different individuals.

Figure 3.4 further demonstrates the non-linearity and non-convexity of face manifolds in a PCA subspace spanned by the first three principal components, where the plots are drawn from real face image data. Each plot depicts the manifolds of three individuals (in three colors). There are 64 original frontal face images for each individual. A certain type of transform is performed on an original face image with 11 gradually varying parameters, producing 11 transformed face images; each transformed image is cropped to contain only the face region; the 11 cropped face images form a sequence. A curve in this figure is the image of such a sequence in the PCA space, and so there are 64 curves for each individual. The three-dimensional (3D) PCA space is projected on three 2D spaces (planes). We can see the non-linearity of the trajectories.

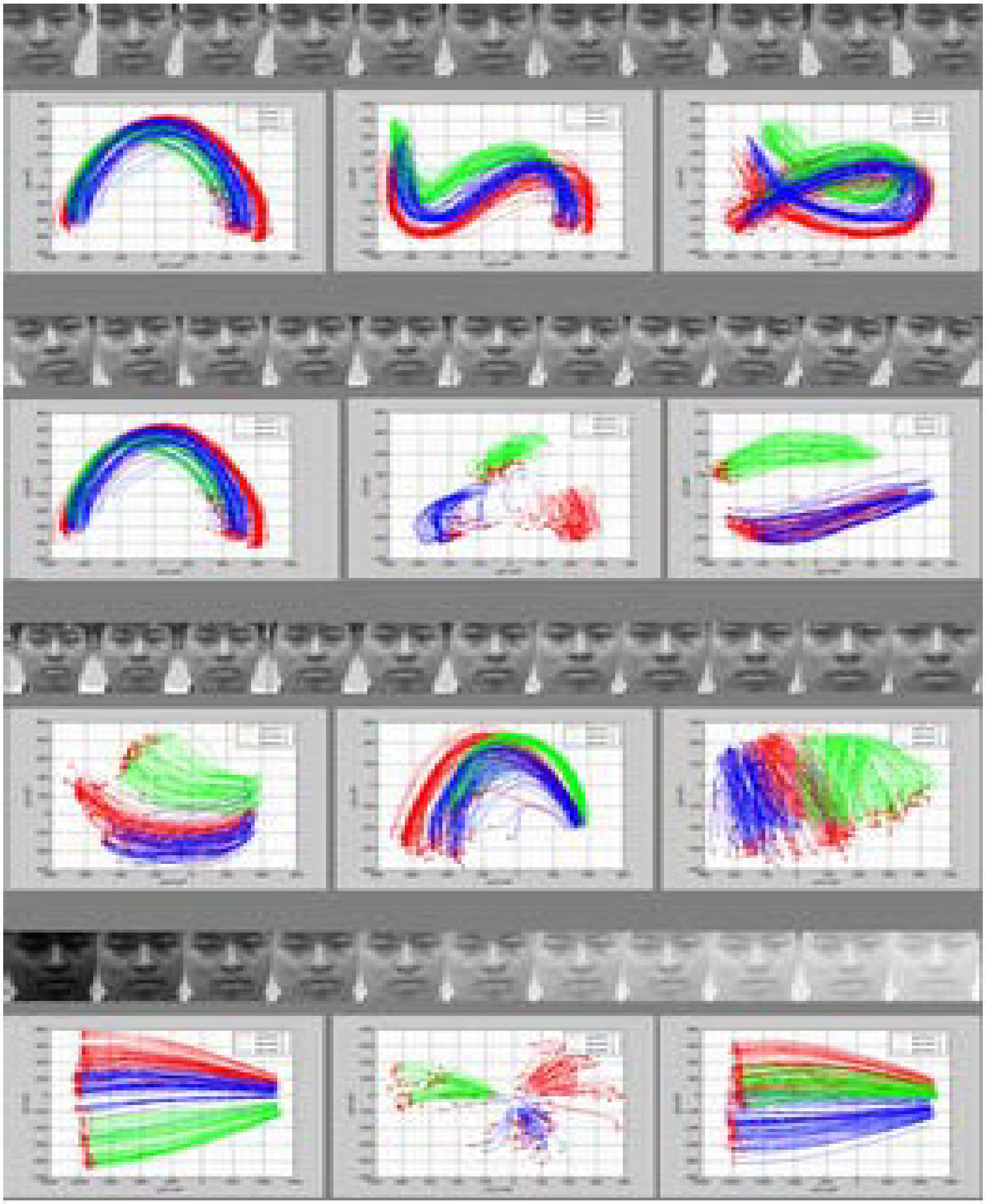


Figure 3.4: Non-linearity and non-convexity of face manifolds under (from top to bottom) translation, rotation , scaling, and Gamma transformations.

Two notes follow: First, while these examples are demonstrated in a PCA space, more complex (nonlinear and non-convex) curves are expected in the original image space. Second, although these examples are subject the geometric transformations in the 2D plane and point wise lighting (gamma) changes, more significant complexity is expected for geometric transformations in 3D (e.g. out-of-plane head rotations) transformations and lighting direction changes.

3.4 Technical Challenges

As shown in Figure 3.3, the classification problem associated with face detection is highly nonlinear and non-convex, even more so for face matching. Face recognition evaluation reports and other independent studies indicate that the performance of many state-of the-art face recognition methods deteriorates with changes in lighting, pose, and other factors. The key technical challenges are summarized below.

Large Variability in Facial Appearance. Whereas shape and reflectance are intrinsic properties of a face object, the appearance (i.e., the texture look) of a face is also subject to several other factors, including the facial pose (or, equivalently, camera viewpoint), illumination, facial expression. Figure 3.5 shows an example of significant intrasubject variations caused by these factors. In addition to these, various imaging parameters, such as aperture, exposure time, lens aberrations, and sensor spectral response also increase intrasubject variations. Face-based person identification is further complicated by possible small intersubject variations (Figure 3.6). All these factors are confounded in the image data, so "the variations between the images of the same face due to illumination and viewing direction are almost always larger than the image variation due to change in face identity". This variability makes it difficult to extract the intrinsic information of the face objects from their respective images.

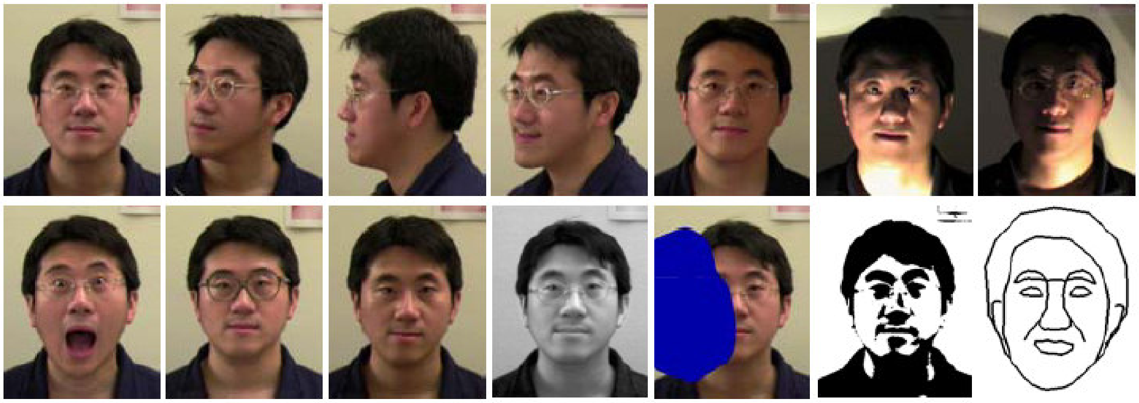


Figure 3.5: Intra subject variations in pose, illumination, expression, occlusion, accessories (e.g., glasses), color, and brightness. (Courtesy of Rein-Lien Hsu)

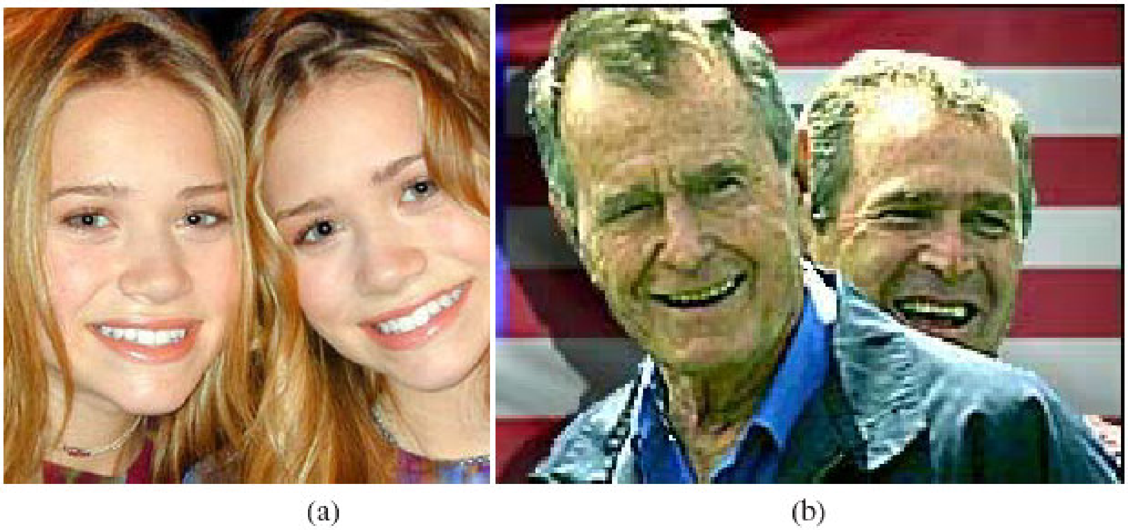


Figure 3.6: Similarity of frontal faces between (a) twins (downloaded from www.marykateandashley.com); and (b) a father and his son (downloaded from BBC news, news.bbc.co.uk).

Highly Complex Nonlinear Manifolds. As illustrated above, the entire face manifold is highly non-convex, and so is the face manifold of any individual under various change. Linear methods such as PCA, independent component analysis (ICA), and linear discriminate analysis (LDA)) project the data linearly from a high-dimensional space (e.g., the image space) to a low-dimensional subspace. As such, they are unable to preserve the non-convex variations of face manifolds necessary to differentiate among individuals. In a linear subspace, Euclidean distance and more generally Mahalanobis distance, which are normally used for template matching, do not perform well for classifying between face and non-face manifolds and between manifolds of individuals (Figure 3.7(a)). This crucial fact limits the power of the linear methods to achieve highly accurate face detection and recognition.

High Dimensionality and Small Sample Size. Another challenge is the ability to generalize, illustrated by Figure 3.7 (b). A canonical face image of 112×92 resides in a 10,304-dimensional feature space. Nevertheless, the number of examples per person (typically fewer than 10, even just one) available for learning the manifold is usually much smaller than the dimensionality of the image space; a system trained on so few examples may not generalize well to unseen instances of the face.

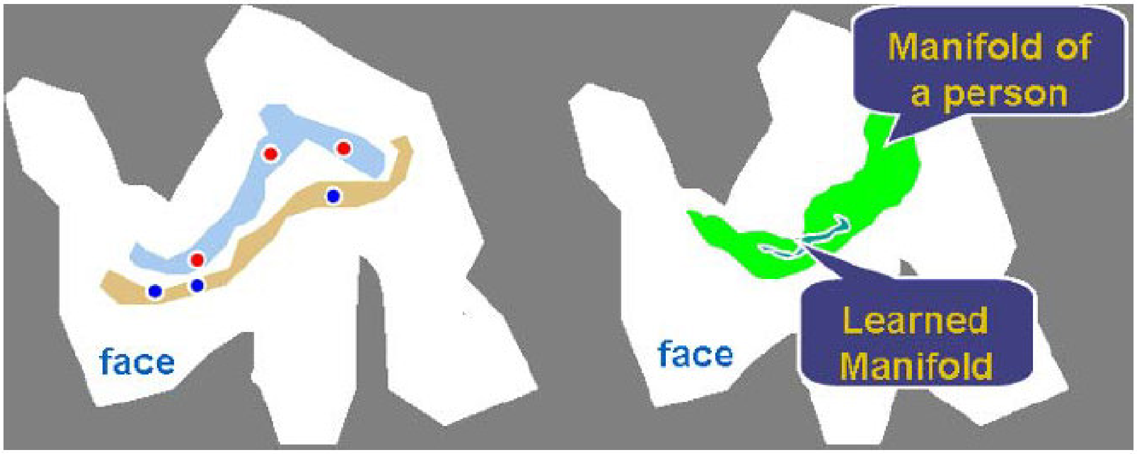


Figure 3.7: Challenges in face recognition from subspace viewpoint. (a) Euclidean distance is unable to differentiate between individuals: In terms of Euclidean distance, an interpersonal distance can be smaller than an intrapersonal one. (b) The learned manifold or classifier is unable to characterize (i.e., generalize to) unseen images of the same individual face.

3.5 Technical Solutions

There are two strategies for dealing with the above difficulties: feature extraction and pattern classification based on the extracted features. One is to construct a “good” feature space in which the face manifolds become simpler i.e., less nonlinear and non-convex than those in the other spaces. This includes two levels of processing: (1) normalize face images geometrically and photometrically, such as using morphing and histogram equalization; and (2) extract features in the normalized images which are stable with respect to such variations, such as based on Gabor wavelets.

The second strategy is to construct classification engines able to solve difficult nonlinear classification and regression problems in the feature space and to generalize better. Although good normalization and feature extraction reduce the non-linearity and non-convexity, they do not solve the problems completely and classification engines able to deal with such difficulties are still necessary to achieve high performance. A successful algorithm usually combines both strategies.

With the geometric feature-based approach used in the early days, facial features such as eyes, nose, mouth, and chin are detected. Properties of and relations (e.g., areas, distances, angles) between the features are used as descriptors for face recognition. Advantages of this approach include economy and efficiency when achieving data reduction and insensitivity to variations in illumination and viewpoint. However, facial feature detection and measurement techniques developed to date are not reliable enough for the geometric feature based recognition, and such geometric properties alone are inadequate for face recognition because rich information contained in the facial texture or appearance is discarded. These are reasons why early techniques are not effective.

The statistical learning approach learns from training data (appearance images or features extracted from appearance) to extract good features and construct classification engines. During the learning, both prior knowledge about face(s) and variations seen in the training data are taken into consideration. Many successful algorithms for face detection, alignment and matching nowadays are learning-based.

The appearance-based approach, such as PCA and LDA based methods, has significantly advanced face recognition techniques. Such an approach generally operates directly on an image-based representation (i.e., array of pixel intensities). It extracts features in a subspace derived from training images. Using PCA, a face subspace is constructed to represent "optimally" only the face object; using LDA, a discriminant subspace is constructed to distinguish "optimally" faces of different persons. Comparative reports show that LDA-based methods generally yield better results than PCA-based ones.

Although these linear, holistic appearance-based methods avoid instability of the early geometric feature-based methods, they are not accurate enough to describe subtleties of original manifolds in the original image space. This is due to their limitations in handling non-linearity in face recognition: there, protrusions of nonlinear manifolds may be smoothed and concavities may be filled in, causing unfavorable consequences.

Such linear methods can be extended using nonlinear kernel techniques (kernel PCA and kernel LDA) to deal with non-linearity in face recognition. There, a nonlinear projection (dimension reduction) from the image space to a feature space is performed; the manifolds in the resulting feature space become simple, yet with subtleties preserved. Although the kernel methods may achieve good performance on the training data, however, it may not be so for unseen data owing to their more flexibility than the linear methods and over fitting thereof.

Another approach to handle the nonlinearity is to construct a local appearance-based feature space, using appropriate image filters, so the distributions of faces are less affected by various changes. Local features analysis (LFA), Gabor wavelet-based features (such as elastic graph bunch matching, EGBM) and local binary pattern (LBP) have been used for this purpose.

Some of these algorithms may be considered as combining geometric (or structural) feature detection and local appearance feature extraction, to increase stability of recognition performance under changes in viewpoint, illumination, and expression. A taxonomy of major face recognition algorithms in Figure 3.8 provides an overview of face recognition technology based on pose dependency, face representation, and features used for matching.

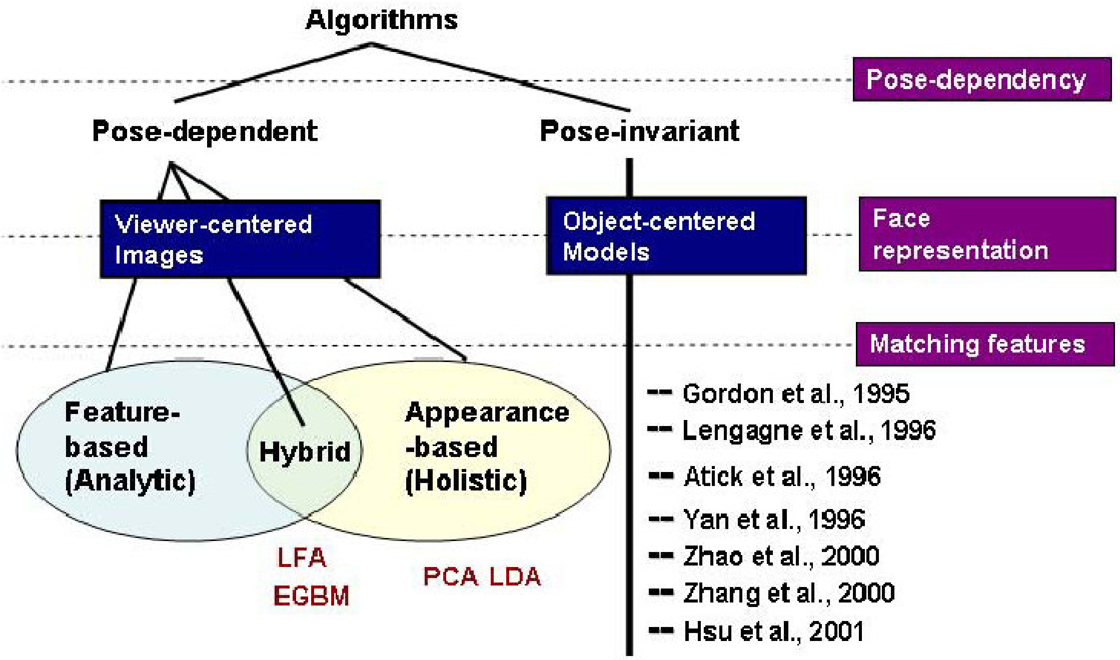


Figure 3.8: Taxonomy of face recognition algorithms based on pose-dependency, face representation, and features used in matching (Courtesy of Rein-Lien Hsu).

A large number of local features can be produced with varying parameters in the position, scale and orientation of the filters. For example, more than 100,000 local appearance features can be produced when an image of 100×100 is filtered with Gabor filters of five scales and eight orientation for all pixel positions, causing increased dimensionality. Some of these features are effective and important for the classification task whereas the others may not be so. AdaBoost methods have been used successfully to tackle the feature selection and nonlinear classification problems. These works lead to a framework for learning both effective features and effective classifiers.

3.6 Current Technology Maturity

As introduced earlier, a face recognition system consists of several components, including face detection, tracking, alignment, feature extraction, and matching. Where are we along the road of making automatic face recognition systems? To answer this question, we have to assume some given constraints namely what the intended situation for the application is and how strong constraints are assumed, including pose, illumination, facial expression, age, occlusion, and facial hair. Real-time face detection and tracking in the normal indoor environment is relatively well solved, whereas more work is needed for handling outdoor scenes. When faces are detected and tracked, alignment can be done as well, assuming the image resolution is good enough for localizing the facial components, face recognition works well for cooperative frontal faces without exaggerated expressions and under illumination without much shadow. Face recognition in an unconstrained daily life environment without the user’s cooperation, such as for recognizing someone in an airport, is currently a challenging task. Many years’ effort is required to produce practical solutions to such problems.

3.7 Face Recognition Technologies

While the internal operations of a facial-scan system are invisible to the deployer, whose primary concern is performance and accuracy, a handful of facial-scan technologies compete within the biometric market, which substantial differences in their operations. Because of their enrollment or verification methods, some types of facial-scan technology are more suitable than others for applications such as forensics, network access and surveillance. Four of the primary methods employed by facial-scan vendors to identify and verify subjects include Eigenface, feature analysis, neural network, and automatic face processing. Other facial-scan technologies based on thermal patterns present under the skin have not yet proven commercially viable.

3.7.1 Eigenface

Eigenface, roughly translated as "one's own face", is a technology patented at MIT that utilize a database of two-dimensional, grayscale facial images (Eigenfaces) from which templates are created during enrollment and verification. These Eigenfaces feature distinctive facial characteristics, and the vast majority of faces can be reconstructed by locating distinctive features from approximately 100 to 125 Eigenfaces. Variations of Eigenface are frequently used as the basis of other face-recognition methods.

Upon enrollment, a subject's facial image is represented using a combination of various Eigenfaces. This reconstruction is then mapped to a series of numbers or coefficients. For 1:1

authentication, in which the image is being used to verify a claimed identity, an individual’s live template is compared against the enrolled template to determine coefficient variation. The degree of variance from the enrollment will determine acceptance or rejection. For 1-to-Many identification, the same principle applies, but with a much larger comparison set. Like all facial recognition technology, Eigenface technology is best utilized in well-lit, frontal image capture situations.

3.7.1.1 Principal Components Analysis (PCA)

Principal Components Analysis (PCA) is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension.

It covers standard deviation, covariance, eigenvectors and eigen values. This background knowledge is meant to make the PCA section very straightforward.

This section will attempt to give some elementary background mathematical skills that will be required to understand the process of Principal Components Analysis. The topics are covered independently of each other, and examples given. It is less important to remember the exact mechanics of a mathematical technique than it is to understand the reason why such a technique may be used, and what the result of the operation tells us about our data. Not all of these techniques are used in PCA, but the ones that are not explicitly required do provide the grounding on which the most important techniques are based.

I have included a section on Statistics that looks at distribution measurements, or, how the data is spread out. The other section is on Matrix Algebra and looks at eigenvectors and eigen values, important properties of matrices that are fundamental to PCA.

The entire subject of statistics is based around the idea that you have this big set of data, and you want to analyze that set in terms of the relationships between the individual points in that data set. We going to look at a few of the measures you can do on a set of data, and what they tell you about the data itself.

Standard Deviation

To understand standard deviation, we need a data set. Statisticians are usually concerned with taking a *sample* of a *population*. To use election polls as an example, the population is all the people in the country, whereas a sample is a subset of the population that the statisticians measure. The great thing about statistics is that by only measuring (in this case by doing a phone survey or similar) a sample of the population, you can work out what is most likely to be the measurement if you used the entire population. In this statistics section, we assume that our data sets are samples of some bigger population. There is a reference later in this section pointing to more information about samples and populations.

Here’s an example set:

$$X = [\ 1\ 2\ 4\ 6\ 12\ 15\ 25\ 45\ 68\ 67\ 65\ 98\ ]$$

We could simply use the symbol X to refer to this entire set of numbers. If we want to refer to an individual number in this data set, we will use subscripts on the symbol X to indicate a specific number. Eg. X_3 refers to the 3rd number in X, namely the number 4. Note that X_1 is the first number in the sequence, not X_0 like you may see in some textbooks. Also, the symbol will be used to refer to the number of elements in the set X.

There are a number of things that we can calculate about a data set. For example, we can calculate the mean of the sample. We assume that the reader understands what the mean of a sample is, and will only give the formula:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

Notice the symbol $\bar{X}$ (said "X bar") to indicate the mean of the set X. All this formula says is "Add up all the numbers and then divide by how many there are". Unfortunately, the mean doesn't tell us a lot about the data except for a sort of middle point. For example, these two data sets have exactly the same mean (10), but are obviously quite different:

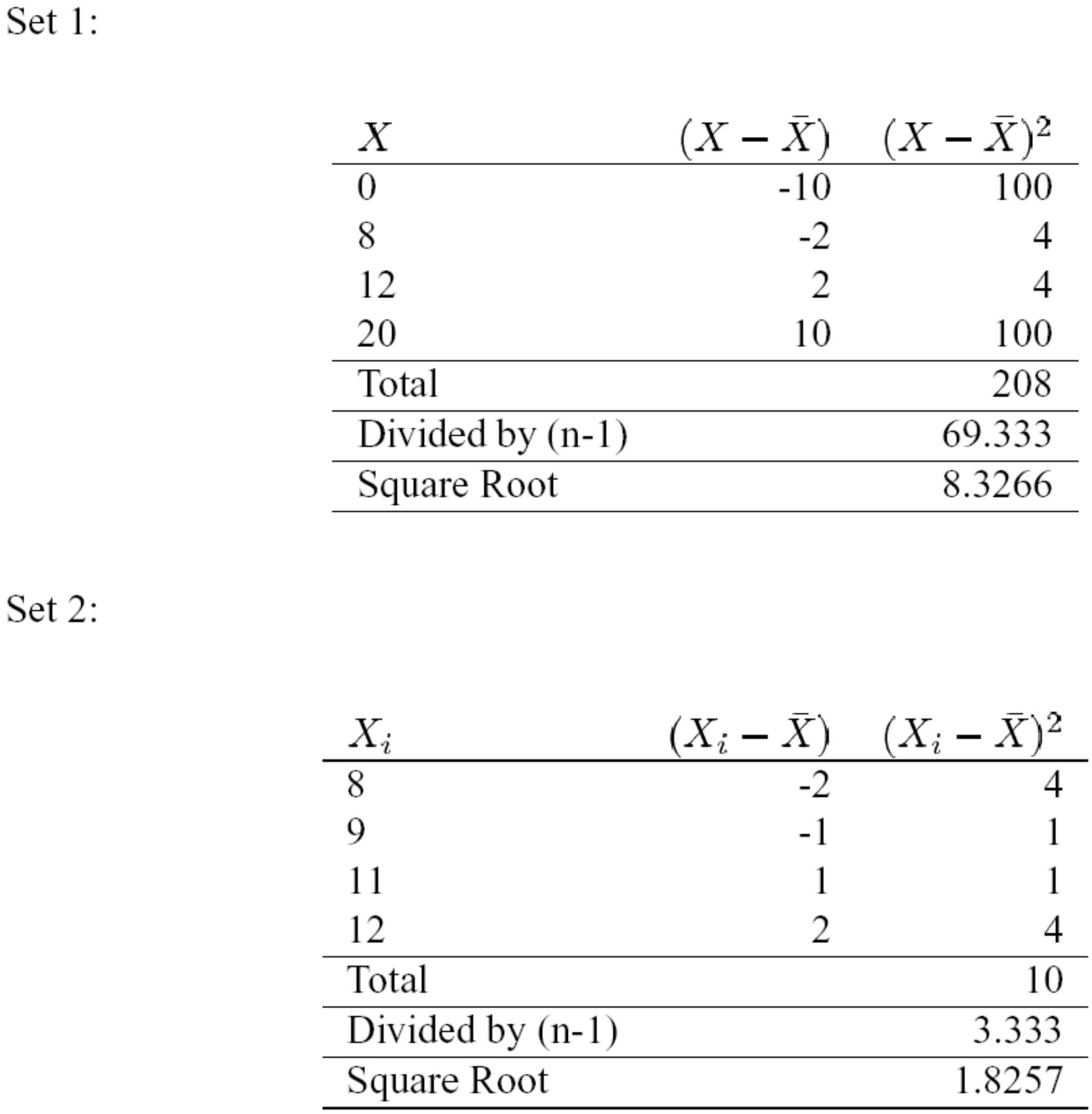
$$[\,0\;8\;12\;20\,]\; and\; [\,8\;9\;11\;12\,]$$

So what is different about these two sets? It is the *spread* of the data that is different. The Standard Deviation (SD) of a data set is a measure of how spread out the data is.

How do we calculate it? The English definition of the SD is: "The average distance from the mean of the data set to a point". The way to calculate it is to compute the squares of the distance from each data point to the mean of the set, add them all up, divide by n - 1 and take the positive square root. As a formula:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}}$$

Where / is the usual symbol for standard deviation of a sample. I hear you asking "Why are you using (n – 1) and not n?" Well, the answer is a bit complicated, but in general, if your data set is a *sample* data set, ie. you have taken a subset of the real-world (like surveying 500 people about the election) then you must use (n – 1) because it turns out that this gives you an answer that is closer to the standard deviation that would result if you had used the *entire* population, than if you'd used n. If, however, you are not calculating the standard deviation for a sample, but for an entire population, then you should divide by n instead of (n – 1) describes standard deviation in a similar way, and also provides an example experiment that shows the difference between each of the denominators. It also discusses the difference between samples and populations.



Set 1:

X	$(X - \bar{X})$	$(X - \bar{X})^2$
0	-10	100
8	-2	4
12	2	4
20	10	100
Total		208
Divided by (n-1)		69.333
Square Root		8.3266

Set 2:

X_i	$(X_i - \bar{X})$	$(X_i - \bar{X})^2$
8	-2	4
9	-1	1
11	1	1
12	2	4
Total		10
Divided by (n-1)		3.333
Square Root		1.8257

Table 3.1 : Calculation of standard deviation

So, for our two data sets above, the calculations of standard deviation are in Table 3.1. And so, as expected, the first set has a much larger standard deviation due to the fact that the data is much more spread out from the mean. Just as another example, the data set:

$$[\ 10\ 10\ 10\ 10\ ]$$

also has a mean of 10, but its standard deviation is 0, because all the numbers are the same. None of them deviate from the mean.

Variance

Variance is another measure of the spread of data in a data set. In fact it is almost identical to the standard deviation. The formula is this:

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

You will notice that this is simply the standard deviation squared, in both the symbol (S^2) and the formula (there is no square root in the formula for variance). S^2 is the usual symbol for variance of a sample. Both these measurements are measures of the spread of the data. Standard deviation is the most common measure, but variance is also used. The reason why I have introduced variance in addition to standard deviation is to provide a solid platform from which the next section, covariance, can launch from.

Covariance

The last two measures we have looked at are purely 1-dimensional. Data sets like this could be: heights of all the people in the room, marks for the last COMP101 exam etc. However many data sets have more than one dimension, and the aim of the statistical analysis of these data sets is usually to see if there is any relationship between the dimensions. For example, we might have as our data set both the height of all the students in a class, and the mark they received for that paper. We could then perform statistical analysis to see if the height of a student has any effect on their mark.

Standard deviation and variance only operate on 1 dimension, so that you could only calculate the standard deviation for each dimension of the data set *independently* of the other dimensions. However, it is useful to have a similar measure to find out how much the dimensions vary from the mean *with respect to each other*. Covariance is such a measure. Covariance is always measured *between* 2 dimensions. If you calculate the covariance between one dimension and *itself*, you get the variance. So, if you had a 3-dimensional data set (x,y,z) then you could measure the covariance between the x and y dimensions, the x and z dimensions, and the y and z dimensions. Measuring the covariance between x and x, or y and y, or z and z would give you the variance of the x ,y and z dimensions respectively.

The formula for covariance is very similar to the formula for variance. The formula for variance could also be written like this:

$$var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

where I have simply expanded the square term to show both parts. So given that knowledge, here is the formula for covariance:

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

It is exactly the same except that in the second set of brackets, the X_ 's are replaced by Y's. This says, in English, "For each data item, multiply the difference between the x value and the mean of x, by the difference between the y value and the mean of y. Add all these up, and divide by (n -1) ".

How does this work? Lets use some example data. Imagine we have gone into the world and collected some 2-dimensional data, say, we have asked a bunch of students how many hours in total that they spent studying COSC241, and the mark that they received. So we have two dimensions, the first is the H dimension, the hours studied, and the second is the M dimension, the mark received. Figure 3.2 holds the imaginary data, and the calculation of cov(H,M) , the covariance between the Hours of study done and the Mark received.

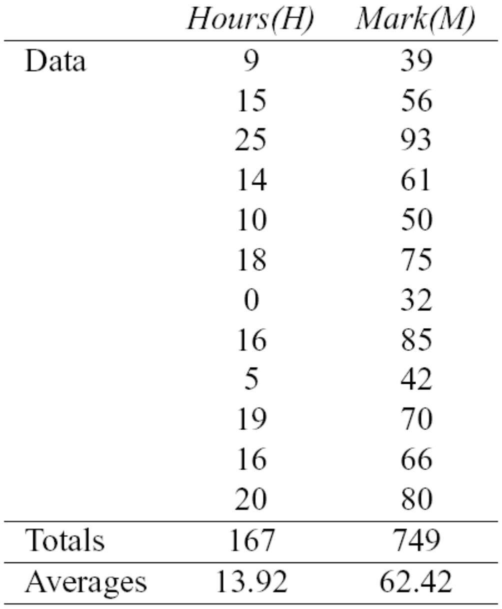
So what does it tell us? The exact value is not as important as it's sign (ie. Positive or negative). If the value is positive, as it is here, then that indicates that both dimensions increase together, meaning that, in general, as the number of hours of study increased, so did the final mark.

If the value is negative, then as one dimension increases, the other decreases. If we had ended up with a negative covariance here, then that would have said the opposite that as the number of hours of study increased the final mark decreased.

In the last case, if the covariance is zero, it indicates that the two dimensions are independent of each other.

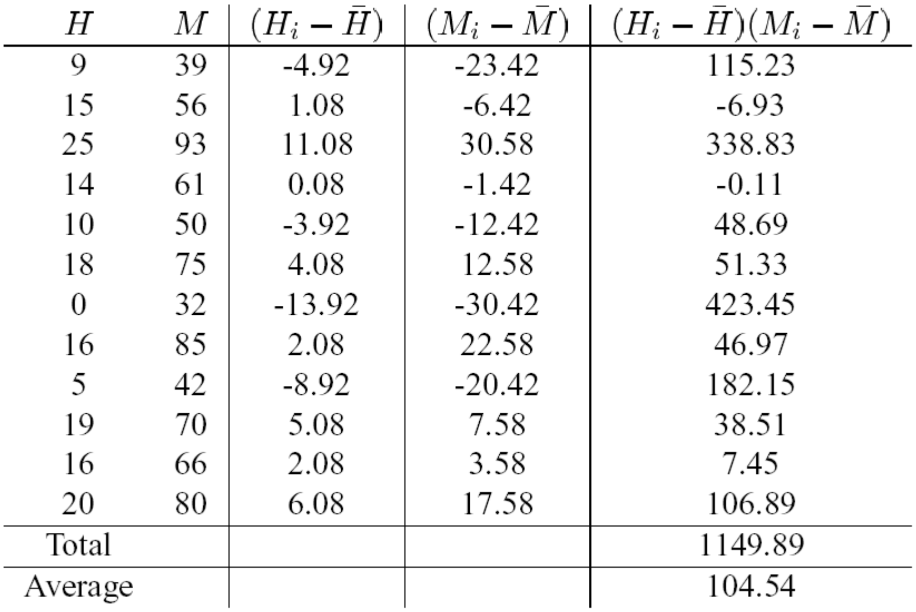
The result that mark given increases as the number of hours studied increases can be easily seen by drawing a graph of the data. However, the luxury of being able to visualize data is only available at 2 and 3 dimensions. Since the covariance value can be calculated between any 2 dimensions in a data set, this technique is often used to find relationships between dimensions in high-dimensional data sets where visualization is difficult.

You might ask ‟is cov(X,Y) equal to cov(Y, X) ”? Well, a quick look at the formula for covariance tells us that yes, they are exactly the same since the only difference between cov(X, Y) and cov(Y,X) is that $(X_i - \bar{X})(Y_i - \bar{Y})$ is replaced by $(Y_i - \bar{Y})(X_i - \bar{X})$. And since multiplication is commutative, which means that it doesn’t matter which way around we multiply two numbers, we always get the same number, these two equations give the same answer.



	Hours(H)	*Mark(M)*
Data	9	39
	15	56
	25	93
	14	61
	10	50
	18	75
	0	32
	16	85
	5	42
	19	70
	16	66
	20	80
Totals	167	749
Averages	13.92	62.42

Covariance:



H	M	$(H_i - \bar{H})$	$(M_i - \bar{M})$	$(H_i - \bar{H})(M_i - \bar{M})$
9	39	-4.92	-23.42	115.23
15	56	1.08	-6.42	-6.93
25	93	11.08	30.58	338.83
14	61	0.08	-1.42	-0.11
10	50	-3.92	-12.42	48.69
18	75	4.08	12.58	51.33
0	32	-13.92	-30.42	423.45
16	85	2.08	22.58	46.97
5	42	-8.92	-20.42	182.15
19	70	5.08	7.58	38.51
16	66	2.08	3.58	7.45
20	80	6.08	17.58	106.89
Total				1149.89
Average				104.54

Table 3.2: 2-dimensional data set and covariance calculation

The covariance Matrix

Recall that covariance is always measured between 2 dimensions. If we have a data set with more than 2 dimensions, there is more than one covariance measurement that can be calculated. For example, from a 3 dimensional data set (dimensions x,y,z) you could calculate cov(x,y), (cov(x,z) and cov(y,z)). In fact, for an n dimensional data set, you can calculate $\frac{n!}{(n-2)!*2}$ different covariance values.

A useful way to get all the possible covariance values between all the different dimensions is to calculate them all and put them in a matrix. We assume in this tutorial that you are familiar with matrices, and how they can be defined. So, the definition for the covariance matrix for a set of data with n dimensions is:

$$C^{n \times n} = (c_{i,j},\ c_{i,j} = cov(Dim_i, Dim_j)),$$

where $C^{n \times n}$ is a matrix with n rows and n columns, and Dim_x is the x^{th} dimension. All that this ugly looking formula says is that if you have an n dimensional data set, then the matrix has n rows and columns (so is square) and each entry in the matrix is the result of calculating the covariance between two separate dimensions. Eg. the entry on row 2, column 3, is the covariance value calculated between the 2nd dimension and the 3rd dimension.

An example. We'll make up the covariance matrix for an imaginary 3 dimensional data set, using the usual dimensions x, y and z. Then, the covariance matrix has 3 rows and 3 columns, and the values are this:

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

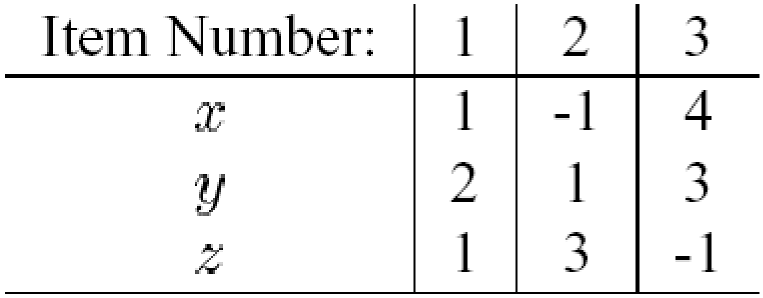
Some points to note: Down the main diagonal, you see that the covariance value is between one of the dimensions and itself. These are the variances for that dimension. The other point is that since cov(a,b) = cov(b,a), the matrix is symmetrical about the main diagonal.

Example

Work out the covariance between the x and y dimensions in the following 2 dimensional data set, and describe what the result indicates about the data.

Item Number:	1	2	3	4	5
x	10	39	19	23	28
y	43	13	32	21	20

Calculate the covariance matrix for this 3 dimensional set of data.



Item Number:	1	2	3
x	1	-1	4
y	2	1	3
z	1	3	-1

Matrix Algebra

This section serves to provide a background for the matrix algebra required in PCA. Specifically We will be looking at eigenvectors and eigenvalues of a given matrix. Again, we assume a basic knowledge of matrices.

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 11 \\ 5 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Figure 3.9: Example of one non-eigenvector and one eigenvector

$$2 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 6 \\ 4 \end{pmatrix} = \begin{pmatrix} 24 \\ 16 \end{pmatrix} = 4 \times \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

Figure 3.10: Example of how a scaled eigenvector is still and eigenvector

Eigenvectors

As you know, we can multiply two matrices together, provided they are compatible sizes. Eigenvectors are a special case of this. Consider the two multiplications between a matrix and a vector in Figure 3.9

In the first example, the resulting vector is not an integer multiple of the original vector, whereas in the second example, the example is exactly 4 times the vector we began with. Why is this? Well, the vector is a vector in 2 dimensional spaces. The vector $\begin{pmatrix}3\\2\end{pmatrix}$ (from the second example multiplication) represents an arrow pointing from the origin, (0,0), to the point (3,2). The other matrix, the square one, can be thought of as a transformation matrix. If you multiply this matrix on the left of a vector, the answer is another vector that is transformed from it's original position.

It is the nature of the transformation that the eigenvectors arise from. Imagine a transformation matrix that, when multiplied on the left, reflected vectors in the line y = x. Then you can see that if there were a vector that lay *on* the line y = x, it's reflection it *itself*. This vector (and all multiples of it, because it wouldn't matter how long the vector was) would be an eigenvector of that transformation matrix.

What properties do these eigenvectors have? You should first know that eigenvectors could only be found for *square* matrices. And, not every square matrix has eigenvectors. And, given an n x n matrix that does have eigenvectors, there are n of them. Given a 3 x 3 matrix, there are 3 eigenvectors.

Another property of eigenvectors is that even if I scale the vector by some amount before I multiply it, I still get the same multiple of it as a result, as in Figure 3.10. This is because if you scale a vector by some amount, all you are doing is making it longer, not changing it's direction. Lastly, all the eigenvectors of a matrix are *perpendicular*, ie. at right angles to each other, no matter how many dimensions you have. By the way, another word for perpendicular, in maths talk, is orthogonal. This is important because it means that you can express the data in terms of these perpendicular eigenvectors, instead of expressing them in terms of the X and Y-axes. We will be doing this later in the section on PCA.

Another important thing to know is that when mathematicians find eigenvectors, they like to find the eigenvectors whose length is exactly one. This is because, as you know, the length of a vector doesn't affect whether it's an eigenvector or not, whereas the direction does. So, in order to keep eigenvectors standard, whenever we find an eigenvector we usually scale it to make it have a length of 1, so that all eigenvectors have the same length. Here's a demonstration from our example above.

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

is an eigenvector, and the length of that vector is

$$\sqrt{(3^2 + 2^2)} = \sqrt{13}$$

so we divide the original vector by this much to make it have a length of 1.

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix} \div \sqrt{13} = \begin{pmatrix} 3/\sqrt{13} \\ 2/\sqrt{13} \end{pmatrix}$$

How does one go about finding these mystical eigenvectors? Unfortunately, it’s only easy(ish) if you have a rather small matrix, like no bigger than about 3 x 3. After that, the usual way to find the eigenvectors is by some complicated iterative method.

Eigenvalues

Eigenvalues are closely related to eigenvectors, in fact, we saw an eigenvalue in Figure 3.10. Notice how, in both those examples, the amount by which the original vector was scaled after multiplication by the square matrix was the same? In that example, the value was 4. 4 is the *eigenvalue* associated with that eigenvector. No matter what multiple of the eigenvector we took before we multiplied it by the square matrix, we would always get 4 times the scaled vector as our result.

So you can see that eigenvectors and eigenvalues always come in pairs. When you get a fancy programming library to calculate your eigenvectors for you, you usually get the eigenvalues as well.

Method of Principal Components Analysis

Finally we come to Principal Components Analysis (PCA). What is it? It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data.

The other main advantage of PCA is that once you have found these patterns in the data, and you compress the data, i.e. by reducing the number of dimensions, without much loss of information. This technique used in image compression, as we will see in a later section.

This chapter will take you through the steps you needed to perform a Principal Components Analysis on a set of data. I am not going to describe exactly *why* the technique works, but I will try to provide an explanation of what is happening at each point so that you can make informed decisions when you try to use this technique yourself.

Method

Step 1: Get some data

In this simple example, we are going to use our own made-up data set. It's only got 2 dimensions, and the reason why we have chosen this is so that we can provide plots of the data to show what the PCA analysis is doing at each step. The data we have used is found in Figure 3.11, along with a plot of that data.

Step 2: Subtract the mean

For PCA to work properly, you have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. So, all the X values have $\bar{x}$ (the mean of the x values of all the data points) subtracted, and all the Y values have $\bar{y}$ subtracted from them. This produces a data set whose mean is zero.

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

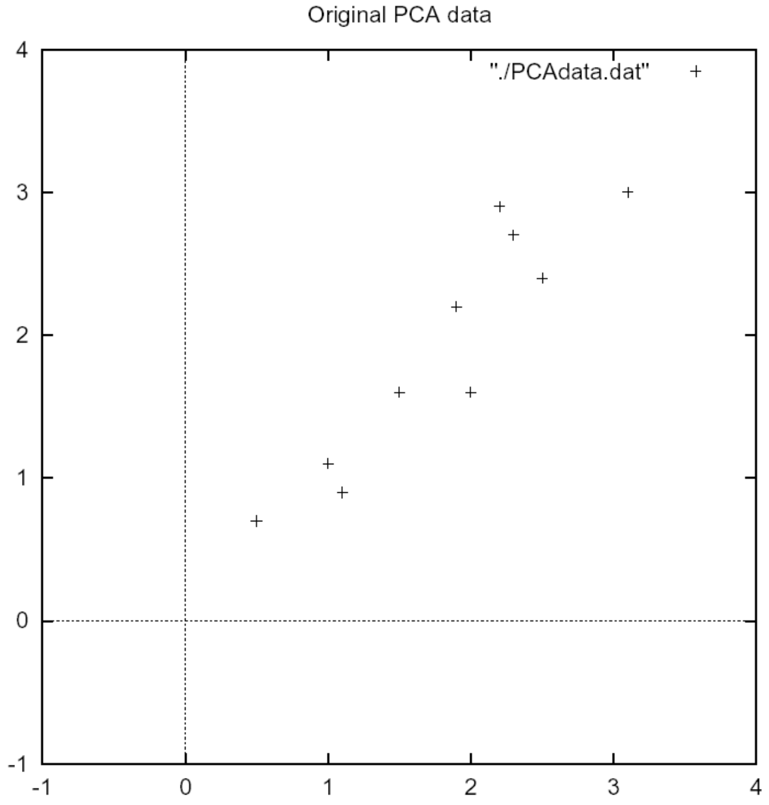


Figure 3.11: PCA example data, original data on the left, data with the means subtracted on the right, and a plot of the data

Step 3: Calculate the covariance matrix

This is done in exactly the same way as was discussed in section The covariance Matrix. Since the data is 2 dimensional, the covariance matrix will be 2 x 2. There are no surprises here, so we have just the result:

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}.$$

So, since the non-diagonal elements in this covariance matrix are positive, we should expect that both the x and y variable increase together.

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Since the covariance matrix is square, we can calculate the eigenvectors and eigenvalues for this matrix. These are rather important, as they tell us useful information about our data. I will show you why soon. In the meantime, here are the eigenvectors and eigenvalues:

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}.$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}.$$

It is important to notice that these eigenvectors are both *unit* eigenvectors i.e. Their lengths are both 1. This is very important for

PCA, but luckily, most maths packages, when asked for eigenvectors, will give you unit eigenvectors.

So what do they mean? If you look at the plot of the data in Figure 3.12 then you can see how the data has quite a strong pattern. As expected from the covariance matrix, they two variables do indeed increase together. On top of the data I have plotted both the eigenvectors as well. They appear as diagonal dotted lines on the plot. As stated in the eigenvector section, they are perpendicular to each other. But, more importantly, they provide us with information about the patterns in the data. See how one of the eigenvectors goes through the middle of the points, like drawing a line of best fit? That eigenvector is showing us how these two data sets are related along that line. The second eigenvector gives us the other, less important, pattern in the data, that all the points follow the main line, but are off to the side of the main line by some amount.

So, by this process of taking the eigenvectors of the covariance matrix, we have been able to extract lines that characterize the data. The rest of the steps involve transforming the data so that it is expressed in terms of them lines.

Step 5: Choosing components and forming a feature vector

Here is where the notion of data compression and reduced dimensionality comes into it. If you look at the eigenvectors and eigenvalues from the previous section, you will notice that the eigenvalues are quite different values. In fact, it turns out that the eigenvector with the highest eigenvalue is the principle component of the data set. In our example, the eigenvector with the larges

152eigenvalue was the one that pointed down the middle of the data. It is the most significant relationship between the data dimensions.

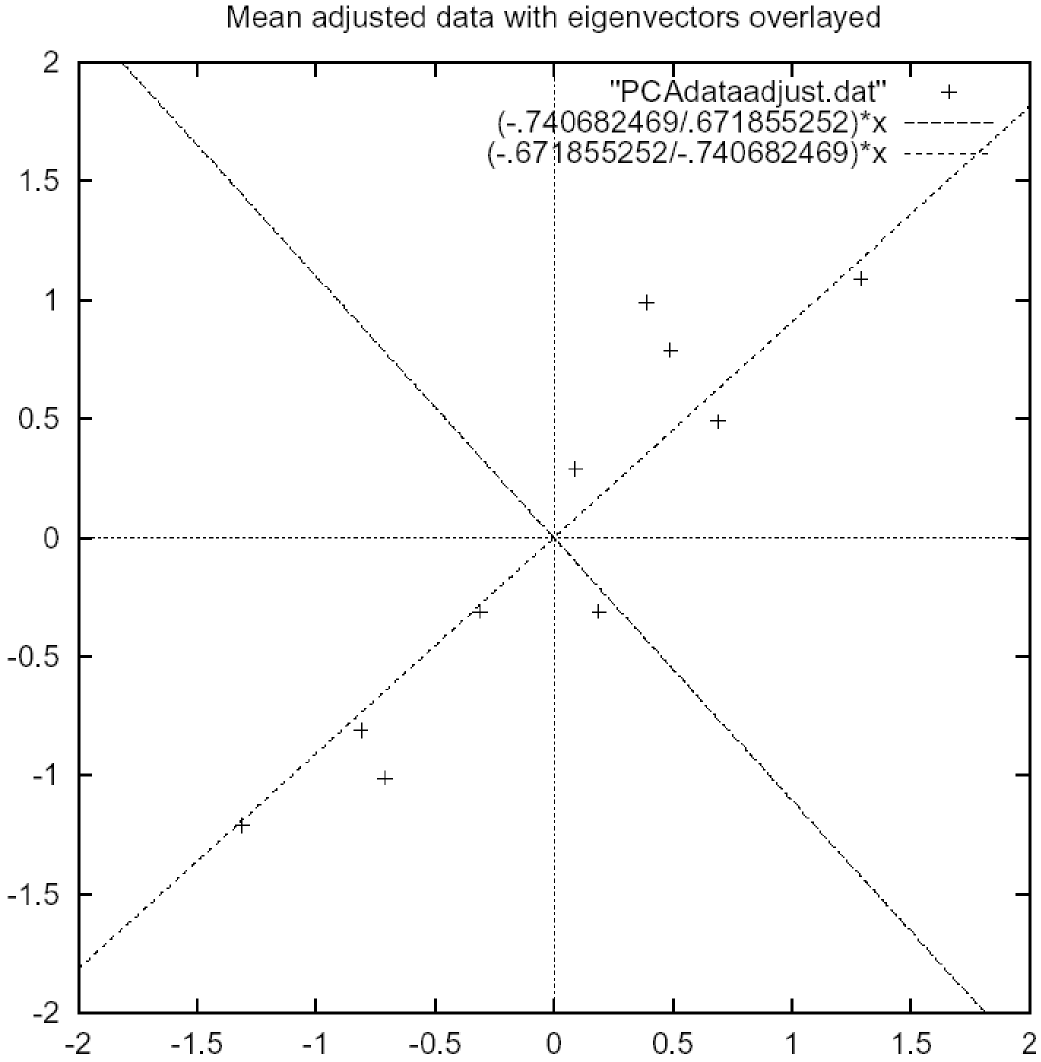


Figure 3.12: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

In general, once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives you the components in order of significance. Now, if you like, you can decide to *ignore* the components of lesser significance. You do lose some information, but if the eigenvalues are small, you don't lose much. If you leave out some components, the final data set will have less dimensions than the original. To be precise, if you originally have n dimensions in your data, and so you calculate n eigenvectors and eigenvalues, and then you choose only the first P eigenvectors, then the final data set has only P dimensions.

What needs to be done now is you need to form a *feature vector*, which is just a fancy name for a matrix of vectors. Taking the eigenvectors that you want to keep from the list of eigenvectors, and forming a matrix with these eigenvectors in the columns construct this.

$$FeatureVector = (eig_1\ eig_2\ eig_3\ ....\ eig_n)$$

Given our example set of data, and the fact that we have 2 eigenvectors, we have two choices. We can either form a feature vector with both of the eigenvectors

$$\begin{pmatrix} -.677873399 & -.735178656 \\ -.735178656 & .677873399 \end{pmatrix}$$

or, we can choose to leave out the smaller, less significant component and only have a single column:

$$\begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

Step 6: Deriving the new data set

This final step in PCA, and is also the easiest. Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the original data set, transposed.

Final Data = RowFeatureVector X RowDataAdjust

Where RowFeatureVector is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and RowDataAdjust is the mean-adjusted data *transposed*, i.e. the data items are in each column, with each row holding a separate dimension. I'm sorry if this sudden transpose of all our data confuses you, but the equations from here on are easier if we take the transpose of the feature vector and the data first, rather that having a little T symbol above their names from now on. Final Data is the final data set, with data items in columns, and dimensions along rows.

What will this give us? It will give us the original data *solely in terms of the vectors we chose*. Our original data set had two axes, x and y, so our data was in terms of them. It is possible to express data in terms of any two axes that you like. If these axes are perpendicular, then the expression is the most efficient. This was why it was important that eigenvectors are always perpendicular to each other. We have changed our data from being in terms of the axes x and y, and now they are in terms of our 2 eigenvectors. In the case of when the new data set has reduced dimensionality, i.e. We have left some of the eigenvectors out, the new data is only in terms of the vectors that we decided to keep.

To show this on our data, I have done the final transformation with each of the possible feature vectors. I have taken the transpose of the result in each case to bring the data back to the nice table-like format. I have also plotted the final points to show how they relate to the components.

In the case of keeping both eigenvectors for the transformation, we get the data and the plot found in Figure 3.13. This plot is basically the original data, rotated so that the eigenvectors are the axes. This is understandable since we have lost no information in this decomposition.

The other transformation we can make is by taking only the eigenvector with the largest eigenvalue. The table of data resulting from that is found in Figure 3.14. As expected, it only has a single dimension. If you compare this data set with the one resulting from using both eigenvectors, you will notice that this data set is exactly the first column of the other. So, if you were to plot this data, it would be 1 dimensional, and would be points on a line in exactly the x positions of the points in the plot in Figure 3.13. We have effectively thrown away the whole other axis, which is the other eigenvector.

So what have we done here? Basically we have transformed our data so that is expressed in terms of the patterns between them, where the patterns are the lines that most closely describe the relationships between the data. This is helpful because we have now classified our data point as a combination of the contributions from each of those lines. Initially we had the simple x and y axes. This is fine, but the x and y values of each data point don't really tell us exactly how that point relates to the rest of the data. Now, the values of the data points tell us exactly where (i.e. above/below) the trend lines the data point sits. In the case of the transformation using *both* eigenvectors, we have simply altered the data so that it is in terms of those eigenvectors instead of the usual axes. But the single-eigenvector decomposition has removed the contribution due to the smaller eigenvector and left us with data that is only in terms of the other.

Method Getting the old data back

Wanting to get the original data back is obviously of great concern if you are using the PCA transform for data compression (an example of which to will see in the next section).

So, how do we get the original data back? Before we do that, remember that only if we took *all* the eigenvectors in our transformation will we get *exactly* the original data back. If we have reduced the number of eigenvectors in the final transformation, then the retrieved data has lost some information.

Recall that the final transform is this:

$$\text{Final Data} = \text{RowFeatureVector X RowDataAdjust}$$

which can be turned around so that, to get the original data back,

$\text{RowDataAdjust} = \text{RowFeatureVector}^{(-1)}$ X Final Data

Where $\text{RowFeatureVector}^{(-1)}$ is the inverse of RowFeatureVector.

However, when we take *all* the eigenvectors in our feature vector, it turns out that the inverse of our feature vector is actually equal to the transpose of our feature vector. This is only true because the elements of the matrix are all the unit eigenvectors of our data set. This makes the return trip to our data easier, because the equation becomes

$\text{RowDataAdjust} = \text{RowFeatureVector}^{T}$ X Final Data

But, to get the actual original data back, we need to add on the mean of that original data (remember we subtracted it right at the start). So, for completeness, RowOriginalData = (RowFeatureVector^T X Final Data) + OriginalMean

This formula also applies to when you do not have all the eigenvectors in the feature vector. So even when you leave out some eigenvectors, the above equation still makes the correct transform.

I will not perform the data re-creation using the *complete* feature vector, because the result is exactly the data we started with. However, I will do it with the reduced feature vector to show you how information has been lost. Figure 3.15 show this plot. Compare
it to the original data plot in Figure 3.11 and you will notice how, while the variation along the principle eigenvector (see Figure 3.12 for the eigenvector overlayed on top of the mean-adjusted data) has been kept, the variation along the other component (the other eigenvector that we left out) has gone.

Transformed Data=

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

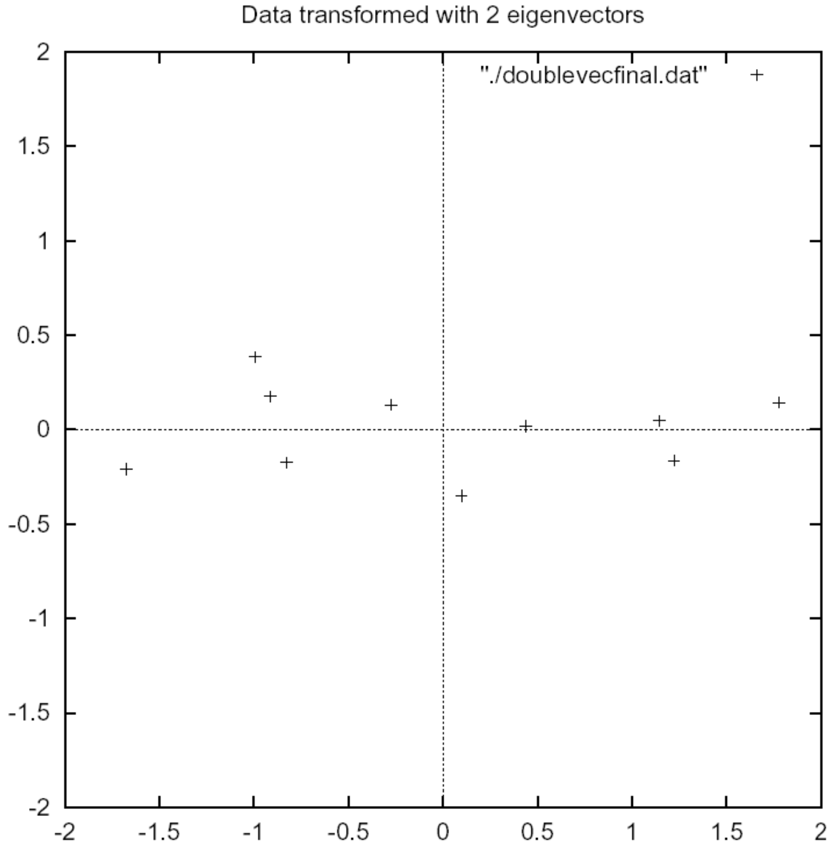
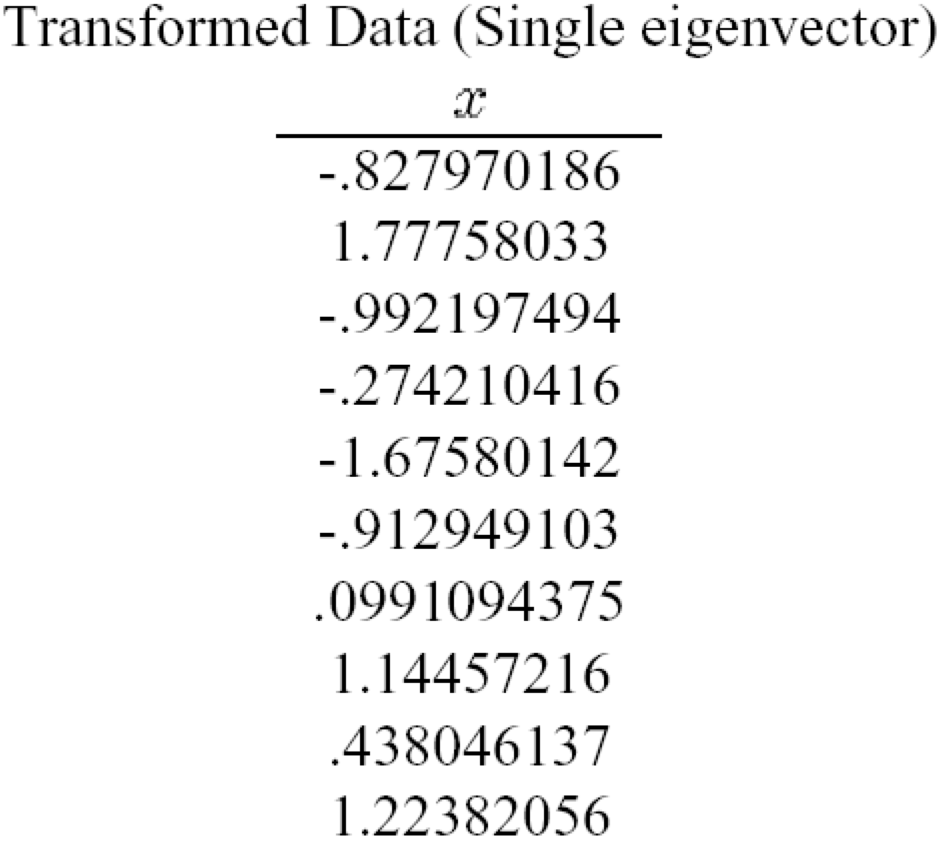


Figure 3.13 : The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.



Transformed Data (Single eigenvector)

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

Figure 3.14: The data after transforming using only the most significant eigenvector

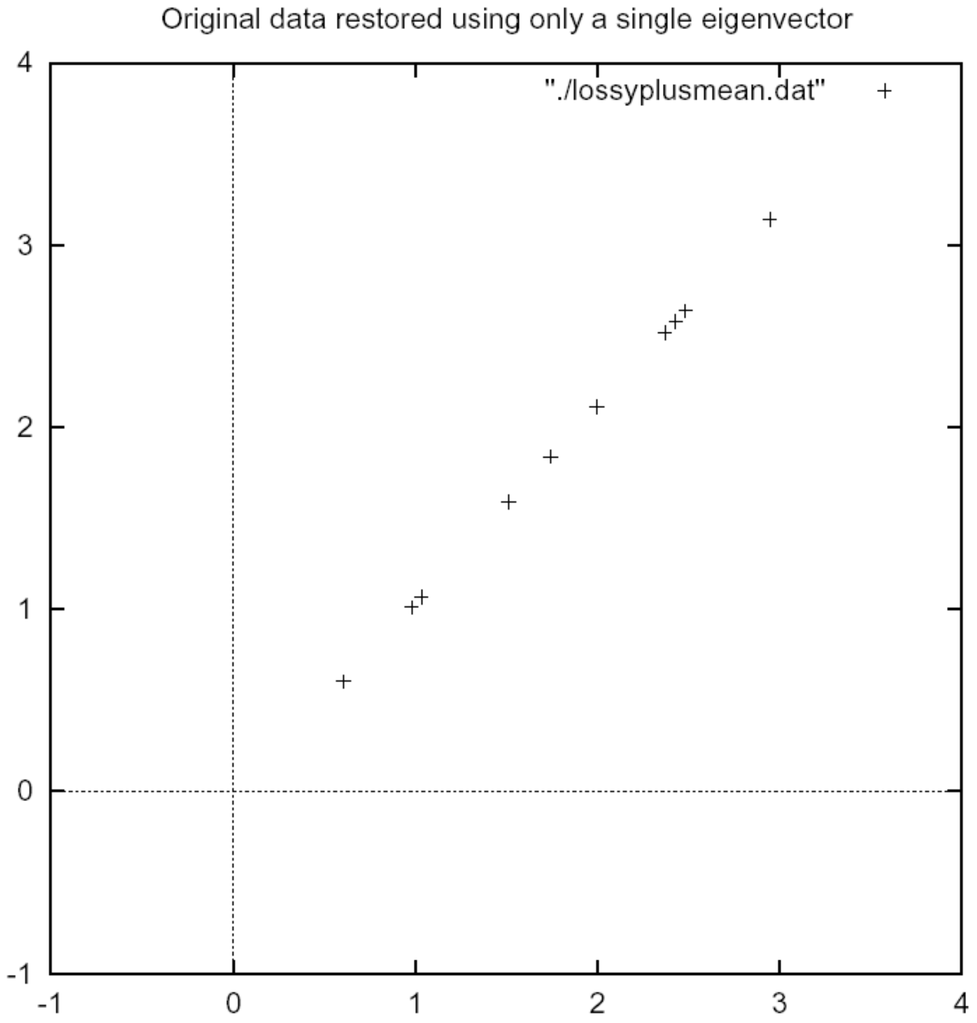


Figure 3.15: The reconstruction from the data that was derived using only a single eigenvector

Application to Computer Vision

This is the outline the way that PCA is used in computer vision, first showing how images are usually represented, and then showing what PCA can allow us to do with those images.

Representation

When using these sorts of matrix techniques in computer vision, we must consider representation of images. A square, N-by-N image can be expressed as an, N^2 -dimensional vector.

$$X = (x_1 \quad x_2 \quad x_3 \quad . \quad . \quad x_{N^2})$$

where the rows of pixels in the image are placed one after the other to form a one dimensional image. E.g. The first N elements (x_1 – x_n) will be the first row of the image, the next N elements are the next row, and so on. The values in the vector are the intensity values of the image, possibly a single greyscale value.

PCA to find patterns

Say we have 20 images. Each image is N pixels high by N pixels wide. For each image we can create an image vector as described in the representation section. We can then put all the images together in one big image-matrix like this:

$$ImagesMatrix = \begin{pmatrix} ImageVec1 \\ ImageVec2 \\ . \\ . \\ ImageVec20 \end{pmatrix}$$

which gives us a starting point for our PCA analysis. Once we have performed PCA, we have our original data in terms of the eigenvectors we found from the covariance matrix. Why is this useful? Say we want to do facial recognition, and so our original images were of people's faces. Then, the problem is, given a new image, whose face from the original set is it? (Note that the new image is not one of the 20 we started with.) The way this is done is computer vision is to measure the difference between the new image and the original images, but not along the original axes, along the new axes derived from the PCA analysis.

It turns out that these axes works much better for recognizing faces, because the PCA analysis has given us the original images in terms of the differences and similarities between them. The PCA analysis has identified the statistical patterns in the data.

Since all the vectors are N^2 dimensional, we will get N^2 eigenvectors. In practice, we are able to leave out some of the less significant eigenvectors, and the recognition still performs well.

PCA for image compression

Using PCA for image compression also knows as the Hotelling, or Karhunen and Leove (KL), transform. If we have 20 images, each with N^2 pixels, we can perform N^2 vectors, each with 20 dimensions. Each vector consists of all the intensity values from the *same* pixel from each picture. This is different from the previous example because before we had a vector for *image*, and each item in that vector was a different pixel, whereas now we have a vector for each *pixel*, and each item in the vector is from a different image.

Now we perform the PCA on this set of data. We will get 20 eigenvectors because each vector is 20-dimensional. To compress the data, we can then choose to transform the data only using, say 15 of the eigenvectors. This gives us a final data set with only 15 dimensions, which has saved us ¼ of the space. However, when the original data is reproduced, the images have lost some of the information. This compression technique is said to be *lossy* because the decompressed image is not exactly the same as the original, generally worse.

3.7.2 Feature Analysis

Feature analysis is perhaps the most widely utilized facial recognition technology. This technology is related to Eigenface, but is more capable of accommodating changes in appearance or facial aspect (smiling versus frowning, for example). Visionics, a prominent facial recognition company, uses Local Feature Analysis (LFA), which can be summarized as a reduction of facial features to an "irreducible set of building elements."

Feature analysis derives enrollment and verification templates from dozens of features from different regions of the face and also incorporates the relative location of these features. The extracted features are building blocks, and both the type of blocks and their arrangement are used for identification and verification. It anticipates that relatively similar movement of adjacent features will accompany the slight movement of a feature located near one's mouth. Since feature analysis is not a global representation of the face, it can accommodate angles up to approximately 15 degrees in the vertical plane. A straight-ahead facial image from distance of 3 feet will be the most accurate.

3.7.3 Neural Network

Neural network systems employ algorithms to determine the similarity of the unique global features of live versus enrolled or reference faces, using as much of the facial images as possible. Neural systems are designed to learn which features are most effective within the body of users that the system is intended to serve. Features from both the enrollment and the verification faces vote on whether there is a match. An incorrect vote, such as false match, prompts the matching algorithm to modify the weight it gives to certain facial features. In this way, neural network systems learn which features are most effective for matching and pragmatically adjust themselves based on the methods that have proven most effective. This method, theoretically, leads to an increased ability to identify faces in difficult conditions.

Other facial technologies have emerged based on more advanced neural models, with detailed cells incorporating thousands of facial images. Since these technologies are capable of learning over time, they may be capable of reducing the time-based performance problems found in many facial-scan systems. However, their extended enrollment process means that they are not well-suited for surveillance applications in which users are matched against watch lists. These watch lists are often generated from static images, not the ideal environment for neural net enrollment.

3.7.4 Automatic Face Processing

Automatic face processing (AFP) is a more rudimentary technology, using distances and distances ratios between easily acquired features such as eyes, end of nose, and corners of mouth. Though overall not as robust as Eigenfaces, feature analysis, or neural network, AFP may be more effective in dimly lit, frontal image-capture situations. It is often used in booking situation applications in which environmental conditions are more controlled.

References:

- Handbook of face recognition, Springer
- Biometrics – Identity Verification in a Network World Wiley Tech Publication, (INDIA)
- Intelligent Biometric Techniques in Fingerprint & Face Recognition, The CRC Press
- www.face-rec.org
- en.wikipedia.org
- openbio.sourceforge.net
- www.cs.otago.ac.nz

Chapter - 4

Multimodal Biometric Authentication System

4.1 Introduction

4.2 Fingerprint based Identification System

4.3 Face Recognition System

4.1 Introduction

Biometric systems have to run with noisy data, and failure to enroll problems, spoof attacks, and unacceptable error rates. In some situations, it may be feasible to install a biometric system that takes advantage of more than one method of identification or authentication to overcome these problems. A biometric device can either be integrated with non-biometric forms of authentication or with other forms of biometric authentication devices. When a biometric device is integrated with other forms of biometric authentication devices, it can be described as a "multi-biometric system". Multi-biometric systems may be more reliable and provide higher verification rates due to the presence of multiple, independent pieces of evidence. Multi-biometric systems address the problem of non-universality, since multiple traits ensure sufficient population coverage, and provide anti-spoofing measures by making it difficult for an intruder to steal multiple biometric traits of a genuine user.

The Problems with Unimodality

The shortcoming of unimodal biometrics is that no one technology is suitable for all applications. Therefore, the presence of a multimodal biometric system helps compensate for the following limitations:

- The usage of certain biometrics makes it susceptible to noisy or bad data, such as inability of a scanner to read dirty fingerprints clearly. This can lead to inaccurate matching, as bad data may lead to a false rejection.
- Unimodal biometrics is also prone to inter-class similarities within large population groups. In case of identical twins, a

facial recognition camera may not be able to distinguish between the two.

- Some biometric technologies are incompatible with a certain subset of the population. Elderly people and young children may have difficulty enrolling in a fingerprinting system, due to their faded prints or underdeveloped fingerprint ridges
- Finally, unimodal biometrics are vulnerable to spoofing, where the data can be imitated or forged.

If there is a weakness in one method of biometrics, then combining it with a biometric method that is stronger with respect to that weakness will improve that problem. For example, it may be feasible to deploy a biometric system in an organization that consists of both fingerprint scanning and face recognition devices. In addition, a multi-biometric system may reduce the false reject rate and the failure to enroll problem.

We must determine the logic used by a multi-biometrics system. Each individual biometric method must be incorporated to logically work with the other biometric method that it is being combined with. The logic of the multi-biometric system may be implemented in an AND configuration or in an OR configuration.

If these two devices must work together to provide continuous authentication using the AND configuration, then they both must output a matching score. It is noted that this type of configuration will reduce the false acceptances achieved by using either device by itself, but it will increase the number of false rejections.

It is possible that these systems may be combined in an OR configuration. In the OR configuration, either device will be able to provide the continuous authentication needed in the organization. If the OR configuration is used then this type of configuration will reduce the number of false rejections, but increase the number of false acceptances. The number of false rejections and false acceptances are based on the matching threshold that the administrators set the device at initially. The matching threshold is used to decide between a genuine user and an impostor.

Usually vendors of biometric devices have suggestions for setting threshold values according to the security level you are trying to achieve. The security level may be labeled as low, medium, and high. Each security level has a threshold value associated with it as well. System performance can be improved by providing separate threshold

values for each user of the system. it is shown that by providing separate threshold values for each user of the system, which consists of a combination of fingerprint, face, and hand geometry, the genuine accept rate is above 96%.

Multimodality is the usage of more than one physiological or behavioral characteristic to identify an individual. It involves the fusion of two or more technologies such as fingerprint, facial recognition, iris scanning, hand geometry, signature verification, or speech recognition.

The fusion is done by running the two (or more) biometric inputs against two (or more) different algorithms, to arrive at a decision. This technique is useful in large-scale civil ID applications, where the identity of thousands of people need to be authenticated at a time. To have an additional method of verification as a backup reduces the possibility of inconveniences caused by the malfunctioning of the primary biometric.

Using multiple biometrics in a system may not be the best solution in some cases. An example is given where fingerprints and voice were used together as one system. The conclusion from this study is that a strong biometric is better alone than in combination with a weaker one. More analysis and testing of multi-biometric systems is needed in order to be able to draw clear conclusions regarding the implementation of such a system.

A multi-biometric system may increase the certainty that the person is who he claims to be and increases the flexibility and circumstances under which someone can be verified. The accuracy and performance of an authentication system may be increased by employing a multi-biometric system if the most compatible methods are combined together to produce a stronger biometric system (i.e. where weaknesses in one method are complemented by the strengths in the other method). If the results of combining different biometric methods are not fully researched, then it is possible that a layered biometric system may be weaker than using only one method.

Information Fusion in Multimodal Biometrics

Multimodal biometric systems can be classified based on four parameters.

(i) Architecture
(ii) Sources that provide multiple evidence
(iii) Level of fusion
(iv) Methodology used for integrating the multiple cues

Generally, these design decisions depend on the application scenario and these choices have a deep influence on the performance of a multimodal biometric system.

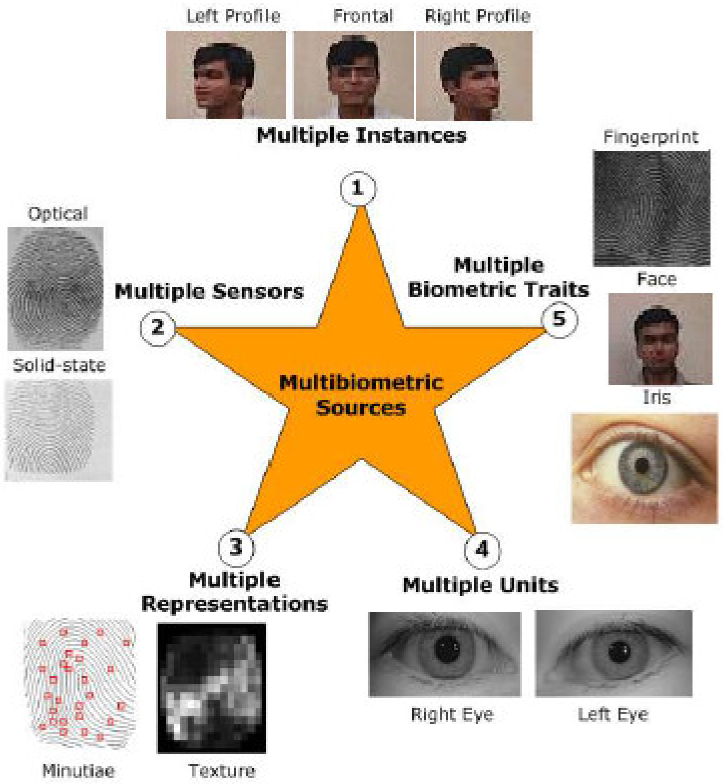


Figure 4.1: Sources of multiple evidence in multimodal biometric systems.

In the first four scenarios, multiple sources of information are derived from the same biometric trait. In the fifth scenario, information is derived from different biometric traits.

Advantages of Multimodality

It is estimated that approximately 5 percent of any population has unreadable fingerprints, either due to scars or aging or illegible prints. In a civil ID scenario, where millions of people have to be enrolled in the system, the segment of the population who are un-enrolable will face inconveniences. Having multimodal biometric technology can overcome this restriction and ensure lower failure to enroll rate (FTE).

Multimodality can also address the problem of aversion to fingerprinting, found in certain parts of the world. Sometimes people associate fingerprints with criminal activity, and are reluctant to submit prints. By having an additional biometric available, a greater number of people can be enrolled into the system

Using multiple biometrics solves the problem of inter-class similarity and the resultant high false acceptance rate (FAR). If people with similar hand sizes or similar looking facial features can gain false acceptance, the presence of another biometric such as signature verification can distinguish between the samples.

Another advantage of using multimodality is that it solves the problem of data distortion. If the quality of one of the biometric samples is unacceptable, the other can make up for it. If a fingerprint has been scarred and the scanner rejects the distorted sample, having another modality like facial recognition can prevent high false rejection rates (FRR).

Unimodal Biometrics can be easily spoofed. Placing a high-resolution picture of a fingerprint under the scanner can deceive some systems. However, by using multiple biometrics, even if one modality could be spoofed, the person would still have to be authenticated using the other biometric. Besides, the effort required for forging two or more biometrics is a deterrent to those who wish to do so.

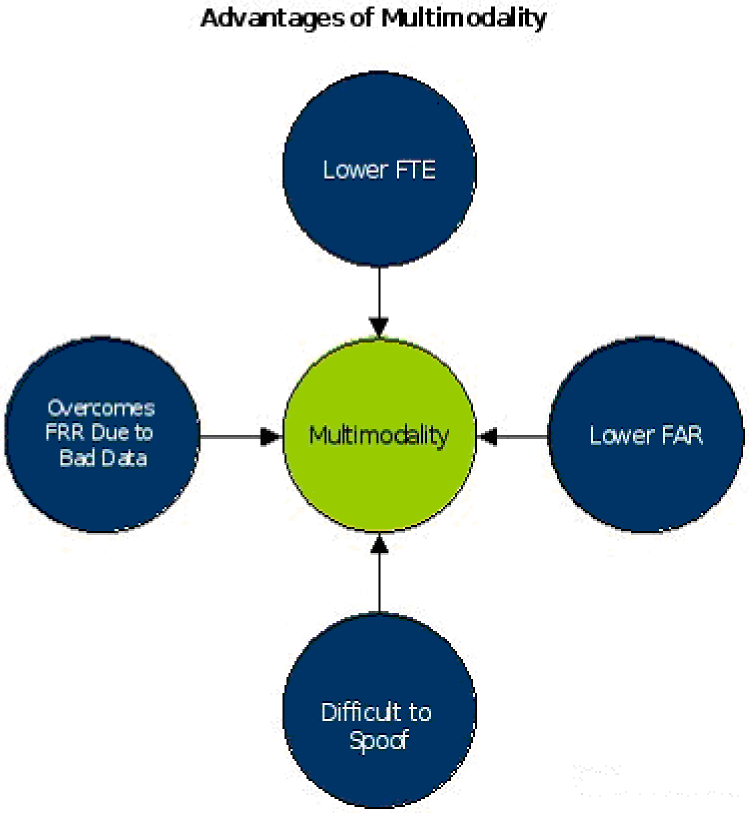
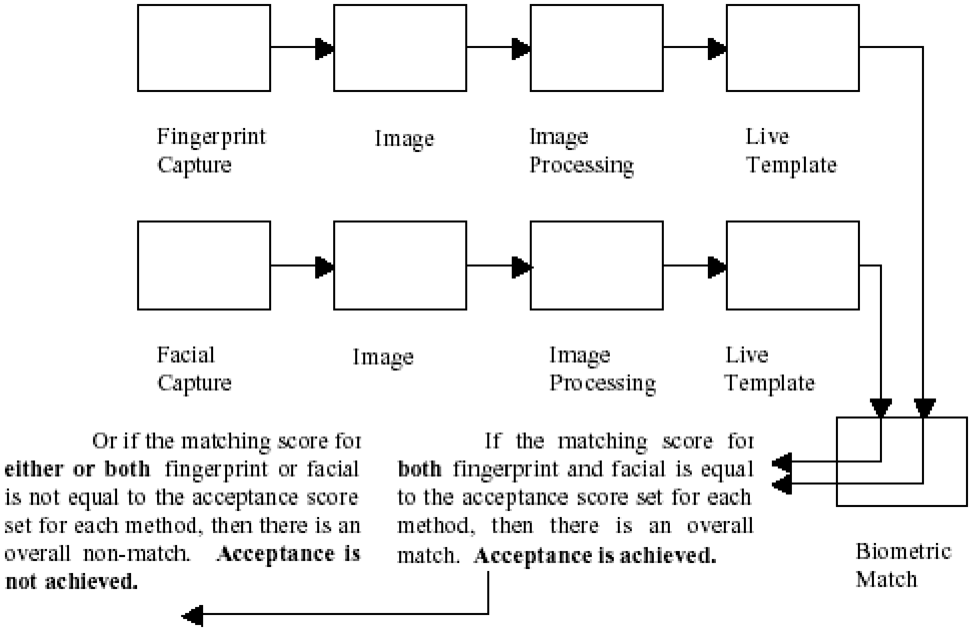


Figure 4.2: Advantages of Multimodality

1. **Multi-Biometric System "AND" Configuration**



Or if the matching score for **either or both** fingerprint or facial is not equal to the acceptance score set for each method, then there is an overall non-match. **Acceptance is not achieved.**

If the matching score for **both** fingerprint and facial is equal to the acceptance score set for each method, then there is an overall match. **Acceptance is achieved.**

Figure 4.3: Multi-Biometric System using the AND configuration

Figure 4.3 depicts a multi-biometric system using the AND configuration. In this configuration, it is necessary that both of the biometric methods achieve a matching score equal to the acceptance score set for the system (which is set up initially). This system would provide high confidence that the person who is introducing their biometric information to the system is who he says he is. Spoofing is more difficult because two biometric characteristics are used. It is possible to set individual biometric thresholds for each method used or to weight one biometric method more than the other throughout the system as a whole.

Some formulas are presented for the false accept and false reject rates in terms of probabilities while using the AND configuration. These error probabilities are denoted as: $PA(FA)$ and $PA(FR)$, where $PA(FA)$ denotes the probability of a false accept while using the AND configuration (PA) and where $PA(FR)$ denotes the probability of a false reject while using the AND configuration (PA).

If the AND configuration is used to combine the two tests 1 and 2, a False Accept can only occur if both tests 1 and 2 produce a False Accept. Thus the combined probability of a False Accept, $PA(FA)$, is the product of its two probabilities for the individual tests:

$PA(FA) = P1(FA)P2(FA)$

This formula indicates that the combined probability of producing a false accept would be lower than either of the methods alone. However, the probability of producing a false reject becomes higher when combining two biometric methods rather than using only one biometric method alone. The formula is:

$PA(FR) = 1-[1-P1(FR)][1-P2\ (FR)] = P1(FR) + P2(FR) - P1(FR)P2(FR)$

This formula shows that the probability of producing a false reject would decrease if one used a single biometric method alone, rather than combining multiple biometric methods, especially if one is considerably stronger than the other. Formulas for the OR configuration are similar except that a false reject can only occur if both biometric methods produce a false reject.

2. Multi-Biometric System “OR” Configuration

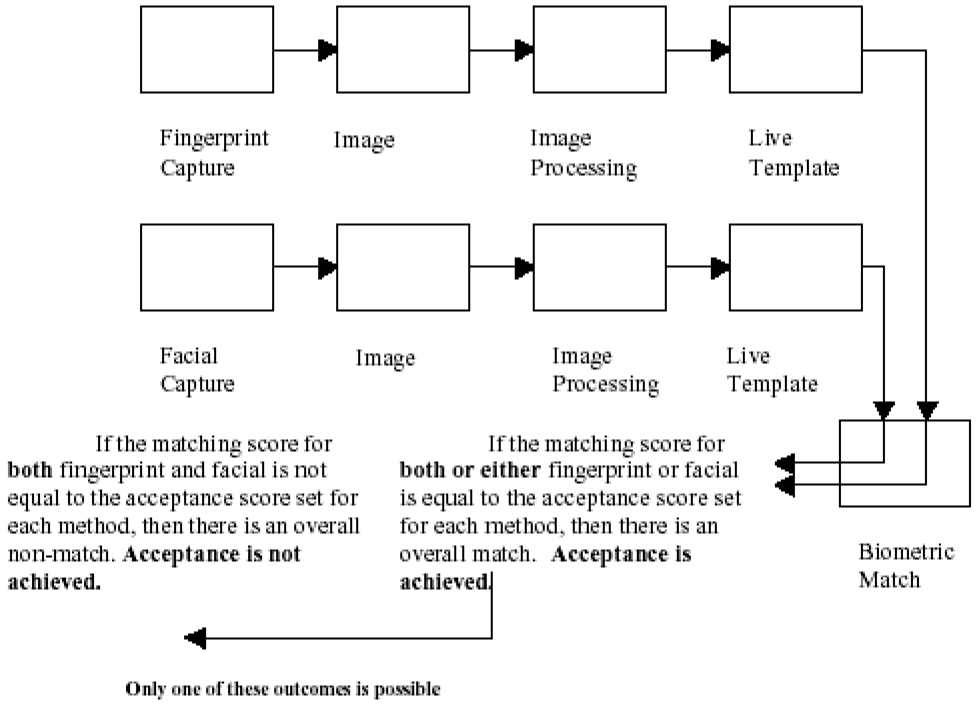

Fingerprint Capture
Image
Image Processing
Live Template
Facial Capture
Image
Image Processing
Live Template
If the matching score for **both** fingerprint and facial is not equal to the acceptance score set for each method, then there is an overall non-match. **Acceptance is not achieved.**
If the matching score for **both or either** fingerprint or facial is equal to the acceptance score set for each method, then there is an overall match. **Acceptance is achieved.**
Biometric Match
Only one of these outcomes is possible

Figure 4.4: Multi-Biometric system using the OR configuration

Figure 4.4 depicts a multi-biometric system using the OR configuration, in this configuration, overall acceptance by the system can be achieved either by both biometric methods possessing a matching score equal to the acceptance score set for the system initially or by either biometric method possessing a matching score equal to the acceptance score set for the system initially. This configuration does not provide the confidence that the person is who they say they are as well as the AND configuration does. This configuration may decrease the false rejection rate overall because the user will be accepted into the system by for example, either their fingerprint template matching the previously stored fingerprint image or by their facial template matching the previously stored facial image or both. Since using this configuration may decrease the false rejection rate, the false acceptance rate will increase, which is not a good idea for highly secured areas.

4.2 Fingerprint Based Identification System

The biometric solutions such as the retina or facial recognition are not so mature and their costs are still too high for a widespread use.

The fingerprint has had a long history of use in police forensic science. Because of this, the authentication by fingerprint is the most convenient biometric element to identify a person. A large variety of solutions are already available and the technology is mature.

With the progress of the technology, the fingerprint is currently to be processed automatically and authenticate a person with a fingerprint reference template. The diversity of applications grows in several fields like the identity card, the driver's license, the security access, etc.

4.2.1 The Fingerprint Features

A fingerprint is composed of valley and ridgelines. They follow a pattern. The general shape of this pattern may be classified according to 5 classes:

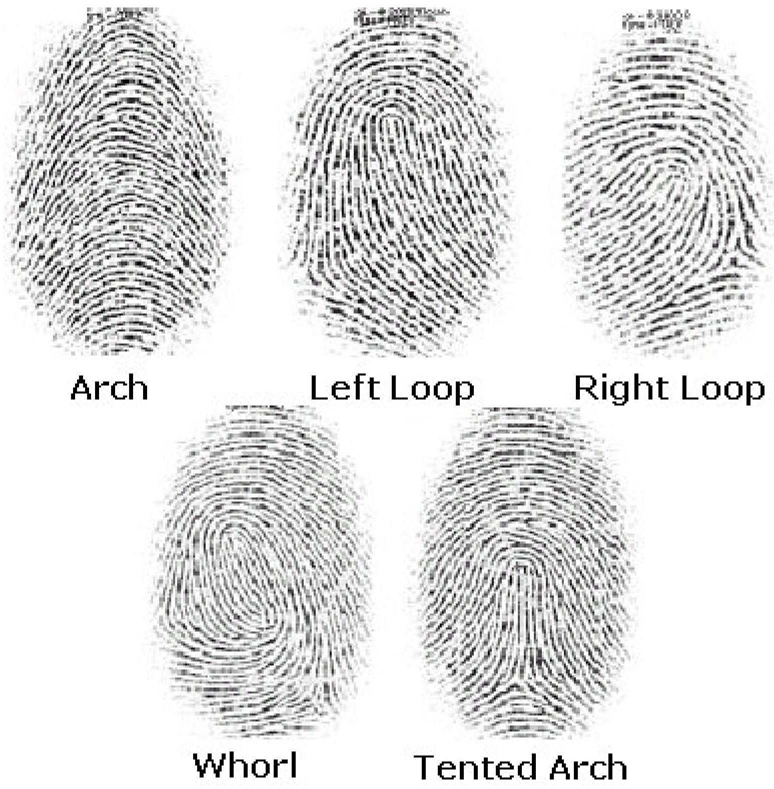
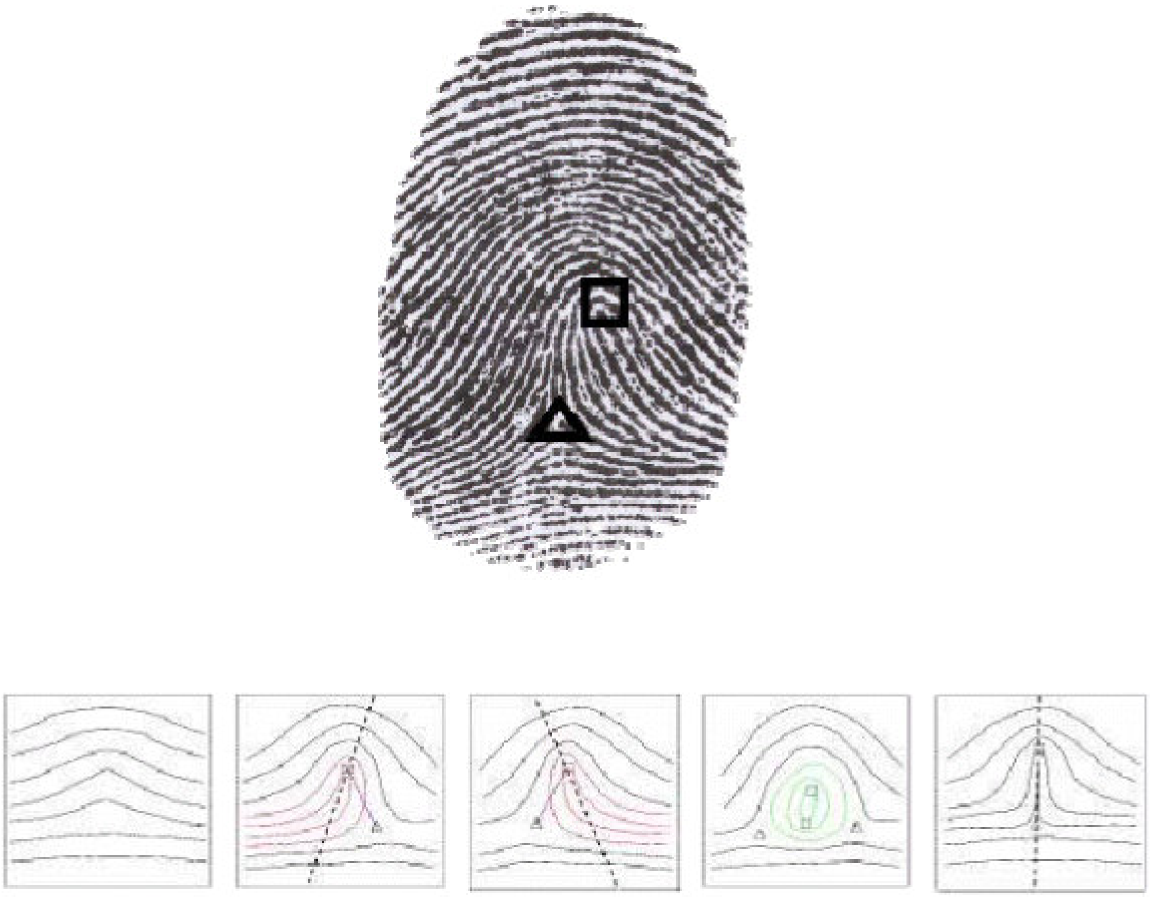


Figure 4.5: The Classes of fingerprint patterns

The second features are the cores and deltas. The core is located by a square while the delta is located by a triangle on the following image. Fingers are then to be sorted in the pattern classification after computing the core and the delta.



Arch

No core
No Delta

Right Loop

One core
Delta at right

Left Loop

One core
Delta at Left

Whorl Loop

Two cores
Two deltas or no delta

Tented Arch

One core
Delta in the Middle

Figure 4.6: Pattern Classification

The features, which give guarantee and uniqueness of a fingerprint, are the minutiae. These points are the ending ridges and the bifurcation when one ridge splits up in two ridges.

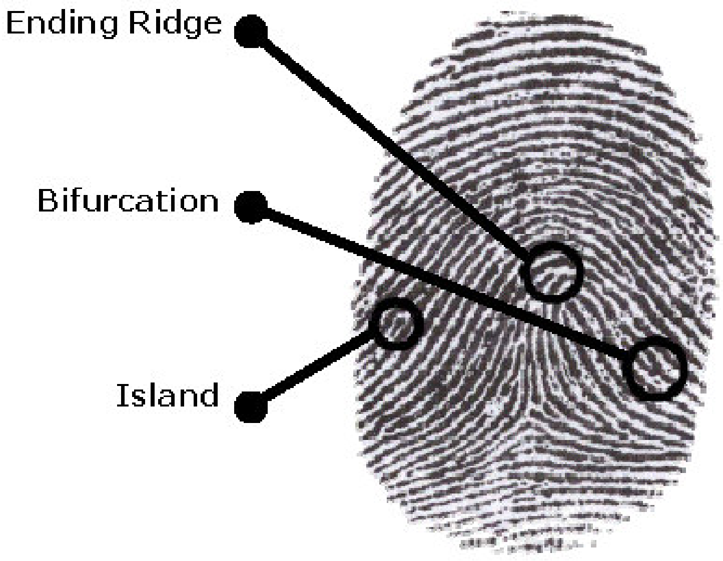


Figure 4.7 : Ridges, Bifurcation and Island

Ridge ending - where a line just stops.
Bifurcation – where a line splits into two.
Enclosure – where the lines make a little island
Island (Ridge dot) – a small dot

The minutiae are characterized by both their X-Y coordinates and the angle of the general direction of the ridge in this point characterize the minutia. Some minutiae are shown on the following fingerprint:

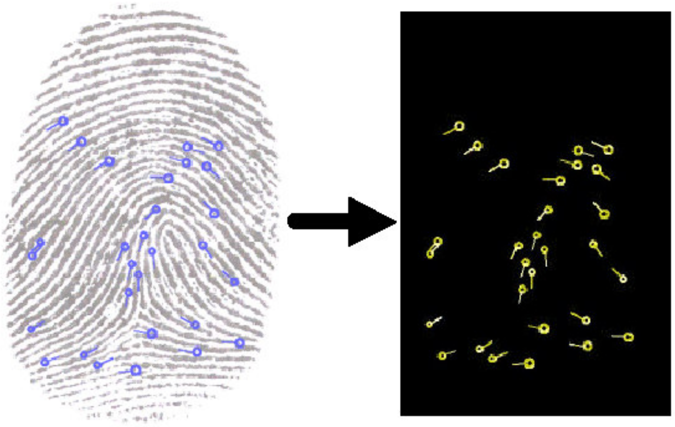
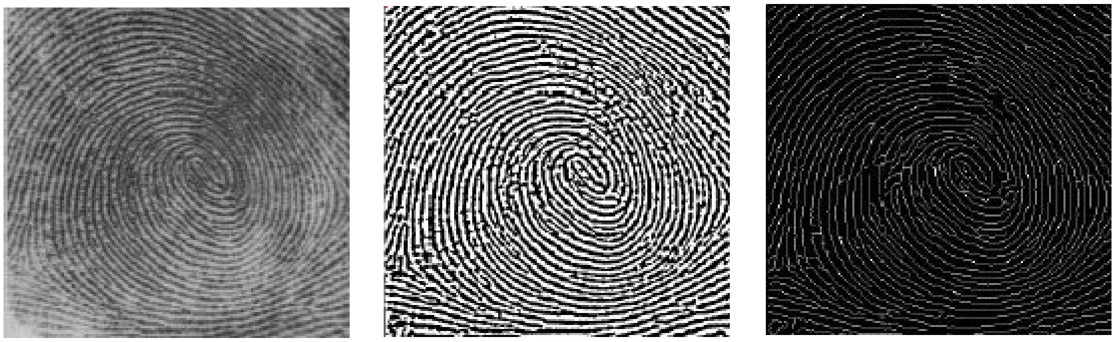


Figure 4.8: Minutiae

This set of minutiae could be the minimum fingerprint template for recognition. In order to increase the performance of this electronic recognition. Each minutia is related to a vector, which describes the frequencies of the ridge, in few directions around the minutia. This vector is used when the number of minutiae is too low. This insures a better matching process. Some other features may be used like the topological configuration between the minutiae, the direction matrix or the general texture vector. These features are used to achieve a better fingerprint classification than the one based on patterns (Arch, Left Loop, Right Loop, whorl, tented arch).

4.2.2 Fingerprint Image Enhancement



Original Fingerprint Image

Enhanced Image

Thinned and Cleaned Ridge map

Figure 4.9: Fingerprint Imaging

When capturing a fingerprint image, the image scan quality can usually significantly affect the performance of an electronic fingerprint system. In order to ensure that the performance of the system will be robust, with respect to the quality of the fingerprint images, it is essential to incorporate a fingerprint enhancement algorithm to filter out image noise and reliably extract ridge and minutiae from the fingerprint image.

Image noise is any condition that prohibits the accurate extraction of ridge and minutiae from the fingerprint image. This noise can come from many conditions, like having dry or wet fingerprints as an example. Dry fingerprints are from the insufficient natural moisture in the skin causing the fingerprint image to appear broken or incomplete.

Wet fingerprints are from the excessive moisture in the skin causing the fingerprint image features to blend together.

Problems with scars, too dry or too moist fingers, or incorrect pressure must also be overcome to get an acceptable image. Therefore, a number of filters, some of which will be described below, are applied to the image.

- **Normalization**

By normalizing an image, the colors of the image are spread evenly throughout the gray scale. A normalized image is much easier to compare with other images, and the quality of the image is easier determined.

- **Binarization**

Making an image binary, transforms the gray scale image into a binary image (black and white). Either a global or localized threshold value is used.

- **Low pass filtering**

The process of low pass filtering smoothens the image to match the pixels nearby so that no points in the image differ from its surroundings to a great extent. By low pass filtering an image, errors and in-correct data are removed, and it simplifies the acquisition process of patterns or minutiae.

- **Quality mark-up**

Redundant information needs to be removed from the image before further analysis can be performed and specific features of the fingerprint can be extracted. Therefore segmentation, i.e. separating the fingerprint image from the background, is needed. Furthermore, any unwanted minutiae (can appear if the print is of bad quality) needs to be removed.

4.2.3 Fingerprint Feature Extraction and Comparison

Many algorithms have been developed to match two different fingerprints and they can be divided into the following groups:

- **Minutia Matching**

Every fingerprint consists of a number of ridges and valleys. Ridges are the upper skin layer segments of the finger and valleys are the lower segments. The ridges form so-called minutia points; ridge endings—where a ridge ends—and ridge bifurcations—where a ridge splits.

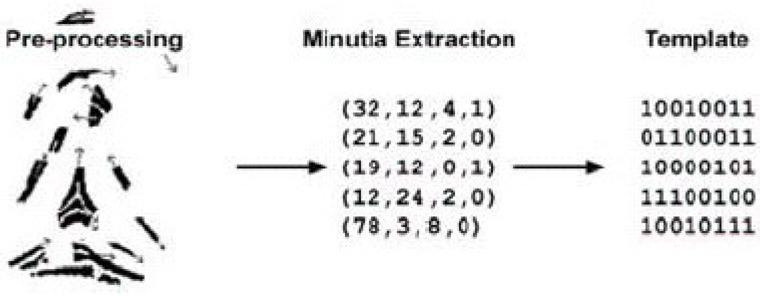


Figure 4.10: Enrolment of minutia points

At registration—enrollment—the minutia points are located and the relative positions to each other and their directions are recorded. This data forms the template, the information later used to authenticate a person. At the matching stage, the incoming fingerprint image is pre-processed and the minutia points are extracted. The minutia points are compared with the registered template, trying to locate as many similar points as possible within a certain boundary. The result of the matching is usually the number of matching minutiae. A threshold is then applied, determining how large this number needs to be for the fingerprint and the template to match.

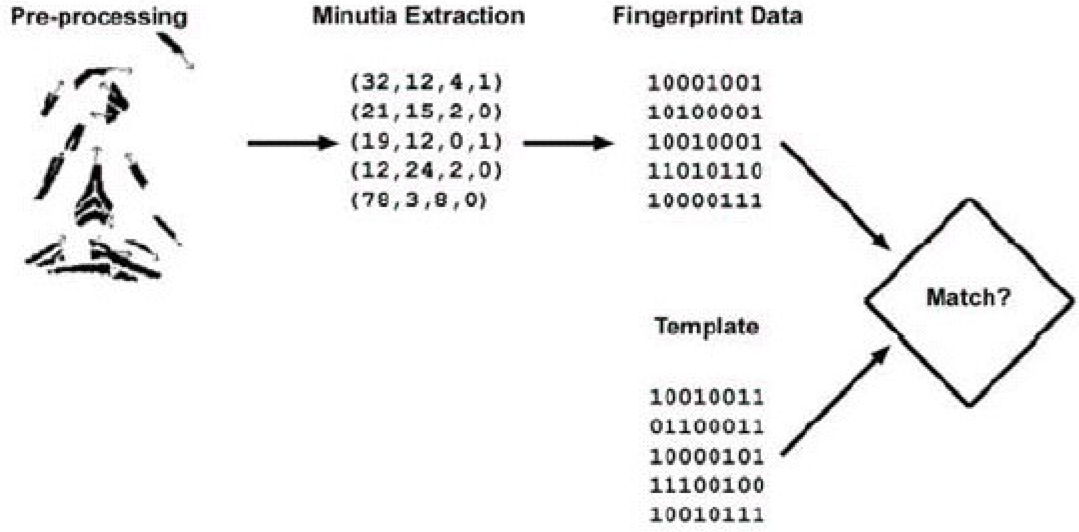


Figure 4.11: Verification using minutia points

- **Pattern Matching**

One intrinsic property of pattern matching algorithms is that overall fingerprint characteristics are taken into account, not only individual points. Fingerprint characteristics can then include sub-areas of certain interest including ridge thickness, curvature, or density. Due to this increased depth of data a pattern-based algorithm is less dependent on the size of the fingerprint sensor and is independent of the number of minutiae points in a fingerprint. Pattern-based algorithms do not, to the same extent as minutia-based methods, suffer from difficulties of recognizing a finger with varying fingerprint quality.

Pattern matching algorithm locates sub-areas of the fingerprint image instead of registering minutia points. Small sections of the fingerprint and their relative distances are extracted from the fingerprint in order to maximize the amount of unique information. Areas of certain interest are for example the area around a minutia point and areas with low curvature radius. The main structure and unusual combinations of ridges are also valuable data.

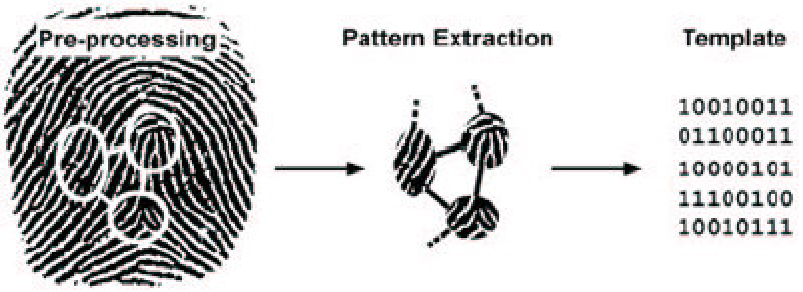


Figure 4.12: Enrolment with pattern-based algorithm

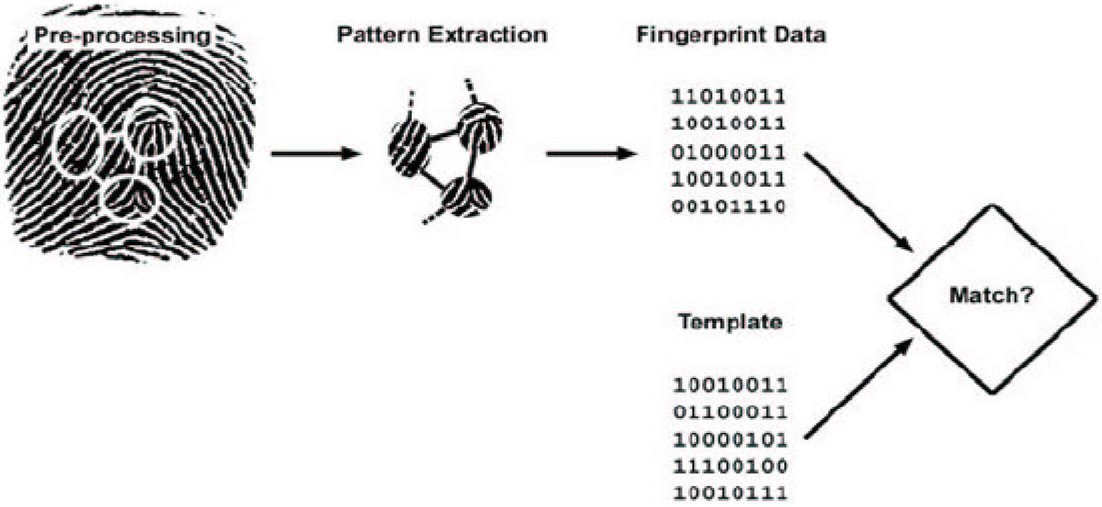


Figure 4.13 : Verification using pattern-based algorithm

The verification procedure begins with the pre-processing of the incoming fingerprint image. The registered small images from the template are then compared with the fingerprint image to determine to what degree the template matches the image. A threshold describing the smallest allowable deviation is then used to decide if the finger matches the stored template.

4.2.4 Fingerprint Scanners

A fingerprint scanner has basically two tasks; to acquire an image of a fingerprint, and to decide whether or not this image matches the image of a previously enrolled fingerprint. Extracting features from the image and then comparing these features to templates stored in a database or a smart card make decision.

The first generation fingerprint scanners appeared on the market in the mid eighties, so the technology is about fifteen years old. Over the past few years the technology for scanning fingerprints for commercial purposes has evolved a lot. While the first generation sensors used optical techniques to scan the finger, current generation sensors are based on a variety of techniques. The following techniques are deployed in commercial products that are currently available:

- Optical sensors with CCD or CMOS cameras
- Ultrasonic sensors
- Solid state electric field sensors
- Solid state capacitive sensors
- Solid state temperature sensors

The techniques will be described in greater detail in this section. The solid-state sensors are so small that they are to be built into virtually any machine. Currently a sensor is in development that will be built in a plastic card the size of a credit card, not only with respect to length and width but also with respect to thickness! It is clear that this type of sensor will give a boost to the number of applications using fingerprint technology.

- **Optical Sensors**

With optical sensors, the finger is placed or pushed on a plate and illuminated by a LED light source. Through a prism and a system of lenses, the image is projected on a camera. This can be either a CCD camera or, its modern successor, a CMOS camera. Using frame grabber techniques, the image is stored and ready for analysis.

- **Ultrasonic Sensors**

Ultrasonic techniques were discovered when it was noticed that there is a difference n acoustic impedance of the skin (the ridges in a fingerprint) and air (in the valleys of a fingerprint). The sensors that are used in these systems are not new; they were already being deployed for many years in the medical world for making echo's. The frequency range, which these sensors use, is from 20kHz to several GigaHertz. The top frequencies are necessary to be able to make a scan of the fingerprint with a resolution of about 500 dots per inch (dpi). This resolution is required to make recognition of minutiae possible.

- **Electric Field Sensors**

This solid-state sensor has the size of a stamp. It creates an electric field with which an array of pixels can measure variations in the electric field, caused by the ridges and valleys in the fingerprint. According to the manufacturer the variations are detected in the conductive layer of the skin, beneath the skin surface or epidermis.

- **Capacitive Sensors**

Capacitive sensors are, just as the electric field sensors, the size of a stamp. When a finger is placed on the sensor an array of pixels measures the variation in capacity between the valleys and the ridges in the fingerprint. This method is possible since there is a difference between skin-sensor and air-sensor contact in terms of capacitive values.

- **Temperature Sensors**

Sensors that measure the temperature of a fingerprint can be smaller than the size of a finger. Although either width or height should exceed the size of the finger, the other dimension can be fairly small since a temperature scan can be obtained by sweeping the finger over the sensor. The sensor contains an array of temperature measurement pixels, which make a distinction between the temperature of the skin (the ridges) and the temperature of the air (in the valleys).

4.2.5 Algorithms in Fingerprint Scanners

A typical fingerprint verification system consists of a scanning device (capture and enhancement), a feature extraction part, and a comparison part where an identification/verification decision is taken.

For very secure applications, where we allow false rejections due to the level of security, the threshold would be set very high. In low security applications, though, we may be able to deal with a few false acceptances because whatever is being protected is of low value or may be protected.

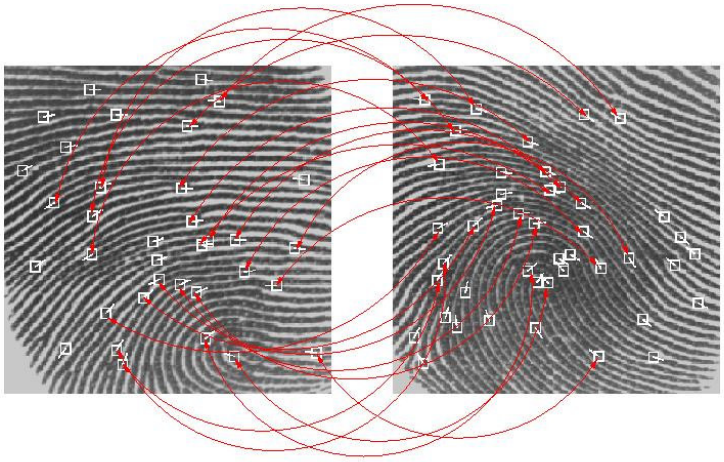


Figure 4.14: Fingerprint Verification

- **False Acceptance Rate (FAR)**

The FAR is the frequency that a non-authorized person is accepted as authorized. Because a false acceptance often leads to damages, FAR is generally a security relevant measure. FAR is a non-stationary statistical quantity, which does not only show a personal correlation, it is to be determined for each individual feature (called personal FAR).

- **False Rejection Rate (FRR)**

The FRR is the frequency that an authorized person is rejected access. FRR is generally thought of as a comfort criteria, because a false rejection is most of all annoying. FRR is a non-stationary statistical quantity, which does not only show a strong personal correlation, it can even be determined for each individual feature.

- **Failure To Enroll rate (FTE or FER)**

The FER is the proportion of people who fail to be enrolled successfully. FER is a non-stationary statistical quantity, which does not only show a strong personal correlation, it can even be determined for each individual feature (called personal FER).

Those who are enrolled yet are mistakenly rejected after much verification / identification attempts count for the Failure To Acquire (FTA) rate. FTA can originate through temporarily not measurable features ("bandage", non-sufficient sensor image quality, etc.). The FTA usually is considered within the FRR and need not be calculated separately.

- **False Identification Rate (FIR)**

The False Identification Rate is the probability in an identification that the biometric feature is falsely assigned to a reference. The exact definition depends on the assignment strategy; namely, after feature comparison, often more than one reference will exceed the decision threshold.

- **False Match Rate (FMR)**

The FMR is the rate which non-authorized people are falsely recognized during the feature comparison. In contrast to the FAR, attempts previously rejected due to poor (image) quality (Failure to Acquire, FTA) are not accounted for. Whether a falsely recognized feature leads to an increase in FAR or FRR depends upon the application. (There are applications that define a successful recognition as a rejection, when, for example, double release of identification cards for a person with a false identity is prevented by comparing the actual reference features with the centrally stored reference features of all cards released so far.)

- **False Non-Match Rate (FNMR)**

The FNMR is the rate at which authorized people are falsely not recognized during feature comparison. In contrast to the FRR, attempts previously rejected due to poor (image) quality (Failure to Acquire, FTA) are not accounted for. Whether a falsely recognized feature leads to increases in FAR or FRR depends upon the application.

- **Equal error rate (EER)**

The common value of the FAR and FRR when the FAR equals the FRR. This is the value where both the FAR and FRR are kept as low as possible at the same time. A low EER value indicates a high accuracy of the system.

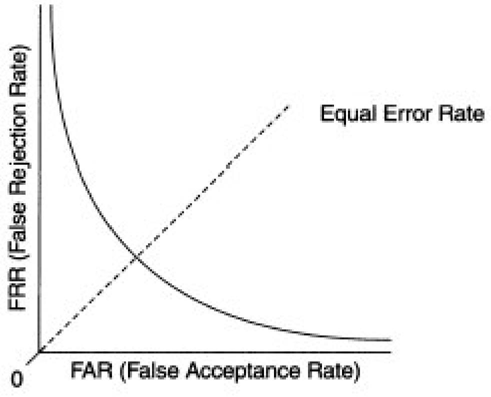


Figure 4.15: EER Measurement

Above figure illustrates the relationship between FRR, FAR, and EER. A big FRR often means a low FAR, and a big FAR often means a low FRR. The small EER value indicates that the security of the system is better.

The algorithm must make a speedy, automated determination of the authenticity of a fingerprint, FAR and FRR must be at or near zero. This way, authentic fingerprints are not rejected and false prints are not accepted.

4.2.6 Fingerprint Accuracy

Different technologies may be appropriate for different applications, depending on perceived user profiles, the need to interface with other systems or databases, environmental conditions, and a host of other application-specific parameters.

From the following table we can determine that the finger print verification is more accurate than any other biometrics technology for the identification system.

Characteristic	Fingerprints	Palmprint	Retina	Iris	Face	Signature	Voice
Ease of Use	High	High	Low	Medium	Medium	High	High
Error incidence	Dryness, dirt, age	Injury, age	Glasses	Poor Lighting	Lighting, age, glasses, hair	Changing signatures	Noise, colds, weather
Accuracy	High	High	Very High	Very High	High	High	High
User acceptance	Medium	Medium	Medium	Medium	Medium	Medium	High
Required security level	High	Medium	High	Very High	Medium	Medium	Medium
Long-term stability	High	Medium	High	High	Medium	Medium	Medium

Table 4.1: Biometrics Technologies Comparison

It is important to note that fingerprint identification works on the principle of a threshold. That is, it is nearly impossible to capture the fingerprint the same way every time it is used for access.

4.3 Face Recognition System

Facial recognition systems analyze facial characteristics. This system requires a digital camera or a camcorder to develop a facial image of the user for identification. The facial recognition technique is one of the fastest growing areas in biometric technologies. Facial recognition software measures characteristics such as the distance between facial features, for example, from pupil to pupil, or the dimensions of the features themselves such as the width of the mouth. Some of these devices also perform a "liveness" test to see how your face moves, so that a photo of the user cannot be used. This "liveness" test would be a necessity essential for good security purpose.

Facial recognition may be generally accepted by users since it uses a digital camera and we are somewhat accustomed to taking photographs or being in a photographic situation (i.e. taking a picture for an ID card or a driver's license). People are used to identifying others by their facial features (i.e. such as viewing a photograph).

For any biometric system there has to be some user knowledge of the device in the first place. If the user does not know how to use the device, for example, that may lead to higher rejection rates by the system. If the user is comfortable with the system and has been trained to properly use it, then the acceptance rates as well as user-to system compatibility will increase.

In the case of facial recognition, it is possible to transparently capture facial images of individuals and compare those images to a database of known criminals, for example. There is a concern regarding transparent capturing of facial images of innocent individuals, mainly due to the fact that they are not aware, or haven't agreed to be part of the "virtual criminal lineup".

4.3.1 How Facial Recognition Works?

There are about 80 nodal points on a human face. Some nodal points that are measured by facial recognition software are the following:

- Width of nose
- Depth of eye sockets
- Width of cheekbones
- Jaw line
- Chin

These nodal points are measured to create a numerical code that represents the face in a database. Facial recognition methods may vary, but they generally involve a series of steps that serve to capture, analyze, and compare your face to a database of stored templates. There are several facial recognition tools currently out in the market, one such example is called the FaceIT® system7. Listed below is the basic process that is used by this system to capture and compare facial images:

- **Detection:** When the system is attached to a video surveillance system, the recognition software searches the field of view of a video camera for faces. If there is a face in the view, it is detected within a fraction of a second. In the case of identification in the flight deck of a plane, for example, the camera would be positioned where there would generally be a face in full view.

- **Alignment:** Once a face is detected, the system determines the heads position, size, and pose. A face needs to be turned at least 35 degrees toward the camera for the system to be able to register it.

- **Normalization:** The image of the head is scaled and rotated so that it can be registered and mapped into an appropriate size and pose.

- **Representation:** The system translates the facial data into a unique code.

- **Matching:** the newly acquired facial data is compared to the stored data and (ideally) linked to at least one stored facial representation.

Raw data, such as an actual photograph, of users' faces is not stored in the system. Instead, the software stores the images as unique codes that only the computer can comprehend. Because unique codes are stored in the system, it is difficult for an attacker to spoof the biometric information. Also, an attacker would not have the ability to extract an actual photograph of the legitimate users of the system. The attacker would only be able to extract numerical codes.

The heart of the FaceIt® facial recognition system is the Local Feature Analysis (LFA) algorithm. This is the mathematical technique the system uses to encode faces. The system maps the face and creates a face print, a unique numerical code for that face. Once the system has stored a face print, it can compare it to the thousands or millions of face prints stored in a database. The system can match multiple face prints at a rate of 60 million per minute from memory or 15 million per minute from hard disk. As comparisons are made, the system assigns a value to the comparison using a scale of one to 10. If a score is above a predetermined threshold, a match is declared.

4.3.2 Facial Recognition: User Influences

Every person carries unique characteristics in their facial features. Factors such as the distance between the eyes and the shape of the nose play an important role in distinguishing a person digitally. The one factor that separates facial recognition from other biometric technologies is the fact that the face is a changeable surface, displaying a variety of expressions, as well as being an active 3D object whose image varies with viewing angle, pose, illumination, accoutrements, and age.

It has been shown that for facial images taken at least one year apart; even the best current algorithms have error rates of 43% - 50%. This error rate range would not be acceptable if it were employed in the flight deck for continuous authentication. The fact that this error rate range corresponds to a one-time authentication step, it is quite possible that this rate may fall well below 10% when it is applied to continuous authentication. It is also possible that there may even be a better algorithm for use in this situation.

When considering facial recognition as a form of identification, there are some user-based influences that must be taken into consideration. Some user-based influences are:

- Beards or moustaches
- Baldness
- Height
- Skin tone

Beards and moustaches play a major role in acceptance rates. It is possible that an appearance or disappearance of facial hair may have an effect on rejection rates for the male population. The same argument can be made about the influence of baldness. A slowly receding hairline may cause rejection by the system, if in fact; the forehead size is a part of the user template. For example, a receding hairline may cause the forehead to appear larger and that person may have to re-enroll their information into the system once again and the same would be true for a man who usually wears a beard or moustache and decides to shave it off completely.

The height of a person may also play a crucial role because the very tall, very short or those in wheelchairs may have difficulty positioning themselves correctly. The height factor will have little effect in the recognition process.

Skin tone may also affect whether the user is accepted or rejected by the system as well. For example, there may be a person whose skin pigment does not register very well with the system and are forced to rejection most of the time. The system should be able to adapt to different skin tones and lighting situations.

The users' behavior may also have an influence on the systems acceptance or rejection rates. Some user behavioral activities that may affect the outcome from the system are:

- Facial expression
- Movement or lack of movement
- Head position
- Distance from camera

Facial expressions can indeed affect the system outcome. For example, if a user initially enrolled into the system with a serious look, they should identify themselves to the camera the same way every time (if at all possible). One should not do things such as widening/squinting the eyes or wrinkling up their nose because it is likely that this type of activity will cause a rejection from the system.

Movement or lack of movement may also cause a rejection from the biometric system. If the user is moving too much, an accurate result may not be possible. The same holds true if the user has lack of movement or if the user has their head tilted to one side. Usually the normalization algorithm used for facial recognition would adjust for activities such as these. Lack of movement may also imply that an intruder is showing a photograph of the legitimate user to the facial scanning device. For this reason, it is important that the system is capable of performing liveness tests.

In the process of facial recognition, the user may be required to stand or sit a certain distance from the camera in order to achieve desired results. If the user is standing or sitting too far or too close to the camera, then the results may be inaccurate and cause a rejection from the system.

User appearance is another issue that must be taken into consideration. Some user appearance factors are:

- Clothing
- Cosmetics and Cosmetic surgery
- Glasses or sunglasses
- Hairstyle or hair color

Some clothing influences may be hats, earrings, or scarves. Cosmetics whether it is caused by user application or surgical procedure may have an effect on acceptance or rejection from the system. Glasses or sunglasses may also affect the result from the system. It is suggested that if the user initially used glasses while enrolling in the system then they must always use those glasses when identifying themselves to the device.

Hairstyles and/or hair color may also affect the users' acceptance or rejection rate. Since hairstyles probably change faster than hair color, it is suggested that the system adapt to these changes or to completely ignore these changes and pay attention to other important attributes of the face. It would become very costly if the users had to re-enroll themselves every time they made a change to their appearance.

In order to be able to implement an effective system, the user influences described here must be taken into consideration. If this type of system is implemented in the flight deck of a plane, some of these influences may be disregarded.

4.3.3 Facial Recognition: Environmental Influences

In addition to user influences, there are also some environmental influences that must be considered. Environmental influences are based on general background, lighting, and weather conditions. These influences are:

- Background, cover
- Other faces
- Lighting or reflections
- Rain or snow

Background scenery or cover around the camera may cause problems when a user is trying to authenticate to the system. If there are other faces that are obstructing or confusing the camera or a faint reflection of another face in the background will have an effect on the acceptance or rejection rates of the system. Lighting and weather conditions such as rain or snow (causing redness in the face) also have an effect on system outcome. By identifying these environmental influences there is a better understanding of what we need to pay attention to if facial recognition is integrated into the designs.

Data quality is the key to achieving satisfactory operational performance of the biometric system. The environment under which enrollment or authentication is taking place will affect the quality of the enrollment or authentication/identification function performed by the system. Since this system will be used by a limited number of people (i.e. rather than by millions of patrons in the airport) it is easier to define the environment that the device will be used in and it makes it easier to determine whether the device is being used the way that it is meant to be used.

4.3.4 Methods of Facial Recognition

The four primary methods employed by facial scan vendors to identify and verify subjects include eigenfaces, feature analysis, neural network, and automatic face processing. Some types of facial scan technology are more suitable than others for applications such as forensics, network access, and surveillance. The process flow of facial scan technology, as with other biometric techniques, contains 4 steps:

- Sample Capture
- Feature Extraction and storage
- Live and stored template comparison prior to matching
- Matching of the live and stored templates to produce a matching score

A system that is based on using local feature analysis uses a camera and computer to identify a person and analyzes pixels that make up the face image.

A flight deck biometric authentication system using facial recognition should be capable of performing liveness tests and a system based on local feature analysis will be able to perform liveness tests. In order to be sure that the eyes, nose, and mouth belong to a living being and not a mannequin, the program looks for eye blinks or other tell tale facial movements.

The Eigenface method examines the face as a whole and is one of the most popular face recognition methods in use today. With a database of headshots on hand, the system compares the face being identified to the composite. The composite is the actual template of the image that is initially stored in the system at the time of enrollment and the target is the live template that is captured at the time of authentication. An algorithm measures how much the target face differs from the composite and generates a 128-digit personal identification number based on the deviation. If the Eigenface method is used, a training set that contains enough number of face examples is needed. The purpose of the training set is to have a number of various templates of the same person. These various templates are expected to cover various conditions such as different head poses, lighting conditions, or facial expressions.

Though overall not as robust as eigenfaces, feature analysis, or neural network, automatic face processing may be more effective in dimly lit, frontal image capture situations. In neural network mapping, the enrollment and verification data are compared and there is a vote on whether there is a match between the two. Neural networks employ an algorithm to determine the similarity of the unique global features of live verses enrolled faces. This method, theoretically, leads to an increased ability to identify faces in difficult conditions.

References:

- Face Processing: Advanced Modeling and Methods
  By Wenyi Zhao, Rama Chellappa, Academic Press
- Handbook of Fingerprint , By Davide Maltoni, Anil K. Jain, Dario Maio, Salil Prabhakar, Springer
- Biometric Systems: Technology, Design and Performance Evaluation, Springer
- www.findbiometrics.com
- www.cl.cam.ac.uk
- www.hitachi.com
- www.face-rec.org

Chapter - 5

BIOMET: A Multimodal Biometric Authentication System

5.1 A Multimodal System using Fingerprint and Face Recognition

5.2 Existing Multimodal Biometric System & Proposed Integrated Model and achievement target

5.3 Hardware and Software used for Fingerprint

5.4 Hardware and Software used for Face Recognition

5.5 Experimental Results of Multimodal Biometric System

5.1 A Multimodal System using Fingerprint and Face Recognition

This research applied multimodal biometric authentication system using fingerprint and face recognition system. It is a serial mode in which fingerprint and face recognition result is taken sequentially (one after another). The experiment has made fusion the results of fingerprint & face.

In the conducted experiment more than 200 live tests of fingerprint & face recognition data were taken and analysis was done.

Architecture of a multibiometric system refers to the sequence in which the multiple cues are acquired and processed. Typically, the architecture of a multimodal biometric system is either serial or parallel. In the serial or cascade architecture, the processing of the modalities takes place sequentially and the outcome of one modality affects the processing of the subsequent modalities. In the parallel design, different modalities operate independently and their results are combined using an appropriate fusion scheme. Both these architectures have their own advantages and limitations.

The cascading scheme can improve the user convenience as well as allow fast and efficient searches in large scale identification tasks. For example, when a cascaded multimodal biometric system has sufficient confidence on the identity of the user after processing the first modality, the user may not be required to provide the other modalities. The system can also allow the user to decide which modality he/she would present first. Finally, if the system is faced with the task of identifying the user from a large database, it can utilize the outcome of each modality to successively trim the database, thereby making the search faster and more efficient. Thus, a cascaded system can be more convenient to the user and generally requires less recognition time when compared to its parallel counterpart. However, it requires robust algorithms to handle the different sequence of events. In this system, face recognition is used to retrieve the top n matching identities and fingerprint recognition is used to verify these identities and make a final identification decision.

A multimodal system designed to operate in the parallel mode generally has a higher accuracy because it utilizes more evidence about the user for recognition. Most of the proposed multibiometric systems have a parallel architecture because the primary goal of system designers has been a reduction in the error rate of biometric systems.

The choice of the system architecture depends on the application requirements. User friendly and less security critical applications like bank ATMs can use a cascaded multimodal biometric system. On the other hand, parallel multimodal systems are more suited for applications where security is of paramount importance (e.g., access to military installations). It is also possible to design a hierarchical (tree-like) architecture to combine the advantages of both cascade and parallel architectures. This hierarchical architecture can be made dynamic so that it is robust and can handle problems like missing and noisy biometric samples that arise in biometric systems.

5.1.1 Generation of the Multimodal Database

The multimodal database used in our experiments was constructed by merging two separate databases of 200 users each. 250 face images were acquired using a CCD camera (640 X 480). 200 fingerprint impressions (of the same finger) were obtained using a Digital Biometrics sensor (512 X 512). The mutual independence assumption of the biometric traits allows us to randomly pair the users from the two sets. The biometric data captured from every user is compared with that of all the users in the database leading to one genuine user and 199 impostor users for each distinct input.

	Finger	**Face**
No. of users	200	200
No. of impressions	4	1
Image Size	512 X 512	640 x 480
Template Size	256 – 1200 Bytes	84 – 2000 Bytes
Image Acquisition	Digital Persona U.are.U. (optical)	Logitech Camera (CCD)
Software	Fingerprint Verification System (FVS) 4.2 Standard SDK	VeriLook 2.0 SDK

Table 5.1: Data table for Fingerprint & Face

5.1.2 Biometric Performance Measurements

The performance of biometric systems is tested usually in terms of false rejection rate (FRR), false acceptance rate (FAR), failure to enroll rate (FER), enrollment time, and verification time. The false acceptance rate is most important when security is a priority whereas low false rejection rates are favored when convenience is the priority.

The biometric system employed must have a low false acceptance rate since security is the priority. If the false acceptance rate is as low as possible then we have a better chance of not allowing unauthorized subjects into the system. The point at which the FAR and FRR meet or crossover is known as the equal error rate. This rate gives a more realistic measure of the performance of the biometric system rather than using either the FAR or FRR individually.

The failure to enroll rate (FER) is the rate, which a subject is unable to introduce his or her biometric to the system that is acceptable to the system. For example, if there is a fingerprint scanning device which is very sensitive to the images presented to it and a subject is not able to provide a clear cut image then he or she will not be able to enroll into the system. Usually, there are systems that will allow the subject several attempts to enroll biometric information into the system.

Both the enrollment and live presentation times are important factors in determining or testing system performance. The enrollment time is that timeline in between and including the capturing of the biometric sample and creating the stored template of that sample.

The verification time is a measurement of the process of live presentation. This process includes the capture of the raw data, live template processing, comparison of the stored template to the live template and the time it takes for the system to provide a decision (i.e. match or non-match). To provide the continuous authentication mechanism desired for the verification time must be near real time for a successful biometric authentication system.

5.2 Existing Multimodal Biometric System & Proposed Integrated Model and achievement target

The design of a multimodal biometric system is strongly dependent on the application scenario. A number of multimodal biometric systems have been that differ from one another in terms of their architecture, the number and choice of biometric modalities, the level at which the evidence is accumulated, and the methods used for the integration or fusion of information.

Four levels of information fusion are possible in a multimodal biometric system. They are fusion at the sensor level, feature extraction level, matching score level and decision level. Sensor level fusion is quite rare because fusion at this level requires that the data obtained from the different biometric sensors must be compatible, which is seldom the case with biometric sensors. Fusion at the feature level is also not always possible because the feature sets used by different biometric modalities may either be inaccessible or incompatible. Fusion at the decision level is too rigid since only a limited amount of information is available. Therefore, integration at the matching score level is generally preferred due to the presence of sufficient information content and the ease in accessing and combining matching scores.

In the context of verification, fusion at the matching score level can be approached in two distinct ways. In the first approach the fusion is viewed as a classification problem, while in the second approach it is viewed as a combination problem. In the classification approach, a feature vector is constructed using the matching scores output by the individual matchers; this feature vector is then classified into one of two classes: “Accept” (genuine user) or “Reject” (impostor). In the combination approach, the individual matching scores are combined to generate a single scalar score, which is then used to make the final decision

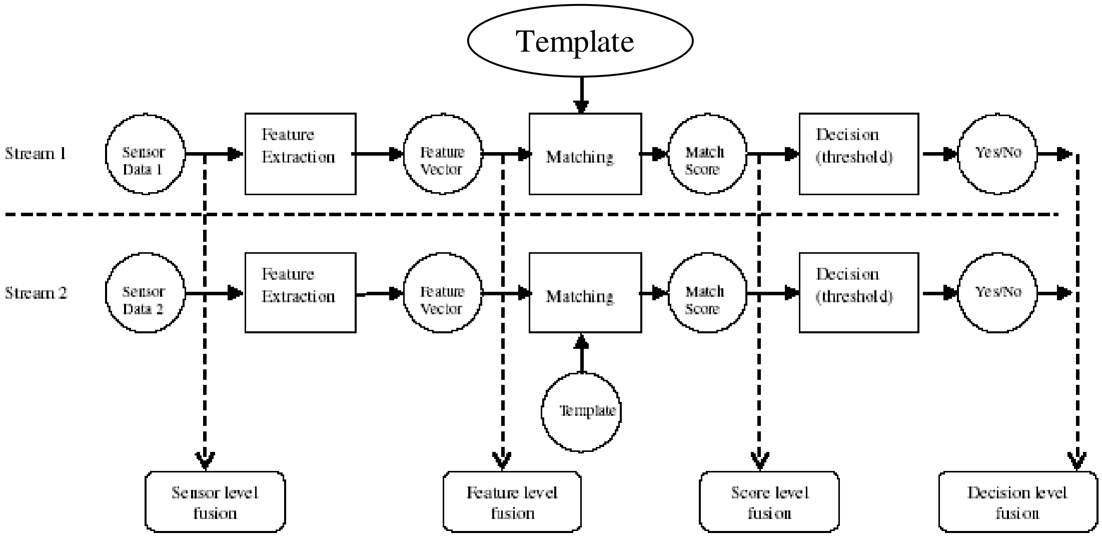


Figure 5.1: Fusion levels in multimodal biometric fusion.

A. Pre-mapping fusion I: Fusion at the sensor level

The raw data, acquired from sensing the same biometric characteristic with two or more sensors, is combined (Figure 5.1). An example of the sensor fusion level is sensing a face data simultaneously with two different cameras. Although fusion at such a level is expected to enhance the biometric recognition accuracy, it can not be used for multimodal biometrics because of the incompatibility of data from different modalities.

B. Pre-mapping fusion II: Fusion at the feature level

Fusion at this level, as shown in Figure 5.1, can be applied to the extraction of different features from the same modality or different multimodalities. An example of a unimodal system is the fusion of instantaneous and transitional spectral information for face recognition. On the other hand, concatenating the feature vectors extracted from face and fingerprint modalities is an example of a multimodal system. It is stated in that fusion at the feature level is expected to perform better in comparison with fusion at the score level and decision level. The main reason is that the feature level contains richer information about the raw biometric data. However, such a fusion type is not always feasible. For example, in many cases the given features might not be compatible due to differences in the nature of modalities. Also such concatenation may lead to a feature vector with a very high dimensionality. This increases the computational load. It is reported that a significantly more complex classifier design might be needed to operate on the concatenated data set at the feature level space.

C. Post-mapping fusion I: Fusion at the matching score level

At this level, it is possible to combine scores obtained from the same biometric characteristic or different ones. Such scores are obtained, for example, on the basis of the proximity of feature vectors to their corresponding reference material (Figure 5.1). The overall score is then sent to the decision module. Currently, this appears to be the most useful fusion level because of its good performance and simplicity. This fusion level can be divided into two categories: combination and classification. In the former approach, a scalar fused score is obtained by normalizing the input matching scores into the same range and then combining such normalized scores. In the latter approach, the input matching scores are considered as input features for a second level pattern classification problem between the two classes of client and the Impostor.

D. Post-mapping fusion II: Fusion at the decision level

In this approach, as shown in Figure 5.1, a separate decision is taken for each biometric type at a very late stage. This seriously limits the basis for enhancing the system accuracy through the fusion process. Thus, fusion at such a level is the least powerful.

The score level fusion techniques are divided into two main categories of fixed rules (rule-based) and trained rules (learning-based). The fixed rules are also referred to as the nonparametric rules while the trained rules are referred to as the parametric rules. The main reason for categorizing the fusion techniques in this way is that trained rules require sample outputs from the individual modalities to train the pattern classifiers. In other words, they use development data to calculate some required parameters. These parameters are then used to appropriately fuse the score data in the test phase. Examples of the trained rules are Weighted Sum rule and Weighted Product rule. On the other hand, fixed rules are applied directly to fuse the given test scores for different modalities. In other words, the contribution of each modality is fixed a priori. Examples of fixed rules are AND rule, OR rule, Maximum rule, Minimum rule, Majority voting, Sum rule, Product rule and Arithmetic Mean rule. Examples of trained rules are Weighted Sum rule, Weighted Product rule, Fisher Linear Discriminant, Quadratic Discriminant Analysis, Logistic Regression, Support Vector Machine, Multi-Layer Perceptrons and Bayesian classifier.

Proposed Design

The intent of the multimodal biometric authentication system is to provide a strong guarantee of identification. The system must provide assurance that the identity of the person is correct and that the identity is unique. Requirements for the multimodal biometric authentication system include reliability, ease of use, and non-intrusiveness. The authentication system should provide continuous and accurate operation. Authorized users should be allowed access and unauthorized users should be prohibited, without interruption or deterioration in performance, accuracy or speed.

Biometric System Process

All biometric systems basically follow the same set of processes for biometric feature matching represented in Figure 3.

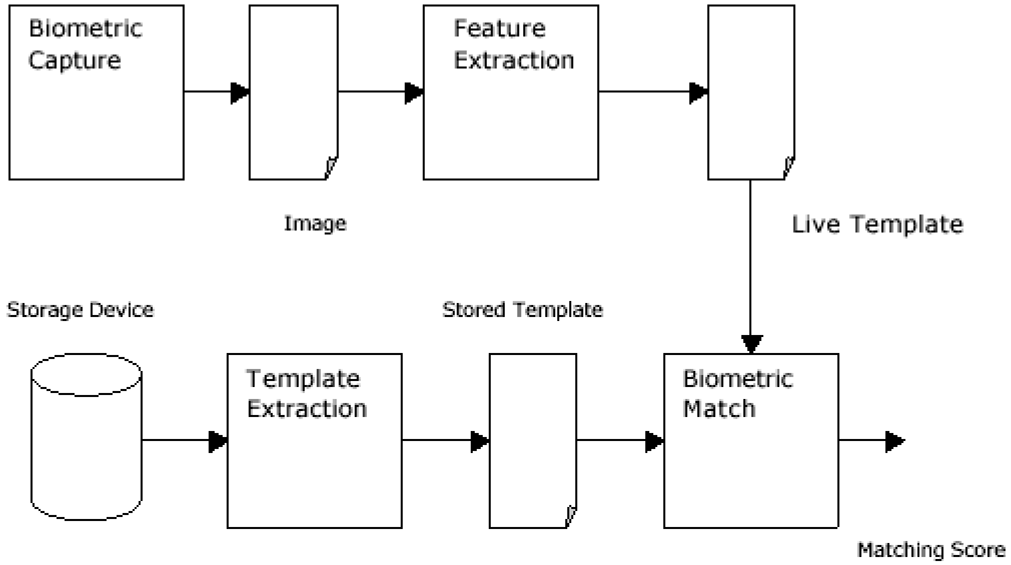


Figure 5.2: Basic Biometric System Process

Biometric capture takes place at the biometric device (i.e. fingerprint scanner). The image of the biometric is processed using specific algorithms tailored for that biometric method to produce a live template. The live template of the biometric is a numerical representation of the currently acquired biometric. From the storage device, the template of the biometric which was stored as part of user enrollment, is retrieved and should match the value from the live template. When this occurs a biometric match is acquired.

PROPOSED DESIGNS FOR MULTIMODAL BIOMETRICS

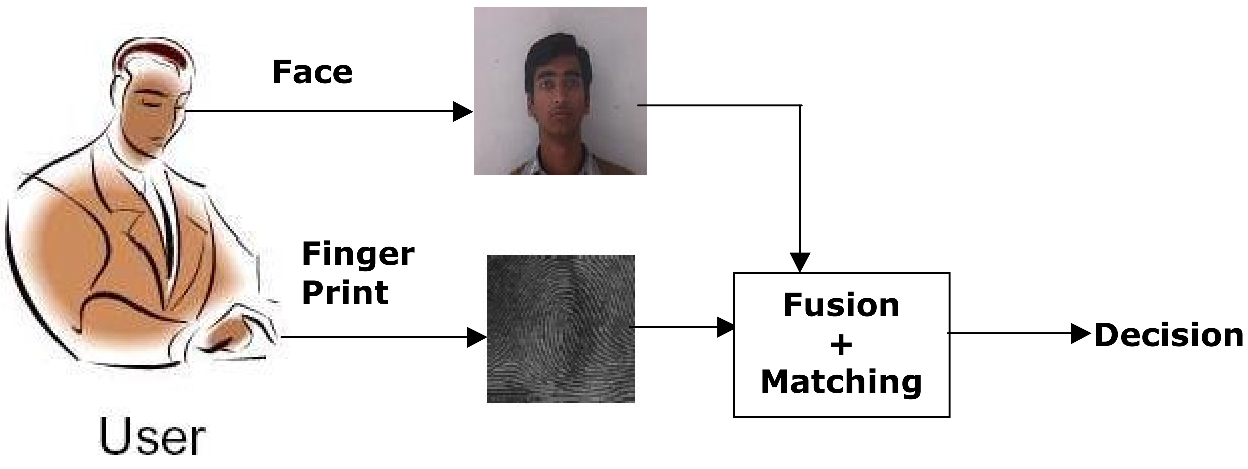


Figure 5.3: Proposed Multimodal Biometric System Design

In this propose design fingerprint and face print taken from the same person using the same biometric devices (finger print reader and web camera). Then the fusion and matching are taken for final decision. The user first identifies him/her using face recognition and then fingerprint recognition. The final result is based on the result of face and fingerprint result. AND configuration is considered for final result.

The templates of fingerprint and face print have been stored in the database. At the time of enrollment the templates have been encrypted and then stored in the database to increase security of templates. The database used for the research is MS – Access 2000. The data dictionary for database has been provide below.

Field Name	**Data type**	**Constraint**
ID	AutoNumber	Primary Key
Features	OLE Object	
FingerprintID	Text(50)	

Table 5.2: Fingerprint Table

Field Name	**Data Type**	**Constraint**
ID	AutoNumber	Primary Key
Features	OLE Object	
FaceID	Text(50)	

Table 5.3: Face Table

The different biometrics systems can be integrated at multi-classifier and multi-modality level to improve the performance of the verification system. However, it can be thought as a conventional fusion problem i.e. can be thought to combine evidence provided by different biometrics to improve the overall decision accuracy.

The multimodal biometric system is developed at multi-modalities level. The following steps are performed

S1: Given a query image as input, features are extracted by the individual recognizers and then an individual comparison algorithm for each recognizer compares the set of features and calculates the matching scores or distances corresponding to each recognizer for various traits.

S2: The scores/distances obtained in S1 are normalized to a common range between 0 to 1.

S3: These scores are then converted from distance to similarity score by subtraction from 1 if it is a dissimilarity score.

S4: The matching scores are further rescaled so that threshold value becomes same for each recognizer.

S5: Then the combined matching score is calculated by fusion of the matching scores of multiple classifiers using sum rule technique.

The multimodal biometric system is developed by integrating two traits (face and fingerprint) at matching score level. Based on the proximity of feature vector and template, each subsystem computes its own matching score. These individual scores are finally combined into a total score, which is passed to the decision module. The same steps for fusion at classifiers level are followed for multiple modalities level i.e., matching scores are computed for each trait followed by normalization to the common scale and distance to similarity score conversion for the two traits. The matching scores are further rescaled so that the threshold value becomes common for all the subsystems. Finally, the sum of score technique is applied for combining the matching scores of two traits. Thus the final score MS_{Final} is given by,

$$MS_{Final} = (a \times MS_{Face} + b \times MS_{Finger}) /2$$

where MS_{Face} = matching score of face, MS_{Finger} = matching score of fingerprint and a, b, are the weights assigned to the various traits. Currently, equal weightage is assigned to each trait so the value of a and b is one. The final matching score (MS_{Final}) is compared against a certain threshold value to recognize the person as genuine or an imposter.

5.3 Hardware and Software used for Fingerprint Recognition

5.3.1 Hardware used for Fingerprint Recognition

Digital Persona U.are.U. Fingerprint Reader

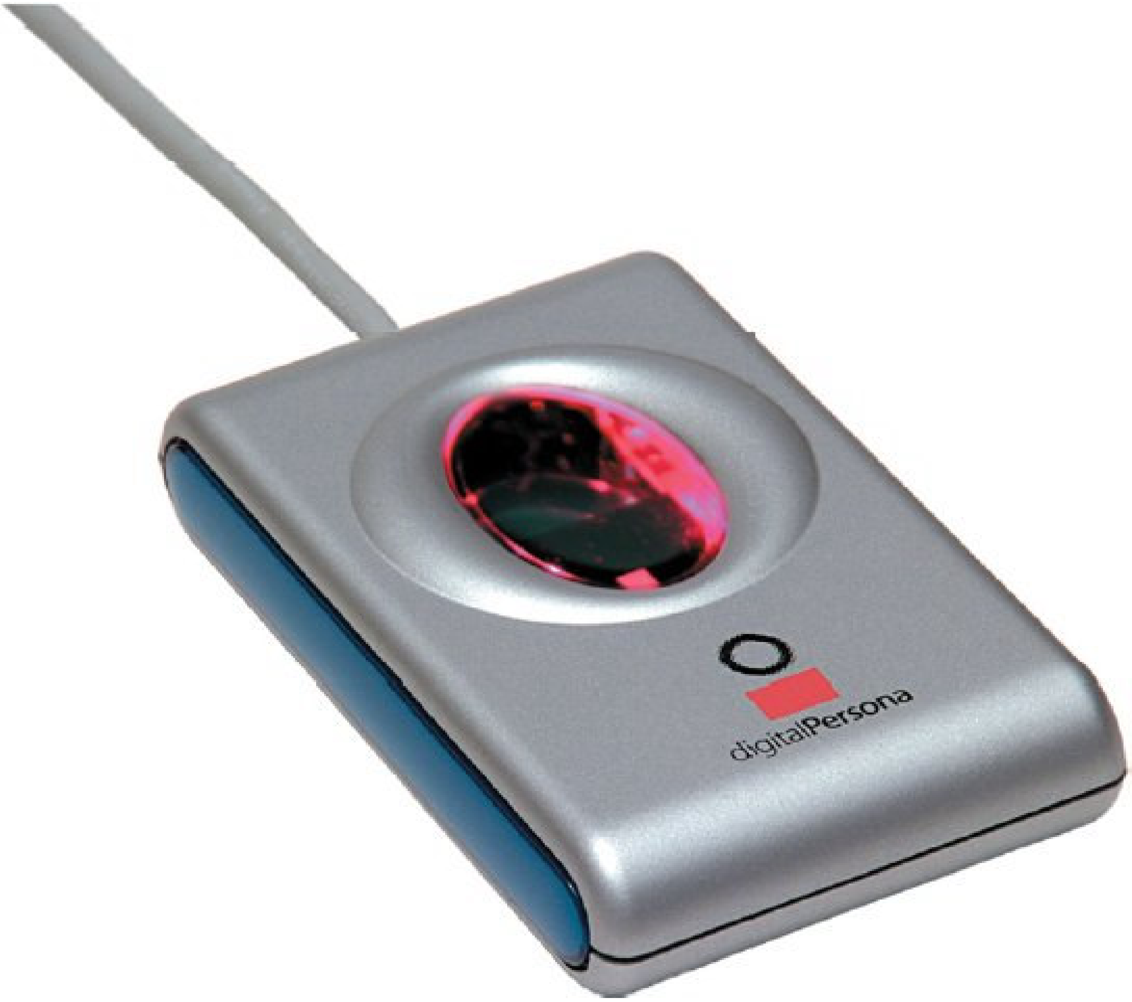


Figure 5.4: Digital Perosna U.are.U. 4000 Fingerprint Reader

System Requirements

Windows® 2000, Windows® XP

Pentium® 4 processor 500 GHz more
128 MB RAM (256 recommended)

Technical Specifications

- Interface : USB 2.0
- Optical Resolution : 512 dpi
- Max. Resolution (Hardware) : 512 X 512 dpi
- Max. Gray Depth : 8-bit (256 Gray Levels)
- Platform : PC
- Operating System: Microsoft Windows XP, Microsoft Windows 2000, Microsoft Windows NT
- Dimensions: 1.93 X 3.11 X 0.75 in (W X D X H)

5.3.2 Software used for Fingerprint Recognition

Fingerprint Verification System (FVS) 4.2 Standard SDK

FVS Standard SDK is intended for most biometric system developers. It allows developing biometric applications for Windows or Linux operating systems. The SDK contains drivers for some of the major fingerprint scanners that allow the developer to obtain data from the scanners without any additional software.

FVS Standard SDK distribution package contains:

- One FVS DLL/library installation license
- MS Windows components
  - o Drivers for image input from DigitalPersona U.are.U, SecuGen Hamster III, BiometriKa FX 2000, OFS sensors
  - o Source codes of FVS DLL usage sample application. Source codes in C/C++ (two samples: Win32 API and MFC), C#, Java, Visual Basic, Visual Basic .Net, Visual Basic for Applications and Delphi 6 are included;
- Linux components
  - o MySQL integration module
  - o Drivers for image input from OFS Sensor, BiometriKa FX 2000, Fujitsu MBF200, fingerprint scanners
  - o Source codes of FVS shared library usage sample application in C/C++
- Documentation.

System requirements for FVS Standard SDK

- PC with Pentium-compatible 500MHz processor or better
- Microsoft Windows 98/ME/NT/2000/XP/2003 or Linux (based on glibc 2.2.5 or compatible)
- 32 MB minimum physical RAM (64 MB physical RAM recommended)
- CD-ROM drive
- 32 MB minimum hard disk space during installation
- Fingerprint scanner driver (users can use the driver, included in FVS Standard SDK, or can obtain the driver from the scanner's manufacturer)

Fingerprint Recognition Operation

The fingerprint recognition operation identifies the processes involved in registering and verifying fingerprints from a developer perspective. You must be familiar with this operation and the related terminology to use the SDK to integrate fingerprint recognition functionality in your application.

The following processes comprise the fingerprint recognition operation:

1 Acquire a fingerprint scan. The first step in the fingerprint recognition operation is to acquire a fingerprint scan. When a user touches the reader, a fingerprint scan—called a raw sample—is compressed and encrypted by the reader and sent to the PC.

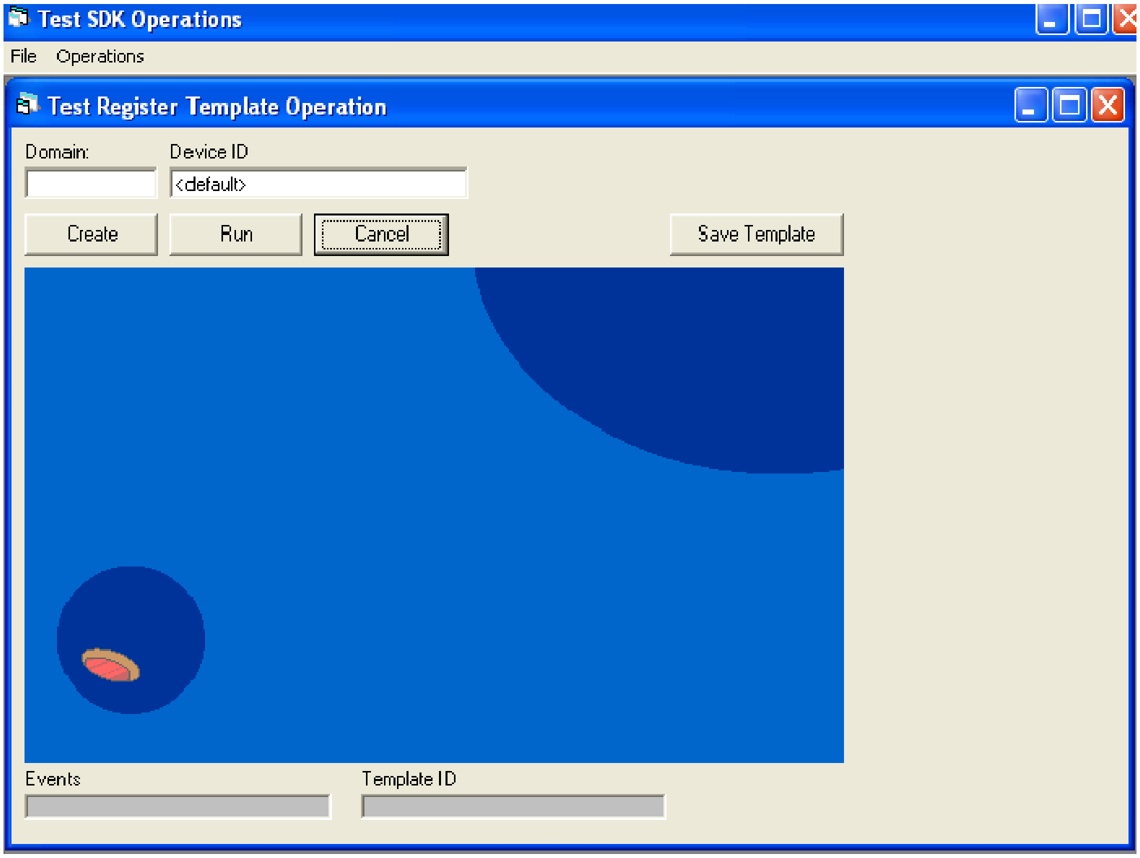
2 Decrypt and decompress the raw sample. When the raw sample is received from the reader, it is decrypted and decompressed into a sample from which features can be extracted to create a template.

3 Create a template. After determining the intended operation—either registration or verification—create the appropriate template. Created from the sample, a template is a mathematical description of the fingerprint characteristics and is assigned one of two types: a pre-registration or verification template.

4 **Perform registration or verification operation.** Following is a description of the registration and verification processes:

- **Register:** If a new fingerprint is being registered, you must acquire four preregistration templates which are used to create a single registration template. The registration template can then be stored for later use during the verification process.

Verify. In the verification operation, a verification template is acquired and compared to an existing registration template for matching.



Domain:

Device ID <default>

Events

Template ID

Figure 5.5 : Register Template Screen

Choosing a Layer

Which layer you choose to implement fingerprint recognition functionality in your application can be based on several factors, ranging from the level of control over the fingerprint recognition operation you require to the degree of experience you have as a programmer.

The Engine Layer is intended for programmers who require control over every process in the fingerprint recognition operation. The Operations Layer is best for those who would benefit from a faster approach to implementation, as well as a less complex one.

Engine Layer

The Engine Layer allows you to facilitate—and control every aspect of—the processes in the fingerprint recognition operation.

Operations Layer

Similar to the Engine Layer, the Operations Layer allows you to facilitate the fingerprint recognition operation. The programmer, however, is shielded from much of the details. You only need to decide which process you want to perform: registration or verification. Then, you write event handlers for the events generated by these processes to control them and provide user feedback. As a result, writing all applications with the Operations Layer is much simpler and faster than with the Engine Layer, although you have less control over the other aspects of the operation.

Adding Security to the Fingerprint Recognition Operation

The Platinum SDK provides security mechanisms that prevent a sample or verification template from being used more than once for matching (known as a replay attack).

The FPRawSample, FPSample and FPTemplate objects contain two properties —SecureMode and Nonce—which are used to add security to the verification process.

Evaluating the SecureMode Property

The SecureMode property of FPRawSample, FPSample and FPTemplate is used to evaluate the level of security applied to the verification process, allowing you to determine whether adequate security measures were in place during the verification process.

When acquiring a raw sample, converting to a sample or performing feature extraction, the SecureMode property will return any combination of the following values:

- Sm_None indicates that no security features were in place during the verification process.
- Sm_DevNonce indicates that the nonce was created and embedded in the raw sample object by the fingerprint recognition device. It is only returned when the nonce is embedded in a FPRawSample object.

- Sm_DevSignature indicates that the raw sample data was signed by the fingerprint recognition device. This value is set by the device and cannot be changed.
- Sm_DevEncryption indicates that the raw sample data was encrypted by the fingerprint recognition device. This value is set by the device and cannot be changed.
- Sm_FakeFingerDetection is returned if the fingerprint recognition device is able to recognize fake fingerprints. This value is set by the device and cannot be changed.
- Sm_NonceNotVerified indicates the nonce was not verified. The object can still be used, but should be considered non-secure.
- Sm_SignatureNotVerified indicates that the signature of the data object was not verified on import. The object can still be used, but should be considered non-secure.

Using a Nonce

A randomly-generated number, or nonce, is used to ensure that when a FPRawSample, FPSample or FPTemplate object is processed, i.e., feature extraction, etc., the return object can be trusted.

A nonce is generated using the GenerateNonce method of the DPDataSecurity component and is set in a processing object using the SetNonce method. When the object is processed, the SecureMode property can be evaluated to determine if the returned object can be trusted. If Sm_NonceNotVerified is returned, the nonce could not be verified in the return object.

5.4 Hardware and Software used for Face Recognition

5.4.1 Hardware used for Face Recognition

Figure 5.6: Logitech Camera

System Requirements

Windows® 2000, Windows® XP

Pentium® 4 processor 1.4 GHz or AMD Athlon ™ 1GHz processor (Pentium® 4 2.4 GHz recommended) 128 MB RAM (256 recommended)

Windows Vista™

Pentium® 4 2.4 GHz (2.8 GHz recommended) 512 MB RAM
(1GB recommended)

- 200 MB hard drive
- CD-ROM drive
- 16-bit color display adapter
- OS compatible sound card and speakers
- or 2.0 USB port

Recommended system requirements are needed to use Logitech Video Effects™, RightSound™ or RightLight™ 2 Technology features.

Software installation required to use RightLight™, RightSound™, and Logitech® Video Effects™.

Technical Specifications

- True 1.3 mega pixel sensor with RightLight™ 2 technology
- Live video: up to 640 x 480 pixels
- Still image capture: 1280 x 960 pixels
- Built-in microphone with RightSound™ technology
- Up to 30 frames per second live video with recommended system
- USB 2.0 high-speed certified
- 6 ft. USB cable
- Fixed focus

5.4.2 Software used for Face Recognition

VeriLook 2.0 SDK

VeriLook SDK is based on the VeriLook Technology and is intended for biometric systems developers and integrators. It allows rapid development of the biometric application using functions from VeriLook library, which ensure high reliability of the face identification, 1:1 and 1:N matching modes, multiple faces' processing, comparison speed up to 80,000 faces per second. VeriLook can be easily integrated into the customer's security system. The integrator has a complete control over SDK data input and output; therefore SDK functions can be used in connection with most cameras and databases. Integrator could develop any user interface.

VeriLook 3.2 Standard SDK distribution package contains

- One VeriLook DLL/library installation license
- Interface for image input from file
- Interface for working with webcam
- Source code of DLL/library usage sample applications in C/C++
- Source code of DLL usage sample application in C# (for Windows only)
- Source code of DLL usage sample application in Visual Basic 6 (for Windows only)
- Source code of DLL usage sample application for MS Access in VBA (for Windows only)
- Documentation

System requirements for VeriLook 2.0 Standard SDK

PC with Pentium-compatible 1 GHz processor or better, 128 MB of RAM, 2MB HDD space for the installation package.

- Optionally, camera or web cam (recommended frame size: 640 x 480);
- Microsoft Windows specific:
  o Microsoft Windows 9x/ME/NT/2000/XP/2003
  o Microsoft DirectX 8.1 or later. Could be downloaded here;
  o Microsoft XML Parser (MSXML) 3.0.
  o Microsoft GDI+ library
- Linux specific:
  o Linux (based on glibc 2.2.5 or compatible)
  o Video4linux

About VeriLook

VeriLook 2.0 is designed with aim to demonstrate the capabilities of VeriLook face recognition engine. The program is a Windows 2000/XP compatible GUI application.

Evaluation software supports image acquisition from the external video source (such as Web cameras) via DirectX library. Also it can read face images from .bmp, .tif, .png, .jpg, .gif files.

The application has 3 operation modes:

1. Enrollment. Software processes the face image, extracts features and writes them to the database.
2. Face enrollment with features generalization. This mode generates the generalized face features collection from a number of the face templates of the same person. Each face image is processed and features are extracted. Then collections of features are analyzed and combined into one generalized features collection, which is written to the database. The face recognition quality increases if faces are enrolled using this mode.
3. Matching. This mode performs new face image matching against face templates stored in the database.

Image quality control

Face recognition is very sensitive to image quality so maximum care should be attributed to image acquisition.

Pose

The frontal pose (full-face) must be used. Rotation of the head must be less than +/- 5 degrees from frontal in every direction – up/down, rotated left/right, and tilted left/right.

Expression

The expression should be neutral (non-smiling) with both eyes open, and mouth closed. Every effort should be made to have supplied images comply with this specification. A smile with closed jaw is allowed but not recommended.

Examples of Non-Recommended Expressions

1. A smile where the inside of the mouth is exposed (jaw open).
2. Raised eyebrows.
3. Closed eyes.
4. Eyes looking away from the camera.
5. Squinting.
6. Frowning.
7. Hair covering eyes.
8. Rim of glasses covering part of the eye.

Face changes

Beard, moustache and other changeable face features influence face recognition quality and if frequent face changes are typical for some individual, face database should contain e.g. face with beard and cleanly shaved face enrolled with identical ID.

Lighting

Lighting must be equally distributed on each side of the face and from top to bottom. There should be no significant direction of the light or visible shadows. Care must be taken to avoid "hot spots". These artifacts are typically caused when one, high intensity, focused light source is used for illumination.

Eyeglasses

There should be no lighting artifacts on eyeglasses. This can typically be achieved by increasing the angle between the lighting, subject and camera to 45 degrees or more. If lighting reflections cannot be removed, then the glasses themselves should be removed. (However this is not recommended as face recognition typically works best when matching people with eyeglasses against themselves wearing the same eyeglasses).

Glasses must be clear glass and transparent so the eyes and irises are clearly visible. Heavily tinted glasses are not acceptable.

Web cameras

As web cameras are becoming one of the most common personal video capturing devices, we have conducted small video image quality check. Most of cheap devices tend to provide 320x240 images of low quality, insufficient for biometrical use. As a general rule, true 640x480 resolution (without interpolation) and a known brand name are recommended.

Images should be enrolled and matched using the same camera, as devices have different optical distortions that can influence face recognition performance.

Liveness Detection

VeriLook algorithm is capable to differentiate live faces from non live faces (e.g. photos). "Use liveness check" checkbox and "Livenes threshold" parameter in the options dialog controls the behavior of liveness check. When "Use liveness check" checkbox is marked, the liveness check is performed while matching. That is the liveness score of collected stream is calculated and checked against the liveness score threshold set in the "Liveness threshold" parameter.

Using liveness check requires a stream of consecutive images. (This check is intended to be used mainly with video stream form a camera). The stream must be at least 10 frames length and the recommended length is 10 - 25 frames. Only one person face should be visible in this stream. If the stream does not qualify as "live" and "Extraction failed" message is displayed in the log window.

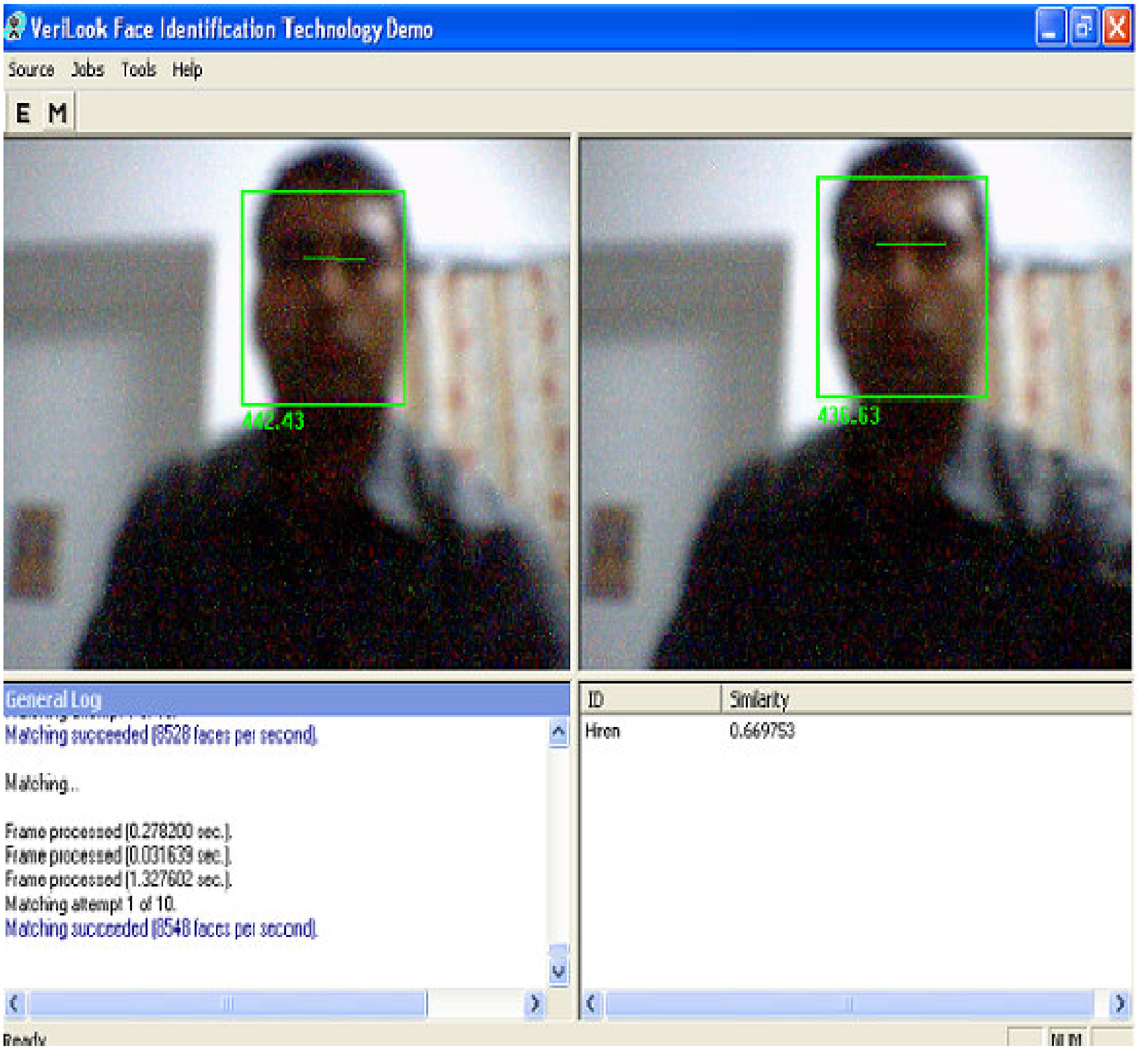
To maximize the liveness score of a face found in an image stream, user should move his head around a bit, tilting it, moving closer to or further from the camera while slightly changing his facial expression. (e.g. User should start with his head panned as far left as possible but still detectable by face detection and start panning it slowly right slightly changing his facial expression until he pans as far right as possible (but still detectable by face detector)).

Application

VeriLook demo application demonstrates VeriLook face recognition algorithm using video and still images.

Main window

Main application window has four-pane layout, where two top panes are used for image display and two bottom panes are used for message logging. Menu commands and two toolbar buttons, used as shortcuts for most accessed commands, control application.



```
Matching succeeded (8528 faces per second).

Matching...

Frame processed (0.278200 sec.).
Frame processed (0.031639 sec.).
Frame processed (1.327602 sec.).
Matching attempt 1 of 10.
Matching succeeded (8548 faces per second).
```

ID	Similarity
Hiren	0.669753

Figure 5.7 : Main application window

Main window panes:

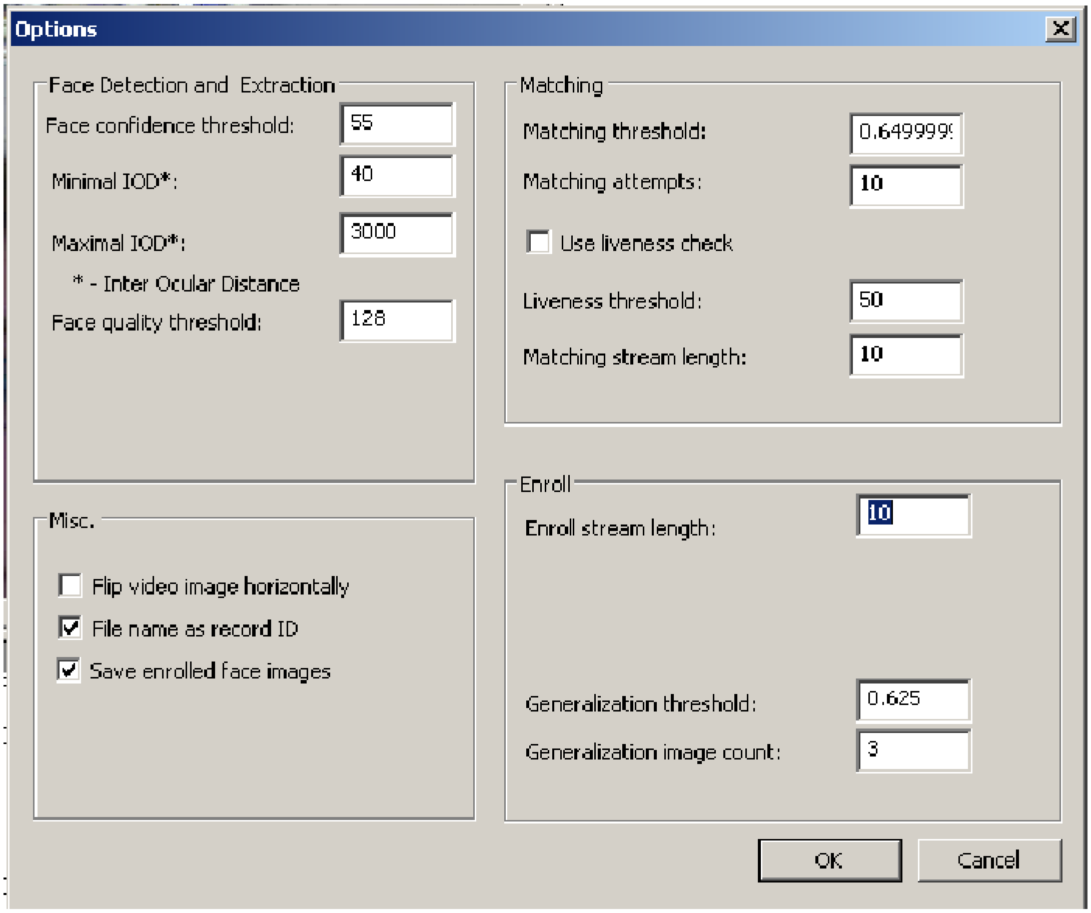
1. Face detection pane, used to display video or still images and result of face detection algorithm overlaid on image.

2. Matching/enrollment pane, used to display images enrolled to face database or used for matching.

3. Application log, used for system information and application progress messages.

4. Match results pane for listing id of the subject in the database, most similar to matched image. Subjects are considered “similar” if their similarity value exceeds matching threshold set via Options dialog. This value is displayed in the second list view column.

Options dialog



Face confidence threshold: 55

Minimal IOD*: 40

Maximal IOD*: 3000

* - Inter Ocular Distance

Face quality threshold: 128

☐ Flip video image horizontally

☑ File name as record ID

☑ Save enrolled face images

Matching threshold: 0.649999!

Matching attempts: 10

☐ Use liveness check

Liveness threshold: 50

Matching stream length: 10

Enroll stream length: 10

Generalization threshold: 0.625

Generalization image count: 3

Figure 5.8: Options dialog

Face confidence threshold – value which controls the requirements for face detection. The greater this value is the more strict rules are applied when looking for faces in an image.

- Minimum IOD – minimum distance between eyes.
- Maximum IOD – maximum distance between eyes.
- Face quality threshold – controls how strict rules are applied when determining the quality of a found face for extraction. If face quality score does not outscore
- Matching threshold – threshold that separates identical and different subjects. Matching threshold is linked to false acceptance

rate (FAR, different subjects erroneously accepted as of the same) of matching algorithm. The higher is threshold, the lower is FAR and higher FRR (false rejection rate, same subjects erroneously accepted as different) and vice a versa.

- Matching attempts – specifies how many times face database will be searched for a match for each newly detected face. Matching will be terminated after finding first subject with similarity value greater than matching threshold.
- Use liveness check – Controls if liveness check is used while matching.
- Liveness threshold – controls the requirements for live faces. The greater this value is the more strict rules are applied to check if face in an image stream is live. (If this value is set to 0 all faces are considered to be live).

- Matching stream length – maximum number of frames to process with face detection algorithm while matching subject using video camera. When liveness check is used this value must be at least 10 or more (Recommended range is 10 - 25)
- Enroll stream length – maximum number of frames to process with face detection algorithm while enrolling subject using video camera.
- Generalization threshold – similarity value that has to be mutually exceeded by each feature template used for generalization.
- Generalization image count – number of images to use for enrollment with generalization.
- Save enrolled images – write to disk images of subjects enrolled to face database.
- Flip video image horizontally – mirror horizontally image received from video camera.

- File name as record ID – when enrolling still image files, use file name without extension as face database record identifiers.

Menu commands

Menu command	Description
Source » ”Camera name”	Choose selected camera as video source.
Source » File	Select an image file as a source.
Jobs » Enroll	Enroll image to face database.
Jobs » Enroll with generalization	Enroll several generalized images to face database.
Jobs » Match	Search for matching image in face database.
Tools » Face detection preview	View face detection result overlaid on images.
Tools » Save image	Save image to disk.
Tools » Clear logs	Clear application log windows.
Tools » Empty database	Empty face database.
Tools » Options…	Display options dialog.
Help » About VeriLook…	Display information about VeriLook demo application.

Simple usage examples

In this section simple basic scenarios of using VeriLook algorithm demo application are described in a step by step fashion.

Enrolling from camera

1. First, camera to be used as the capture device must be selected from "source" menu in the toolbar. After that camera video input should be visible in the upper left pane of the program.

2. Faces found in the video stream are outlined in the capture image by colorful rectangles (the green rectangle outlines the face that best fits the matching requirements of the VeriLook algorithm in addition this face has its eyes marked by the program, and yellow rectangles show other faces found in the image).

3. To enroll a face from a video stream, "enroll" button in the toolbar can be used or option "enroll" from a system menu "jobs" can be selected. For this operation to succeed at least one face in the image must be present. Program will process a few frames and will enroll face into the database of the demo program from these frames and a dialog asking for the person to be enrolled id will be shown.

Matching from camera

1. First, camera to be used as the capture device must be selected from "source" menu in the toolbar. After that camera video input should be visible in the upper left pane of the program.

2. Faces found in the video stream are outlined in the capture image by colorful rectangles (the green rectangle outlines the face that best fits the matching requirements of the VeriLook algorithm in addition this face has its eyes marked by the program, and yellow rectangles show other faces found in the image).

3. To identify a face "match" button must be clicked or option "match" must be selected from a system menu "jobs". After this the face, that best suits the matching requirements of the VeriLook algorithm (it should be outlined by a green rectangle in the video input pane) will be matched against the database of the demo program and most probable candidate will be displayed in the bottom right pane of the program window.

Enrolling from file

1. First, file input as the capture device must be selected from "source" menu in the toolbar.

2. To enroll a face from a file "enroll" button must be pressed or "enroll" option selected from the system menu "jobs". After that a file open dialog should open in which a file to be opened must be selected. The image in the file will be displayed in the upper left pane of the window, with the found face outlined by a green rectangle (if such rectangle is absent it means that no face suitable for enrollment was found in the image) and a dialog asking for the person to be enrolled id will be shown. The outlined face will be enrolled to the demo program database.

Matching from file

1. First, file input as the capture device must be selected from "source" menu in the toolbar.

2. To identify a face from a file "match" button must be pressed or "match" option selected from the system menu "jobs". After that a file open should open in which a file to be opened must be selected. The image in the file will be displayed in the upper left pane of the window, with the found face outlined by a green rectangle (if such rectangle is absent, it means that no face, suitable for matching was found in the image). The outlined image will be matched agains the demo program database and most probable candidate will be displayed in the bottom right pane of the window.

Enrolling with generalization

Generalization enables face feature extraction from multiple images of the same person thus allowing more details to be precisely extracted, increasing the reliability of matching operations. To perform enrollment using generalization follow these steps:

1. First, select your desired input either file or web camera from the "source" menu in the toolbar.

2. From "jobs" menu in the toolbar select "enroll with generalization".

3. If you chose camera as your input source, the program will attempt number of distinct face detections from the video stream. If file as input was selected, program will open a standard file open dialog asking to select number of images of the same person. The number of files the program will ask or try to capture from video stream is set in the options dialog "generalization image count".

4. After the input images have been captured, the program will process all of them and extract generalized features. The last input image will be displayed in the top left pane of the window with the found face outlined by a green rectangle (if such rectangle is absent it means that no face suitable for enrollment was found in the images) and a dialog asking for the person to be enrolled id will be shown. Template generated from these input images will be enrolled to the programs database.

Matching threshold and similarity

VeriLook features matching algorithm provides value of features collections similarity as a result. The higher is similarity, the higher is probability that features collections are obtained from the same person.

Matching threshold is linked to false acceptance rate (FAR, different subjects erroneously accepted as of the same) of matching algorithm. The higher is threshold, the lower is FAR and higher FRR (false rejection rate, same subjects erroneously accepted as different) and vice a versa.

5.5 Experimental Results of Multimodal Biometric System

The performance metrics of a biometric system such as accuracy, throughput, and scalability can be estimated with a high degree of confidence only when the system is tested. The multimodal systems have been tested on databases containing more than 200 individuals. Further, multimodal biometric databases can be either true or virtual. In a true multimodal database, different biometric traits are collected from the same individual. Virtual multimodal databases contain records which are created by consistently pairing a user from one unimodal database with a user from another database. The creation of virtual users is based on the assumption that different biometric traits of the same person are independent.

The data has taken from over 200 different users vary from the ages 18-60 which includes both male and females.

The following table & chart show this research experiment result.

System	Failure to Enroll Rate
Fingerprint	2.0%
Face	1.0%

Table 5.4: Failure to Enroll Rate

The false acceptance and false rejection rates are calculated as follows:

FAR (t) = (1 – FTA) FMR (t)

FRR (t) = (1 – FTA) FNMR (t) + FTA

Where FTA is the failure to acquire rate, FNMR is the false non- match rate, and FMR is the false match rate. The false match and non-match rates are used to measure the accuracy of the matching process. *t* represents the decision threshold. The decision threshold is the value, set initially, to determine whether a user is accepted or rejected by the system, according to their matching score. The failure to acquire *rate* measures the proportion of attempts for which the system is unable to capture or locate a sufficient quality image. This may happen simply when the image that was captured doesn't meet the quality requirements of the system.

System	Failure to Acquire Rate
Fingerprint	1.5%
Face	1.0%

Table 5.5: Failure to Acquire Rate

The failure to acquire rate occurs such as not correctly positioning fingers on the fingerprint device. Failure to acquire rate is low. Face recognition has low failure to acquire rate compare to fingerprint recognition.

System	**False Acceptance Rate**
Fingerprint	2.5%
Face	1.5%

Table 5.6: False Acceptance Rate

The False Acceptance Rate (FAR) measures the proportion of falsely accepted person using fingerprint and face print. Face recognition has low FAR compare to fingerprint recognition.

System	**False Rejection Rate**
Fingerprint	2.5%
Face	6%

Table 5.7: False Rejection Rate

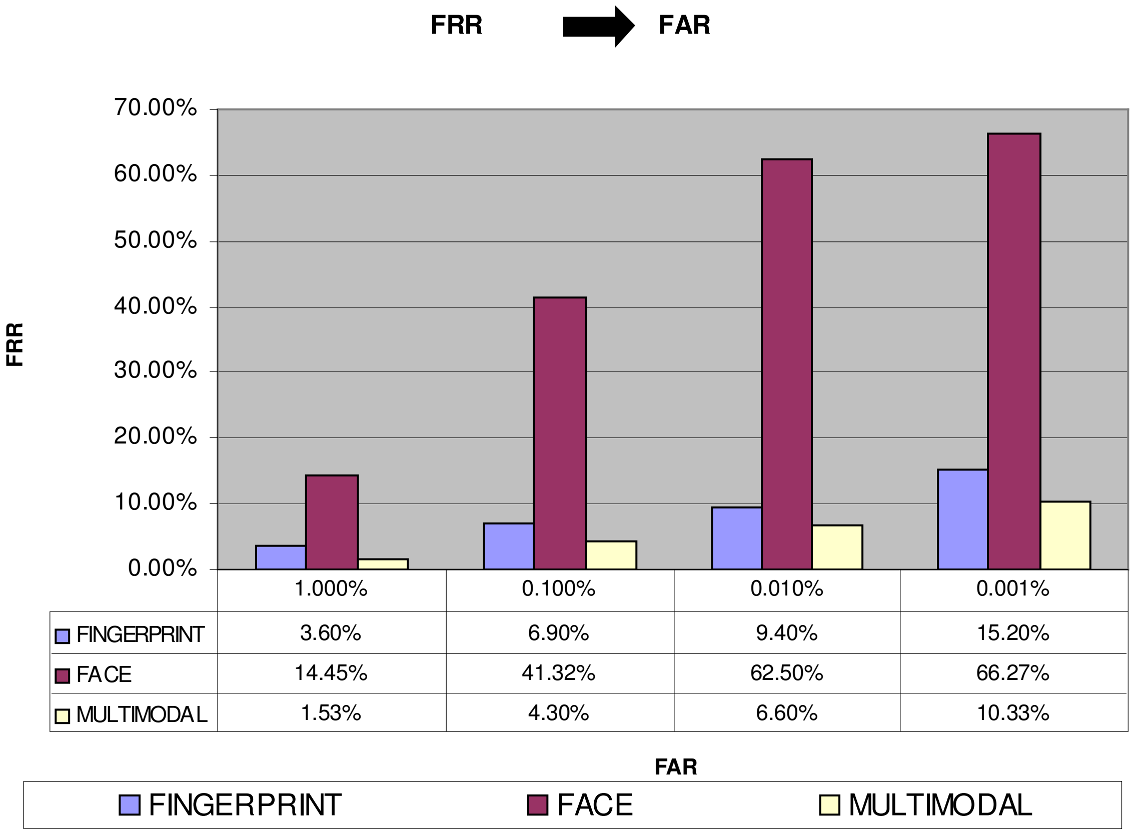
The False Rejection Rate (FRR) measures the proportion of falsely rejected person using fingerprint and face print. Face recognition has high FRR compare to fingerprint recognition.

The existing multimodal biometric authentication system provides accuracy up to 85.3% at a FAR of 0.001% as per the research study undertaken by Anil Jain.

False Accept Rate (FAR)	False Reject Rate (FRR)		
	Fingerprint	Face	Integration
1%	3.6%	14.45%	1.53%
0.1%	6.9%	41.32%	4.30%
0.01%	9.4%	62.5%	6.6%
0.001%	15.2%	66.27%	10.33%

Table 5.8: False Reject Rate vs. False Accept Rate in an integrated system

The above table shows result for single biometric trait and then integration of these two single multiple biometric traits. As the data shows single biometric has a high False Rejection Rate (FRR) while the integration of fingerprint and face has low FRR for the same False Acceptance Rate. The following chart shows a comparison of the data presented in the table. As from the chart we can say that the multimodal (integration) of the biometric trait has significantly improve the performance.



	1.000%	0.100%	0.010%	0.001%
FINGERPRINT	3.60%	6.90%	9.40%	15.20%
FACE	14.45%	41.32%	62.50%	66.27%
MULTIMODAL	1.53%	4.30%	6.60%	10.33%

Figure 5.9: A chart showing False Acceptance Rate(FAR) and False Rejection Rate (FRR) for Fingerprint, Face Recognition and Multimodal Biometric using fingerprint and face recognition.

The chart above shows the false reject rates (FRR) for various values of false accept rates (FAR) for face, fingerprint, and integrated face/fingerprint. The false rejection rate is lower for every false accept rate value for an integrated system.

References:

- Guide to Biometrics
  By Ruud Bolle, Jonathan H. Connell, Sharath Pankanti, Nalini K. Ratha, Andrew W. Senior, Springer
- Handbook of Face Recognition
  By S. Z. Li, Anil K. Jain, Springer
- Practical Biometrics: from Aspiration to Implementation
  By Julian Ashbourn, Springer
- Handbook of Fingerprint , By Davide Maltoni, Anil K. Jain, Dario Maio, Salil Prabhakar, Springer
- Newbold's Biometric Dictionary: For Military and Industry
  By Richard D. Newbold
- www.security.iitk.ac.in
- www.findbiometrics.com
- www.bayometric.com
- www.face-rec.org
- www.iiit.ac.in

Conclusion

This research has made exhaustive study of existing biometric authentication system. The study has concluded various challenges in the identification and verification of a human being. Although biometrics is becoming an integral part of the identity management systems, current biometric systems do not have 100% accuracy. Some of the factors that impact the accuracy of biometric systems include noisy input, non-universality, lack of invariant representation and non-distinctiveness. Further, biometric systems are also vulnerable to security attacks. A multimodal biometric system that integrates multiple biometric traits can overcome some of these limitations and achieve better performance.

Biometric methods used in research are fingerprint and facial recognition. Apart from fingerprint and face there are many newer biometric methods, which may be used, for identification and verification have not been included in this study. As single biometric devices may not suffice for authentication, so the use of multi-biometrics improves an authentication system.

Prior to choosing an adequate biometric method, one needs to carefully research biometric performance measurements. These measurements are important when we are balancing security and convenience. Biometric susceptibility is defined so that they can be moderate before clever attackers use them. This document serves to introduce and define security considerations for the use of biometric authentication system.

Further research for use of biometric systems should be done in the area of multi-biometrics. If additional research and testing (on combining different biometric methods together) is done in this area, we would then have sufficient information that would be useful in choosing the best biometric methods to combine together to form a strong system overall.

This thesis has examined the problem of authentication and verification of human being an organization. Several biometric techniques were reviewed. We analyze and design designs a multimodal biometric system that use fingerprint and face recognition for authentication and verification. Expansion of the designs proposed here is possible to accommodate advances in the area biometric technology and biometric authentication systems. Future developments in multimodal biometric technology should make one of these designs feasible and highly reliable.

The research has studied the existing biometric systems and identified the challenges in them. To overcome the negative sides of the challenges, this research focused on providing improved secured identity considering fingerprint & face recognition together under the banner of multimodel biometric authentication system. The research has given a proposed model to satisfy the research objectives and requirements. The experimental part of the research was the setup of multimodal biometric authentication system. This experimental prototype was tested with a sample of data to create database and analyze the data therein. The analysis of the data revealed

- Multimodal biometric authentication system gives better result than unimodal biometric authentication system.
- The false rejection rate (FRR) is lower for every false accept rate (FAR) value for an integrated system.
- The proposed design improves the FAR and FRR significantly.

Scope of future work

The experimental analysis, in this thesis, involves the two biometric trait: fingerprint and face. The results show that the performance of multimodal biometric systems can benefit from score level fusion.

The research documented in this thesis may be extended to build robust multimodal biometric system taking following considerations.

- Further research of multimodal biometric systems should be done with more biometric traits like fingerprint, face, voice, signature.
- It can be possible to done research using the same person's multiple instances like all the fingers of both hand of the same person.
- The research can extend by applying different algorithm for the same trait and then make fusion of it.
- One can consider different types of hardware devices like different types of fingerprint reader (optical, solid-state etc.) for identification and verification of a person.
- Fusion at the matching score level is the most popular approach to multimodal biometrics due to the ease in accessing and consolidating the scores generated by multiple matchers. However, fusion at the feature extraction level is expected to be more effective due to the richer source of information available at this level. Therefore, it is important to study the possibility of fusing information at this level.
- Soft biometric can be combined with multimodal biometric authentication system to make it better.