February, 2007
Thesis for the Degree of Doctor

Design of Multi-modal Biometrics System and Sensor Network Security System based on USN(Ubiquitous Sensor Network)

Graduate School of Chosun University
Department of Electronics Engineering
Jin Soo NOH

Design of Multi-modal Biometrics System and Sensor Network Security System based on USN(Ubiquitous Sensor Network)

유비쿼터스 센서 네트워크 기반의 다중생체인식 시스템과
센서 네트워크의 보안 시스템 설계

February, 2007

Graduate School of Chosun University

Department of Electronics Engineering

Jin Soo NOH

Design of Multi-modal Biometrics System and Sensor Network Security System based on USN(Ubiquitous Sensor Network)

指導敎授　李 康 鉉

이 論文을 工學博士學位申請 論文으로 提出함.

2006年　10月

朝鮮大學校 大學院

電 子 工 學 科

魯　鎭　守

魯鎭守의 博士學位論文을 認准함

委員長	朝鮮大學校 電子情報工科大學	敎 授 工學博士	朴 光 採	印
委員	漢陽大學校 情報通信大學, 大韓電子工學會	敎 授 工學博士 會 長	鄭 正 和	印
委員	JAPAN, University of the RyuKyu, College of Information Engineering	Professor Ph.D.	MORIKAZU NAKAMURA	印
委員	JAPAN, University of the RyuKyu, College of Information Engineering	Professor Ph.D.	姜 東 植	印
委員	朝鮮大學校 電子情報工科大學	敎 授 工學博士	李 康 鉉	印

2006年 12月

朝鮮大學校 大學院

Table of Contents

List of Tables

List of Figures

초 록

유비쿼터스 센서 네트워크 기반의 다중생체인식 시스템과 센서 네트워크의 보안 시스템 설계

이 름 : 노 진 수

지도교수 : 이강현 공학박사

소 속 : 조선대학교 대학원 전자공학과

최근 정보통신 기술의 비약적인 발전은 기존의 계산기로서의 컴퓨터가 아닌 정보단말기로서의 컴퓨터로 발전하여 더욱더 우리의 생활에 밀접한 영향을 주고 있다. 이런 기술의 진보는 유비쿼터스 컴퓨팅(Ubiquitous Computing)이라는 새로운 정보통신 혁명을 이끌게 되었고, 이런 사회발전의 흐름과 끊임없이 환경을 인간 친화적으로 바꾸고 싶어 하는 인간의 욕구가 맞물려 무선 센서 네트워크 (WSN: Wireless Sensor Network)의 필요성이 증대되어지고 있다. 무선 센서 네트워크란 센서를 통하여 정보를 수집하고 수집된 정보를 가공할 수 있는 프로세서가 달려 있으며 이를 전송할 수 있는 무선 송수신기를 갖춘 소형장치, 즉, 센서 노드로 구성된 네트워크를 의미하며, 기존의 네트워크와 다르게 의사소통의 수단이 아니라 환경에 대한 정보를 수집하는 것을 그 목적으로 한다. 이에 따라, 인간, 사물 그리고 컴퓨터가 유기적으로 연계되어 다양하고 편리한 서비스를 제공해 주는 유비쿼터스 컴퓨팅 환경에서, 외부 환경의 감지와 제어 기능을 수행하는 센서 네트워크 기술이 활발히 연구되고 있다.

센서 네트워크 기술의 발전에 따라 물체의 위치추적, 사물인식 및 생체 정보를

이용한 개인 식별 기술 등이 발전하고 있다. 그 중 개인 식별 기술은 정보화 기술의 발달로 기존의 개인인증번호(Personal Identification Number)나 패스워드(Password)를 이용한 사용자 인증방식의 문제점 및 한계점이 노출됨에 따라 이를 해결하기 위한 기술로 연구가 활발히 진행 중이다. 개인 식별에 사용할 수 있는 신체적 특징은 각 사람마다 유일해야 하고 측정 시간에 관계없이 항상 불변하는 특성을 가져야 한다. 이러한 조건을 충족하는 신체적 특징으로는 지문을 비롯하여 얼굴, 눈의 홍채 및 망막, 손등의 정맥, 장문, 음성 등 다양한 것들이 존재할 수 있으며, 현재 이러한 특징들을 수집 할 수 있는 여러 가지 센서와 인증을 위한 생체인식 알고리즘이 연구되고 있다.

이러한 연구를 바탕으로 하여 본 논문에서는 유비쿼터스 센서 네트워크 기반에서 활용되어질 수 있는 다중 생체인식 시스템과 센서 네트워크의 보안성능을 향상시킬 수 있는 시스템을 제안하였다. 제안된 다중 생체인식 시스템을 구현하기 위하여 UStar-2400을 기본 플랫폼으로 사용하여 음성 신호를 수집할 수 있는 무선 음성인식 센서를 설계 및 구현하였으며, TX-32CS 무선 카메라를 사용하여 영상 신호를 수집할 수 있는 시스템을 구현 하였다. 구현된 하드웨어 플랫폼 상에서 동작되어지는 다중 생체인식 시스템은 주성분 분석법을 이용한 얼굴인식, Hu 불변모멘트를 이용한 장문인식 그리고 심리음향 모델을 이용한 음성 인식으로 나누어진다. 또한 BIBD(Balanced Incomplete Block Design) 코드와 웨이블렛(Wavelet) 필터를 사용하여 센서 네트워크의 보안성능을 향상시킬 수 있는 알고리즘을 제안하였다.

본 논문에서는 제안된 다중 생체인식 시스템의 성능 평가를 위해 얼굴, 장문 그리고 음성 인식 시스템의 FAR과 FRR을 측정하였으며, 센서 네트워크의 보안성능 측정을 위하여 공모공격 및 외부잡음에 대한 강인성을 측정하였다. 이를 위해 2장에서는 유비쿼터스 센서 네트워크 환경에서 이루어지는 생체 인식 시스템에 대하여 알아보고, 3장에서는 본 논문에서 얼굴, 장문 그리고 음성 인식에 사용된 알고리즘에 대하여 설명하겠다. 4장에서는 제안 알고리즘의 하드웨어와 소프트웨어의 구현 방법을 기술한다. 그리고 5장에서 제안된 알고리즘의 성능 측정 및 결과 검토를 하고 마지막 6장에서 결론과 향후 연구방향에 대해 고찰한다.

Chapter 1. Introduction

Recently a breakthrough in information and communications technology has transformed computers used in computations into information terminals that affect our lives intimately. Such advances led to a new kind of revolution in information and communications, called ubiquitous computing. Need for a WSN (Wireless Sensor Network) is being raised to keep up with these social advances and to satisfy basic human desires to continually change their environment to make it more hospitable. A wireless sensor network is made up of sensor nodes which are small devices that consist of a sensor, a processor for handling the sensed information and a wireless transmitter and receiver to transmit this information. Unlike existing networks that are used for communication, wireless sensor networks are used to collect information about the environment. As a result, active researches are being carried out for sensor network technology that can sense the external environments and carry out the control functions [1,2] in an ubiquitous computing environment which provide diverse and convenient services by organically linking humans, objects and computers.

Due to advances in wireless communication technology and electronic device technology, there has been a rapid increase in interest of wireless sensor networks that are made up of low cost, low power, multi-function sensor nodes [3,4]. Wireless sensor network is a technology that can replace the existing sensor network with a wireless network in which each sensor node contains sensing, data processing and multi-hop networking functions. This means an improvement in information processing capability has been added to existing sensors [5～7]. Due to such improvements in information processing capability, WSN is being applied in sensing and tracking of military information, environmental monitoring, patient observation, smart environmental areas, electronic commerce and biometric authentication.

The rapid growth in the use of e-commerce and online banking applications requires

reliable and automatic personal identification for effective security control. Traditional, automatic, personal identification can be divided into two categories: token-based, such as a physical key, an ID card, and a passport, and knowledge-based, such as a password. However, these approaches have some limitations. In the token-based approach, the "token" can be easily stolen or lost. In the knowledge-based approach, to some extent, the "knowledge" can be guessed or forgotten. Therefore, biometric personal identification is highlighted as a powerful method for solving these problems [8]. The physiological characteristics used in personal identification should be only available for each individual and they should have features that are consistent over time. The biometric system is concerned with identifying a person by the physiological characteristics or using some aspect of the behavior such as fingerprint, iris pattern, retina, face, palmprint, voice, gait, signature, and gesture. Currently, there are a wide variety of ongoing studies on biometrics system which recognize an individual by identifying the specific characteristics for that individual [9].

Based on these studies, we proposed multi-modal biometrics system that can be used in the ubiquitous sensor network and channel establishment algorithm that can improve the performance of sensor network security. To implement the proposed system, we designed wireless speech recognition sensor which can collect speech signals and image capture system using TX-32CS [10] wireless camera. Ustar-2400 [11] is used as a basic platform in this study. The implemented multi-modal biometrics on the hardware platform are consist of PCA (Principle Component Analysis) [12] used in face recognition and HU invariant moment [13] used in Palmprint recognition. And psychoacoustic model [14] was used for the speech recognition. Moreover, BIBD (Balanced Incomplete Block Design) [15] code and wavelet filter [16] were used to improve the ability of sensor network security.

In this paper, FAR (False Acceptance Rate) and FRR (False Rejection Rate) were measured to evaluate the performance of the proposed multi-modal biometrics system

such as face, palmprint and speech. The robustness towards collusion attack and external noise generated in wireless environment was measured to evaluate the security ability of sensor network. Biometrics system which was used in ubiquitous sensor network will be explained in Section 2. We will explain some algorithms for speech, face and palmprint authentication in Section 3. H/W and S/W implementation method for the recognition sensors and the proposed algorithm will be described in Section 4. The performance results of proposed algorithm will be examined in Section 5. Finally, the conclusion and future research direction will be considered in Section 6.

Chapter 2. Related Works

2.1 Biometric System

A biometric system is essentially a pattern recognition system that recognizes a person by determining the authenticity of a specific physiological and/or behavioral characteristic possessed by that person. An important issue in designing a practical biometric system is to determine how an individual is recognized. Depending on the application context, a biometric system may be called either a verification system or an identification system [17]:

- A verification system authenticates a person's identify by comparing the captured biometric characteristic with her own biometric template pre-stored in the system. It conducts one-to-one comparison to determine whether the identity claimed by the individual is true. A verification system either rejects or accepts the submitted claim of identity.
- An identification system recognizes an individual by searching the entire template database for a match. It conducts one-to-many comparisons to establish the identity of the individual. In an identification system, the system establishes a subject's identity without the subject having to claim an identity.

The term authentication is also frequently used in the biometric field, sometimes as a synonym for verification; actually, in the information technology language, authenticating a user means to let the system know the user identity regardless of the mode [18,19].

The block diagrams of a verification system and an identification system are depicted in figure 1; user enrollment, which is common to both tasks is also graphically illustrated. The enrollment module is responsible for registering individuals in the

biometric system database. During the enrollment phase, the biometric characteristic of an individual is first scanned by a biometric reader to produce a raw digital representation of the characteristic. A quality check is generally performed to ensure that the acquired sample can be reliably processed by successive stages. In order to facilitate matching, the raw digital representation is usually further processed by a feature extractor to generate a compact but expressive representation, called a template. Depending on the application, the template may be stored in the central database of the biometric system or be recorded on a magnetic card or smartcard issued to the individual. The verification task is responsible for verifying individuals at the point of access. During the operation phase, the user's name of PIN (Personal Identification Number) is entered through a keyboard; the biometric reader captures the characteristic of the individual to be recognized and converts it to a digital format, which is further processed by the feature extractor to produce a compact digital representation. The resulting representation is fed to the feature matcher, which compares it against the template of a single user (retrieved from the system DB based on the user's PIN). In the identification task, no PIN is provided and the system compares the representation of the input biometric against the templates of all the users in the system database; the output is either the identity of an enrolled user or an alert message such as "user not identified." Because identification in large databases is computationally expensive, classification and indexing techniques are often deployed to limit the number of templates that have to be matched against the input [20].

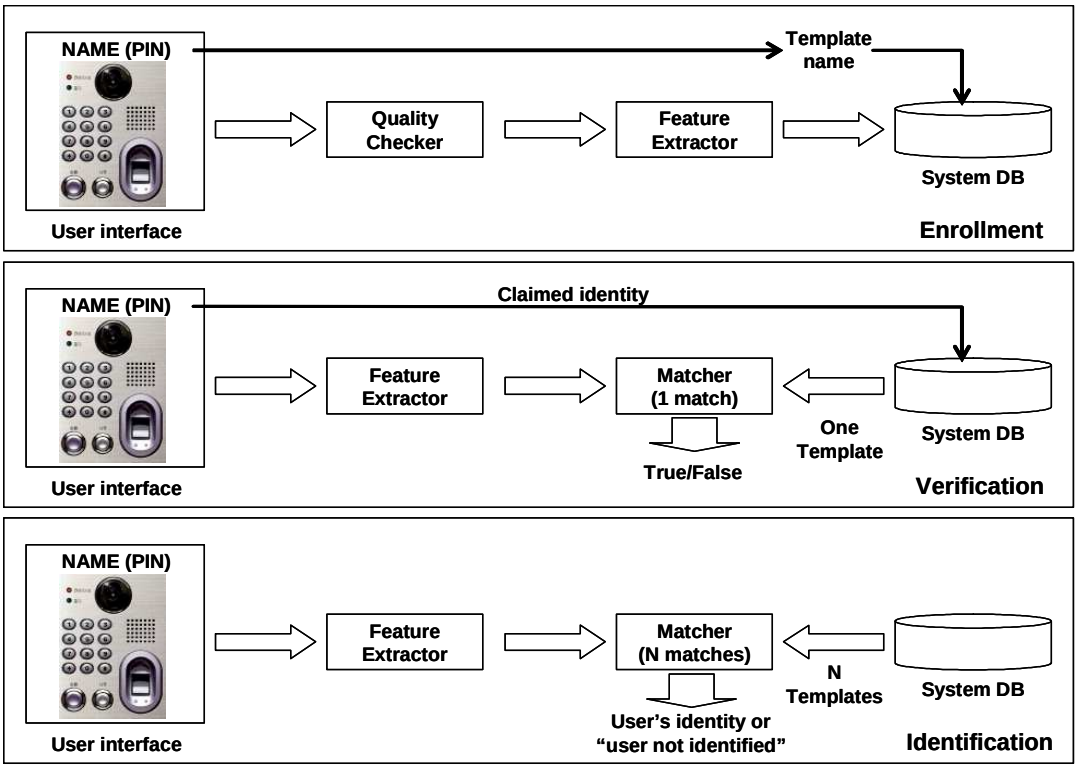


Figure 1. Block diagrams of enrollment, verification, and identification tasks.

Depending on the application domain, a biometric system could operate either as an on-line system or an off-line system. An on-line system requires the recognition to be performed quickly and an immediate response is imposed (e.g., a computer network logon application). On the other hand, an off-line system usually does not require the recognition to be performed immediately and a relatively long response delay is allowed (e.g., an employee background check application). Typically, on-line systems are fully automatic and require that the biometric characteristic be captured using a live-scan scanner, the enrollment process be unattended, there be no quality control, and the matching and decision be fully automatic. Off-line systems, however, are typically semi-automatic, where the biometric acquisition could be through an off-line scanner (e.g., scanning a fingerprint image form a latent or inked fingerprint card), the enrollment may be supervised (e.g., when a criminal is "booked," a forensic expert or a

police officer may guide the fingerprint acquisition process), a manual quality check may be performed to ensure good quality acquisition, and the matcher may return a list of candidates which are then manually examined by a forensic expert to arrive at a final decision[21～23].

An application could operate either in a positive or a negative recognition mode:

- In a positive recognition application, the system establishes whether the person is who he (implicitly or explicitly) claims to be. The purpose of a positive recognition is to prevent multiple people from using the same identity. For example, if only Alice is authorized to enter a certain secure area, then the system will grant access only to Alice. If the system fails to match the enrolled template of Alice with the input, a rejection results; otherwise, an acceptance results;
- In a negative recognition application, the system establishes whether the person is who he denies being. The purpose of negative recognition is to prevent a single person from using multiple identities. For example, if Alice has already received welfare benefits and now she claims that she is Becky and would like to receive the welfare benefits of Becky, the system will establish that Becky is not who she claims to be. If the system fails to match the input biometric of Becky with a database of people who have already received benefits, an acceptance results; otherwise, a rejection results.

Note that although the traditional methods of user authentication such as passwords, PINs, keys, and tokens may work for positive recognition, negative recognition can only be established through biometrics. Furthermore, positive recognition application can operate both in verification or identification mode, but negative recognition applications cannot work in verification mode: in fact, the system has to search the entire archive

to prove that the given input is not already present [24].

A biometric system can be classified according to a number of other application-dependent characteristics. Wayman (1999a) [25] suggests that all the biometric applications may be classified into categories based on their characteristics:

1. Cooperative versus non-cooperative,
2. Overt versus covert,
3. Habituated versus non-habituated,
4. Attended versus non-attended,
5. Standard versus non-standard operating environment,
6. Public versus private, and
7. Open versus closed.

Cooperative versus non-cooperative dichotomy refers to the behavior of the impostor in interacting with the system. For example, in a positive recognition system, it is in the best interest of an impostor to cooperate with the system to be accepted as a valid user. On the other hand, in a negative recognition system, it is in the best interest of the impostor not to cooperate with the system so that she does not get recognized. Electronic banking is an example of a cooperative application whereas an airport application to identify terrorists who will try to break the system is an example of a non-cooperative application.

If a user is a ware that he is being subjected to a biometric recognition, the application is categorized as over. If the user is unaware, the application is covert. Facial recognition can be used in a covert application while fingerprint recognition cannot be used in this mode (except for criminal identification based on latent fingerprints). Most commercial uses of biometrics are overt, whereas government, forensic, and surveillance applications are typically covert. Also, most verification

applications are overt whereas identification applications generally fall in the covert category.

Habituated versus non-habituated use of a biometric system refers to how often the enrolled users are subjected to biometric recognition. For example, a computer network logon application typically has habituated users (after an initial "habituation" period) due to their use of the system on a regular basis. However, a driver's license application typically has non-habituated users since a driver's license is renewed only once in several years. This is an important consideration when designing a biometric system because the familiarity of users with the system affects recognition accuracy.

Attended versus non-attended classification refers to whether the process of biometric data acquisition in an application is observed, guided, or supervised by a human (e.g., a security officer). Furthermore, an application may have an attended enrollment but non-attended recognition. For example, a banking application may have a supervised enrollment when an ATM card is issued to a user but the subsequent uses of the biometric system for ATM transactions will be non-attended. Non-cooperative applications generally require attended operation.

Standard versus non-standard environments refer to whether the system is being operated in a controlled environment (such as temperature, pressure, moisture, lighting conditions, etc.). Typically, indoor applications such as computer network logon operate in a controlled environment whereas outdoor applications such as keyless car entry or parking lot surveillance operate in a non-standard environment. This classification is also important for the system designer as a more rugged biometric sensor is needed for a non-standard environment. Similarly, infrared face recognition may be preferred over visible-band face recognition for outdoor surveillance at night.

Public or private dichotomy refers to whether the users of the system are customers or employees of the organization deploying the biometric system. For example, a network logon application is used by the employees and managed by the information

technology manager of the same company. Thus it is a private application. The use of biometric data in conjunction with electronic identity cards is an example of a public application.

Closed versus open systems refers to whether a person's biometric template is used for a single or multiple applications. For example, a user may use a fingerprint-based recognition system to enter secure facilities, for computer network logon, electronic banking, and ATM. Should all these applications use separates (databases) for each application, or should they all access the same template. A closed system may be based on a proprietary template whereas an open system will need standard data formats and compression methods to exchange and compare information between different systems [26,27].

Note that the most popular commercial applications have the following attributes: cooperative, overt, habituated, attended enrollment and non-attended recognition, standard environment, close, and private.

2.1.1 A comparison of Various Biometrics

Any human physiological and/or behavioral characteristic can be used as a biometric identifier to recognize a person as long as it satisfies these requirements [28]:

- Universality: which means that each person should have the biometric.
- Distinctiveness: which indicates that any two persons should be sufficiently different in terms of their biometric identifiers.
- Permanence: which means that the biometric should be sufficiently invariant over a period of time.
- Collectability: which indicates that the biometric can be measured quantitatively.

However, in a practical biometric system, there are a number of other issues that

should be considered, including:

- Performance: which refers to the achievable recognition accuracy, speed, robustness, the resource requirements to achieve the desired recognition accuracy and speed, as well as operational or environmental factors that affect the recognition accuracy and speed.
- Acceptability: which indicates the extent to which people are willing to accept a particular biometric identifier in their daily lives.
- Circumvention: which reflects how easy it is to fool the system by fraudulent methods.

A practical biometric system should have acceptable recognition accuracy and speed with reasonable resource requirements, harmless to the users, accepted by the intended population, and sufficiently robust to various fraudulent methods.

A number of biometric identifiers are in use in various applications. Each biometric has its strengths and weaknesses and the choice typically depends on the application. No single biometric is expected to effectively meet the requirements of all the applications. The match between a biometric and an application is determined depending upon the characteristics of the application and the properties of the biometric.

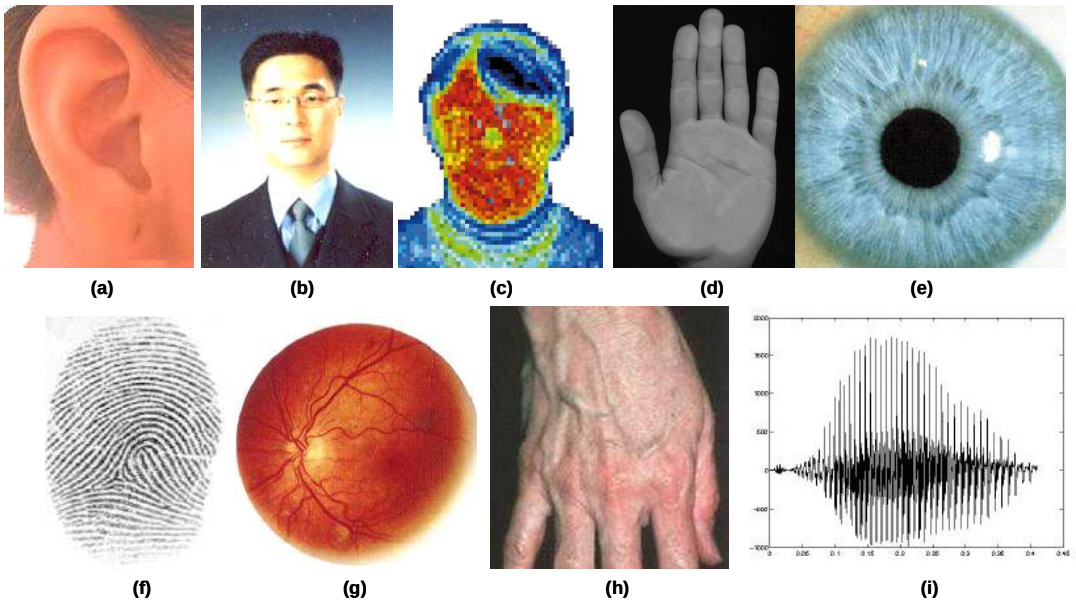


Figure 2. Some of the biometrics are shown: a) ear, b)face, c)facial thermogram, d)palmprint, e)iris, f)fingerprint, g)retina, h)hand vein, and i) voice.

When choosing a biometric for an application the following issues have to be addressed:

- Does the application need verification or identification? If an application requires an identification of a subject from a large database, it needs a scalable and relatively more distinctive biometric (e.g., fingerprint, iris, palmprint, or DNA).
- What are the operational modes of the application? For example, whether the application is attended or unattended, whether the users are habituated to the given biometrics, whether the application is covert or overt, whether subjects are cooperative or non-cooperative, and so on.
- What is the storage requirement of the application? For example, an application that performs the recognition at a remote server may require a small template size.
- How stringent are the performance requirements? For example, an application that demands very high accuracy needs a more distinctive biometric.

- What types of biometrics are acceptable to the users? Different biometrics are acceptable in applications deployed in different demographics depending on the cultural, ethical, social, religious, and hygienic standards of that society. The acceptability of a biometric in an application is often a compromise between the sensitivity of a community to various perceptions/taboos and the value/convenience offered by biometrics-based recognition.

A brief introduction to the most common biometrics is provided below.

- DNA: Deoxyribo Nucleic Acid (DNA) is the one-dimensional ultimate unique code for one's individuality, except for the fact that identical twins have identical DNA patterns. It is, however, currently used mostly in the context of forensic applications for person recognition. Several issues limit the utility of this biometric for other applications:

  i) Contamination and sensitivity: it is easy to steal a piece of DNA from an unsuspecting subject that can be subsequently abused for an ulterior purpose.
  ii) Automatic real-time recognition issues: the present technology for DNA matching requires cumbersome chemical methods involving an expert's skills and is not geared for on-line non-invasive recognition.
  iii) Privacy issues: information about susceptibilities of a person to certain diseases could be gained from the DNA pattern and there is a concern that the unintended abuse of genetic code information may result in discrimination, for example, in hiring practices.

- Ear: It is known that the shape of the ear and the structure of the cartilaginous tissue of the pinna are distinctive. The features of an ear are not expected to be

unique to an individual. The ear recognition approaches are based on matching the distance of salient points on the pinna from a landmark location on the ear.

- Face: The face is one of the most acceptable biometrics because it is one of the most common methods of recognition that humans use in their visual interactions. In addition, the method of acquiring face images is non intrusive. Facial disguise is of concern in unattended recognition applications. It is very challenging to develop face recognition techniques that can tolerate the effects of aging, facial expressions, slight variations in the imaging environment, and variations in the pose of the face with respect to the camera.
- Facial, hand, and hand vein infrared thermograms: The pattern of heat radiated by the human body is a characteristic of each individual body and can be captured by an infrared camera in an unobtrusive way much like a regular photograph. The technology could be used for covert recognition and could distinguish between identical twins. A thermogram-based system is non-contact and non-invasive but sensing challenges in uncontrolled environments, where heat-emanating surfaces in the vicinity of the body, such as, room heaters and vehicle exhaust pipes, may drastically affect the image acquisition phase. A related technology using near-infrared imaging is used to scan the back of a clenched fist to determine hand vein structure. Infrared sensors are prohibitively expensive which is a factor inhibiting widespread use of the thermograms.
- Gait: Gait is the peculiar way one walks and is a complex spatio-temporal biometric. Gait is not supposed to be very distinctive, but is sufficiently characteristic to allow verification in some low-security applications. Gait is a behavioral biometric and may not stay invariant, especially over a large period of time, due to large fluctuations of body weight, major shift in the body weight, major injuries involving joints or brain, or due to inebriety. Acquisition of gait is similar to acquiring facial pictures and hence it may be an acceptable biometric.

Because gait-based systems use a video-sequence footage of a walking person to measure several different movements of each articulate joint, it is computing and input intensive.

- Hand and finger geometry: Some features related to a human hand are relatively invariant and peculiar to an individual. The image acquisition system requires cooperation of the subject and captures frontal and side view images of the palm flatly placed on a panel with outstretched fingers. The representational requirements of the hand are very small, which is an attractive feature for bandwidth-and memory-limited systems. Due to its limited distinctiveness, hand geometry-based systems are typically used for verification and do not scale well for identification applications. Finger geometry systems may be preferred because of their compact size.
- Iris: Visual texture of the human iris is determined by the chaotic morphogenetic processes during embryonic development and is posited to be distinctive for each person and each eye (Daugman, 1999a). An iris image is typically captured using a non-contact imaging process. Capturing an iris image involves cooperation from the user, both to register the image of iris in the central imaging area and to ensure that the iris is at a predetermined distance from the focal plane of the camera. The iris recognition technology is believed to be extremely accurate and fast.
- Keystroke dynamics: It is hypothesized that each person types on a keyboard in a characteristic way. This behavioral biometric is not expected to be unique to each individual but it offers sufficient discriminatory information to permit identity verification. Keystroke dynamics is a behavioral biometric; for some individuals, one may expect to observe large variations from typical typing patterns. The keystrokes of a person using a system could be monitored unobtrusively as that person is keying in information.

- Odor: It is known that each object exudes an odor that is characteristic of its chemical composition and could be used for distinguishing various objects. A whiff of air surrounding an object is blown over an array of chemical sensors, each sensitive to a certain group of compounds. A component of the odor emitted by a human body is distinctive to a particular individual. It is not clear if the invariance in the body odor could be detected despite deodorant smells and varying chemical composition of the surrounding environment.
- Signature: The way a person signs his name is known to be a characteristic of that individual. Although signatures require contact and effort with the writing instrument, they seem to be acceptable in many government, legal, and commercial transactions as a method of verification. Signatures are a behavioral biometric that change over a period of time and are influenced by physical and emotional conditions of the signatories. Signatures of some people vary a lot: even successive impressions of their signature are significantly different. Furthermore, professional forgers can reproduce signatures to fool the unskilled eye.
- Voice: Voice capture is unobtrusive and voice print is an acceptable biometric in almost all societies. Voice may be the only feasible biometric in applications requiring person recognition over a telephone. Voice is not expected to be sufficiently distinctive to permit identification of an individual from a large database of identities. Moreover, a voice signal available for recognition is typically degraded in quality by the microphone, communication channel, and digitizer characteristics. Voice is also affected by a person's health, stress, emotions, and so on. Besides, some people seem to be extraordinarily skilled in mimicking others.

These various biometric identifiers described above are compared in Table 1.

Table 1. Comparison of biometric technologies.

Biometric identifier	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention
DNA	H	H	H	L	H	L	L
Ear	M	M	H	M	M	H	M
Face	H	L	M	H	L	H	H
Facial thermogram	H	H	L	H	M	H	L
Fingerprint	M	H	H	M	H	M	M
Gait	M	L	L	H	L	H	M
Hand geometry	M	M	M	H	M	M	M
Hand vein	M	M	M	M	M	M	L
Iris	H	H	H	M	H	L	L
Keystroke	L	L	L	M	M	M	M
Odor	H	H	H	L	M	M	L
Retina	H	H	M	L	H	L	L
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

The data are based on the perception of the authors. High, Medium, and Low are denoted by H, M, and L.

2.1.2 Biometric System Errors

For simplicity of exposition, the following discussion focuses on fingerprints, although it is valid for any other biometric identifier. The response of a matcher in a fingerprint recognition system is typically a matching scores (without loss of generality, ranging in

the interval [0, 1]) that quantifies the similarity between the input and the database template representations. The closer the score is to 1, the more certain is the system that the two fingerprints come from the same finger; the closer the score is to 0, the smaller is the system confidence that the two fingerprints come from the same finger. The system decision is regulated by a threshold *t*: pairs of fingerprints generating scores higher than or equal to *t* are inferred as matching pairs (i.e., belonging to the same finger); pairs of fingerprints generating scores lower than *t* are inferred as non-matching pairs [29,30].

A typical biometric verification system commits two types of errors: mistaking biometric measurements from two different fingers to be from the same finger and mistaking two biometric measurements from the same finger to be from two different fingers. Note that these two types of errors are also often denoted as false acceptance and false rejection; a distinction has to be made between positive and negative recognition; in positive recognition systems a false match determines the false acceptance of an impostor, whereas a false non-match causes the false rejection of a genuine user. On the other hand, in a negative recognition application, a false match results in rejecting a genuine request, whereas a false non-match results in falsely accepting an impostor attempts. The notation "false match/false non-match" is not application dependent and therefore, in principle, is preferable to "false acceptance/false rejection." However, the use of false acceptance rate (FAR) and false rejection rate (FRR) is more popular and largely used in the commercial environment.

2.1.2.1 Verification system errors

From the design perspective, the biometric verification problem can be formulated as follows. Let the stored biometric template of a person be represented as *T* and the acquired input for recognition be represented by *I*. Then the null and alternate hypotheses are [31]:

$H_0 : I \neq T$, input does not come from the same person as the template;

$H_1 : I \neq T$, input comes from the same person as the template.

The associated decisions are as follows.

D_0: person is not who she claims to be;

D_1: person is who she claims to be.

The verification involves matching *T* and *I* using a similarity measure *s (T,I)*. If the matching score is less than the system threshold *t*, then decide D_0, else decide D_1. The above terminology is borrowed from communication theory, where the goal is to detect a message in the presence of noise H_0 is the hypothesis that the received signal is noise alone, and H_1 is the hypothesis that the received signal is message plus the noise. Such a hypothesis testing formulation inherently contains two types of errors:

Type I : false match (D_1 is decided when H_0 is true);

Type II : false non-match (D_0 is decided when H_1 is true);

False Match Rate (FMR) is the probability of type I error and False Non-Match Rate (FNMR) is the probability of type II error:

$$FMR = P(D_1|H_0 = true);$$

$$FNMR = P(D_0|H_1 = true).$$

Note that (1-FNMR*)* is also called the power of the hypothesis test.

To evaluate the accuracy of a biometric system one must collect scores generated from a number of fingerprint pairs from the same finger, and scores generated from a number of fingerprint pairs from different fingers. Figure 3 graphically illustrates the

computation of FMR and FNMR over genuine and impostor distributions:

$$FNMR = \int_0^t p\,(s|H_1 = true\,)ds,$$

$$FMR = \int_t^1 p\,(s|H_0 = true\,)ds,$$

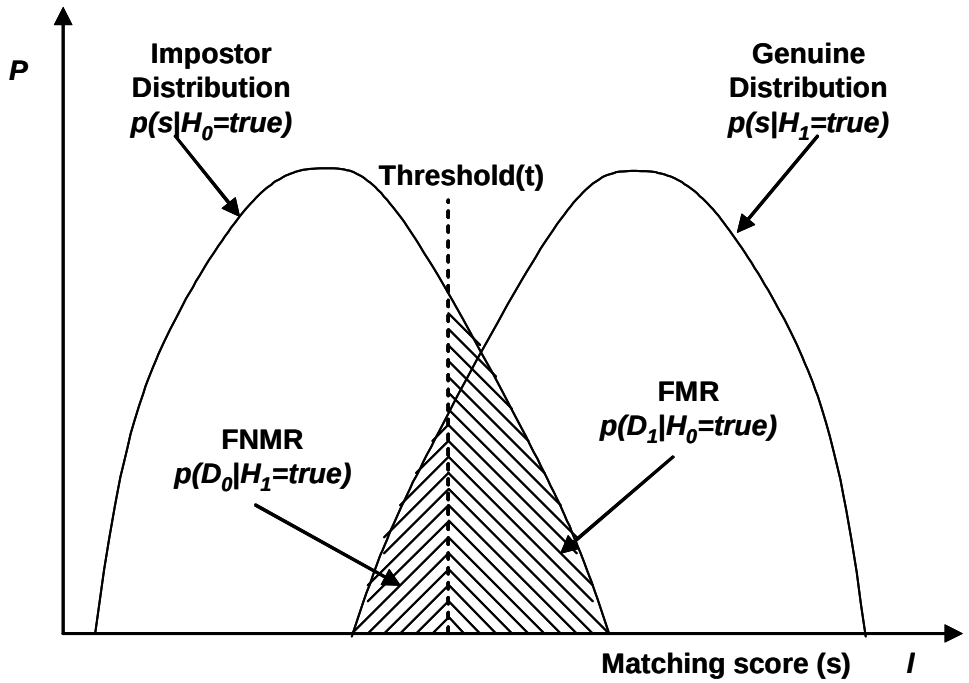


Figure 3. FMR and FNMR for a given threshold *t* are displayed over the genuine and impostor score distributions.

There is a strict tradeoff between FMR and FNMR in every biometric system. In fact, both FMR and FNMR are functions of the system threshold *t*, and we should, therefore, refer them as FMR*(t)* and FNMR*(t)*, respectively. If *t* is decreased to make the system more tolerant with respect to input variations and noise, then FMR*(t)* increases; vice versa, if *t* is raised to make the system more secure, then FNMR*(t)*

increases accordingly. A system designer may not know in advance the particular application for which the system may be used. So it is advisable to report system performance at all operating points. This is done by plotting a Receiver Operating Characteristic (ROC) curve. A ROC curve is a plot of FMR against of (1-FNMR) for various decision thresholds. In the figure 4, a through c show examples of score distributions, FMR*(t)* and FNMR*(t)* curves, and a ROC curve, respectively.

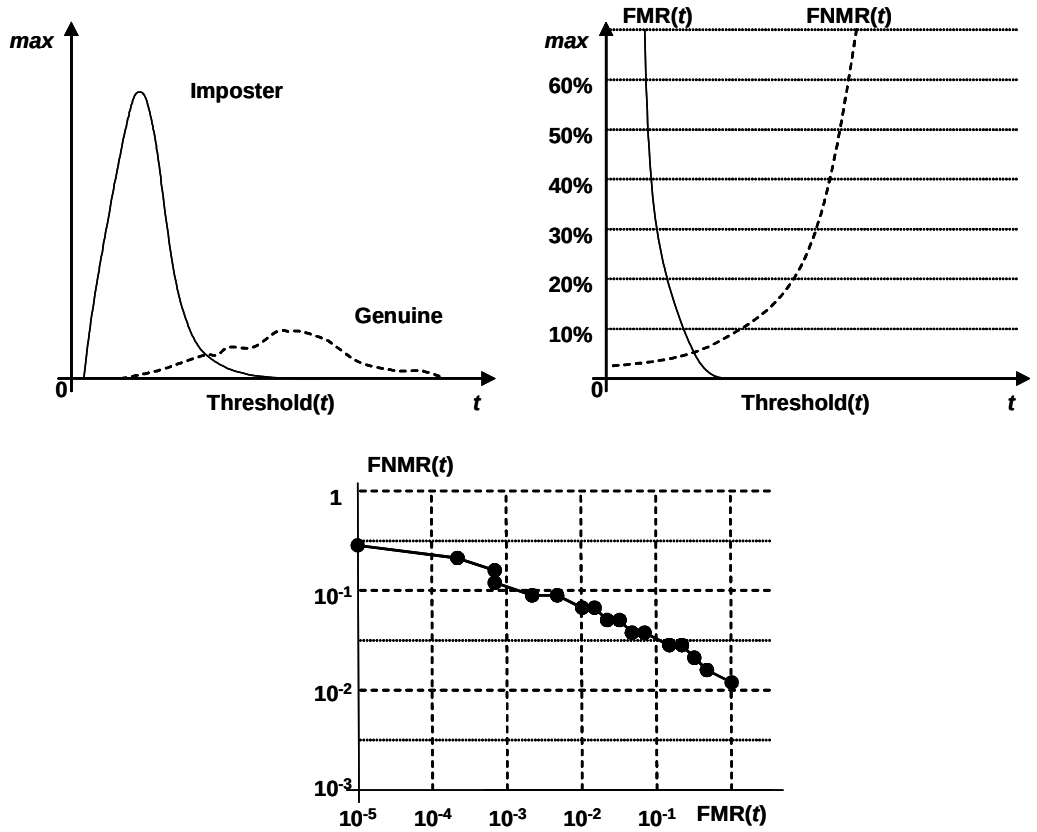


Figure 4. Evaluation of a fingerprint verification algorithm over FVC2002 (Maio.et al., 2002b) [36] database DB1: a) genuine and impostor distributions were computed from 2800 genuine pairs and 4950 impostor pairs; b) FMR*(t)* and FNMR*(t)* are derived from the score distribution in a); c) ROC curve is derived from the FMR*(t)* and FNMR*(t)* curves in b).

The practical performance requirements of a biometric system are bery mech application related. From the viewpoint of system accuracy, an extremely low false non-match rate may be the primary objective. For example, in some forensic applications such as criminal identification, it is the false non-match rate that is a major concern and not the false match rate: that is, we do not want to miss a criminal even at the risk of manually examining a large number of potential matches identified by the biometric system. In forensic applications, it is the human expert that will make the final decision anyway. At the other extreme, a very low false match rate may be the most important factor in a highly secure access control application, where the primary objective is not to let in any impostors although we are concerned with be possible inconvenience to legitimate users due to a high false non-match rate. In between these two extremes are several civilian applications, where both false match rate and false non-match rate need to be considered. For example, in applications such as an ATM card verification a false match means a loss of several hundred dollars whereas a high false non-match rate may irritate the customers. Figure 5 graphically depicts the FMR and FNMR tradeoff preferred by different types of applications [32,33].

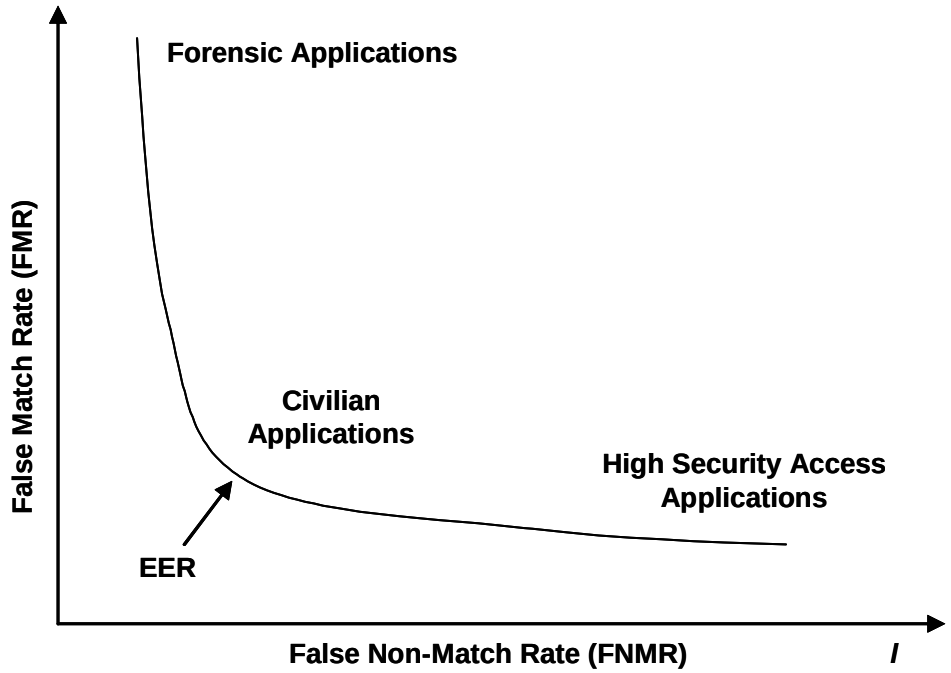


Figure 5. Typical operating points of different applications displayed on an ROC curve.

2.1.2.2 Identification System Errors

How do the definitions of errors introduced above for fingerprint verification extend to fingerprint identification? Under some simplifying assumptions, an estimation of the performance in the identification mode can be inferred by the error estimates in the verification mode.

Let us assume that no indexing/retrieval mechanism is available, and that a single template for each user is present in the database. Let $FNMR_N$ and FMR_N denote the identification false non-match rate and false match rate, respectively, then:

- $FNMR_N$ = FNMR; in fact, the probability of falsely non-matching the input against the user template is the same as in verification mode;
- $FMR_N = 1-(1-FMR)^N$; in fact, a false match occurs when the input falsely matches one or more templates in the database. FMR_N is then computed as one minus the

probability that no false match is made with any of the database templates, In the above expression (1-FMR) is the probability that the input does not falsely match a single template, and $(1\text{-FMR})^N$ is the probability that it does not falsely match any of the database templates. If FMR is very small, then the above expression can be approximated by $FMR_N \cong N \bullet FMR$, and therefore we can state that the probability of false match increases linearly with the size of the database.

This result has serious implications for the design of large-scale identification systems. Usually, computation speed is perceived as the biggest problem in scaling an identification application. Actually, accuracy scales even worse than speed: in fact, consider an identification application with 10,000 users. We can certainly find a combination of a fast algorithm plus a fast architecture capable of carrying out an identification in a few seconds. On the other hand, suppose that, for an acceptable FNMR, the FMR of the chosen algorithm is 10^{-5} (i.e., just one false match in 100,000 matches). Then the probability of falsely accepting an individual during identification is $FMR_N \cong 10\%$, and everyone has a good chance of gaining access to the system by trying to get in with all the ten fingers in their two hands. Multimodal biometric systems seems to be the only obvious solution to accuracy scalability in large-scale automatic identification.

2.1.2.3 Evaluating biometric system

Phillips et al. (2000) [34] define three types of evaluation of biometric systems: technology evaluation, scenario evaluation, and operational evaluation.

- Technology evaluation: The goal of a technology evaluation is to compare competing algorithms from a single technology. Only algorithm compliant with a given input/output protocol are tested. Testing of all the algorithm is carried out

on one or more databases. Although sample data may be distributed for developmental or tuning purposes prior to the test, the actual testing must be done on data that have not previously been seen by algorithm developers. Because the database is fixed, the results of technology tests are repeatable . FVC2000 (Maio et al., 2002a) [35] and FVC2002 (Mario et al., 2002b) [36] are examples of technology evaluations of fingerprint verification algorithm.

- Scenario evaluation: The goal of scenario evaluation is to determine the overall system performance in a prototype or simulated application. Testing is performed on a complete system in an environment that models a real-world target application. Each tested system has its own acquisition device. Data collection across all tested systems has to be carried out in the same environment with the same population. Test results are repeatable only to the extent that the modeled scenario can be carefully controlled (UKBWG, 2002) [37].
- Operational evaluation: The goal of operational testing is to determine the performance of a complete biometric system in a specific application environment with a specific target population. In general, operational test results are not repeatable because of unknown and undocumented differences between operational environments.

In scenario and operational evaluations, the accuracy of a biometric system depends heavily on several variables: the composition of the population, the environment, the system operational mode, and other application-specific constraints. In an ideal situation, one would like to characterize the application-independent performance of a recognition system and be able to predict the real operational performance of the system based on the application. Rigorous and realistic modeling techniques characterizing data acquisition and matching processes are the only way to grasp and extrapolate the performance evaluation results.

The performance evaluation of a biometric system is empirical and the resulting measures cannot be completely understood/compared without carefully considering the methods that were used to acquire the underlying test data. Fortunately, the biometric community is making efforts towards establishing best practices guidelines for performance evaluation so that egregious mistakes in data collection can be avoided and the test results presented in a consistent and descriptive manner.

2.2 Sensor Networks

Smart environments represent the next evolutionary development step in building, utilities, industrial, home, shipboard, and transportation systems automation. Like any sentient organism, the smart environment relies first and foremost on sensory data from the real world. Sensory data comes from multiple sensors of different modalities in distributed locations. The smart environment needs information about its surroundings as well as about its internal workings; this is captured in biological systems by the distinction between exteroceptors and proprioceptors

The challenges in the hierarchy of: detecting the relevant quantities, monitoring and collecting the data, assessing and evaluating the information, formulating meaningful user displays, and performing decision-making and alarm functions are enormous. The information needed by smart environments is provided by Distributed Wireless Sensor Networks, which are responsible for sensing as well as for the first stages of the processing hierarchy. The importance of sensor networks is highlighted by the number of recent funding initiatives, including the DARPA SENSIT program, military programs, and NSF Program Announcements.

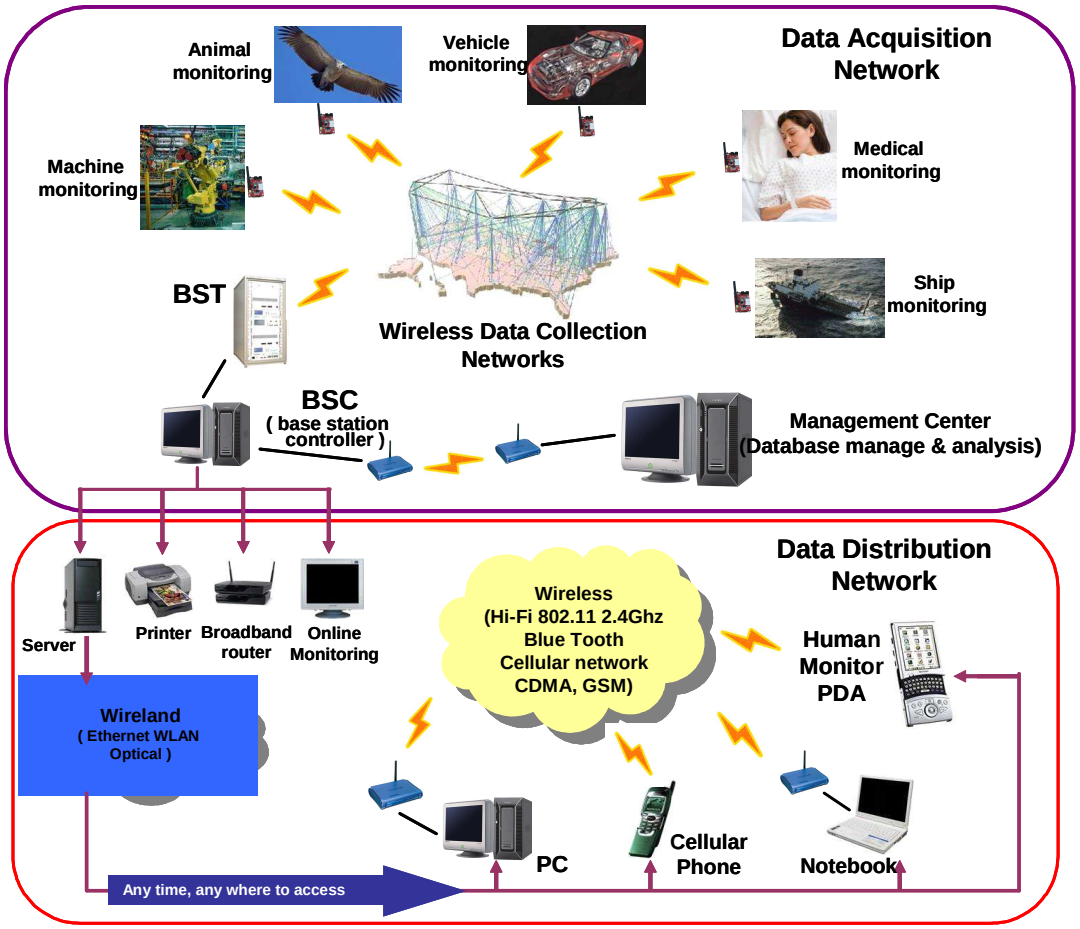


Figure 6. Wireless Sensor Networks model.

The figure 6 shows the complexity of wireless sensor networks, which generally consist of a data acquisition network and a data distribution network, monitored and controlled by a management center. The plethora of available technologies makes even the selection of components difficult, let alone the design of a consistent, reliable, robust overall system.

The study of wireless sensor networks is challenging in that it requires an enormous breadth of knowledge from an enormous variety of disciplines. In this chapter we outline communication networks, wireless sensor networks and smart sensors, physical transduction principles, commercially available wireless sensor systems, self-organization,

signal processing and decision-making, and finally some concepts for home automation.

2.2.1 Historical Development and Standards

Much of this information is taken from [38], which contains a thorough summary of communication network standards, topologies, and components.

Ethernet. The Ethernet was developed in the mid 1970's by Xerox, DEC, and Intel, and was standardized in 1979. The Institute of Electrical and Electronics Engineers (IEEE) released the official Ethernet standard IEEE 802.3 in 1983. The Fast Ethernet operates at ten times the speed of the regular Ethernet, and was officially adopted in 1995. It introduces new features such as full-duplex operation and auto-negotiation. Both these standards use IEEE 802.3 variable-length frames having between 64 and 1514-byte packets.

Token Ring. In 1984 IBM introduced the 4Mbit/s token ring network. The system was of high quality and robust, but its cost caused it to fall behind the Ethernet in popularity. IEEE standardized the token ring with the IEEE 802.5 specification. The Fiber Distributed Data Interface (FDDI) specifies a 100Mbit/s token-passing, dual-ring LAN that uses fiber optic cable. It was developed by the American National Standards Institute (ANSI) in the mid 1980s, and its speed far exceeded current capabilities of both Ethernet and IEEE 802.5.

Gigabit Ethernet. The Gigabit Ethernet Alliance was founded in 1996, and the Gigabit Ethernet standards were ratified in 1999, specifying a physical layer that uses a mixture of technologies from the original Ethernet and fiber optic cable technologies from FDDI.

Client-Server networks became popular in the late 1980's with the replacement of large mainframe computers by networks of personal computers. Application programs for distributed computing environments are essentially divided into two parts: the client or front end, and the server or back end. The user's PC is the client and more

powerful server machines interface to the network.

Peer-to-Peer networking architectures have all machines with equivalent capabilities and responsibilities. There is no server, and computers connect to each other, usually using a bus topology, to share files, printers, Internet access, and other resources.

Peer-to-Peer Computing is a significant next evolutionary step over P2P networking. Here, computing tasks are split between multiple computers, with the result being assembled for further consumption. P2P computing has sparked a revolution for the Internet Age and has obtained considerable success in a very short time. The Napster MP3 music file sharing application went live in September 1999, and attracted more than 20 million users by mid 2000.

802.11 Wireless Local Area Network. IEEE ratified the IEEE 802.11 specification in 1997 as a standard for WLAN. Current versions of 802.11 (i.e. 802.11b) support transmission up to 11Mbit/s. WiFi, as it is known, is useful for fast and easy networking of PCs, printers, and other devices in a local environment, e.g. the home. Current PCs and laptops as purchased have the hardware to support WiFi. Purchasing and installing a WiFi router and receivers is within the budget and capability of home PC enthusiasts.

Bluetooth was initiated in 1998 and standardized by the IEEE as Wireless Personal Area Network (WPAN) specification IEEE 802.15. Bluetooth is a short range RF technology aimed at facilitating communication of electronic devices between each other and with the Internet, allowing for data synchronization that is transparent to the user. Supported devices include PCs, laptops, printers, joysticks, keyboards, mice, cell phones, PDAs, and consumer products. Mobile devices are also supported. Discovery protocols allow new devices to be hooked up easily to the network. Bluetooth uses the unlicensed 2.4GHz band and can transmit data up to 1Mbit/s, can penetrate solid non-metal barriers, and has a nominal range of 10m that can be extended to 100m. A master station can service up to 7 simultaneous slave links. Forming a network of these

networks, e.g. a piconet, can allow one master to service up to 200 slaves.

Currently, Bluetooth development kits can be purchased from a variety of suppliers, but the systems generally require a great deal of time, effort, and knowledge for programming and debugging. Forming piconets has not yet been streamlined and is unduly difficult.

Home RF was initiated in 1998 and has similar goals to Bluetooth for WPAN. Its goal is shared data/voice transmission. It interfaces with the Internet as well as the Public Switched Telephone Network. It uses the 2.4GHz band and has a range of 50 m, suitable for home and yard. A maximum of 127 nodes can be accommodated in a single network. IrDA is a WPAN technology that has a short-range, narrow-transmission-angle beam suitable for aiming and selective reception of signals.

2.2.2 Wireless Sensor Networks

Sensor networks are the key to gathering the information needed by smart environments, whether in buildings, utilities, industrial, home, shipboard, transportation systems automation, or elsewhere. Recent terrorist and guerilla warfare countermeasures require distributed networks of sensors that can be deployed using, e.g. aircraft, and have self-organizing capabilities. In such applications, running wires or cabling is usually impractical. A sensor network is required that is fast and easy to install and maintain.

2.2.2.1 IEEE 1451 and smart sensors

Wireless sensor networks satisfy these requirements. Desirable functions for sensor nodes include: ease of installation, self-identification, self-diagnosis, reliability, time awareness for coordination with other nodes, some software functions and DSP, and standard control protocols and network interfaces [39].

There are many sensor manufacturers and many networks on the market today. It is

too costly for manufacturers to make special transducers for every network on the market. Different components made by different manufacturers should be compatible. Therefore, in 1993 the IEEE and the National Institute of Standards and Technology (NIST) began work on a standard for Smart Sensor Networks. IEEE 1451, the Standard for Smart Sensor Networks was the result. The objective of this standard is to make it easier for different manufacturers to develop smart sensors and to interface those devices to networks.

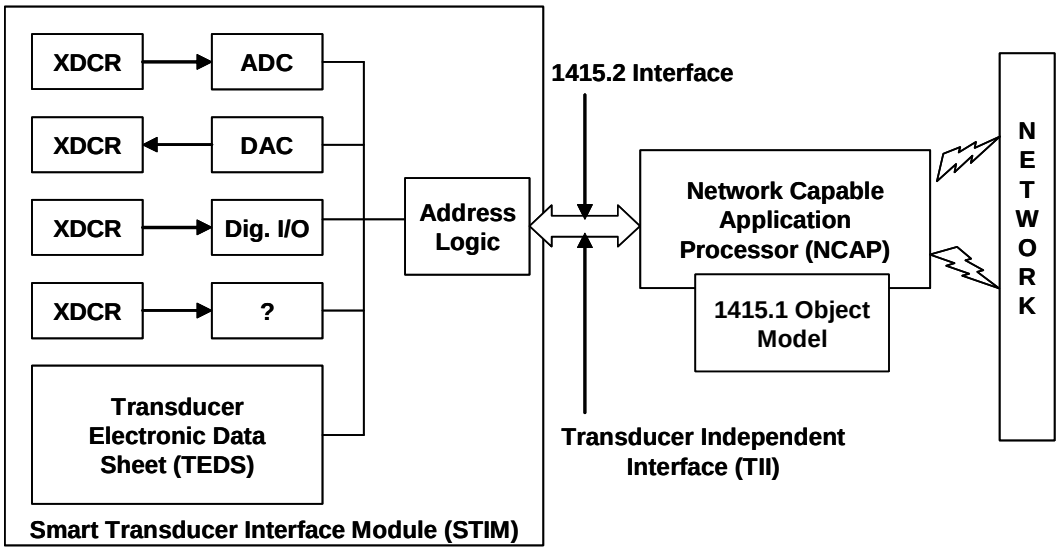


Figure 7. The IEEE 1415 standard for smart sensor networks.

Smart Sensor, Virtual Sensor. The figure 7 shows the basic architecture of IEEE 1451 [Conway and Hefferman 2003] [40]. Major components include STIM, TEDS, TII, and NCAP as detailed in the figure 7. A major outcome of IEEE 1451 studies is the formalized concept of a Smart Sensor. A smart sensor is a sensor that provides extra functions beyond those necessary for generating a correct representation of the sensed quantity [41]. Included might be signal conditioning, signal processing, and decision-making/alarm functions. A general model of a smart sensor is shown in the

figure 8. Objectives for smart sensors include moving the intelligence closer to the point of measurement; making it cost effective to integrate and maintain distributed sensor systems; creating a confluence of transducers, control, computation, and communications towards a common goal; and seamlessly interfacing numerous sensors of different types. The concept of a Virtual Sensor is also depicted. A virtual sensor is the physical sensor/transducer, plus the associated signal conditioning and digital signal processing (DSP) required to obtain reliable estimates of the required sensory information. The virtual sensor is a component of the smart sensor.

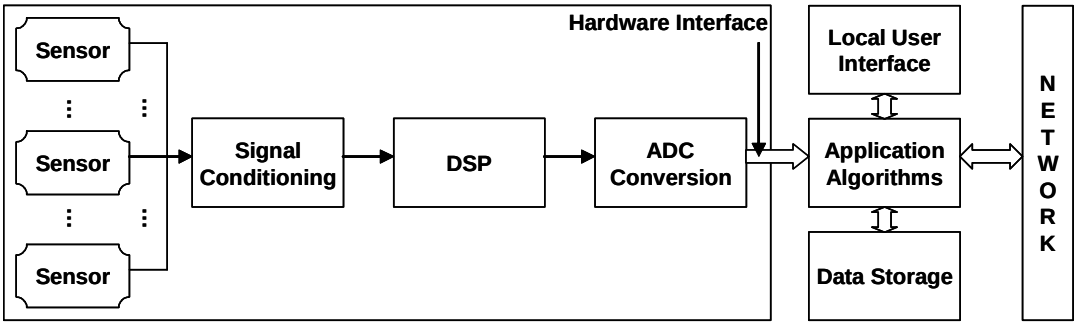


Figure 8. A general model of a smart sensor.

2.2.2.2 Transducers and physical transduction principles

A transducer is a device that converts energy from one domain to another. In our application, it converts the quantity to be sensed into a useful signal that can be directly measured and processed. Since much signal conditioning (SC) and digital signal processing (DSP) is carried out by electronic circuits, the outputs of transducers that are useful for sensor networks are generally voltages or currents. Sensory transduction may be carried out using physical principles, some of which we review here. Micro_electro_mechanical Systems (MEMS) sensors are by now very well developed and are available for most sensing applications in wireless networks.

Figure 9. Sensory Transducer.

- Mechanical Sensors include those that rely on direct physical contact.

  i) The Piezoresistive Effect converts an applied strain to a change in resistance that can be sensed using electronic circuits such as the Wheatstone Bridge (discussed later). Discovered by Lord Kelvin in 1856, the relationship is $\Delta R/R = S\epsilon$, with R the resistance, ϵ the strain, and S the gauge factor which depends on quantities such as the resistivity and the Poisson' ratio of the material. There may be a quadratic term in ϵ for some materials. Metals and semiconductors exhibit Piezoresistivity. The Piezoresistive Effect in silicon is enhanced by doping with boron (p-type silicon can have a gauge factor up to 200). With semiconductor strain gauges, temperature compensation is important.

  ii) The Piezoelectric Effect, discovered by the Curies in 1880, converts an applied stress (force) to a charge separation or potential difference. Piezoelectric materials include barium titanate, PZT, and single-crystal quartz. The relation between the change in force F and the change in voltage V is given by $\Delta V = k\Delta F$, where k is proportional to the material charge sensitivity coefficients and the crystal thickness, and inversely proportional to the crystal area and the material relative permittivity. The piezoelectric effect is reversible, so that a change in voltage also generates a force and a corresponding change in thickness. Thus the same device can be both a sensor and an actuator. Combined sensor/actuators are an intriguing topic of current research.

  iii) Tunneling Sensing depends on the exponential relationship between the tunneling current I and the tip/surface separation z given by $I = I_0 e^{-kz}$, where

k depends on the tunnel barrier height in ev. Tunneling is an extremely accurate method of sensing nanometer-scale displacements, but its highly nonlinear nature requires the use of feedback control to make it useful.

iv) Capacitive Sensors typically have one fixed plate and one movable plate. When a force is applied to the movable plate, the change in capacitance C is given as $\Delta C = \epsilon A/\Delta d$, with Δd the resulting displacement, A the area, and ϵ the dielectric constant. Changes in capacitance can be detected using a variety of electric circuits and converted to a voltage or current change for further processing. Inductive sensors, which convert displacement to a change in inductance, are also often useful.

- Magnetic and Electromagnetic Sensors do not require direct physical contact and are useful for detecting proximity effects [42].

i) The Hall Effect, discovered by Edwin Hall in 1879, relies on the fact that the Lorentz Force deflects flowing charge carriers in a direction perpendicular to both their direction of flow and an applied magnetic field (i.e. vector cross product). The Hall voltage induced in a plate of thickness T is given by $V_H = RI_xB_z/T$, with R the Hall coefficient, I_x the current flow in direction x, and B_z the magnetic flux density in the z direction. R is 4-5 times larger in semiconductors than in most metals. The Magnetoresistive Effect is a related phenomenon depending on the fact that the conductivity varies as the square of the applied flux density.

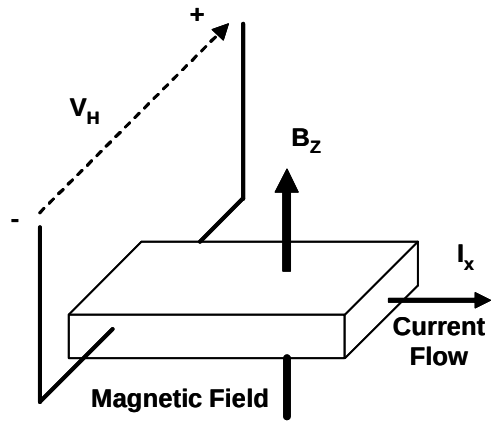


Figure 10. The hall effect.

ii) Magnetic Field Sensors can be used to detect the remote presence of metallic objects. Eddy-Current Sensors use magnetic probe coils to detect defects in metallic structures such as pipes.

- Thermal Sensors are a family of sensors used to measure temperature or heat flux. Most biological organisms have developed sophisticated temperature sensing systems [42].

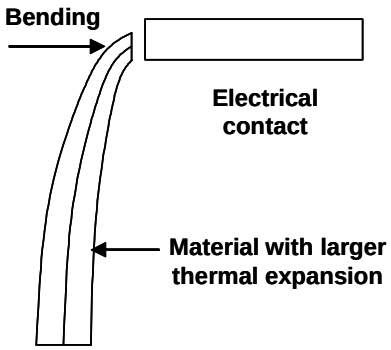


Figure 11. Thermal Bimorph.

i) Thermo-Mechanical Transduction is used for temperature sensing and regulation in homes and automobiles. On changes in temperature T, all materials exhibit (linear) thermal expansion of the form $\Delta L/L = \alpha \Delta T$, with L the length and α the coefficient of linear expansion. One can fabricate a strip of two joined materials with different thermal expansions. Then, the radius of curvature of this thermal bimorph depends on the temperature change.

ii) Thermoresistive Effects are based on the fact that the resistance *R* changes with temperature *T*. For moderate changes, the relation is approximately given by for many metals by $\Delta R/R = \alpha_R \Delta T$, with α_R the temperature coefficient of resistance. The relationship for silicon is more complicated but is well understood. Hence, silicon is useful for detecting temperature changes.

iii) Thermocouples are based on the thermoelectric Seebeck effect, whereby if a circuit consists of two different materials joined together at each end, with one junction hotter than the other, a current flows in the circuit. This generates a Seebeck voltage given approximately by $V \approx \alpha(T_1 - T_2) + \gamma(T_1^2 - T_2^2)$ with T_1, T_2 the temperatures at the two junctions. The coefficients depend on the properties of the two materials. Semiconductor thermocouples generally have higher sensitivities than do metal thermocouples. Thermocouples are inexpensive and reliable, and so are much used. Typical thermocouples have outputs on the order of $50\mu V/^\circ C$ and some are effective for temperature ranges of -270 $^\circ C$ to 2700 $^\circ C$.

iv) Resonant Temperature Sensors rely on the fact that single-crystal SiO2 exhibits a change in resonant frequency depending on temperature change. Since this is a frequency effect, it is more accurate than amplitude-change effects and has extreme sensitivity and accuracy for small temperature changes.

• Optical Transducers convert light to various quantities that can be detected [42]. These are based on one of several mechanisms. In the photoelectric effect

(Einstein, Nobel Prize, 1921) one electron is emitted at the negative end of a pair of charged plates for each light photon of sufficient energy. This causes a current to flow. In photoconductive sensors, photons generate carriers that lower the resistance of the material. In junction-based photosensors, photons generate electron-hole pairs in a semiconductor junction that causes current flow. This is often misnamed the photovoltaic effect. These devices include photodiodes and phototransistors. Thermopiles use a thermocouple with one junction coated in a gold or bismuth black absorber, which generates heat on illumination.

i) Solar cells are large photodiodes that generate voltage from light. Bolometers consist of two thermally sensitive resistors in a Wheatstone bridge configuration, with one of them shielded from the incident light. Optical transducers can be optimized for different frequencies of light, resulting in infrared detectors, ultraviolet detectors, etc.

ii) Various devices, including accelerometers, are based on optical fiber technology, often using time-of-flight information.

- Chemical And Biological Transducers [42] cover a very wide range of devices that interact with solids, liquids, and gases of all types. Potential applications include environmental monitoring, biochemical warfare monitoring, security area surveillance, medical diagnostics, implantable biosensors, and food monitoring. Effective use has been shown for NOx (from pollution), organophosphorus pesticides, nerve gases (Sarin, etc), hydrogen cyanide, smallpox, anthrax, COx, SOx, and others.

  i) Chemiresistors have two interdigitated finger electrodes coated with specialized chemical coatings that change their resistance when exposed to certain chemical challenge agents. The electrodes may be connected directly to an FET, which amplifies the resulting signals in situ for good noise rejection. This device is known as an interdigitated-gate electrode FET (IGEFET). Arrays of

chemiresistors, each device with a different chemically active coating, can be used to increase specificity for specific challenge agents [43]. Digital signal processing, including neural network classification techniques, is important in correct identification of the agent.

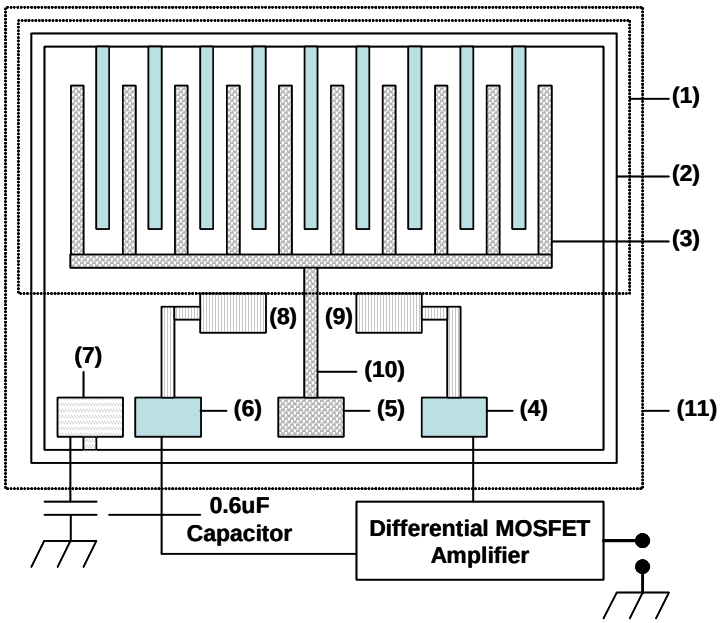


Figure 12. IGEFET structure.

ii) Metal-Oxide Gas Sensors rely on the fact that adsorption of gases onto certain semiconductors greatly changes their resistivities. In thin-film detectors, a catalyst such as platinum is deposited on the surface to speed the reactions and enhance the response. Useful as sensors are the oxides of tin, zinc, iron, zirconium, etc. Gases that can be detected include CO2, CO, HsS, NH3, and ozone. Reactions are of the form $O_2 + 2e^- \rightarrow 2O^-$ so that adsorption effectively produces an electron trap site, effectively depleting the surface of mobile carriers and increasing its resistance.

iii) Electrochemical Transducers rely on currents induced by oxidation or reduction

of a chemical species at an electrode surface. These are among the simplest and most useful of chemical sensors. An electron transfer reaction occurs that is described by $O + ze^{-} \Leftrightarrow R$, with O the oxidized species, R the reduced species, and z the charge on the ion involved. The resulting current density is given in terms of z by the Butler-Volmer equation [42].

iv) Biosensors of a wide variety of types depend on the high selectivity of many biomolecular reactions, e.g. molecular binding sites of the detector may only admit certain species of analyte molecules. Unfortunately, such reactions are not usually reversible so the sensor is not reusable. These devices have a biochemically active thin film deposited on a platform device that converts induced property changes (e.g. mass, resistance) into detectable electric or optical signals. Suitable conversion platforms include the IGEFET (above), ion-sensitive FET (ISFET), SAW (below), quartz crystal microbalance (QCM), microcantilevers, etc. To provide specificity to a prescribed analyze measurand, for the thin film one may use proteins (enzymes or antibodies), polysaccharide, nucleic acid, oligonucleotides [44], or an ionophore (which has selective responses to specific ion types). Arrays of sensors can be used, each having a different biochemically active film, to improve sensitivity. This has been used in the so-called 'electronic nose.'

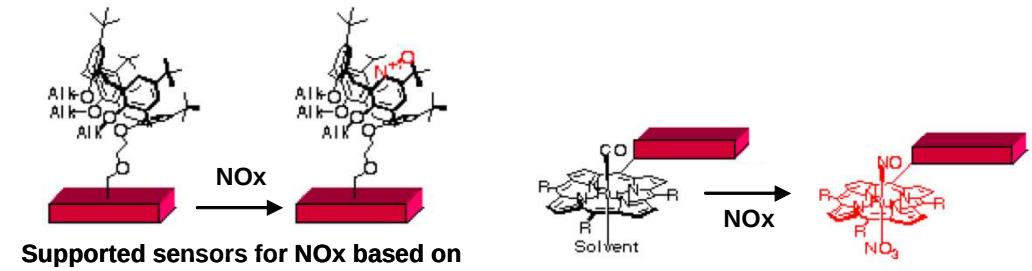


Figure 13. Biosensors based on molecular recognition [45].

- The Electromagnetic Spectrum can be used to fabricate Remote Sensors of a wide variety of types. Generally the wavelength suitable for a particular application is selected based on the propagation distance, the level of detail and resolution required, the ability to penetrate solid materials or certain mediums, and the signal processing difficulty. Doppler techniques allow the measurement of velocities. Millimeter waves have been used for satellite remote monitoring. Infrared is used for night vision and sensing heat. IR motion detectors are inexpensive and reliable. Electromagnetic waves can be used to determine distance using time of flight information. Radar uses RF waves and Lidar uses light (laser). The velocity of light is $c = 299.8 \times 10^6 m/s$. GPS uses RF for absolute position localization. Visible light imaging using cameras is used in a broad range of applications but generally requires the use of sophisticated and computationally expensive DSP techniques including edge detection, thresholding, segmentation, pattern recognition, motion analysis, etc.

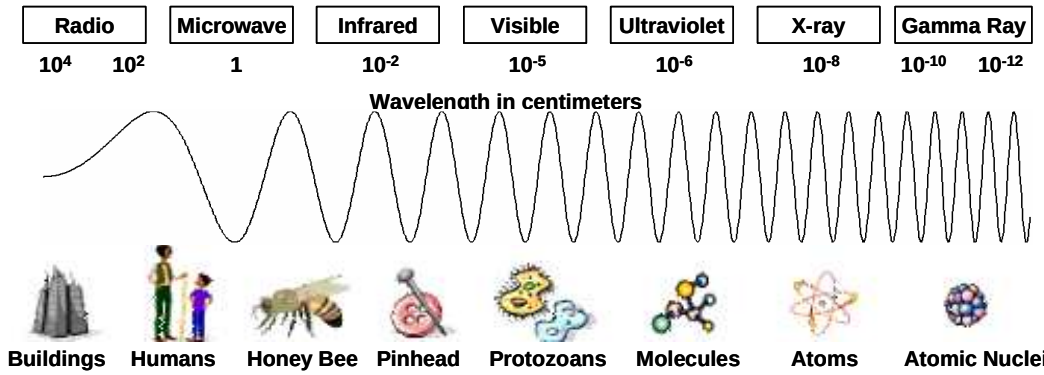


Figure 14. The electromagnetic spectrum.

- Acoustic Sensors include those that use sound as a sensing medium. Doppler techniques allow the measurement of velocities. Ultrasound often provides more information about mechanical machinery vibrations, fluid leakage, and impending

equipment faults than do other techniques. Sonar uses sound to determine distance using time-of-flight information. It is effective in media other than air, including underwater. Caution should be used in that the propagation speed of acoustic signals depends on the medium. The speed of sound at sea level in a standard atmosphere is $c_s = 340.294 m/s$. Subterranean echoes from earthquakes and tremors can be used to glean information about the earth's core as well as about the tremor event, but deconvolution techniques must be used to remove echo phenomena and to compensate for uncertain propagation speeds.

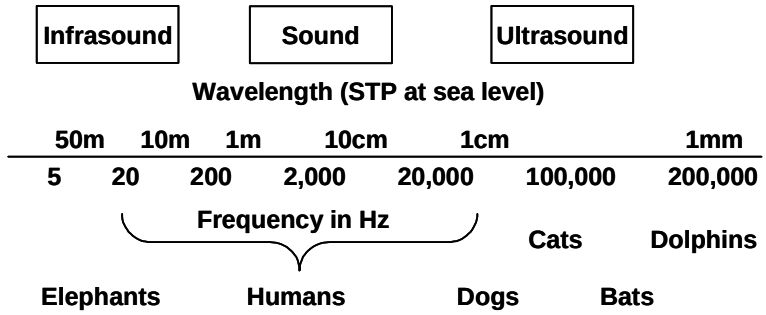


Figure 15. The acoustic spectrum.

i) Acoustic Wave Sensors are useful for a broad range of sensing devices [42]. These transducers can be classified as surface acoustic wave (SAW), thickness-shear mode (TSM), flexural plate wave (FPW), or acoustic plate mode (APM). The SAW is shown in the figure 15 and consists of two sets of interdigitated fingers at each end of a membrane, one set for generating the SAW and one for detecting it. Like the IGEFET, these are useful platforms to convert property changes such as mass into detectable electrical signals. For instance, the surface of the device can be coated with a chemically or biologically active thin film. On presentation of the measurand to be sensed, adsorption might cause the mass m to change, resulting in a frequency shift

given by the Sauerbrey equation $\Delta f = k f_0^2 \Delta m / A$, with f_0 the membrane resonant frequency, constant k depending on the device, and A the membrane area.

2.2.2.3 Sensors for smart environment

Many vendors now produce commercially available sensors of many types that are suitable for wireless network applications. See for instance the websites of SUNX Sensors, Schaevitz, Keyence, Turck, Pepperl & Fuchs, National Instruments, UE Systems (ultrasonic), Leake (IR), CSI (vibration). The table shows which physical principles may be used to measure various quantities. MEMS sensors are by now available for most of these measurands.

Table 2. Measurements for Wireless Sensor Networks.

	Measurand	*Transduction Principle*
Physical Properties	Pressure	Piezoresistive, capacitive
	Temperature	Thermistor, thermo-mechanical, Thermocouple
	Humidity	Resistive, capacitive
	Flow	Pressure change, thermistor
Motion Properties	Position	E-mag, GPS, contact sensor
	Velocity	Doppler, Hall effect, optoelectronic
	Angular velocity	Optical encoder
	Acceleration	Piezoresistive, piezoelectric, optical fiber
Contact Properties	Strain	Piezoresistive
	Force	Piezoelectric, piezoresistive
	Torque	Piezoresistive, optoelectronic
	Slip	Dual torque
	Vibration	Piezoresistive, piezoelectric, optical fiber, sound, ultrasound
Presence	Tactile/contact	Contact switch, capacitive
	Proximity	Hall effect, capacitive, magnetic, seismic, acoustic, RF
	Distance/range	E-mag(sonar, radar, lidar), magnetic, tunneling
	Motion	E-mag, IR, acoustic, seismic(vibration)
Biochemical	Biochemical agents	Biochemical transduction
Identification	Personal features	Vision
	Personal ID	Fingerprints, retinal scan, voice, heat plume, vision motion analysis

2.2.2.4 Commercially available wireless sensor systems

Many commercially available wireless communications nodes are available including Lynx Technologies, and various Bluetooth kits, including the Casira devices from Cambridge Silicon Radio, CSR.

- Crossbow Berkeley Motes may be the most versatile wireless sensor network

devices on the market for prototyping purposes. Crossbow (http://www.xbow.com/) makes three Mote processor radio module families MICA [MPR300] (first generation), MICA2 [MPR400] and MICA2-DOT [MPR500] (second generation). Nodes come with five sensors installed- Temperature, Light, Acoustic (Microphone), Acceleration/Seismic, and Magnetic. These are especially suitable for surveillance networks for personnel and vehicles. Different sensors can be installed if desired. Low power and small physical size enable placement virtually anywhere. Since all sensor nodes in a network can act as base stations, the network can self configure and has multi-hop routing capabilities. The operating frequency is ISM band, either 916MHz or 433MHz, with a data rate of 40 Kbits/sec. and a range of 30ft to 100ft. Each node has a low power microcontroller processor with speed of 4MHz, a flash memory with 128 Kbytes, and SRAM and EEPROM of 4K bytes each. The operating system is Tiny-OS, a tiny micro-threading distributed operating system developed by UC Berkeley, with a NES-C (Nested C) source code language (similar to C). Installation of these devices requires a great deal of programming. A workshop is offered for training.

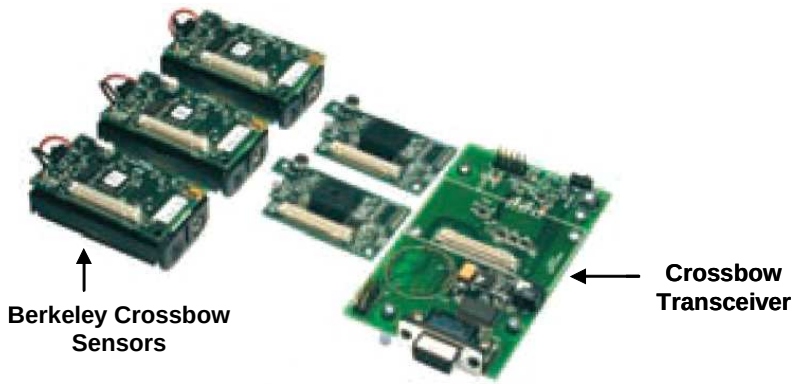


Figure 16. Berkeley Crossbow Motes.

- Microstrain's X-Link Measurement System (http://www.microstrain.com/) may be

the easiest system to get up and running and to program. The frequency used is 916MHz, which lies in the US license-free ISM band. The sensor nodes are multi-channel, with a maximum of 8 sensors supported by a single wireless node. There are three types of sensor nodes S-link (strain gauge), G-link (accelerometer), and V-link (supports any sensors generating voltage differences). The sensor nodes have a pre-programmed EPROM, so a great deal of programming by the user is not needed. Onboard data storage is 2MB. Sensor nodes use a 3.6-volt lithium ion internal battery (9V rechargeable external battery is supported). A single receiver (Base Station) addresses multiple nodes. Each node has a unique 16-bit address, so a maximum of 216 nodes can be addressed. The RF link between Base Station and nodes is bi-directional and the sensor nodes have a programmable data logging sample rate. The RF link has a 30 meter range with a 19,200 baud rate. The baud rate on the serial RS-232 link between the Base Station and a terminal PC is 38400. LabVIEW interface is supported.

Figure 17. Microstrain Wireless Sensors.

2.2.3 Building and Home Automation

The figure 18 shows how networks of various sorts might interact in the smart home

environment. An excellent reference for this section is [46]. There are many available protocols for networking of the smart home, and it is not necessary to develop new protocols on one's own for commercially acceptable systems. The BACnet protocol has been developed by the building automation industry to provide a standard for interconnecting networks for building sensing and control. Networks that can be used include Ethernet, MS/TP, and LonWorks. Building energy management standards are being developed by the American Society of Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE). A major driver for the smart home is the power distribution industry, which could save enormous sums with demand-side regulation and automated remote meter reading. The Intelligent Building Institute has been a force in developing appropriate standards.

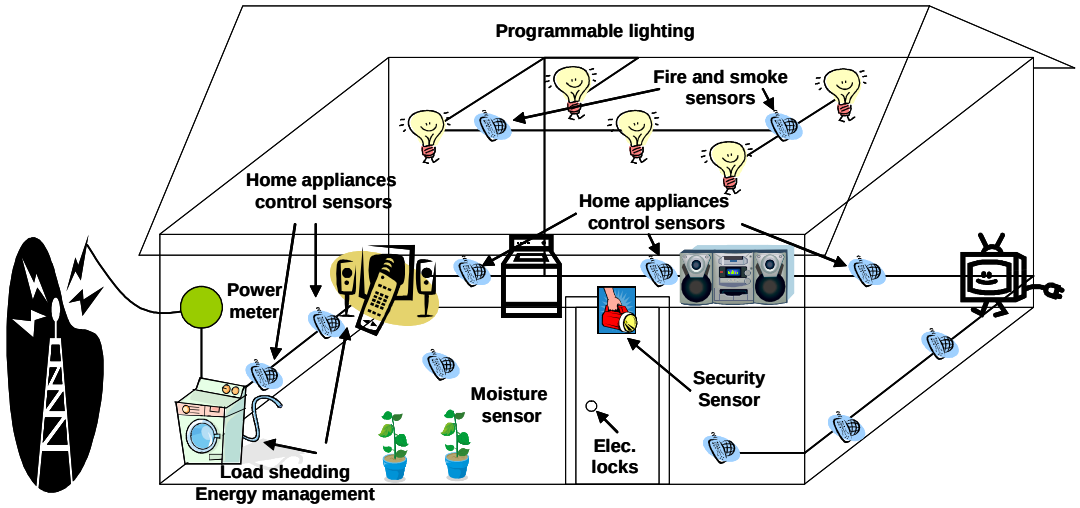


Figure 18. Smart home networks.

The X-10 protocol is used for lamp and appliance controls. The more recently developed Smart House Applications Language (SHAL) includes over 100 message types for specific sensing and control functions. However, SHAL requires dedicated multiconductor wiring. The Consumer Electronics bus (CEBus), initiated by the

Electronic Industries Association, provides both data and control channels and handles up to 10Kbps. It is useful for the utility industry.

Several automotive protocols have been developed, and some of these are useful also for building control. CAN is a serial communications protocol developed for automotive multiplex wiring systems, and has been adopted in industrial applications by manufacturers such as Allen-Bradley (in the DeviceNET system) and Honeywell (in SDS). CAN supports distributed real-time control with a high level of security, and is a multimaster protocol that allows any node in the network to communicate with any other node. Supported are user-defined message prioritization, multiple access/collision resolution, and error detection.

The LonWorks protocol, developed by Echelon Corp (http://www.echelon.com/products/lonworks/default.htm) is very convenient for industrial and consumer applications. It supports all seven layers of the OSI/RM model, and supports fieldbus requirements, arbitration, and message coding. LonWorks operates on a peer-to-peer bus network basis. Devices in a LonWorks network communicate using LonTalk. This language provides a set of services that allow the application program in a device to send and receive messages from other devices over the network without needing to know the topology of the network or the names, addresses, or functions of other devices. The LonWorks protocol can optionally provide end-to-end acknowledgement of messages, authentication of messages, and priority delivery to provide bounded transaction times. Support for network management services allows for remote network management tools to interact with devices over the network, including reconfiguration of network addresses and parameters, downloading of application programs, reporting of network problems, and start/stop/reset of device application programs. LonWorks networks can be implemented over basically any medium, including power lines, twisted pair, radio frequency (RF), infrared (IR), coaxial cable and fiber optics.

Chapter 3. Theoretical Background

3.1 Principal Components Analysis

In statistics, principal components analysis (PCA) is a technique for simplifying a dataset, by reducing multidimensional datasets to lower dimensions for analysis [47].

Technically speaking, PCA is a linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA can be used for dimensionality reduction in a dataset while retaining those characteristics of the dataset that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" aspects of the data. But this is not necessarily the case, depending on the application.

PCA is also called the (discrete) Karhunen-Loève transform (or KLT, named after Kari Karhunen and Michel Loève) or the Hotelling transform (in honor of Harold Hotelling). PCA has the distinction of being the optimal linear transformation for keeping the subspace that has largest variance. This advantage, however, comes at the price of greater computational requirement if compared, for example, to the discrete cosine transform. Unlike other linear transforms, the PCA does not have a fixed set of basis vectors. Its basis vectors depend on the data set.

Assuming zero empirical mean (the empirical mean of the distribution has been subtracted from the data set), the principal component w_1 of a dataset x can be defined as equation (1).

$$w_1 = arg\,max_{\|w\|=1} E\{(w^T x)^2\} \tag{1}$$

With the first $k-1$ components, the *k*-th component can be found by subtracting the first $k-1$ principal components from x and can be defined as equation (2).

$$\hat{x}_{k-1} = x - \sum_{i=1}^{k-1} w_i w_i^T x \tag{2}$$

And by substituting this as the new dataset to find a principal component in equation (3).

$$w_k = arg\,max_{\|w\|=1} E\{(w^T \hat{x}_{k-1})^2\} \tag{3}$$

The Karhunen-Loève transform is therefore equivalent to finding the singular value decomposition of the data matrix X which can be defined equation (4).

$$X = W \Sigma V^T \tag{4}$$

and then obtaining the reduced-space data matrix Y by projecting X down into the reduced space defined by only the first *L* singular vectors, W_L. Y can be defined equation (5).

$$Y = W_L^T X = \Sigma_L V_L^T \tag{5}$$

The matrix W of singular vectors of X is equivalently the matrix W of eigenvectors of the matrix of observed covariances $C = XX^T$,

$$XX^T = W \Sigma^2 W^T \tag{6}$$

The eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset.

PCA is equivalent to empirical orthogonal functions (EOF) and it is a popular technique in pattern recognition. But it is not optimized for class separability. An alternative is the linear discriminant analysis, which does take this into account. PCA optimally minimizes reconstruction error under the L2 norm.

Table 3. Table of symbol and abbreviations.

Symbol	*Meaning*
$X = \{x[m,n]\}$	Data matrix, consisting of the set of all data vectors, one vector per column
N	The number of column vectors in the data set
M	The number of elements in each column vector
L	The number of dimensions in the dimensionally reduced subspace
$u = \{u[m]\}$	Vector of empirical means, one mean for each row m of the data matrix
$s = \{s[m]\}$	Vector of empirical standard deviations, one standard deviation for each row m of the data matrix
$h = \{h[n]\}$	Vector of all 1's
$B = \{B[m,n]\}$	Deviations from the mean of each row m of the data matrix
$Z = \{Z[m,n]\}$	Z-scores, computed using the mean and standard deviation for each row m of the data
$C = \{C[p,q]\}$	Covariance matrix
$R = \{R[p,q]\}$	Correlation matrix
$V = \{V[p,q]\}$	Matrix consisting of the set of all eigenvectors of C, one eigenvector per column
$D = \{D[p,q]\}$	Diagonal matrix consisting of the set of all eigenvalues of C along its principal diagonal, and 0 for all other elements
$W = \{W[p,q]\}$	Matrix of basis vectors, one vector per column, where each basis vector is one of the eigenvectors of C
$Y = \{Y[m,n]\}$	Matrix consisting of N column vectors, where each vector is the projection of the corresponding data vector from matrix X

3.2 Moment

3.2.1 Geometric Moments

Let *I(x, y)* be a continuous image function. Its Geometric moment of order *p+q* is defined as equation (7) [48].

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q I(x,y)dxdy \tag{7}$$

Geometric moments provide rich information about the image and are popular features for pattern recognition. Their information content stems from the fact that moments provide an equivalent representation of an image, in the sense that an image can be reconstructed from its moments. Thus, each moment coefficient conveys a certain amount of the information residing in an image.

It is by now commonplace to state that a desirable property in pattern recognition is invariance in geometric transformations. Moments, as defined in equation (8), depend on the coordinates of the object of interest within an image; thus, they lack the invariance property. This problem can be circumvented by defining appropriate combinations of normalized versions of the moments. Specifically, our goal will be to define moments that are invariant to:

Translations:

$$x' = x + a, \quad y' = y + b$$

Scaling:

$$x' = ax, \quad y' = ay$$

Rotations:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

To this end, let us define central moments as equation (8).

$$\mu_{pq} = \int \int I(x,y)(x-\overline{x})^p (y-\overline{y})^q dxdy \tag{8}$$

where

$$\overline{x} = \frac{m_{10}}{m_{00}}, \quad \overline{y} = \frac{m_{01}}{m_{00}}$$

Central moments are invariant to translations.

Equation (9) is the normalized central moments

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{r}}, \quad r = \frac{p+q+2}{2} \tag{9}$$

These are easily shown to be invariant to both translation and scaling.

The seven moments of HU: HU has defined a set of seven moments that are invariant under the actions of translation, scaling, and rotation. The first six of these are also invariant under the action of reflection, while $\varnothing_7$ changes sign. The values of these quantities can be quite different. In practice, in order to avoid precision problems, the logarithms of their absolute values are usually used as features. A number of other moment-based features that are invariant to more general transformations have also been proposed.

For a digital image *I(i,j)*, with *i* = 0, 1, 2,..., N_x-1, *j* = 0, 1, 2,..., N_y-1, the

preceding moments can be approximated by replacing integrals by equation (10).

$$m_{pq} = \sum_i \sum_j I(i,j) i^p j^q \tag{10}$$

In order to keep the dynamic range of the moment values consistent for different sized images, a normalization of the x - y axis can be performed, prior to the computation of the moments. The moments are then approximated by equation (11).

$$m_{pq} = \sum_i I(x_i, y_j) x_i^p y_i^q \tag{11}$$

where the sum is over all image pixels. Then x_i, y_i are the coordinates of the center point of the i th pixel and are no longer integers but real numbers in the interval $x_i \in [-1, +1]$, $y_i \in [-1, +1]$. For digital images, the invariance properties of the moments we have defined are only approximately true.

3.2.2 Zernike Moments

The geometric moments defined in equation (8) can also be viewed as projections of $I(x,y)$ on the basis functions formed by the monomials $x^p y^q$. These monomials are not orthogonal; thus, the resulting geometric moment features are not optimal from an information redundancy point of view. We will derive moments based on alternative complex polynomial functions, known as Zernike polynomials. These form a complete orthogonal set over the interior of the unit circle $x^2 + y^2 \leq 1$ and are defined as equation (12).

$$V_{pq}(x,y) = V_{pq}(\rho,\theta) = R_{pq}(\rho)exp(jq\theta) \tag{12}$$

where:

p is a nonnegative integer

q is an integer subject to the constraint p - $|q|$ even, $|q| \leq p$

$\rho = \sqrt{x^2 + y^2}$

$\theta = tan^{-1}\frac{y}{x}$

$$R_{pq}(\rho) = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s[(p-s)!]\rho^{p-2s}}{s!\,(\frac{p+|q|}{s} - s)!\,(\frac{p-|q|}{2} - s)!}$$

The Zernike moments of a function *I(x,y)* are given by equation (13).

$$A_{pq} = \frac{p+1}{\pi}\int\int_{x^2+y^2} i(x,y)\,V^*(\rho,\theta)dxdy \tag{13}$$

where the "*" denotes complex conjugation. For a digital image, the respective Zernike moments are computed as equation (14).

$$A_{pq} = \frac{p+1}{\pi}\sum I(x_i,y_i)\,V^*(\rho_i,\theta_i),\ x_i^2 + y_i^2 \leq 1 \tag{14}$$

where *i* runs over all the image pixels. The computation of the corresponding moments of an image considers the center of the image as the origin and pixels are mapped into the unit circle, that is, $x_i^2 + y_i^2 \leq 1$. The pixels falling outside the unit circle are not taken into consideration. The magnitude of the Zernike moments is invariant to rotations.

3.2.3 Moment-Based Features

In equation (8) and (12) the geometric moments and central moments were defined. If in the place of *I(i, j)* we consider the sequence

$$I(i,j) = \begin{cases} 1 & (i,j \in C \\ 0 & (i,j) otherwise \end{cases}$$

where *C* is the set of points *(i, j)* inside the object of interest, then we obtain a way to describe the shape of the object through the moments. Indeed. in such a case only the limits in the summations are taken into account, whereas the details inside the object do not participate, Hence

$$m_{pq} = \sum_i \sum_j i^p j^q, \quad (i,j) \in C$$

with m_{00} = *N*, the total number of pixels inside the region. The features

$$\bar{x} = \frac{m_{10}}{m_{00}} \text{ and } \bar{y} = \frac{m_{10}}{m_{00}}$$

define the center of mass $(\bar{x}, \bar{y})$. The respective central moments become

$$\mu_{pq} = \sum_i \sum_j (i - \bar{x})^p (j - \bar{y})^q, \quad (i,j) \in C$$

The invariant moments can in turn be computed and used. whenever appropriate. Two useful quantities that are related to these moments and provide useful discriminatory information are:

1. Orientation

$$\theta = \frac{1}{2}\tan^{-1}\left[\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right]$$

which is the angle between the axis with the least moment of inertia and the x-coordinate axis

2. Eccentricity

$$\epsilon = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}}{area}$$

Another representation of the eccentricity is via the ratio $\frac{R_{\max}}{R_{\min}}$ of the maximum to the minimum distance of the center of mass $(\bar{x}, \bar{y})$ from the object's boundary.

3.3 Psychoacoustic Model

The psychoacoustic model is based on many studies of human perception. These studies have shown that the average human does not hear all frequencies the same. Effects due to different sounds in the environment and limitations of the human sensory system lead to facts that can be used to cut out unnecessary data in an audio signal.

The two main properties of the human auditory system that make up the psychoacoustic model are:

- absolute threshold of hearing

- auditory masking

Each provides a way of determining which portions of a signal are inaudible and indiscernible to the average human, and can thus be removed from a signal.

3.3.1 Absolute Threshold of Hearing

Humans can hear frequencies in the range from 20Hz to 20,000Hz. However, this does not mean that all frequencies are heard in the same way. One could make the assumption that a human would hear frequencies that make up speech better than others; this is a good guess. Furthermore, one could also hypothesize that hearing a tone becomes more difficult as its frequency nears either of the extremes. Again, this is true.

One other observation forms the basis for modeling. Because humans hear lower frequencies, like those making up speech, more than others, like high frequencies around 20KHz, the ear probably has better capability in detecting differences in pitch at lower frequencies than at high ones. This, too, is true. For example, a human has an easier time telling the difference between 500Hz and 600Hz than he does determining whether something is 17,000Hz or 18,000Hz. After many studies, scientists found that the frequency range from 20Hz to 20,000Hz can be broken up into critical bandwidths, which are non-uniform, non-linear, and dependent on the heard sound. Signals within one critical bandwidth are hard to separate for a human observer.

A more uniform measure of frequency based on critical bandwidths is the Bark. From the earlier discussed observations, one would expect a Bark bandwidth to be smaller at low frequencies (in Hz) and larger at high ones. Indeed, this is the case.

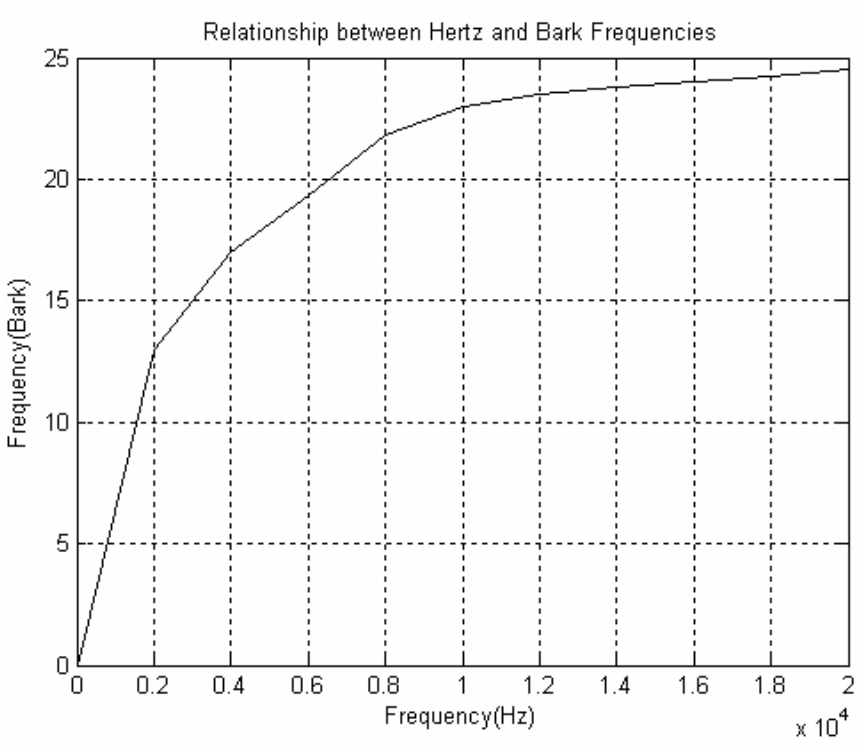


Figure 19. Relationship between Hertz and Bark Frequencies.

The Bark frequency scale can be approximated by the following equation (15).

$$Barks = 13 \times tan^{-1}(0.00076 \times Hz) + 3.5 \times tan^{-1}((f/7500)^2) \quad (15)$$

To determine the effect of frequency on hearing ability, scientists played a sinusoidal tone at a very low power. The power was slowly raised until the subject could hear the tone. This level was the threshold at which the tone could be heard. The process was repeated for many frequencies in the human auditory range and with many subjects. As a result, the following plot was obtained.

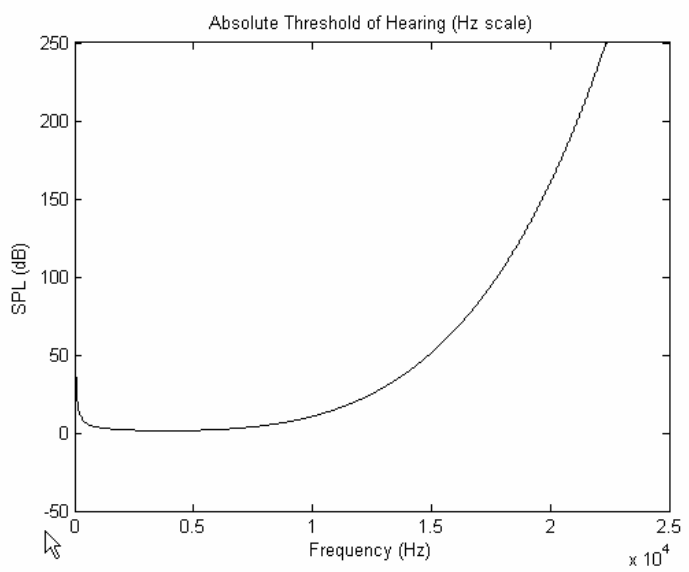


Figure 20. Absolute threshold of hearing (Hz scale).

This experimental data can be modeled by the following equation, where f is frequency in Hertz:

$$ATH(f) = 3.64 \times (\frac{f}{1000})^{-0.8} - 6.5e^{(-0.6\times((f/1000)-33)^2)} + 10^{-3} \times (\frac{f}{1000})^4 \quad (16)$$

Thus, we can make the following jump for the purposes of compression. If a signal has any frequency components with power levels that fall below the absolute threshold of hearing, then these components can be discarded, as the average listener will be unable to hear those frequencies of the signal anyway.

3.3.2 Auditory Masking

Humans do not have the ability to hear minute differences in frequency. For example, it is very difficult to discern a 1,000Hz signal from one that is 1,001Hz. This becomes even more difficult if the two signals are playing at the same time.

Furthermore, the 1,000Hz signal would also affect a human's ability to hear a signal that is 1,010Hz, or 1,100Hz, or 990Hz.

This concept is known as masking. If the 1,000Hz signal is strong, it will mask signals at nearby frequencies, making them inaudible to the listener. For a masked signal to be heard, its power will need to be increased to a level greater than that of a threshold that is determined by the frequency of the masker tone and its strength.

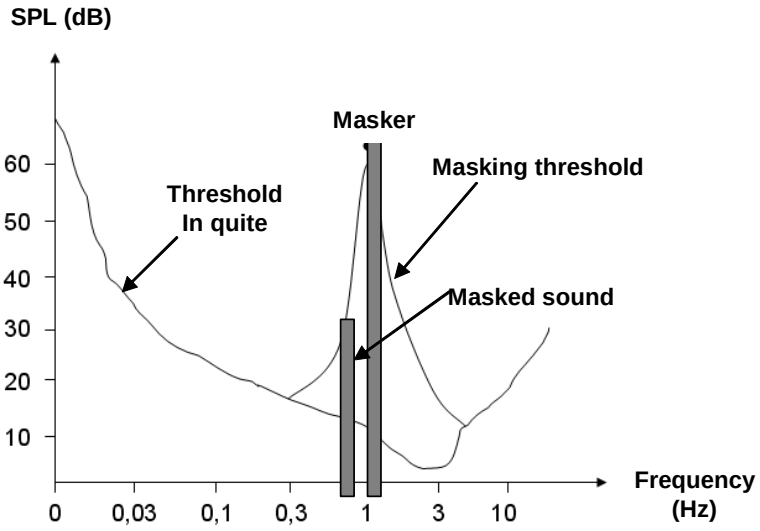


Figure 21. Masking threshold and masking effect.

It turns out that noise can be a masker as well. If noise is strong enough, it can mask a tone that would be clear otherwise. For example, a jet engine, which is very noisy, can drown out music easily.

In a compression algorithm, therefore, one must determine:

- Tone maskers
- Noise maskers
- Masking effect due to these maskers.

If any frequency components around these maskers fall below the masking threshold, they can be discarded.

3.3.2.1 Tone maskers

Determining whether a frequency component is a tone requires knowing whether it has been held constant for a period of time, as well as whether it is a sharp peak in the frequency spectrum, which indicates that it is above the ambient noise of the signal.

For the purposes of this project, only the second criterion is considered. Determining whether a certain frequency is a tone (masker) can be done with the following definition:

A frequency f (with FFT index k) is a tone if its power P[k] is:

1. Greater than P[k-1] and P[k+1], i.e., it is a local maxima
2. 7 dB greater than the other frequencies in its neighborhood, where the neighborhood is dependent on *f*

3.3.2.2 Noise maskers

If a signal is not a tone, it must be noise. Thus, one can take all frequency components that are not part of a tone's neighborhood and treat them like noise. Combining such components into maskers, though, takes a little more thought.

Since humans have difficulty discerning signals within a critical band, the noise found within each of the bands can be combined to form one mask. Thus, the idea is to take all frequency components within a critical band that do not fit within tone neighborhoods, add them together, and place them at the geometric mean location within the critical band. Repeat this for all critical bands.

3.3.2.3 Masking effect

The maskers which have been determined affect not only the frequencies within a critical band, but also in surrounding bands. Studies show that the spreading of this masking has an approximate slope of +25 dB/Bark before and -10 dB/Bark after the masker.

There is a slight difference in the resulting mask that depends on whether the mask is a tone or noise. As a result, the masks can be modeled by the following equations, with the same variables as described above:

For tones: $Ttm(i,j) = Ptm(j) - 0.275z(j) + SF(i,j) - 6.025$ (dB SPL)

For noise: $Tnm(i,j) = Pnm(j) - 0.175z(j) + SF(i,j) - 2.025$ (dB SPL)

The following are plots of various levels of tone and noise maskers.

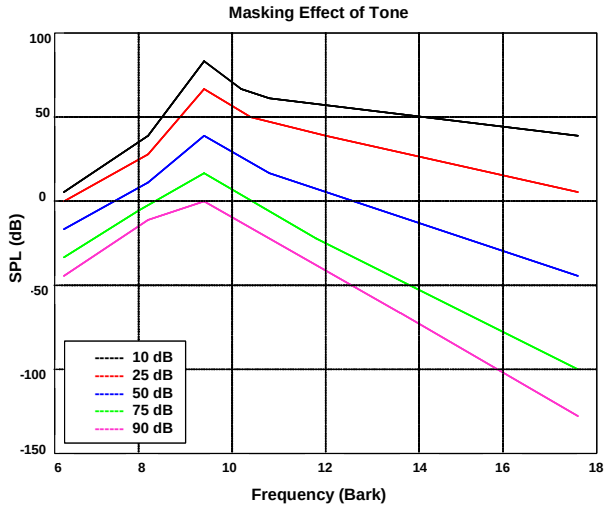


Figure 22. The levels of tone maskers.

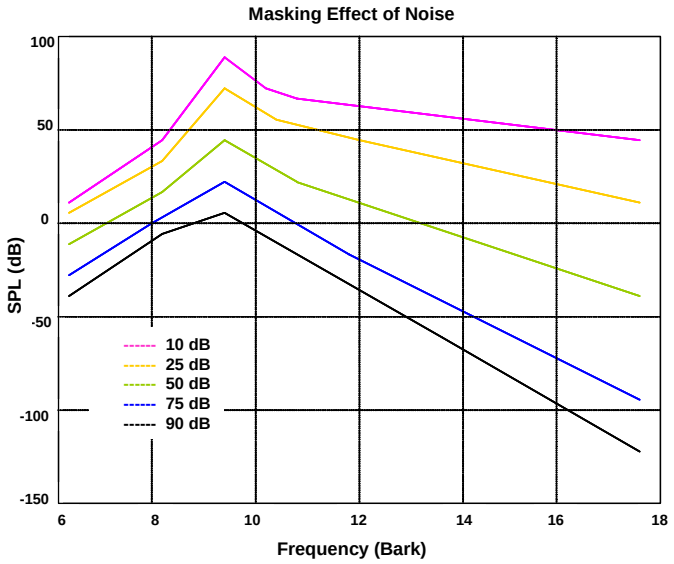


Figure 23. The levels of noise maskers.

The final plot compares a tone and noise masker at the same frequency and of the same power.

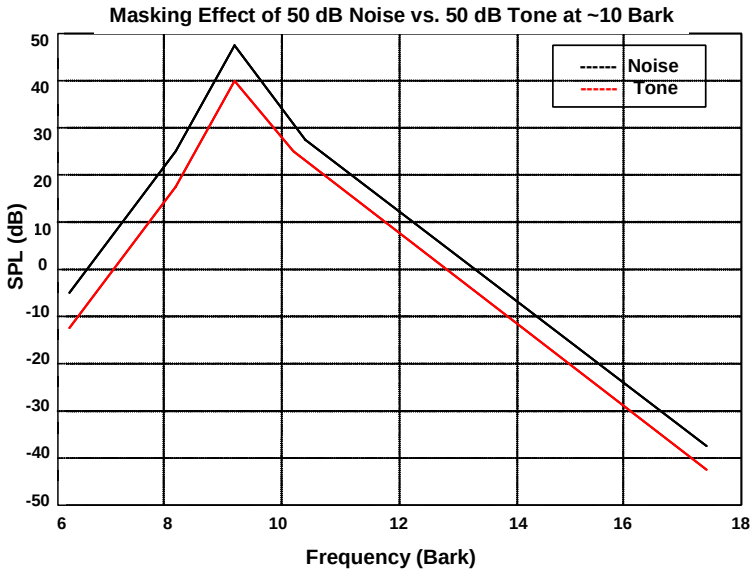


Figure 24. Masking effect of 50dB noise vs. 50dB tone.

Naturally, if there are multiple noise and tone maskers, the overall effect is a little harder to determine. In this project, the assumption is made that the effects are power additive. This is a reasonable assumption to make, but note that there is a definitely an interplay that can occur between maskers that would lower or increase thresholds.

3.4 Balanced Incomplete Block Design (BIBD)

Combinational problems can generate the matrix that satisfies a restricted condition using a matrix model. The BIBD code generates Incidence Matrix which satisfies the restricted condition of anti-collusion and can thus analyze a partly matrix symmetry. That is, the anti-collusion code has robustness against collusion attack. Among *n* code vectors, the combination of less than (*n*-1) code vectors differ each other and can therefore detect less than (*n*-1) colluders.

The BIBD code is generated with 5 parameters (*v*, *b*, *r*, *k* and λ)

Where :

v: Number of treatments

b: Number of blocks

r: Number of times each treatment is run

k: Number of treatments per block

λ: Number of times a pair of elements of v appear in the same block.

$$vr = bk \tag{17}$$

$$r(k-1) = \lambda(v-1) \tag{18}$$

Five parameters satisfy a restrict condition of equation (17) and (18) and simply present Incidence Matrix $v \times b$ represented with (v, k, λ) using equation (19) and (20).

$$b = \frac{v(v-1)\lambda}{k(k-1)} \tag{19}$$

$$r = \frac{\lambda(v-1)}{k-1} \tag{20}$$

In Incidence Matrix, if $b=v$ or $r=k$, then the BIBD code is to be a symmetric square matrix. Incidence Matrix M with $v \times b$ is defined by equation (21) and satisfies equation (22).

$$M = [m_{ij}]$$
$$m_{ij} = \begin{cases} 1 \text{ if } & j_{th} \quad blocks \in i_{th} \quad elements \\ 0 & otherwise \end{cases} \tag{21}$$

$$MM^t = (r-\lambda)I + \lambda J \tag{22}$$

I: $v \times v$ Identity Matrix

J: $v \times v$ Unit Matrix

As a result, the row vector of Incidence Matrix *M* will be a fingerprint code and given to *b* users and this Incidence Matrix *M* can be used as an anti-collusion code.

3.5 Low Density Parity Check (LDPC) codes

3.5.1 A Bit of History

Low-density parity-check (LDPC) codes were invented by R. G. Gallager (Gallager 1963; Gallager 1962) in 1962. He discovered an iterative decoding algorithm which he applied to a new class of codes. He named these codes low-density parity-check codes since the parity-check matrices had to be sparse to perform well. Yet, LDPC codes have been ignored for a long time due mainly to the requirement of high complexity computation, if very long codes are considered.

In 1993, C. Berrou et. al. invented the turbo codes (Berrou, Glavieux, and Thitimajshima 1993) and their associated iterative decoding algorithm. The remarkable performance observed with the turbo codes raised many questions and much interest toward iterative techniques.

In 1995, D. J. C. MacKay and R. M. Neal (MacKay and Neal 1995; MacKay and Neal 1996; Mackay 1999) rediscovered the LDPC codes, and set up a link between their iterative algorithm to the Pearl's belief algorithm (Pearl 1988), from the artificial intelligence community (bayesian networks). At the same time, M. Sipser and D. A. Spielman (Sipser and Spielman 1996) used the first decoding algorithm of R. G. Gallager (algorithm A) to decode expander codes.

The articles of MacKay and Neal have been the kick off of a great work in the field of LDPC codes. Most of the main articles related to the LDPC codes are gathered in a special issue of the IEEE's Transactions on Information Theory (IEEE 2001): irregular codes, density evolution, design of capacity approaching codes, etc.

Meanwhile, turbo decoding of turbo codes is shown to be an instance of the Pearl's belief algorithm by McEliece et. al. (McEliece, MacKay, and Cheng 1998), collecting under the same model (belief propagation) the last 2 best classes of codes. Graphs are becoming a standard representation of error correcting codes: F. R. Kschischang denotes

by factor graphs (Kschischang and Frey 1998) a wide class of graph associated with the sum-product algorithm, which aim at describing many different algorithms by the same formalism. This work have its origin in the work of Tanner (Tanner 1981), and N. Wiberg et. al. (Wiberg 1996; Wiberg, Loegliger, and Kotter 1995).

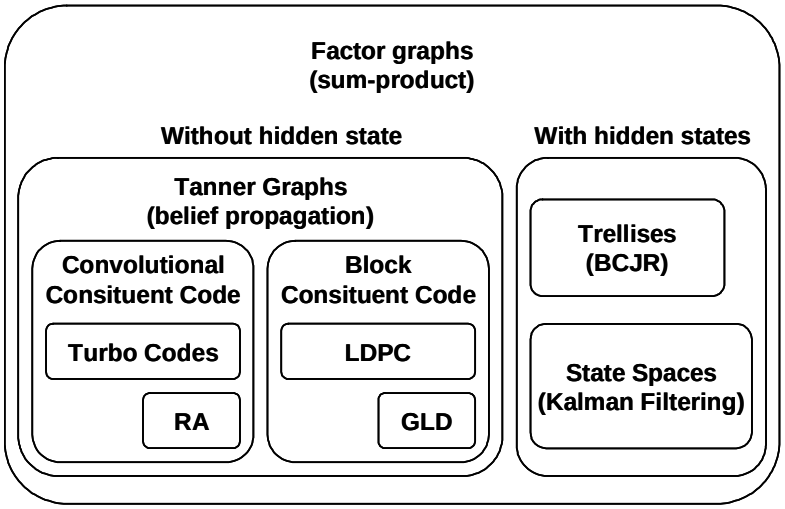


Figure 25. classical decoding algorithm as a particular instance of the sum-product algorithm in a factor graph.

Hence, LDPC codes are at the confluence of two major revolutions in the channel coding community: the graph-based code-description, and the iterative decoding techniques.

3.5.2 Classes of LDPC Codes

R. Gallager (Gallager 1962) defined an (N, j, k) LDPC codes as a block code of length N having a small fixed number (j) of ones in each column of the parity check H, and a small fixed number (k) of ones in each rows of H. This class of codes is then to be decoded by the iterative algorithm.

This algorithm computes exact a posteriori probabilities, provided that the Tanner

graph of the code is cycle free. Generally, LDPC codes do have cycles (Etzion, Trachtenberg, and Vardy 1999). The sparseness of the parity check matrix aims at reducing the number of cycles and at increasing the size of the cycles.

Moreover, as the length N of the code increases, the cycle free hypothesis becomes more and more realistic. The iterative algorithm is processed on these graphs. Although it is not optimal, it performs quite well.

Since then, LDPC codes class have been enlarged to all sparse parity check matrices, thus creating a very wide class of codes, including the extension to codes in GF(q) (Davey and MacKay 1998) and irregular LDPC codes (Luby et al. 2001).

3.5.2.1 Irregularity

In the Gallager's original LDPC code design, there is a fixed number of ones in both the rows (*k*) and the columns (*j*) of the parity check matrix: it means that each bit is implied in j parity check constraints and that each parity check constraint is the exclusive-OR (XOR) of k bits. This class of codes is referred to as regular LDPC codes.

On the contrary, irregular LDPC codes do not have a constant number of non-zero entries in the rows or in the columns of H. They are specified by the distribution degree of the bit $\lambda(x)$ and of the parity check constraints $\rho(x)$, using the notations of (Luby et al. 1997), where:

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1} \tag{23}$$

$$\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1} \tag{24}$$

λ_i (resp. ρ_i) denotes the proportion of non-zero entries of *H* which belongs to the columns (resp. rows) of *H* of weight i. Note that by definition, $\lambda(1) = \rho(1) = 1$. If Γ denotes the number of non-zero entries in *H*, $\lambda_i \Gamma$ is the total number of ones in the columns of weight *i*. So $\lambda_i \Gamma / i$ is the total number of columns of weight *i*, and $\Sigma_i \Gamma \lambda_i / i$ is the total number of columns in *H*. So the proportion of the columns of weight *i* is:

$$\widetilde{\lambda_i} = \frac{\Gamma \lambda_i / i}{\Sigma_j \Gamma \lambda_j / j} = \frac{\lambda_i / i}{\Sigma_j \lambda_j / j} \tag{25}$$

Similarly, denoting by $\widetilde{\rho_i}$ the proportion of rows having weight *i*:

$$\widetilde{\rho_i} = \frac{\rho_i / i}{\Sigma_j \rho_j / j} \tag{26}$$

3.5.2.2 Code rate

The rate *R* of LDPC codes is defined by $R \geq R_d \cong 1 - M/N$ where R_s is the design code rate (Gallager 1962). $R_d = R$ if the parity check matrix has full rank. The authors of (Miller and Cohen 2003) have shown that as *N* increases, the parity-check matrix is almost sure to be full rank. Hereafter, we will assume that $R_d = R$ unless the contrary is mentioned.

The rate *R* is then linked to the other parameters of the class by:

$$R = 1 - \frac{\Sigma_i \rho_i / i}{\Sigma_i \lambda_i / i} = 1 - \frac{j}{k} \tag{27}$$

Note that in general, for random constructions, when j is odd:

$$R = 1 - \frac{M}{N} \tag{28}$$

and when j is even:

$$R = 1 - \frac{M-1}{N} \tag{29}$$

3.5.3 Encoding and Decoding of LDPC codes

LDPC code is an error correction code that has been receiving the most attention recently. This code was proposed in early 1960s by Gallager. It is defined to be a code for which number of non zero elements in the parity check matrix is considerably low compared to the code length. It is an error correcting code that most closely approaches the Shannon limit. Along with turbo code, it is regarded as an excellent error correcting code that can be used in 4th generation mobile communication systems.

Equation (30) shows the LDPC encoding process.

$$\begin{aligned} &H = (A_p^{-1} \cdot A) \bmod 2 = [I \quad A_2] \\ &G = \left(\frac{A_2}{I} \right) \\ &c = (G \cdot m) \bmod 2 \end{aligned} \tag{30}$$

A, H : parity check matrix

A_p^{-1} : inverse pivot matrix

G : generator matrix

m : transmission message

c : code word

After generator matrix *G* is created from parity check matrix *A* and *H*, code word *c* is generated using *G* and message *m*.

Equation (31) shows the LDPC decoding process.

$$q_n(x) = \alpha P(c_n = x|r_n) \Pi_{m \in n} P(z_m = 0|c_n = x, r) \quad (31)$$

$q_n(x)$: pseudo posterior probability

$\alpha P(c_n = x|r_n)$: intrinsic probability

$\Pi_{m \in n} P(z_m = 0|c_n = x, r)$: extrinsic probability

z_m : parity check bits

c_n : code word

n, m : row, column index

r : total received data

Code word *c* that has passed through the transmission channel is combined with noise and error components and is received as *r*. The received signal is decoded by calculating the posterior probability using equation (31).

3.6 Wavelet

Wavelets, wavelet analysis, and the wavelet transform refers to the representation of a signal in terms of a finite length or fast decaying oscillating waveform (known as the mother wavelet). This waveform is scaled and translated to match the input signal. In formal terms, this representation is a wavelet series, which is the coordinate representation of a square integrable function with respect to a complete, orthonormal set of basis functions for the Hilbert space of square integrable functions. Note that the wavelets in the JPEG2000 standard are biorthogonal wavelets, that is, the coordinates in the wavelet series are computed with a different, dual set of basis functions.

Wavelet theory is applicable to several other subjects. All wavelet transforms may be considered to be forms of time-frequency representation and are, therefore, related to the subject of harmonic analysis. Almost all practically useful *discrete wavelet transforms* make use of filterbanks containing finite impulse response filters. The wavelets forming a CWT are subject to Heisenberg's uncertainty principle and, equivalently, discrete wavelet bases may be considered in the context of other forms of the uncertainty principle.

3.6.1 Continuous Wavelet Transforms

In the continuous wavelet transform, a given signal of finite energy is projected on a continuous family of frequency bands (or similar subspaces of the function space $L^2(\mathbb{R})$), for instance on every frequency band of the form [*f*,2*f*] for all positive frequencies *f>0*. By a suitable integration over all the thus obtained frequency components one can reconstruct the original signal.

The frequency bands or subspaces are scaled versions of a subspace at scale *l*. This subspace in turn is in most situations generated by the shifts of one generating function $\psi \in L^2(\mathbb{R})$, the *mother wavelet*. For the example of the scale one frequency band with the (normalized) sinc function.

$$\psi(t) = 2sinc(2t) - sinc(t) = \frac{sn(2\pi t) - sin(\pi t)}{\pi t} \tag{32}$$

Other example mother wavelets are:

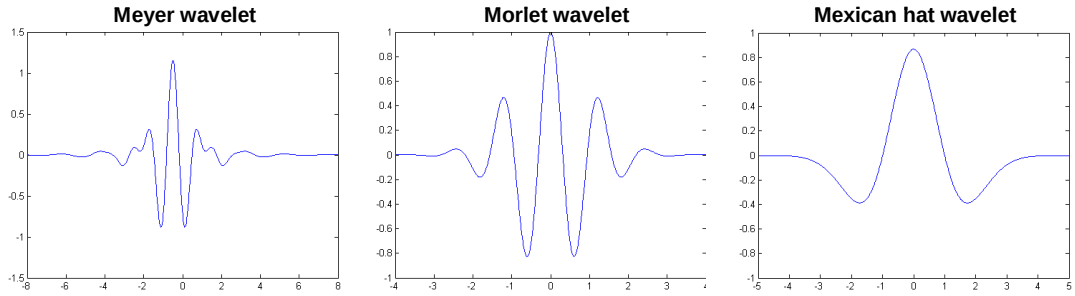


Figure 26. Other example mother wavelets.

The subspace of scale *a* or frequency band [1/a, 2/a] is generated by the functions (sometimes called *baby wavelets*)

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}) \tag{33}$$

where *a* is positive and defines the scale and *b* is any real number and defines the shift. The pair *(a,b)* defines a point in the upper halfplane $\mathbb{R}_+ \times \mathbb{R}$.

The projection of a function *x* onto the subspace of scale *a* has then the form

$$x_a(t) = \int_{\mathbb{R}} WT_\phi\{x\}(a,b) \cdot \psi_{a,b}(t)db \tag{34}$$

with wavelet coefficients

$$WT_\phi\{x\}(a,b) = < x, \psi_{a,b} > = \int_{\mathbb{R}} x(t)\overline{\psi_{a,b}(t)}\,dt \tag{35}$$

For the analysis of the signal *x*, one can assemble the wavelet coefficients into a scale of the signal.

3.6.2 Discretized Wavelet Transform

It is computationally impossible to analyze a signal using all wavelet coefficients. So one may wonder if it is sufficient to pick a discrete subset of the upper halfplane to be able to reconstruct a signal from the corresponding wavelet coefficients. One such system is the affine system for some real parameters *a>1*, *b>0*. The corresponding discrete subset of the halfplane consists of all the points $(a^m, na^m b)$ with integers $m, n \in Z$. The corresponding baby wavelets are now given as

$$\psi_{m,n}(t) = a^{-m/2}\psi(a^{-m}t - nb) \tag{36}$$

A sufficient condition for the reconstruction of any signal *x* of finite energy by the equation (37).

$$x(t) = \sum_{m \in Z}\sum_{n \in Z} < x, \psi_{m,n} > \cdot \psi_{m,n}(t) \tag{37}$$

Equation (37) is that the functions $\{\psi_{m,n} : m, n \in Z\}$ form a tight frame of $L^2(R)$.

3.6.3 MRA based Discrete Wavelet Transforms

In each instance of the disretised wavelet transform, there are only a finite number of wavelet coefficients for each bounded rectangular region in the upper halfplane. Still, each coefficient requires the evaluation of an integral. To avoid this numerical complexity one needs one auxiliary function, the *father wavelet* $\phi \in L^2(R)$. Further, one has to restrict *a* to be an integer number. A typical choice is *a*=2 and *b*=1. The most famous pair of father and mother wavelets is the Daubechies 4 tap wavelet.

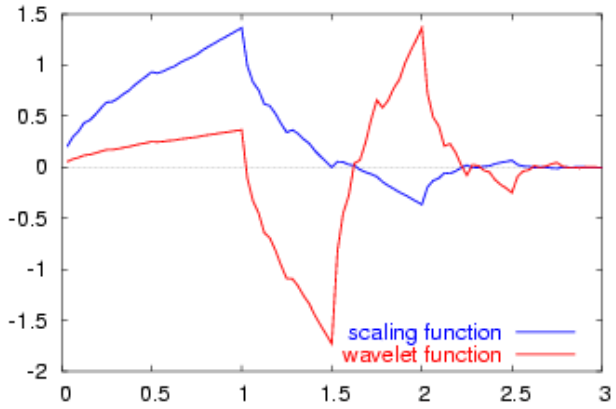


Figure 27. Daubechies 4 tap wavelet.

From the mother and father wavelets one constructs the subspaces

$$V_m = span(\phi_{m,n} : n \in Z), \text{ where } \phi_{m,n}(t) = 2^{-m/2}\phi(2^{-m}t - n)$$

$$W_m = span(\psi_{m,n} : n \in Z), \text{ where } \psi_{m,n}(t) = 2^{-m/2}\psi(2^{-m}t - n)$$

From these one requires that the sequence

$$\{0\} \subset \cdots \subset V_1 \subset V_0 \subset V_{-1} \subset \cdots \subset L^2(R)$$

Forms a multiresolution analysis of $L^2(R)$ and that the subspaces $\cdots, W_1, W_0, W_{-1}, \cdots$ are the orthogonal "differences" of the above sequence, that is, W_m ist the orthogonal complement of V_m inside the subspace V_{m-1}. In analogy to the sampling theorem one may conclude that the space V_m with sampling distance 2^m more or less covers the frequency baseband from 0 to 2^{-m-1}. As orthogonal complement, W_m roughly covers the band $[2^{-m-1}, 2^{-m}]$.

From those inclusions and orthogonality relations follows the existence of sequences $h = \{h_n\}_{n \in Z}$ and $g = \{g_n\}_{n \in Z}$ that satisfy the identities

$$h_n = < \phi_{0,0}, \phi_{1,n} > \text{ and } \phi(t) = \sqrt{2} \sum_{n \in Z} h_n \phi(2t - n) \tag{38}$$

$$g_n = < \psi_{0,0}, \phi_{1,n} > \text{ and } \psi(t) = \sqrt{2} \sum_{n \in Z} h_n \phi(2t - n) \tag{39}$$

The second identity of the first pair is a refinement equation for the father wavelet ϕ. Both pairs of identities form the basis for the algorithm of the fast wavelet transform.

3.6.4 Mother Wavelet

For practical applications one prefers for efficiency reasons continuously differentiable functions with compact support as mother (prototype) wavelet (functions). However, to satisfy analytical requirements (in the continuous WT) and in general for theoretical reasons one chooses the wavelet functions from a subspace of the space $L^1(R) \cap L^2(R)$. This is the space of measurable functions that are both absolutely and square integrable:

$$\int_{-\infty}^{\infty} |\psi(t)| dt < \infty, \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \tag{40}$$

Being in this space ensures that one can formulate the conditions of zero mean and square norm one:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \tag{41}$$

Equation (41) is the condition for zero mean

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1 \tag{42}$$

Equation (42) is the condition for square norm one.

For Ψ to be a wavelet for the continuous wavelet transform (see there for exact statement), the mother wavelet must satisfy an admissibility criterion (loosely speaking, a kind of half-differentiability) in order to get a stably invertible transform.

For the discrete wavelet transform, one needs at least the condition that the wavelet series is a representation of the identity in the space $L^2(R)$. Most constructions of discrete WT make use of the multiresolution analysis, which defines the wavelet by a scaling function. This scaling function itself is solution to a functional equation.

In most situations it is useful to restrict Ψ to be a continuous function with a higher number *M* of vanishing moments, i.e. for all integer $m < M$

$$\int_{-\infty}^{\infty} t^m \psi(t) dt = 0 \tag{43}$$

The mother wavelet is scaled (or dilated) by a factor of *a* and translated (or shifted) by a factor of *b* to give (under Morlet's original formulation):

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \tag{44}$$

For the continuous WT, the pair *(a,b)* varies over the full halfplane $R_+ \times R$; for

the discrete WT this pair varies over a discrete subset of it, which is also called affine group.

These functions are often incorrectly referred to as the basis functions of the (continuous) transform. In fact, as in the continuous Fourier transform, there is no basis in the continuous wavelet transform. Time-frequency interpretation uses a subtly different formulation.

3.7 Associative Memories

3.7.1 Hopfield Model

The Hopfield model was proposed by John Hopfield of the California Institute of Technology during the early 1980s. The publication of his work in 1982 significantly contributed to the renewed interest in research in artificial neural networks. He showed how an ensemble of simple processing units can have fairly complex collective computational abilities and behavior.

The dynamics of the Hopfield model is different from that of the linear associator model in that it computes its output recursively in time until the system becomes stable. Below is a Hopfield model with six units, where each node is connected to every other node in the network.

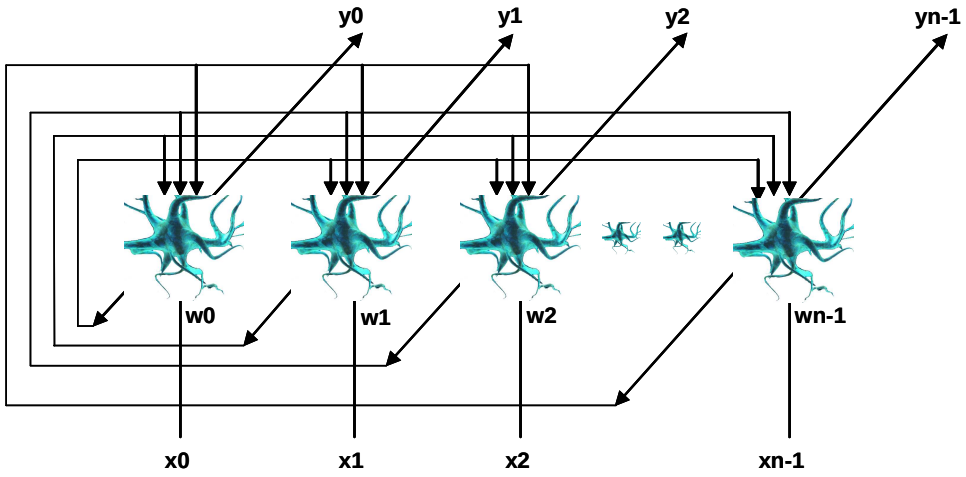


Figure 28. Hopfield model.

Unlike the linear associator model which consists of two layers of processing units, one serving as the input layer while the other as the output layer, the Hopfield model

consists of a single layer of processing elements where each unit is connected to every other unit in the network other than itself. The connection weight matrix W of this type of network is square and symmetric.

$$w_{ji} = w_{ij} \tag{45}$$

for $i, j = 1, 2, \cdots, m.$

Each unit has an extra external input I_i. This extra input leads to a modification in the computation of the net input to the units:

$$input_j = \sum_{i=1}^{m} x_i w_{ij} + I_j \tag{46}$$

for $j = 1, 2, \cdots, m.$

Unlike the linear associator, the units in the Hopfield model act as both input and output units. But just like the linear associator, a single associated pattern pair is stored by computing the weight matrix as follows:

$$w_k = X_k^T Y_k \quad \text{where} \quad Y_k = X_k \tag{47}$$

$$W = a \sum_{k=1}^{p} W_k \tag{48}$$

To store *p* different associated pattern pairs. Since the Hopfield model is an autoassociative memory model, patterns, rather than associated pattern pairs, are stored in memory.

After encoding, the network can be used for decoding. Decoding in the Hopfield

model is achieved by a collective and recursive relaxation search for a stored pattern given an initial stimulus pattern. Given an input pattern X, decoding is accomplished by computing the net input to the units and determining the output of those units using the output function to produce the pattern X'. The pattern X' is then fed back to the units as an input pattern to produce the pattern X''. The pattern X'' is again fed back to the units to produce the pattern X'''. The process is repeated until the network stabilizes on a stored pattern where further computations do not change the output of the units.

If the input pattern X is an incomplete pattern or if it contains some distortions, the stored pattern to which the network stabilizes is typically one that is most similar to X without the distortions. This feature is called *pattern completion* and is very useful in many image processing applications.

During decoding, there are several schemes that can be used to update the output of the units. The updating schemes are *synchronous* (or parallel as termed in some literatures), *asynchronous* (or sequential), or a combination of the two (*hybrid*).

Using the synchronous updating scheme, the output of the units are updated as a group prior to feeding the output back to the network. On the other hand, using the asynchronous updating scheme, the output of the units are updated in some order (e.g. random or sequential) and the output are then fed back to the network after each unit update. Using the hybrid synchronous-asynchronous updating scheme, subgroups of units are updated synchronously while units in each subgroup updated asynchronously. The choice of the updating scheme has an effect on the convergence of the network.

Hopfield (1982) demonstrated that the maximum number of patterns that can be stored in the Hopfield model of *m* nodes before the error in the retrieved pattern becomes severe is around 0.15*m*. The memory capacity of the Hopfield model can be increased as shown by Andrecut (1972).

In spite of this seemingly limited memory capacity of the Hopfield model, several

applications can be listed (Chaudhuri, Dai, deMenezes, Garcia, Poli, Soper, Suh, Tsai). Discussions on the application of the Hopfield model to combinatorial optimization problems can be found in (Siu, Suh).

Various widely available text and technical papers vy Hassoun, Hertz (et al), Hopfield, McEliece, Ritter, and Volk can be consulted for a mathematical discussion on the memory capacity of the Hopfield model.

3.7.2 Discrete Hopfield Model

In the discrete Hopfield model, the units use a slightly modified bipolar output function where the states of the units, i.e., the output of the units remain the same if the current state is equal to some threshold value:

$$x_i(t+1) = \begin{cases} +1 & \text{if} \quad input_i > \theta_i \\ x_{i(t)} & \text{if} \quad input_i = \theta_i \\ -1 & \text{if} \quad input_i < \theta_i \end{cases} \tag{49}$$

for $i = 1, 2, \cdots, m$ and where t denotes the discrete time.

An interesting property of recurrent type networks is that their state can be described by an *energy function*. The energy function is used to prove the stability of recurrent type networks. For the discrete Hopfield model with $w_{ii} = 0$ and $w_{ij} = w_{ji}$ using the asynchronous updating scheme, the energy function E according to Hopfield (1982) is defined as:

$$E = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} x_i w_{ij} x_j - \sum_{i=1}^{m} x_i I_i + \sum_{i=1}^{m} x_i \theta_i \tag{50}$$

where the local minima of the energy function correspond to the energy of the stored

patterns. Hopfield (1982) has shown that the energy of the discrete Hopfield model decreases or remains the same after each unit update. Therefore, the network will eventually converge to a local minimum that corresponds to a stored pattern. The stored pattern to which the network converges depends on the input pattern and the connection weight matrix.

The energy of the discrete Hopfield model is bounded from below by:

$$E = -\sum_{i=1}^{m}\sum_{j=1}^{m}|w_{ij}| - \sum_{i=1}^{m}|I_i| + \sum_{i=1}^{m}|\theta_i| \tag{51}$$

for all $X_k, k = 1, 2, \cdots, p$. Since the energy is bounded from below, the network will eventually converge to a local minimum corresponding to a stored pattern.

3.7.3 Continuous Hopfield Model

The continuous Hopfield model is just a generalization of the discrete case. Here, the units use a continuous output function such as the sigmoid or hyperbolic tangent function. In the continuous Hopfield model, each unit has an associated capacitor C_i and resistance r_i that model the capacitance and resistance of real neuron's cell membrane, respectively. Thus the state equation of each unit is now:

$$C_j = \frac{dinput_j}{dt} = \sum_{i=1}^{m} x_i w_{ij} - \frac{input_j}{R_j} + I_j \tag{52}$$

where

$$\frac{1}{R_j} = \frac{1}{p_j} + \sum_{i=1}^{m} w_{ij} \tag{53}$$

Just like in the discrete case, there is an energy function characterizing the continuous Hopfield model. The energy function due to Hopfield (1984) is given by:

$$E = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} x_i w_{ij} x_j - \sum_{i=1}^{m} x_i I_i + \sum_{i=1}^{m} \left(\frac{1}{R_i}\right) \int_0^{x} f^{-1}(x)\, dx \qquad (54)$$

where f is the output function of the units.

It can be shown that $dE/dt = 0$ when $w_{ij} = w_{ji}$. Therefore, the energy of the continuous Hopfield model decreases or remains the same. The minimum energy of the continuous Hopfield model also exists using analogous computation as that in the discrete Hopfield model.

Chapter 4. The Proposed Algorithm and Implementation

In this paper, a speech recognition module, a face recognition module and a palmprint recognition module for authenticating persons who enter and exit a building using the designed sensor modules was proposed. Also, communication channel establishment algorithm was proposed between cluster head and sensors. Figure 29 shows the total block diagram of the proposed system.

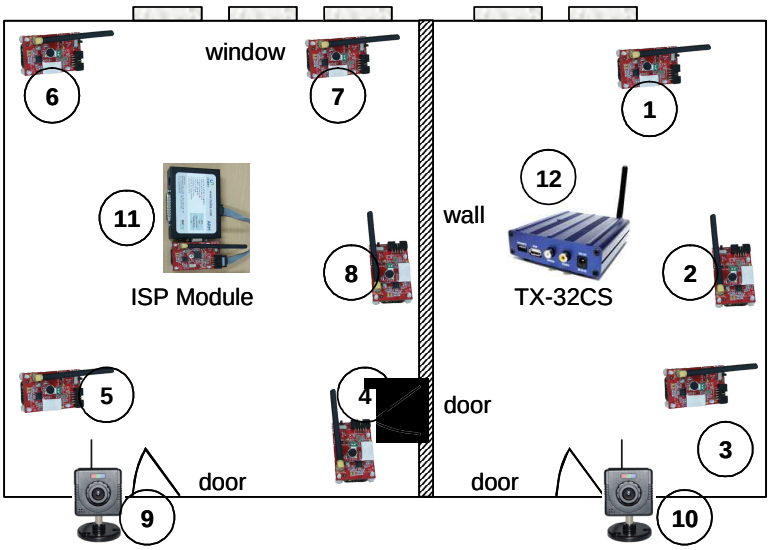


Figure 29. Total block diagram.

In the figure 29:

- Wireless audio sensors : 1～8
- Wireless image sensors : 9, 10
- Audio data collecting module (ISP 2400) : 11
- Image data collecting module (TX-32CS) : 12

Overall block diagram of the proposed system is as shown in figure 30.

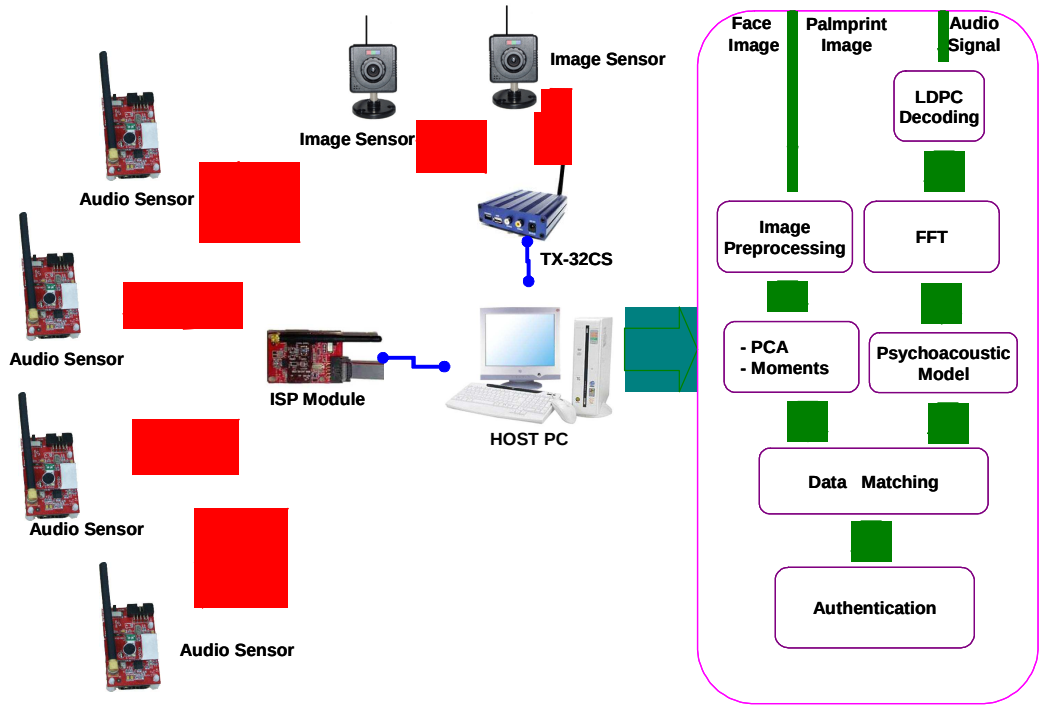


Figure 30. Block diagram of the proposed system.

4.1 Wireless Speech Recognition System

In this paper, a speech recognition module for authenticating persons who enter and exit a building using the designed speech recognition sensor modules was proposed.

Speech signal that is collected through sensors are transformed into frequency components using FFT. Then the Psychoacoustic model is applied to extract the 24 speech characteristic values that are effective for speech recognition. In this system, persons who enter and exit are identified by examining the coefficient of correlation between the extracted data and database stored in the HOST PC. The whole system is designed with a JAVA base. The proposed system is shown in figure 31.

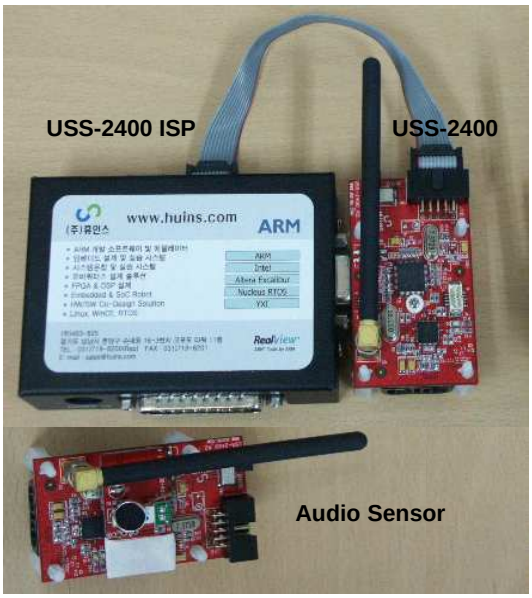


Figure 31. Implemented system module.

4.2 Wireless Palmprint Recognition System

The palmprint authentication algorithm suggested in this paper is composed of 5 systematic steps listed as followings; the histogram equalization, the smoothing filter, the Otsu binarization, the invariant moment and the search algorithm.

The histogram equalization and the smoothing filter allow the accurate distinctions for background regions and wrinkles as well as principle lines of palmprints which are needed in the palmprint identification system. Moreover, they also provide subtle protections against the changes occurring in brightness or the damages in palmprints. The Otsu binarization process is the process to transfer the palmprint image into the binary data of '0' and '1' thus allow the calculations of invariant moment. This invariant moment calculation would provide an original value of palmprints for distinguishing each individual.

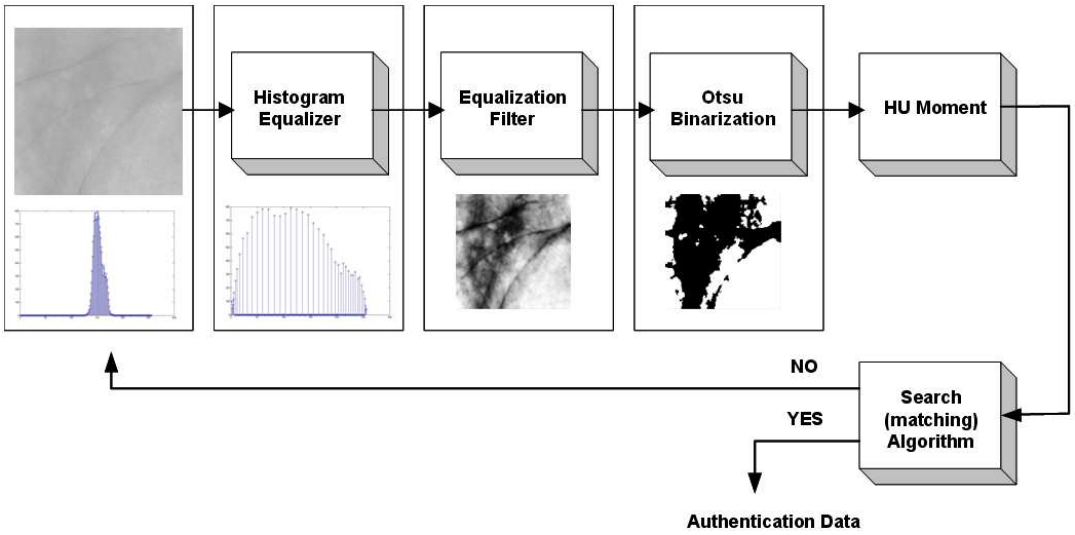


Figure 32. The Palmprint Authentication Algorithm Block Mapping.

4.2.1 Histogram Equalization

The human eye is more likely to increase its recognition level when the level of contrast increased rather than the level of absolute brightness increased. It does not necessary mean that the visibility should improve as the overall brightness of image is lightened, and the equal distributions in values of brightness provide better means of identification. By considering these aspects, the palmprint identification algorithm also employed the histogram equalization technique to accurately distinguish wrinkles and principle lines in a palmprint image.

The histogram equalization composed of 3 steps as following;

Step 1: generate the histogram regarding the brightness values of the original image.

Step 2: transform the generated histogram into the normalization histogram.

Step 3: execute the re-mapping sequence for input images using the regulated histogram.

Equation (55) represents the normalization equation to equalize the histogram and figure 33 demonstrates palmprint image at before and after the histogram procedure.

$$h(i) = \frac{G_{\max}}{N_t} H(i) \tag{55}$$

H(i): the accumulation histogram of input image

h(i): the normalization histogram

G_{max}: the maximum brightness level of image (255)

Ni: the number of total pixel

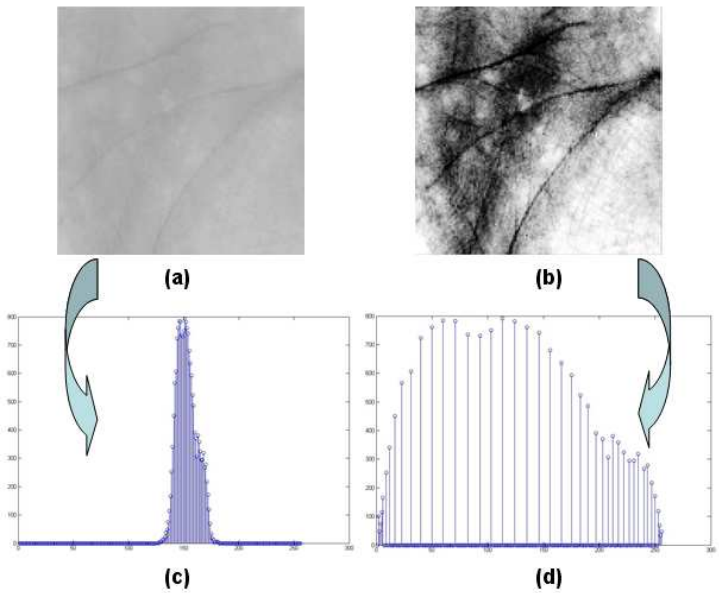


Figure 33. The image before and after the histogram equalization.

Figure 33 (a) and (c) represent the palmprint image and the histogram obtained by applying the palmprint acquisition devices whereas (b) and (d) are the equalized palmprint images and histograms which being generated by employing equation (32).

4.2.2 Smoothing Filter

The smoothing filter is generally applied to eliminate the noise of images. However, this technique was used to eliminate those image components which might generate possible errors

in identifying the palmprint because of the unaccounted appearances of small wrinkles and scars on the palmprint.

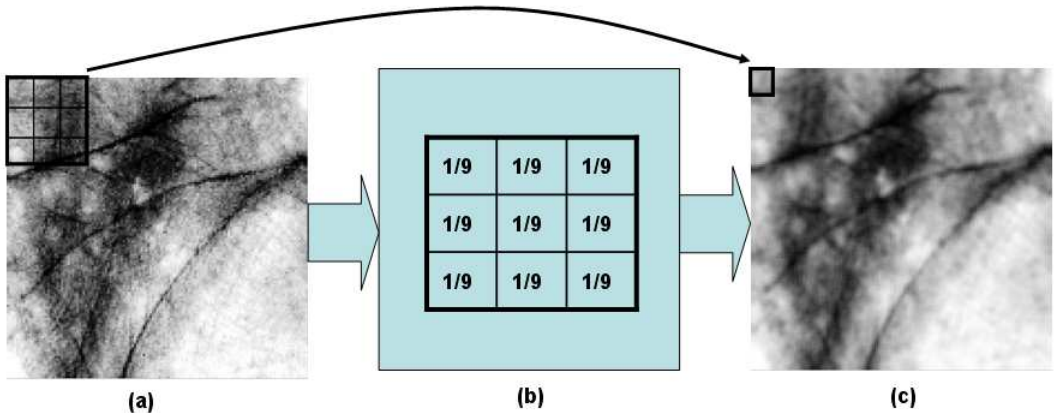


Figure 34. The Procedures of smoothing filter process.

Figure 34 (a) represents the image which already went through the histogram equalization and (b) is the filter coefficient used in smoothing filter. As a result, (c) which passed through the smoothing filter to rid of small wrinkles is generally left with the large principle lines.

4.2.3 Otsu Binarization

As a method to calculate the critical values of the histogram which composed of two fixed points, the maximum variance binarization technique suggested by Otsu was considered. As shown in figure 35 below, it is a technique to divide the histogram by selecting the critical value to maximize the variance between the divided regions when the histogram of the two fixed points were divided with a critical value as a standard.

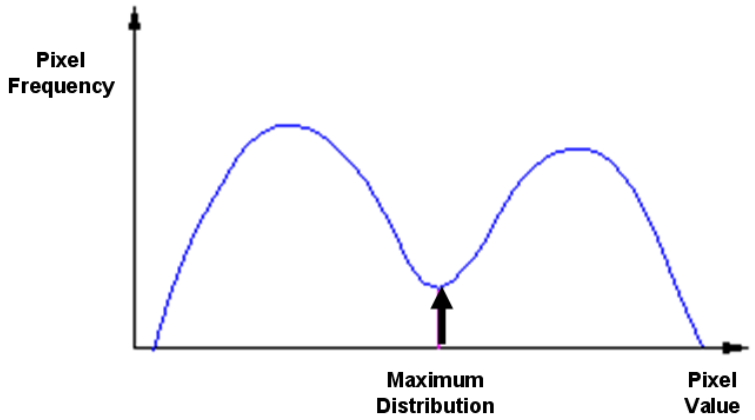


Figure 35. The Maximum Variance of Histogram.

By defining 1 ~ m as the pixel value of an image and ni as the frequency of the pixel value of 'i', the overall pixel value and each pixel value could be expressed as the equation (56) and (57).

$$N = \sum_{i=1}^{m} n_i \tag{56}$$

$$P_i = \frac{n_i}{N} \tag{57}$$

N: the total number of pixel

Pi: the frequency of each pixel value

When the histogram was divided into two groups as $C_0 = \{1 \sim k\}$, $C_1 = \{(k+1) \sim m\}$ by using k, the pixel value, appearance probability and average values of C0 and C1 could be represented as the equation from (58) to (61).

$$e_0 = \sum_{i=1}^{k} P_i = e(k) \tag{58}$$

$$e_1 = \sum_{i=k+1}^{m} P_i = 1 - e(k) \tag{59}$$

$$f_0 = \sum_{i=1}^{k} ip(i|C_0) = \sum_{i=1}^{k} \frac{iP_i}{e_0} = \frac{f(k)}{e(k)} \tag{60}$$

$$f_1 = \sum_{i=k+1}^{m} ip(i|C_1) = \sum_{i=k+1}^{m} \frac{iP_i}{e_1} = \frac{f - f(k)}{1 - e(k)} \tag{61}$$

e0: the appearance probability of C0

e1: the appearance probability of C1

f0: the average value of C0

f1: the average value of C1

$f = \sum_{i=1}^{m} ip_i$, $f(k) = \sum_{i=1}^{k} ip_i$ is the average value of the total sample values.

Therefore f could be expressed as equation (62).

$$f = e_0 f_0 + e_1 f_1 \tag{62}$$

The variance between the two regions could be expressed as equation (63).

$$\sigma^2(k) = e_0(f_0 - f)^2 + e_1(f_1 - f)^2 = e_0 e_1 (f_1 - f_0)^2 = \frac{[f e(k) - f(k)]^2}{e(k)[1 - e(k)]} \tag{63}$$

The constant, *k* which grants the maximum value of $\sigma_2(k)$ can be calculated by applying equation (63) and the histogram can be separated by considering this constant as the critical value.

In order to minimize the amount of necessary calculations, the overall image was binarized into 0 and 1 by considering the maximum variance point as the standard while identifying the palmprint. Figure 36 (a) represents the maximum variance point of the histogram and (b) is the image which is binarized by considering the maximum variancepoint as the standard.

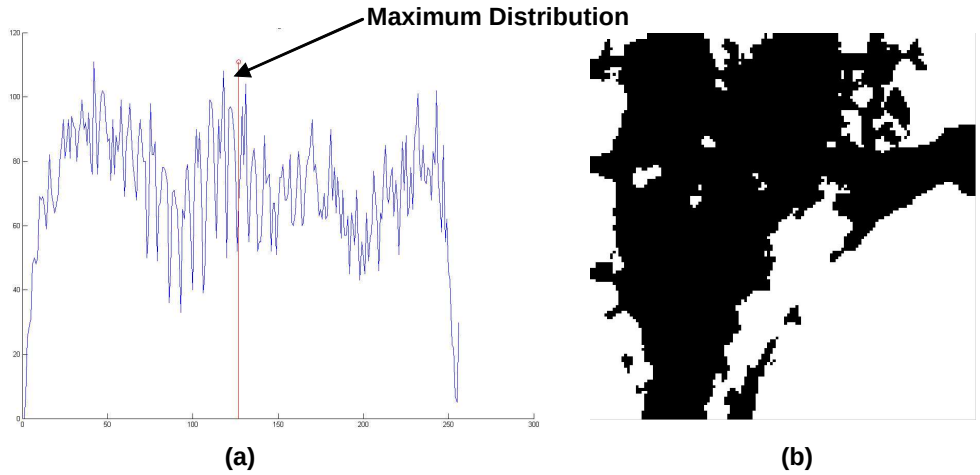


Figure 36. The Otsu binarization image of Palmprint.

4.2.4 Invariant Moment

According to the uniqueness theorem [49], in case of the image composed of 2-dimension continuous function *f(x, y)*, if *f(x, y)* was continuous through the nodes and certain section of *x-y* surface carried a value other than 0, the moment of all degrees would exist and the moment, m_{pq} of degree, *(p+q)* would be determined by *f(x, y)*. And, *f(x, y)* would be uniquely determined from m_{pq} as well. The moment can be expressed as equation (64) under the condition if the digital image *f(i, j)* were a binarization image.

$$m_{pq} = \sum_{i}\sum_{j} i^{p} j^{q} \quad (p,q = 1, 2, 3,\cdots) \tag{64}$$

The central moment can be defined as equation (65) and it features with the invariant characteristic against the movement.

$$\mu_{pq} = \sum_{i}\sum_{j}(i - \bar{x})^{p}(j - \bar{y})^{q} \tag{65}$$

The indexes of *i* and *j* in equation (65) represent the location of the horizontal *x*-axis and the vertical *y*-axis, respectively. The moment, m_{00} represents the binary objective region and the central moment up to the level 3 moment sustains the similar relationship as equation (66).

$$\begin{aligned}
\mu_{00} &= m_{00} = \mu \\
\mu_{10} &= \mu_{01} = 0 \\
\mu_{20} &= m_{20} - \mu\bar{x}^{2} \\
\mu_{11} &= m_{11} - \mu\overline{xy} \\
\mu_{02} &= m_{02} - \mu\bar{y}^{2} \\
\mu_{03} &= m_{03} - 3m_{02}\bar{y} + 2\mu\bar{y}^{3} \\
\mu_{30} &= m_{30} - 3m_{20}\bar{x} + 2\mu\bar{x}^{3} \\
\mu_{21} &= m_{21} - m_{20}\bar{y} - 2m_{11}\bar{x} + 2\mu\bar{x}^{2}\bar{y}^{2} \\
\mu_{12} &= m_{12} - m_{02}\bar{x} - 2m_{11}\bar{y} + 2\mu\overline{xy}^{2}
\end{aligned} \tag{66}$$

The physical definitions of central moment can be organized as below.

U20: abscissa variance

U02: ordinate variance

U11: covariance of abscissa and ordinate

U12: distribution intensity towards the left side compared to the right side in abscissa

U21: distribution intensity towards the lower side compared to the upper side in ordinate

U30: abscissa skew intensity

U03: ordinate skew intensity

Moreover, the normalization moment divides the central moment into a certain size of values as shown in equation (67) and this sizing grants the invariant characteristic[50].

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{r}}, \quad \gamma = \frac{p+q}{2} + 1 \tag{67}$$

In this paper, HU invariance moment[51, 52] was extracted by applying equation (66) and (67) and then it was used for the palmprint identification algorithm. HU invariance moment is constituted of level 2 and 3 central moments as demonstrated in equation (68).

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\
&\quad - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
&\quad \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\
&\quad - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \\
&\quad \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned} \tag{68}$$

HU moment defined in equation (68) can be explained as followings;

Ø1: the sum of horizontal and vertical directed variance, more distributed towards horizontal and vertical axes, the values are enlarged.

Ø2: the covariance value of vertical and horizontal axes when the variance intensity of

vertical axis and horizontal axis were similar.

Ø3: the result emphasizing the values inclined to left/right and upper/lower axes.

Ø4: the result emphasizing the values counterbalancing to left/right and upper/lower axes.

Ø5, Ø6, Ø7: the extraction of values invariant against size, rotation, and location

4.2.5 Search Algorithm

5^{th} and 7^{th} moments of the HU moments applied in search algorithm have very small changes in terms of values thus they scarcely affect the palmprint identification. Moreover, they tend to accompany increase in amount of calculations thus they were exempted from the palmprint searching algorithm.

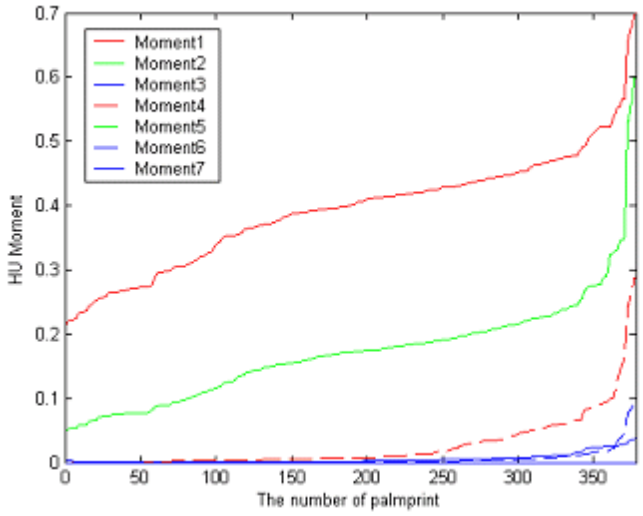


Figure 37. HU invariance moment distribution chart of Palmprint.

The search algorithm generally was based on the minimum Euclidean distance calculation and this was practiced by using equation (69).

$$V_M = \left| \frac{x_{dbM}}{x_{inM}} \right|$$
$$V_T = \frac{\sum_{M=0}^{M-1} V_M - 1}{M} \quad (69)$$
$$result = coeff \ \geqq \ V_T$$

M : Moment order

XinM : HU moment of input data

XdbM : HU moment of DB data

VM : Normalize data

VT : Distance data

coeff : Critical value(0.001 ~ 0.005)

Viz., this is an algorithm searching for the data which its distance value between the HU invariance moments of palmprint is below the certain value for 5 units of HU invariance moment embedded within the database and authentication.

4.3 Wireless Face Recognition System

The proposed face identification algorithm consists of the following 5 steps: face normalization, average face calculation, covariance matrix calculation of face, extraction of Eigen face and the search algorithm. Figure 38 shows the proposed face identification algorithm.

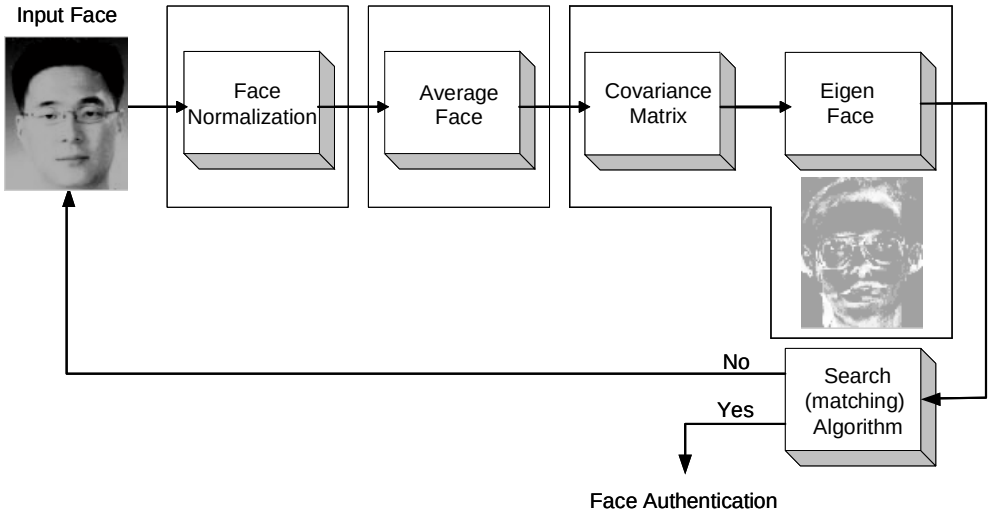


Figure 38. The proposed algorithm of face identification.

4.3.1 Face Normalization

Face image normalization clearly distinguishes the face background, wrinkles and lines for identification and confers robustness against changes in contrast caused by external noise or face damage. Equation (70) shows the normalization process using standard deviation. The assumption average was set to 100 and the assumption standard deviation was set to 80 after testing about 50 faces.

$$P = (I - \bar{I}) \times \frac{ustd}{std} + um \tag{70}$$

P = Normalized Face

I = Input Data

$\bar{I}$ = Average of I

um = Assumption Average

$ustd$ = Assumption Standard Deviation

Figure 39. Normalized face.

Figure 39. shows normalized face images after applying equation (70).

4.3.2 Eigen Face

When each pixel of a face image is to be formed on the face space, the Eigen face represents the basis vector forming the face space. It is a basis vector that best represents the common characteristics of all recognition candidate faces. In other words, Eigen face is the differential vector between the average face image of all recognition candidate face images and each candidate face image. It is the Eigen vector ofthe covariance matrix. The Eigen value of covariance matrix shows the degree of distribution of the average face image so that the Eigen face composed of the Eigen vector corresponding to the largest Eigen value is the closest face. As Eigen value decreases, the face characteristics decrease. Equation (71) shows the covariance matrix of a face.

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T,$$
$$A = \frac{1}{\sqrt{M}}[\Phi_1, \Phi_2, \Phi_3, \cdots, \Phi_M] \tag{71}$$

C : Covariance matrix

M : Number of the recognition candidate face vector

In equation (71), Φ is the differential vector of the recognition candidate face vector and the average face vector. Expression for Φ is given by equation (72). Ψ is the average face vector expressed as equation (73).

$$\Phi_i = \Gamma_i - \Psi \tag{72}$$

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n \tag{73}$$

Γ in equations (72) and (73) is the resulting vector when the vector recognition candidate face is realigned as a row vector.

From covariance matrix *C*, the Eigen value λ and the corresponding Eigen vector μ are calculated. Here, the Eigen value represents the degree of distribution of the average face image. Eigen vector μ can be calculated using equation (74). Eigen vector is the basis vector for each candidate face and is stored in the database.

$$\mu_l = \frac{1}{M}\sum_{k=1}^{M}\lambda_{lk}\Phi_k \qquad l = 1, 2, \cdots, M \tag{74}$$

Figure 40 is the Eigen face calculated using equation (74).

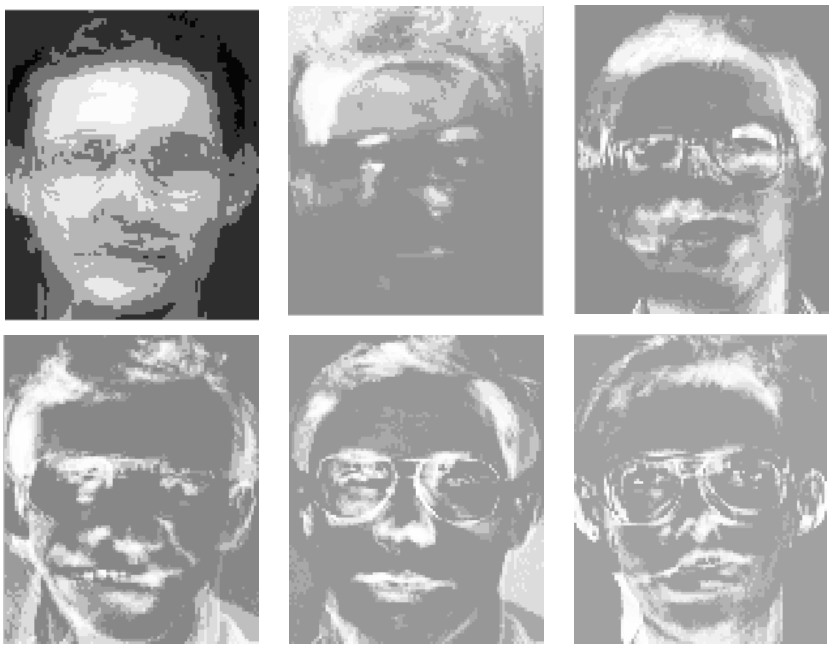


Figure 40. Eigen face.

4.3.3 Face Authentication

Once the face is input for identification, projection for the Eigen face is taken to calculate its value. Taking a projection means that an inner product is taken and face similarity is distinguished by taking a weighted sum of the basis vector component of the face. Each Eigen face component value for the input face image is calculated as shown in equation (75).

$$w_k = \mu_k^T (\Gamma - \Psi) \qquad k = 1, 2, \cdots, M \tag{75}$$

μ: Basis vector of face

Γ: Input face vector

Ψ: Average face vector

Figure 41 shows the face reconstructed using the input face and the basis vector

calculated from the input face.

(a) Input Face Image (b) Reconstruction Face Image

Figure 41. Input face and Reconstruction face using basis vector.

Using the weight calculated in equation (75), the Eigen face component vector ($\Omega = [w_1, w_2, w_3, \cdots, w_M]$) expressing the input face can be calculated. Once Ω is calculated,

$$\epsilon_k = |\Omega - \Omega_k|^2 \qquad k = 1, 2, \cdots, M \tag{76}$$

As shown in equation (76), since the face with the shortest Euclidean distance between the face to be identified and the weights of candidate face images becomes the face closest to the input face, this image candidate is determined to be the identification result.

4.4 USN Channel Establishment Algorithm

The proposed algorithm established communication channel between cluster and sensors using wavelet filter bank and prevented the collision of channels used BIBD code. In this paper, there is two type of structure. One is that a base station can

collect data from all of the sensors and the other structure is extended cluster head model. Each of these two architectures can use a wavelet transformation on each sensor signal, treated as a one dimensional time series. Different wavelet transformation methods have been developed, more or less complex. In this paper, we have chosen one of the simplest methods Haar wavelet transform.

4.4.1 One Cluster Head Collecting All Sensor Data

The proposed structure, base station had filter bank (channel number), BIBD code and the allocated code values of each sensor. If selection signal transmits through filter bank of the sensor which wants to do communication, the sensor which was allocated filter bank channel transmits BIBD code to the base station. This structure can select sensor efficiently because of using wavelet filter bank and base station can check external collusion attack or collision between sensors by having eigen-BIBD code. Figure 42 shows that total block diagram of the proposed algorithm.

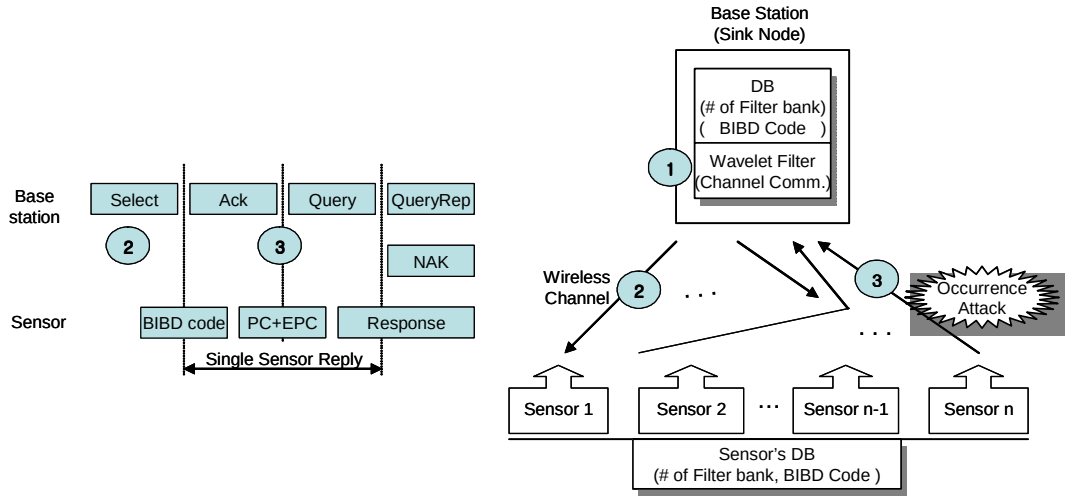


Figure 42. Cluster head collecting all sensor data.

In the figure 42, '①' is a part of filter bank to selects channel of each sensor by

using wavelet. Figure 43 shows that channel selection module using wavelet.

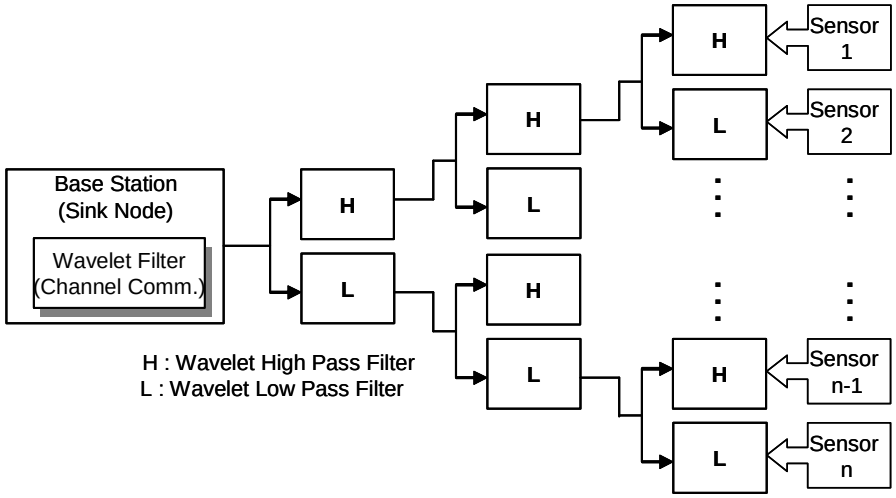


Figure 43. Channel selection module using wavelet.

Sensor transmits to base station using channel that is selected through figure 43 and base station can establish channel without collision or collusion attack by detecting BIBD code.

Wavelet filter that was implemented by VHDL is inserted in the audio sensor. Figure 44 shows the synthesized wavelet filter on the hardware level and figure 45 shows the RTL level simulation result.

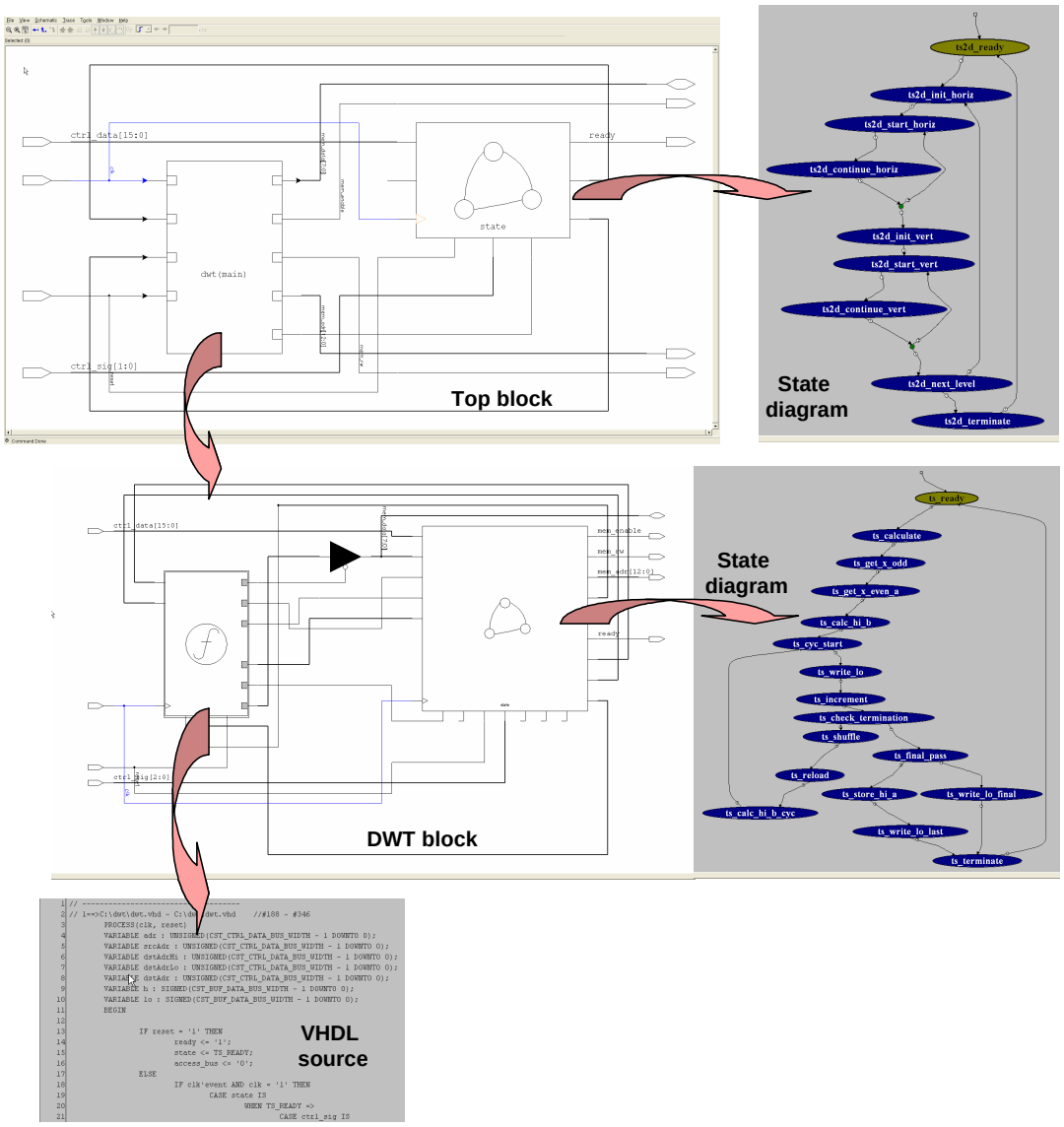


Figure 44. Synthesized wavelet filter on the hardware tool.

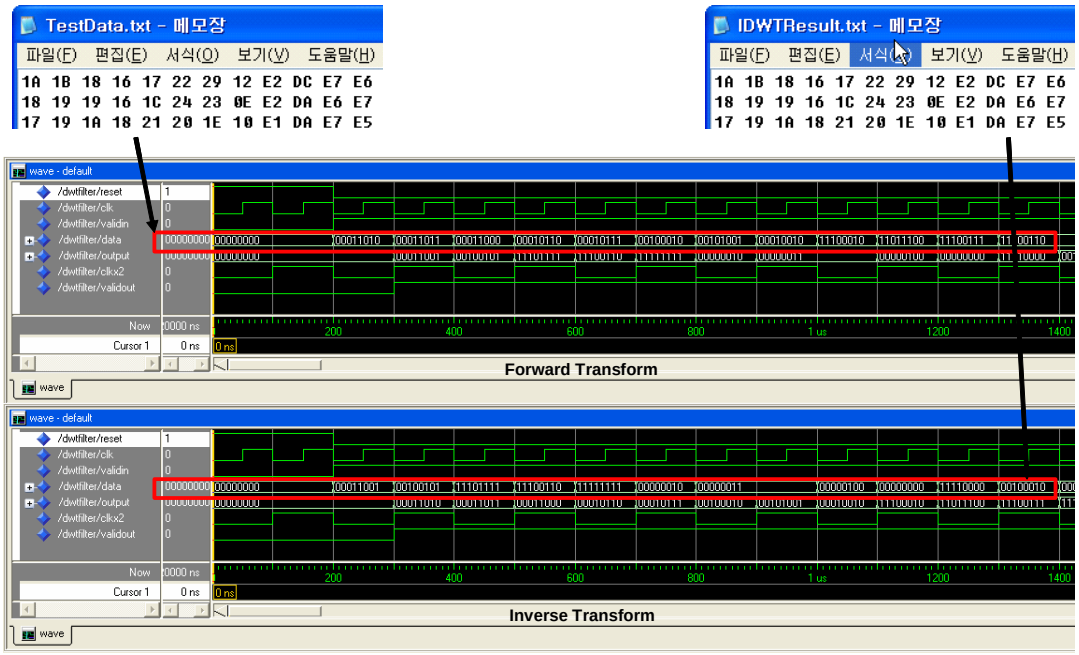


Figure 45. RTL level simulation waveform.

We verified that the designed wavelet filter operated correctly.

4.4.2 Extended Cluster Head Model

We proposed the extended cluster head model that was extended from one cluster head model. This model is possible to communicate between sink node and base station. Figure 46 shows that the extended cluster head model.

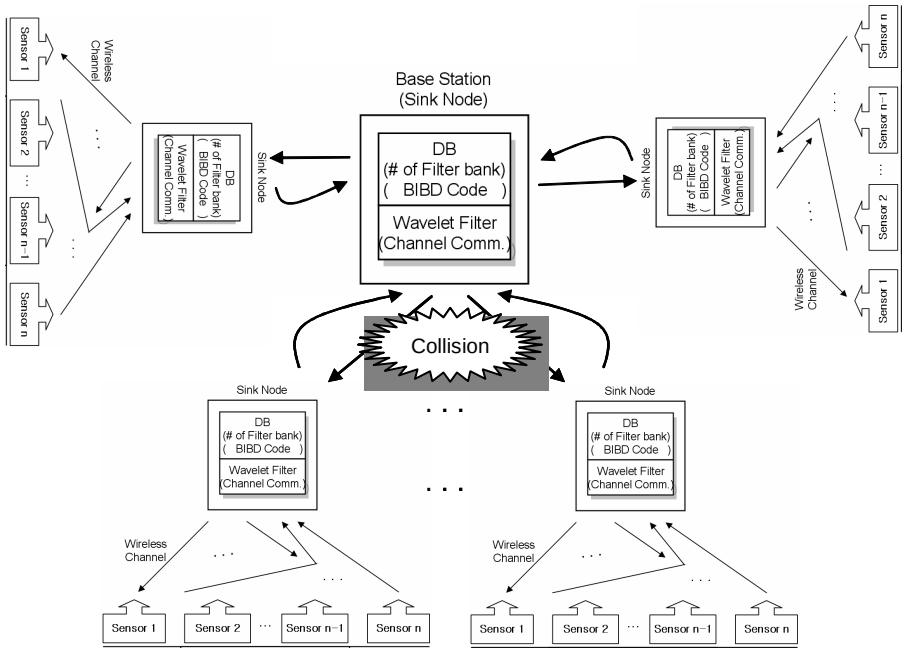


Figure 46. Extended cluster head model.

The proposed model had structure that allocate each channel using filter bank to transmit data between base station and sink node and in the figure 47, this algorithm shows that can detect data collision and collusion attack.

m_{ij}	b_1	b_2	b_3	b_4	b_5	b_6	b_7	
v_1	0	1	0	1	0	1	0	
v_2	1	0	0	1	1	0	0	
v_3	0	0	1	1	0	0	1	← a
v_4	1	1	1	0	0	0	0	
v_5	0	1	0	0	1	0	1	← b
v_6	1	0	0	0	0	1	1	← c
v_7	0	0	1	0	1	1	0	

			a		b	c	
Sink Node 1 :	0	1	0	1	0	1	0
Sink Node 6 :	1	0	0	0	0	1	1
Averaged :	0.5	0.5	0	0.5	0	1	0.5
AND Attack :	0	0	0	0	0	1	0
OR Attack :	1	1	0	1	0	1	1

Collision Code : b_1, b_6

Figure 47. Anti-collision BIBD code.

Chapter 5. Experiment Result

In this chapter, the performances of the proposed multi-modal biometrics system (speech recognition, palmprint recognition, face recognition and USN channel establishment algorithm) were measured respectively.

5.1 Speech Recognition

Figure 48 shows the block diagram for the wireless audio signal transmission sensor designed in this paper. It is made up of a USS-2400 [11] data transmission module, an ADC, an audio sensor and a LDPC generation algorithm.

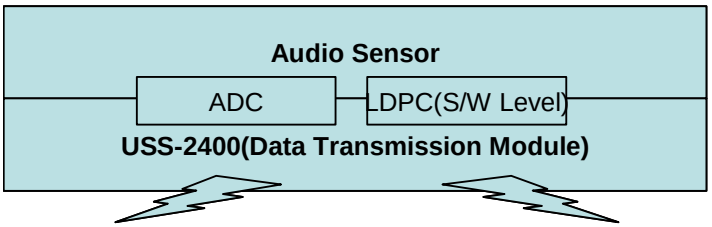


Figure 48. Audio sensor block diagram.

LDPC block was generated using matrix values generated from equation (35) and digital signal that passed through the ADC. The decoder was realized within the JAVA program. Figure 49 shows the PCB board designed in this paper.

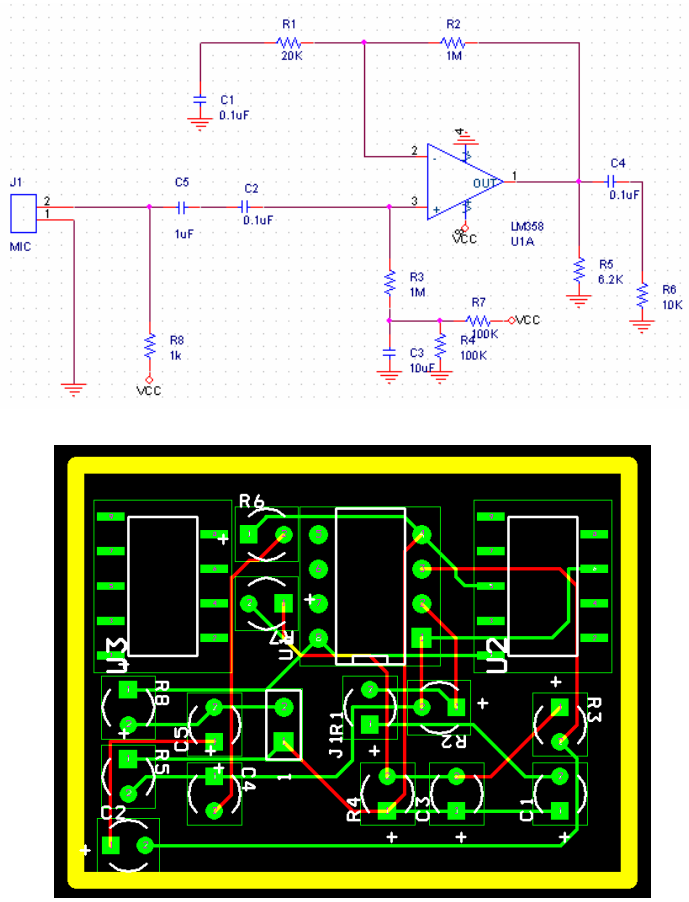


Figure 49. Designed PCB for the audio sensor.

Speech recognition was realized by using Fourier transform and the psychoacoustic model. When the subject for speech recognition utters his name, it is sent to the HOST PC through the sensors. After processing the transmitted signal using FFT, the Psychoacoustic model is applied to extract the 24 inherent speech characteristic values that each person possesses and compared with values stored in the DB to determine whether a match has occurred. Psychoacoustic model used is shown in equation (21). Table 4 shows the frequency range of channels used to extract the inherent characteristic values.

Table 4. Frequency range of channels

Channel No.	*Freq.[Hz]*	*Channel No.*	*Freq. [Hz]*	*Channel No.*	*Freq. [Hz]*
1	100	9	1,270	17	4,400
2	200	10	1,480	18	5,300
3	300	11	1,720	19	6,400
4	400	12	2,000	20	7,700
5	510	13	2,320	21	9,500
6	630	14	2,700	22	12,000
7	770	15	3,150	23	15,500
8	920	16	3,700	24	End

The collected speech signal can be verified using an oscilloscope realized in JAVA. In this paper, speech recognition experiments were carried out by arranging 8 audio sensors.

Data collected by the ISP module is transmitted to the HOST PC. After passing through the LDPC decoder block, realized in JAVA, the data coming in through 8 channels can be verified with an oscilloscope. Finally, the speech recognition system operates at the JAVA program level. When speech recognition results in a match, an ACK signal is sent through the ISP module and the LED in the Audio Sensor is activated.

Sensor operates in sleep mode when surrounding sound is below a certain level. It will only collect sound signal that is above a certain level. Figure 50 shows the audio signal that is displayed on the oscilloscope after LDPC decoding and figure 51 shows the speech recognition process.

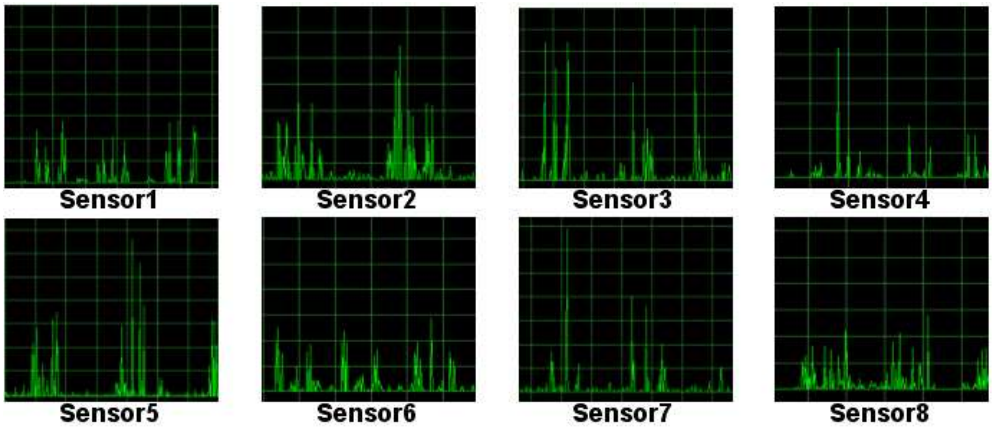


Figure 50. Collected speech signal from sensors.

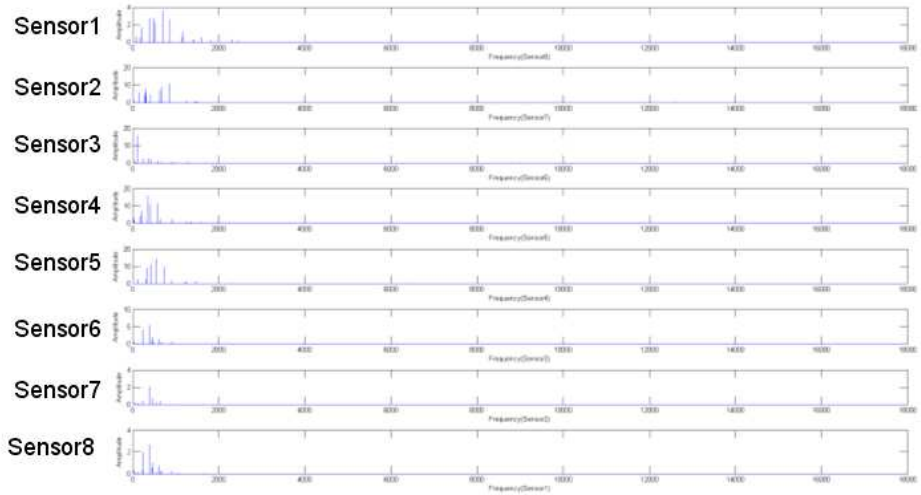


Figure 51. Extracted value from the speech signal.

Similarity between 24 characteristic values calculated from the speech signal collected by each sensor and characteristic values stored in the database are compared to determine whether a match has occurred. For comparison of similarity, coefficient of correlation in equation (77) was used.

$$k = \frac{1/n \sum_{m-1}^{n} (a_m - \bar{a})(b_m - \bar{b})}{\sigma_a \sigma_b} \tag{77}$$

k: coefficient of correlation($-1 \le k \le 1$)

$\bar{a}, \bar{b}$: average

σ_a, σ_b: standard deviation

Figure 52 shows the coefficient of correlation that was generated as a result of a matching simulation between the subject and another person for speech input through the sensors. From this result, speech recognition critical value was set to 0.45 and experiment was carried out.

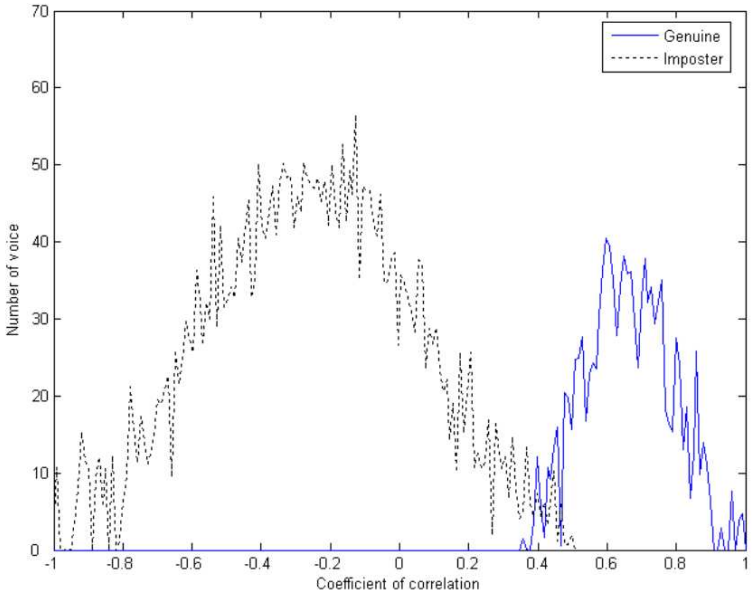


Figure 52. The coefficient of correlation of speech.

For performance evaluation, False Acceptance Ratio (FAR) and False Rejection Ratio (FRR) were used in this paper. FAR is the ratio of incorrectly identifying a speech as

being the subject's speech when the input speech is compared with the database data. FRR is the ratio of incorrectly rejecting a speech as not being the subject's speech. FAR and FRR used in this paper are shown in equation (78).

$$FAR = \frac{r_a}{r_b} \qquad FRR = \frac{\delta_a}{\delta_b} \tag{78}$$

r_a: Frequency of false acceptance

r_b: Matching frequency of user to unknown user

δ_a: Frequency of false rejection

δ_b: Matching frequency of user to user

Table 5 shows FAR and FRR for the system for changing critical values.

Table 5. FAR and FRR performances according to critical values.

Critical Value	γ_a	*FAR[%]*	δ_a	*FRR[%]*
0.37	11	0.048	5	12.5
0.42	27	0.117	8	20.0
0.45	32	0.139	11	27.5
0.48	62	0.270	14	35.0

FAR and FRR values in table 5 indicate unsatisfactory overall performance. This is because a speech recognition result was processed as a mistake when there were more that two data points that were below the critical value. In order to improve performance, the system was designed so that speech recognition was carried out again when more than two values were within the critical value. Table 6 shows FAR and FRR values for re-identification result data.

Table 6. FAR and FRR performances after re-identification.

Critical Value	γ_a	*FAR[%]*	δ_a	*FRR[%]*
0.37	10	0.035	2	5.0
0.42	25	0.109	2	5.0
0.45	29	0.126	3	7.5
0.48	49	0.213	6	15.0

As shown in table 6, we obtained FAR and FRR of 0.126 [%] and 7.5 [%] respectively from data that went through a re-identification process when critical value was 0.45.

In addition, a difference in speech recognition rate proportional to distance between the sensor and the speech source was verified. Experiment results show that there is little difference in speech recognition rate within 1.5m. When distance is increased beyond 2m, however, the recognition rate decreases rapidly.

5.2 Palmprint Recognition

5.2.1 Palmprint Database

The performance of proposed algorithm was measured by collecting total 378 units of palmprints from 189 people by using the palmprint acquisition device designed in this paper. The palmprints were provided by 168 males and 21 females between the ages of 21 to 28. The acquired palmprints are in the size of 135×135 pixels and any possible changes that might occur rotationally and positionally were minimized by using the palmprint fixed equipment.

In order to establish the palmprint database, one palmprint was measured 3 times and 5 units of the HU invariance moment were collected from each palmprint. And then the average of 3 moments was calculated to complete the database. Figure 53 shows

the basic formation of database constituted of the palmprint name and 5 units of HU invariance moment.

```
DB = cell(378,2);                                   NAME
DB{1,1} = 'Binod_Right';
DB{1,2} = [0.4398 0.2018 0.0022 0.0062 0.0003];

DB{2,1} = 'Binod_Left';
DB{2,2} = [0.4138 0.1926 0.0021 0.0051 0.0007];
.                                                   HU Moment
.
.
DB{377,1} = 'Minhuk_Right';
DB{377,2} = [0.3897 0.1532 0.0048 0.0060 0.0003];

DB{378,1} = 'Minhuk_Left';
DB{378,2} = [0.3698 0.1467 0.0121 0.0293 0.0041];

SAVE DB DB;
```

Figure 53. Database structure.

Table 7 shows maximum, minimum, average and pixel brightness of Otsu binarization value of palmprint input to the database.

Table 7. Pixel brightness & Otsu binarization value of database.

	Max. Value	*Min. Value*	*Average*	
			Max.	*Min.*
Brightness of Pixel	181	106	172	122
Otsu Binarization Value	131	119	126	

5.2.2 Palmprint Authentication

The matching sequence of the HU invariance moments stored in palmprint database and those moments of input palmprints sent from the palmprint acquisition device was

selected as the palmprint authentication technique. The input palmprint was authenticated as the valid palmprint if the Euclidean distance between the stored and input was below the certain value.

The general method to evaluate the performance of the palmprint authentication system is based on the False Acceptance Ratio and the False Rejection Ratio. Where, the FAR is the ratio to falsely accept the input palmprint from an unknown user as the print of a user in the system. The FRR is the ratio to falsely reject the user in the system. In this paper, FAR and FRR was based on equation (79) and GAR (Genuine Acceptance Rate) was calculated to obtain the data comparative to [8].

$$FAR = \frac{r_a}{r_b} \qquad FRR = \frac{\delta_a}{\delta_b} \qquad GAR = \frac{\delta_b - \delta_a}{\delta_b} = 1 - FRR \tag{79}$$

r_a: Frequency of false acceptance

r_b: Matching frequency of user to unknown user

δ_a: Frequency of false rejection

δ_b: Matching frequency of user to user

The performance measurement of the palmprint authentication algorithm based on the 378 prints which was stored in the database scored 0% each for FAR and FRR since the HU invariance moments had matched perfectly. In this paper, the palmprints authentication was executed with the palmprints stored in the database and the palmprint data collected from the palmprint acquisition device. Therefore, those palmprints with any changes occurring in their positions of acquired palmprint and the brightness of palmprint image were input to the data. In this investigation, the overall matching frequency of palmprint was 142,884 (378×378) and the user to unknown user matching frequency was 142,506 and the user to user frequency was 378.

The performance of the suggested algorithm was compared to the authentication

system [8] after selecting the most advanced system among the authentication systems based on the hand shape, the finger prints and the palmprint. Table 8 shows the values of FAR and GAR and the marked section would be optimally suggested FAR and GAR in [8].

Table 8. FAR and GAR of [8]

FAR[%]	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08
GAR[%]	97.0	97.9	97.8	98.0	98.2	98.3	98.4	98.5

Table 9 shows FAR and GAR relative to the changes in coeff. of the palmprint search algorithm based on equation (79).

Table 9. FAR and GAR performance measured using 1st run

Critical Value(Coeff.)	γ_a	*FAR[%]*	δ_a	*GAR[%]*
0.001	104	0.072	70	81.5
0.002	225	0.157	48	87.3
0.003	276	0.193	31	91.8
0.004	290	0.203	24	93.7
0.005	306	0.214	12	96.8

The overall FAR and GAR readings from table 9 noted that they were worse than the reading from table 8 of the algorithm suggested from [8]. This was due to the fact that when there were more than 2 units of data entered within the range of the critical value, the palmprint authentication was regarded as the errorneous frequency. In order to resolve this problem, the re-authentication structure was designed to re-acquire the palmprint once more when there were more than two values detected within the range of critical values and this re-authentication was limited to 3 times. In case when the re-authentication required more than 3 times, it was processed as the authentication

failure and designed to follow through the authentication procedures from the start. Table 10 shows the data obtained from the second run authentication.

Table 10. FAR and GAR performance measured using 2nd run

Critical Value(Coeff.)	γ_a	*FAR[%]*	δ_a	*GAR[%]*
0.001	61	0.042	32	91.5
0.002	78	0.054	21	94.5
0.003	86	0.060	19	95.0
0.004	135	0.094	14	96.3
0.005	194	0.136	9	97.6

When the critical value of the searching algorithm was selected at 0.001 in table 10, there was 0.042% and 91.5% for FAR and GAR, respectively. Lastly, the data collected after following through 3rd run authentication process scored 0.038% and 98.1% for FAR and GAR, respectively. This result suggested that FAR and GAR had improved respectively by 0.002% and 0.1% than [8].

Table 11. FAR and GAR performance measured using 3rd run

Critical Value(Coeff.)	γ_a	*FAR[%]*	δ_a	*GAR[%]*
0.001	55	0.038	7	98.1
0.002	71	0.049	7	98.1
0.003	79	0.055	6	98.4
0.004	125	0.087	4	98.9
0.005	177	0.124	4	98.9

Figure 54 is the graphical representation of FAR and GAR relative to the changes in the re-authentication frequency and the critical value. FAR and GAR tend to increase as the number of re-authentication was increasing.

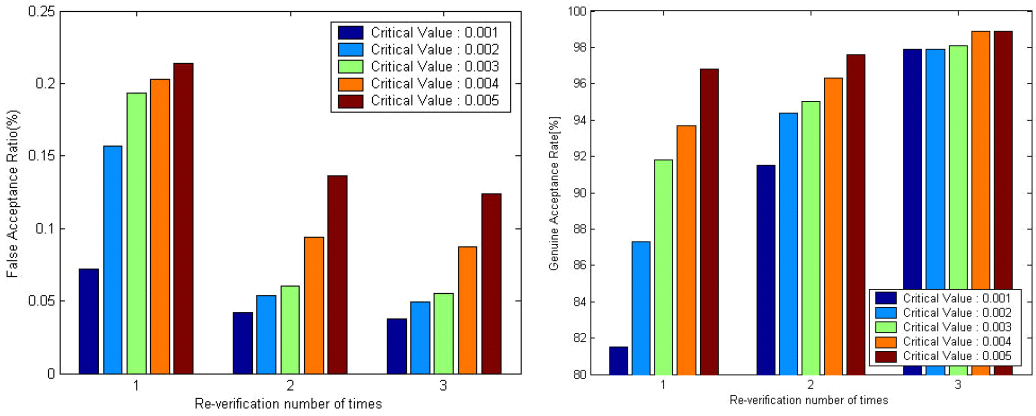


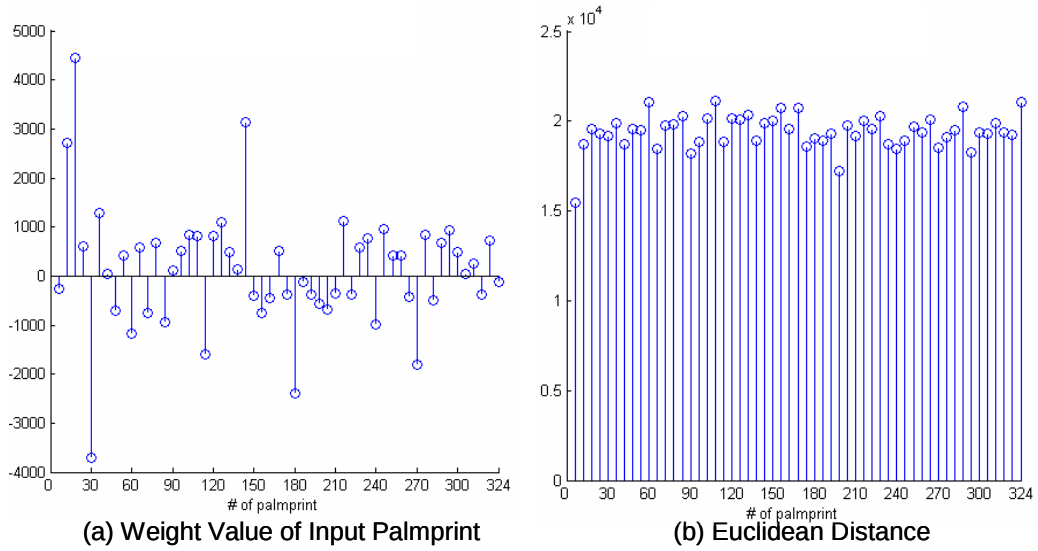
Figure 54. Comparative graph of FAR and GAR.

5.3 Face Recognition

Performance of the proposed algorithm was measured using a total of 365 faces that were consist of foreigners who were obtained 300 persons from [53] and korean were obtained 65 persons using the face acquisition system designed in this paper. The subjects were 323 men and 42 women between the ages of 20 and 55. The acquired faces had pixel size of 92×115.

In order to prepare the face database, one face was measured 3 times. From each face, the basis vector composing the Eigen face was calculated to come up with the database composed of average values.

As shown in figure 55, face identification is done as follows. The basis vector of Eigen face in the face database and the basis vector of the input face from the face acquisition device were matched. The face with the Euclidean distance within a certain range of the input face is recognized as being the same as the input face.



(a) Weight Value of Input Palmprint (b) Euclidean Distance

Figure 55. The procedure of the face identification.

The typical criteria used to evaluate the performance of a face identification system are the false acceptance ratio and false rejection ratio. Here, FAR is the ratio of incorrectly identifying a face as being the subject's face when the input face is compared with the database face. FRR is the ratio of incorrectly rejecting a face as not being the subject's face. In order to obtain the comparative data for [54, 55], the genuine acceptance rate was calculated.

face identification was carried out using the faces in the database and that obtained from the face acquisition device, in real time. Therefore, data with changes in the acquired face location and contrast in the face image is input and the face acquisition device was designed to minimize this. The total number of face matching in this study was 133,225 (365×365). The number of face matching of the subject and someone else was 132,860. The number of matching the subject and subject was 365.

Figure 56 shows the Euclidean distance produced as a result of carrying out subjects vs. other matching simulation of 365 faces. Using this result, simulation was done by setting the critical value of face identification from 1.62×10^4 to 1.65×10^4.

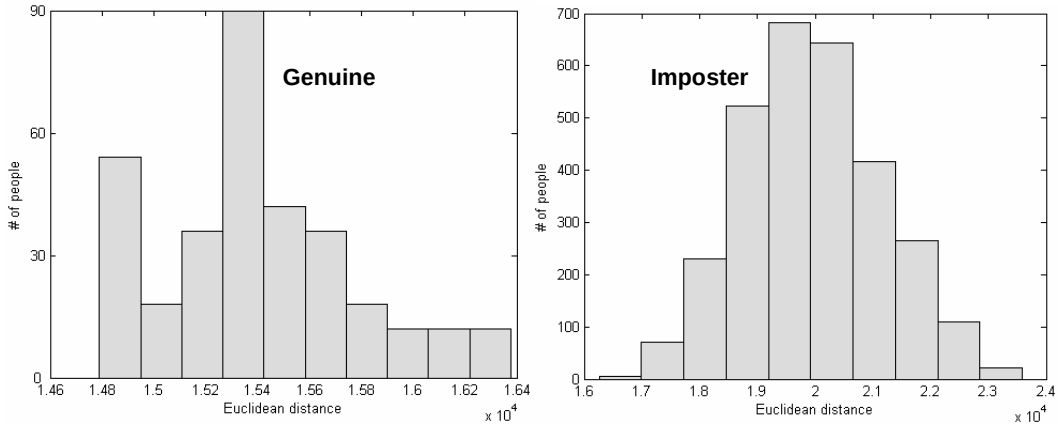


Figure 56. The Euclidean distance of face.

Table 12 shows FAR and GAR of the proposed algorithm with changing critical values.

Table 12. The FAR and GAR performances of the first identification.

Critical value	γ_a	$FAR(\gamma_a/\gamma_b)[\%]$	δ_a	$GAR(1-\delta_a/\delta_b)[\%]$
1.62×104	57	0.043	14	96.0
1.63×104	69	0.052	13	96.6
1.64×104	78	0.059	10	97.2
1.65×104	137	0.103	7	98.1

Although FAR and GAR performance was high overall in table 12, FAR was 10% lower compared with [54, 55]. This was because face identification result was processed as being incorrect when there were more than two data points that had values that were lower than the critical value range. Thus, in order to correct this problem, the system was designed so that the face identification is carried out one more time with a newly acquired face when more than two values fall into the critical value range. Table 13 is the processed data after the second identification.

Table 13. The FAR and GAR performances of the second identification.

Critical Value	γ_a	$FAR(\gamma_a/\gamma_b)[\%]$	δ_a	$GAR(1-\delta_a/\delta_b)[\%]$
1.62×104	41	0.031	7	98.1
1.63×104	43	0.033	7	98.1
1.64×104	47	0.036	5	98.5
1.65×104	69	0.052	4	98.8

As shown in table 13, when the critical value was 1.64×10^4 in the processed data after the second identification, FAR and GAR were 0.036[%] and 98.5[%], respectively.

Table 14. FAR and GAR of proposed algorithm and [12, 13].

[12]	FAR[%]	0.040	GAR[%]	98.0
[13]	FAR[%]	0.038	GAR[%]	98.1
Proposed Algorithm	FAR[%]	0.036	GAR[%]	98.5

Table 14 shows the optimized values in this study and those proposed in [54, 55]. Improvement of 5.3% for FAR and 0.4% for GAR can be seen.

5.4 USN Channel Establishment Algorithm

The platform for the experiments, from which the data analyzed in this paper were obtained, is a collection of 'USN-AP-Zigbee'. One this unit embodies a sensor module (audio), and a communication module, which are interconnected. Figure 57 shows the hardware architecture which used in simulation.

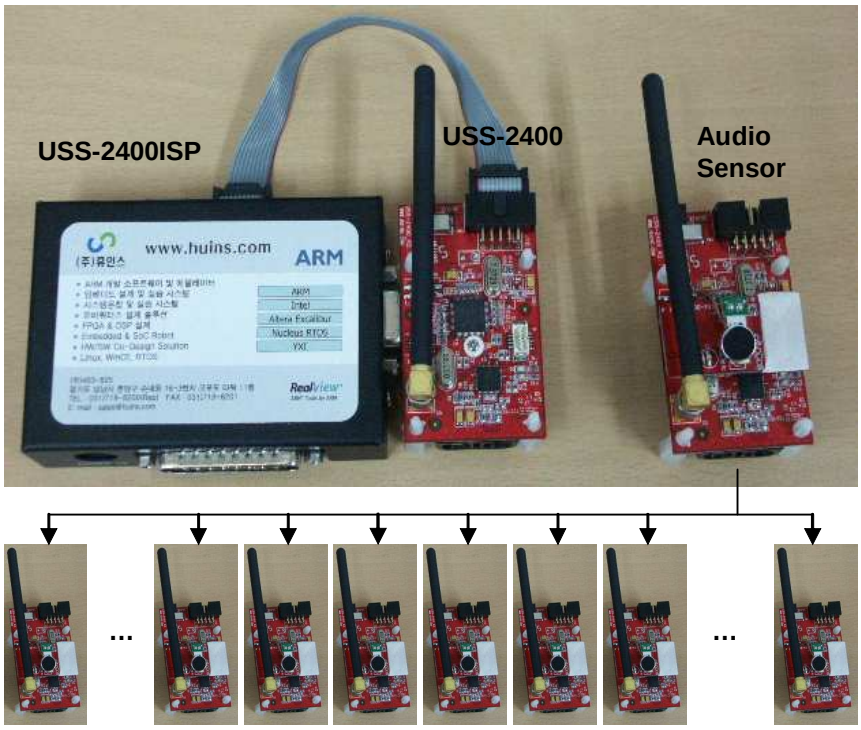


Figure 57. Hardware architecture used in simulation.

In simulation, the condition of BIBD code parameter { v, k, λ} is generated with {7,3,1}, {11,5,2}, {15,7,3}, {19,9,4}, {23,11,5}, {31,15,7}, {39,20,9} and {47,24,11}. As collusion/collision attack, a collusion/collision number of possible combination is presented in table 15. The collusion/collision detection of experiment, measured the robustness of anti-collision and channel establishment.

Table 15. The number of collusion cases by the number of colluder.

Number of colluders	*Number of collusion cases*			
	{7,3,1} code	*{15,7,3} code*	*{22,11,5} code*	*{31,15,7} code*
2	21	105	231	465
3	35	455	1540	4495
4	35	1365	7315	31465
5	21	3003	26334	169911
6	7	5005	74613	736281

The correlation coefficient of collision code and codebook in the base station is computed, that result is upper critical value then it defines collision. Correlation coefficient is computed by equation (80).

$$r = \frac{1/n\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y}, \quad -1 \le r \le 1 \tag{80}$$

$\bar{x}, \bar{y}$: Average

σ_x, σ_y: Standard deviation

Table 16 shows the result of collision code detection against collision averaging attack by means of the algorithm proposed in this paper. As a result, collision was 100% detected only for the collision averaging attack.

Table 16. The number of the detected collision by attack on communication channel.

	Number of Collisions							
BIBD code	6	10	14	18	22	30	38	46
{7,3,2}	6	–	–	–	–	–	–	–
{11,5,2}	6	10	–	–	–	–	–	–
{15,7,3}	6	10	14	–	–	–	–	–
{19,9,4}	6	10	14	18	–	–	–	–
{23,11,5}	6	10	14	18	22	–	–	–
{31,15,7}	6	10	14	18	22	30	–	–
{39,20,9}	6	10	14	18	22	30	38	–
{47,24,11}	6	10	14	18	22	30	38	46

Figure 58 shows the correlation coefficient of collusion and anti-collusion code and table 17 shows the number of the detected colluders by changing AWGN.

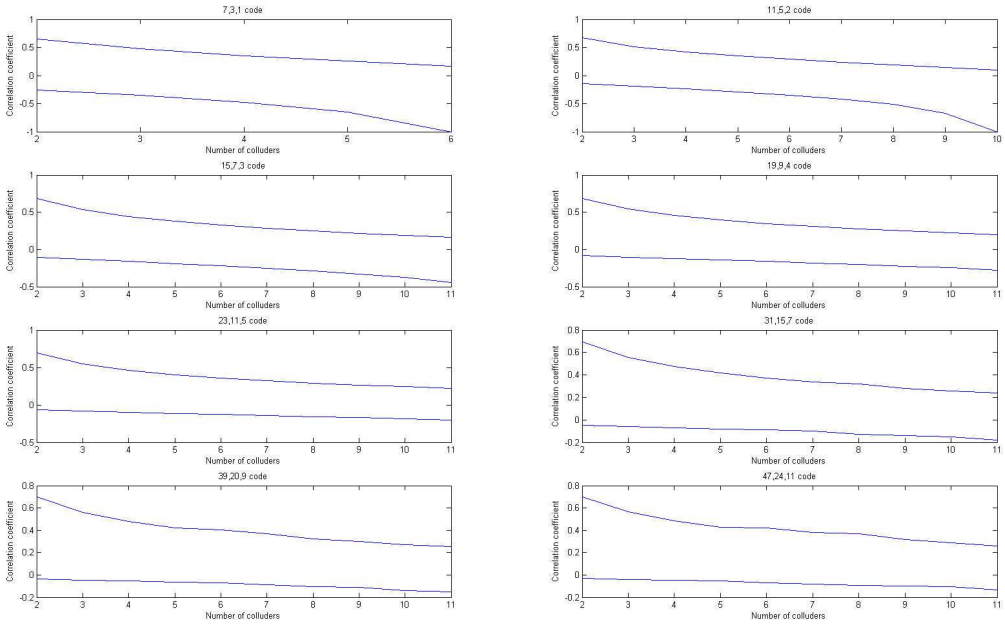


Figure 58. Correlation coefficient of collusion and anti-collusion code.

Table 17. The number of the detected colluders by changing AWGN.

BIBD code	AWGN[dB]						
	4	2	0	-1	-2	-3	-4
{7,3,2}	0	0	0	0.2	3.1	4.7	13.7
{11,5,2}	0	0	0	0.7	3.2	6.8	14.4
{15,7,3}	0	0	0	0.9	3.2	9.3	18.7
{19,9,4}	0	0	0	1.3	4.3	11.2	20.2
{23,11,5}	0	0	0	1.4	4.6	11.6	21.3
{31,15,7}	0	0	0	1.4	4.9	12.5	23.9
{39,20,9}	0	0	0	1.8	5.4	13.5	25.2
{47,24,11}	0	0	0	2.1	7.1	15.9	25.9

In this paper, hopfield network is applied to the cluster head to enhance detection ratio in AWGN environment. Figure 59 shows the hopfield network and table 18 shows the final result.

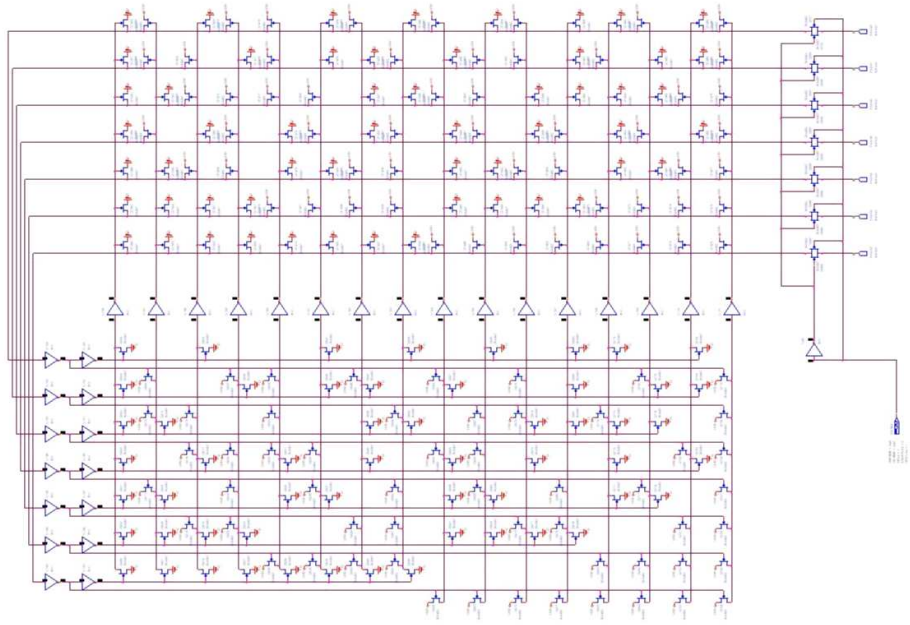


Figure 59. Error correction circuit using Hopfield network.

Table 18. Number of the detected colluders in fingerprint code that passed Hopfield network.

BIBD코드	AWGN[dB]			
	-1	-2	-3	-4
{7,3,2}	0.00	1.55	4.28	13.19
{11,5,2}	0.02	1.60	6.52	14.12
{15,7,3}	0.05	1.62	8.69	18.41
{19,9,4}	0.11	2.16	10.96	19.17
{23,11,5}	0.13	2.37	11.10	20.99
{31,15,7}	0.17	2.49	11.97	22.98
{39,20,9}	0.20	3.58	12.91	24.06
{47,24,11}	0.24	5.80	15.15	25.44

Chapter 6. Conclusion

The necessity of human interface technologies is increasing in ubiquitous environments such as speech, face and palmprint recognition, etc. Accordingly, a system that can recognize persons who enter or leave a building by using wireless audio sensors and wireless cameras, was developed in this paper. In addition, this paper presents a channel establishment algorithm in ubiquitous sensor networks, which is for sensor's authentication and anti-collision in the wireless field.

The proposed system consists of the wireless audio sensor designed to identify speech, the wireless image capture camera (TX-32CS) to identify palmprint/face and the LDPC encoder/decoder designed to reduce channel noise. Software is composed of the psychoacoustic model, PCA (Principal Components Analysis), HU invariant moment, Wavelet filter, FFT (Fast Fourier Transform) and the data matching algorithm.

Performances of the speech recognition system are as follows: FAR 0.126%, FRR 7.5% within a distance of 1.5*m*. This is equivalent to a recognition rate of 92.5%. However, the performance decreases rapidly according to increasing distance. And the performances of palmprint recognition algorithm were calculated after acquiring 378 units of palmprint data from 198 students. As a result, FAR and GAR are 0.038% and 98.1% respectively while maintaining the critical value at 0.001. The face recognition algorithm consists of the face acquisition system and the face identification system. The proposed system uses the face acquisition algorithm to improve accuracy of the acquired face image. By using standard deviation, face images were normalized to confer robustness against the changes, in contrast, caused by the external noise. The proposed system extracts the basis vector of eigen face from the input image and seeks the euclidean distance by integrating the database and data weight for face identification. As a result, when the critical value was 1.64 in the processed data , FAR and GAR were 0.36% and 98.5% respectively.

In this paper, two different USN channel establishment architectures were proposed. One of the proposed architectures, one cluster head, that collect the clustering outputs from the other units provides a big dimensionality reduction and saving additional communication ability in the same time, since only classification IDs (small binaries) are passed to the cluster head instead of all input samples. As a result, the proposed algorithm based on BIBD code has 100% detection of collision of sensor. Also, filter bank using wavelet transform can be incorporated as a preprocessing unit of the USN giving the ability to extract the important features in the data like abrupt changes at the various scales.

The proposed system is dependent on the base station for most processing. It increases the overhead of base station and as the distance between the sensor and the sound source increases, the performance will also fall rapidly. And data matching time will increase with size of the database. Further research needs to be carried out to solve such problems. And channel establishment algorithm using other ACC (Anti Collusion Code) will be studied.

Reference

[1] Paramvir Bahl and Venkata Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," Proc. of IEEE INFOCOM, vol. 2, p.p. 775-784. March 2000.

[2] Nissanka B., Priyantha, Anit Chakraborty, Hari Balakrishnan, "The cricket location-support system," Proc. of MOBICOM 2000, p.p.32-43, Boston, MA, Aug. 2000, ACM, ACM Press.

[3] Ian D. Chakeres and Luke Klein-Berndt, "AODVjr, AODV Simplified," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6, no. 3, Jul. 2002, p.p.100-101.

[4] R. Chellappa, G. Qian, and Q. Zheng, "Vehicle detection and tracking using acoustic and video sensors," in Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, Utah, May 2001.

[5] E. H. Callaway, "Wireless Sensor Networks Architectures and Protocols," Auebach, 2003.

[6] Wen Hu et al. "The Design and Evaluation of a Hybrid Sensor Network For Cane-toad Monitoring,"

[7] H. Wang, D. Estrin, and L. Girod, "Preprocessing in a tiered sensor network for habitat monitoring," EURASIP JASP special issue of sensor networks, pp. 392-401, 2003.

[8] D. Zhang, W. K. Kong, J. You and M. Wong, “Online Palmprint Identification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25. NO.9, pp.1041-1050, 2003.

[9] J. S. Noh, K. H. Rhee, "Eigen Palmprint Identification Algorithm using PCA(Principal Components Analysis)," Journal of The Institute of Electronics Engineers of Korea, Vol. 43-CI, NO. 3, pp.82-89, May 2006.

[10] Phenix Information System, http://w-cctv.co.kr.

[11] Huins Intelligent System, http://www.huins.com.

[12] J. S. Noh, K. H. Rhee, "Eigen Palmprint Authentication System using Dimension Reduction of Singular Vector," FSKD, LNAI 4223, pp.1167-1175, 2006.

[13] J. S. Noh, K. H. Rhee, "Palmprint Identification Algorithm Using Hu Invariant Moments," FSKD, LNAI 3614, pp.91-94, 2005.

[14] J. S. Noh, K. H. Rhee, "High Quality Audio Watermarking using Spread Spectrum and Psychoacoustic Model," Journal of The Institute of Electronics Engineers of Korea, , Sept. 2006.

[15] J. S. Noh, K. H. Rhee, "Detection of Colluded Multimedia Fingerprint by Neural Network," Journal of The Institute of Electronics Engineers of Korea, Vol. 43-CI, NO. 4, pp.80-87, July 2006.

[16] J. S. Noh, K. G. Shin and K. H. Rhee, "Watermarking of Gray Logo & Color Image based on Human Visual System," Journal of The Institute of Electronics Engineers of Korea, Vol. 42-CI, NO. 3, pp.73-82, May 2005.

[17] Bernard S., Boujemaa N., Vitale D., and Bricot C., “Fingerprint Segmentation Using the Phase of Multiscale Gabor Wavelets," in Proc. Asian Conf. Computer Vision, 2002.

[18] Davide Maltoni, Dario Maio, Anil K. Jain and Salil Prabhakar, Handbook of Fingerprint Recognition, Springer, 2003.

[19] Willis A.J. and Myers L., “A Cost-Effective Fingerprint Recognition System for Use with Low-Quality Prints and Damaged Fingertips," Pattern Recognition, vol. 34, no. 2, pp. 255-270, 2001.

[20] Almansa A. and Lindeberg T., “Fingerprint Enhancement by Shape Adaptation of Scale-Space Operators with Automatic Scale Selection," IEEE Transactions on Image Processing, vol. 9, 12, pp. 2027-2024, 2000.

[21] Brunelli R. and Falavigna D., “Personal Identification Using Multiple Cues," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 10, pp. 955-966, 1995.

[22] Bazen A.M. and Gerez S.H., “Systematic Methods for the Computation of the Directional Fields and Singular Points of Fingerprints," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 905-919, 2002.

[23] Bolle R.M., Serior A.W., Ratha N.K., and Pankanti S., “Fingerprint Minutiae: A Constructive Definition," in Proc. Workshop on Biometric Authentication (in ECCV 2002), LNCS 2359, pp. 58-66, Springer Verlag, New York, 2002.

[24] Kovacs-Vajna Z.M., “A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 1266-1276, 2000.

[25] Wayman J.J., “Technical Testing and Evaluation of Biometric Identification Devices," in Biometrics: Personal Identification in a Networked Society, A.K. Jain,

R. Bolle, and S. Pankanti (Eds.), pp. 345-368, Kluwer, New York, 1999.

[26] Chang J.H. and Fan K.C., “Fingerprint Ridge Allocation in Direct Gray-Scale Domain," Pattern Recognition, vol. 34, no. 10, pp 1907-1925, 2001.

[27] Dorai C., Ratha N.K., and Bolle R.M., “Detecting Dynamic Behavior in Compressed Fingerprint Videos: Distortion," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 320-326, 2000.

[28] Greenberg S., Aladjem M., Kogan D., and Dimitrov I., “Fingerprint Image Enhancement Using Filtering Techniques," in Proc. Int. Conf. on Pattern Recognition (15th), vol. 3, pp. 326-329, 2000.

[29] Hong L. and Jain A.K., “Integrating Faces and Fingerprints for Personal Identification," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 20, no 12, pp. 1295-1307, 1998.

[30] Dickinson A., McPherson R., Mendis S., and Ross P.C., “Capacitive Fingerprint Sensor with Adjustable Gain," US patent 6049620, 2000.

[31] Jain A.K. and Pankanti S., “Biometrics Systems: Anatomy of Performance," IEICE Transactions on Information and Systems (Special Issue on Biometrics), vol. E84-D, no. 7, pp. 788-799, 2001.

[32] Frischholz R.W. and Dieckmann U., “BioId: A Multimodal Biometric Identification System," IEEE Computer, pp. 64-68, Feb. 2000.

[33] Yamazaki Y. and Komatsu N., “A Secure Communications System Using Biometric Identity Verification," IEICE Transactions on Information and Systems, vol. E84-D, no. 7, pp. 879-884, 2001.

[34] Phillips P.J., Martin A., Wilson C.L., and Przybocki M., “An Introduction to Evaluating Biometric Systmes," IEEE Computer Magazine, Feb. 2000.

[35] Maio D., Maltoni D., Cappelli R., Wayman J.L., and Jain A.K., “FVC2000: Fingerprint Verification Competition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp. 402-412, 2002.

[36] Maio D., Maltoni D., Cappelli R., Wayman J.L., and Jain A.K., “FVC2002: Second Fingerprint Verification Competition," in Proc. Int. Conf. on Pattern Recognition (16th), vol. 3, pp. 811-814, 2002.

[37] United Kingdom Biometric Working Group, “Best Practices in Testing and Reporting Biometric Device Performance," Tech. Report: Version 2.01, Aug. 2002, available at: http://www.cesg.gov.uk/technology/biometrics/.

[38] R. Jordan and C.A. Abdallah, "Wireless communications and networking: an

overview," Report, Elect. and Comp. Eng. Dept., Univ. New Mexico, 2002.

[39] IEEE 1451, A Standard Smart Transducer Interface, Sensors Expo, Philadelphia, Oct. 2001, http://ieee1451.nist.gov/Workshop_04Oct01/ 1451_overview.pdf.

[40] Conway and Heffernan, Univ. Limerick, 2003, http://wwww.ul.ie/~pei.

[41] R. Frank, Understanding Smart Sensors, 2nd Ed., Artech House, Norwood, MA, 2000.

[42] G.T.A. Kovacs, Micromachined Transducers Sourcebook, McGraw-Hill, Boston, 1998.

[43] E.S. Kolesar, C.P. Brothers, C.P. Howe, et al., "Integrated circuit microsensor for selectively detecting nitrogen dioxide and diisopropyl methylphosphate," Thin Solid Films, vol. 220, pp. 30-37. 1992.

[44] J. Choi, C. Conrad, C. Malakowsky, J. Talent, C.S. Yuan, and R.W. Gracy, "Flavones from Scutellaria baicalensis Georgi attenuate apoptosis and protein oxidation in neuronal cell lines," Biochemica et Biophysica Acta, 1571: 201-210 (2002).

[45] D.M. Rudkevich, J. Scheerder, and D.N. Reinhoudt, "Anion recognition by natural receptors," in Molecular Design and Bioorganic Catalysis, ed. C.S. Wilcox, pp. 137-162, Kluwer, Boston, 1996.

[46] R. Frank, Understanding Smart Sensors, 2nd Ed., Artech House, Norwood, MA, 2000.

[47] PCA, http://en.wikipedia.org/wiki/Principal_components_analysis#_note-0

[48] Sergios Theodoridis, and Konstantinos Koutroumbas, “Pattern Recognition," ACADEMIC PRESS, 1998.

[49] Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw Hill, 1965.

[50] Cho-Huak Teh and Roland T. Chin, “On Digital Approximation of Moment invariants", Computer Vision, Graphics, And Image Processing, Vol. 33, pp. 318-326, 1986.

[51] M. K. Hu, “Pattern recognition by moment invariants," Proc. IEEE, vol. 49, no. 9, pp. 1428, Sept. 1961.

[52] M. K. Hu, “Visual pattern recognition by moment invariants," IRE Transactions on Information Theory, vol. 17-8, no. 2, pp. 179-187, Feb. 1962.

[53] Ricardo Gutierrez-Osuna: http://research.cs.tamu.edu/prism/rgo.htm.

[54] H. A. Rowley, S. Baluja, and T. Kanade, "Human face detection in visual scenes,"

School of Computer Science, Carnegie Mellon University, Technocal report CMU-CS-95-158R, 1995.

[55] K. K. Sung and T. Poggio, "Learning human face detection in cluttered scenes," Computer Analysis of Image and Patterns, 432-439, 1995.

APPENDIX

DWT(Discrete wavelet Transform) VHDL Source

```
PROCESS(clk, reset)
variable h : signed(CST_Buf_DataBus_Width - 1 DOWNTO 0);
variable lo : signed(CST_Buf_DataBus_Width - 1 DOWNTO 0);
variable adr : unsigned(CST_Ctrl_DataBus_WIDTH - 1 DOWNTO 0);
variable srcAdr : unsigned(CST_Ctrl_DataBus_WIDTH - 1 DOWNTO 0);
variable dstAdrHi : unsigned(CST_Ctrl_DataBus_WIDTH - 1 DOWNTO 0);
variable dstAdrLo : unsigned(CST_Ctrl_DataBus_WIDTH - 1 DOWNTO 0);
variable dstAdr : unsigned(CST_Ctrl_DataBus_WIDTH - 1 DOWNTO 0);
BEGIN

IF reset = '1' THEN
        ready <= '1';
        state <= Ts_Ready;
        access_bus <= '0';
ELSE
IF clk'event AND clk = '1' THEN
CASE state IS
WHEN Ts_Ready =>CASE ctrl_sig IS
WHEN DwtCtrlVector(TCtrl_Src_start_offset) =>src_start_offset <= unsigned(ctrl_data);
WHEN DwtCtrlVector(TCtrl_Src_Step) =>src_step <= unsigned(ctrl_data);
WHEN DwtCtrlVector(TCtrl_Dst_Start_Offset_LO) =>
        dst_start_offset_lo <= unsigned(ctrl_data);
WHEN DwtCtrlVector(TCtrl_Dst_Start_Offset_HI) =>
        dst_start_offset_hi <= unsigned(ctrl_data);
WHEN DwtCtrlVector(TCtrl_Dst_Step_LO) =>dst_step_lo <= unsigned(ctrl_data);
WHEN DwtCtrlVector(TCtrl_Dst_Step_HI) =>dst_step_hi <= unsigned(ctrl_data);
WHEN DwtCtrlVector(TCtrl_Elem_Count) =>elem_count <= unsigned(ctrl_data);
        state <= Ts_Calculate;
        ready <= '0';
WHEN OTHERS =>  END CASE;

WHEN Ts_Calculate =>srcAdr := src_start_offset + Shift_Left(src_step, 1);
        ReadMemReq(enable, rw, m_adr, srcAdr);
        CheckReadClkCyc(counter, enable);
        IF (IsReadCycOver(counter) = '1') THEN
                state <= Ts_Get_X_Odd;
        END IF;
        x_even_b <= signed(mem_data);
```

```
WHEN Ts_Get_X_Odd =>srcAdr := src_start_offset + src_step;
        ReadMemReq(enable, rw, m_adr, srcAdr);
        CheckReadClkCyc(counter, enable);
        IF (IsReadCycOver(counter) = '1') THEN
                state <= Ts_Get_X_Even_A;
        END IF;
        x_odd <= signed(mem_data);

WHEN Ts_Get_X_Even_A =>srcAdr := src_start_offset;
        ReadMemReq(enable, rw, m_adr, srcAdr);
        CheckReadClkCyc(counter, enable);
        IF (IsReadCycOver(counter) = '1') THEN
                state <= Ts_CAlC_Hi_B;
        END IF;
        x_even_a <= signed(mem_data);

WHEN Ts_CAlC_Hi_B =>h := filter_5_3_odd(x_odd, x_even_a, x_even_b);
        hi_b <= h;
        hi_a <= h;
        i <= 1;
        state <= Ts_Cyc_Start;
        srcAdr := src_start_offset + src_step;
        dstAdrHi := dst_start_offset_hi;
        dstAdrLo := dst_start_offset_lo;

WHEN Ts_Cyc_Start =>
        WriteMemReq(enable, rw, m_adr, dstAdrHi, data, TYPE_DATA(hi_b), access_bus);
        CheckWriteClkCyc(counter, enable, access_bus);
        IF (IsWriteCycOver(counter) = '1') THEN
                state <= Ts_Write_LO;
        END IF;
        lo := filter_5_3_even(x_even_a, hi_a, hi_b);

WHEN Ts_Write_LO =>
        WriteMemReq(enable, rw, m_adr, dstAdrLo, data, TYPE_DATA(lo), access_bus);
        CheckWriteClkCyc(counter, enable, access_bus);
        IF (IsWriteCycOver(counter) = '1') THEN
                state <= TS_INCREMENT;
        END IF;

WHEN TS_INCREMENT =>
        i <= i + 2;
        srcAdr := srcAdr + Shift_Left(src_step, 1);
        dstAdrHi := dstAdrHi + dst_step_hi;
        dstAdrLo := dstAdrLo + dst_step_lo;
```

```
            state <= TS_CHECK_TERMINATION;

WHEN TS_CHECK_TERMINATION =>
            IF i >= TO_INTEGER(elem_count - 1) THEN
                        state <= TS_FINAL_PASS;
            ELSE
                        state <= TS_SHUFFLE;
            END IF;

WHEN TS_SHUFFLE =>
            hi_a <= hi_b;
            x_even_a <= x_even_b;
            ReadMemReq(enable, rw, m_adr, srcAdr);
            CheckReadClkCyc(counter, enable);
            IF (IsReadCycOver(counter) = '1') THEN
                        state <= TS_RELOAD;
            END IF;
            x_odd <= signed(mem_data);

WHEN TS_RELOAD =>
            adr := srcAdr + src_step;
            ReadMemReq(enable, rw, m_adr, adr);
            CheckReadClkCyc(counter, enable);
            IF (IsReadCycOver(counter) = '1') THEN
                        state <= Ts_CAlC_Hi_B_CYC;
            END IF;
            x_even_b <= signed(mem_data);

WHEN Ts_CAlC_Hi_B_CYC =>
            hi_b <= filter_5_3_odd(x_odd, x_even_a, x_even_b);
            state <= Ts_Cyc_Start;

WHEN TS_FINAL_PASS =>
            IF (elem_count(0) = '1') THEN
                        lo := filter_5_3_even(x_even_b, hi_b, hi_b);
                        state <= Ts_Write_LO_FINAL;
            ELSE
                        ReadMemReq(enable, rw, m_adr, srcAdr);
                        CheckReadClkCyc(counter, enable);
                        x_odd <= signed(mem_data);
                        hi_a <= filter_5_3_odd(signed(mem_data), x_even_b, x_even_b);
            IF (IsReadCycOver(counter) = '1') THEN
                        state <= TS_STORE_HI_A;
                        END IF;
            END IF;
```

```
WHEN Ts_Write_LO_FINAL =>
        WriteMemReq(enable, rw, m_adr, dstAdrLo, data, TYPE_DATA(lo), access_bus);
        CheckWriteClkCyc(counter, enable, access_bus);
        IF (IsWriteCycOver(counter) = '1') THEN
                state <= TS_TERMINATE;
        END IF;

WHEN TS_STORE_HI_A =>
        lo := filter_5_3_even(x_even_b, hi_b, hi_a);
        WriteMemReq(enable, rw, m_adr, dstAdrHi, data, TYPE_DATA(hi_a), access_bus);
        CheckWriteClkCyc(counter, enable, access_bus);
        IF (IsWriteCycOver(counter) = '1') THEN
                state <= Ts_Write_LO_LAST;
        END IF;

WHEN Ts_Write_LO_LAST=>
        WriteMemReq(enable, rw, m_adr, dstAdrLo, data, TYPE_DATA(lo), access_bus);
        CheckWriteClkCyc(counter, enable, access_bus);
        IF (IsWriteCycOver(counter) = '1') THEN
                state <= TS_TERMINATE;
        END IF;

WHEN TS_TERMINATE =>
        ready <= '1';
        state <= Ts_Ready;
END CASE;
END IF;
END IF;

END PROCESS;
```

IDWT(Inverse Discrete wavelet Transform) VHDL Source

```
PROCESS(clk, reset)
variable x : signed(CST_Buf_DataBus_Width - 1 DOWNTO 0);
variable adr : unsigned(CST_Ctrl_DataBus_WIDTH - 1 DOWNTO 0);
variable dstAdr : unsigned(CST_Ctrl_DataBus_WIDTH - 1 DOWNTO 0);
variable srcAdrHi : unsigned(CST_Ctrl_DataBus_WIDTH - 1 DOWNTO 0);
variable srcAdrLo : unsigned(CST_Ctrl_DataBus_WIDTH - 1 DOWNTO 0);
BEGIN

IF reset = '1' THEN
        ready <= '1';
        state <= Ts_Ready;
ELSE
IF clk'event AND clk = '1' THEN
CASE state IS
WHEN Ts_Ready =>
        CASE ctrl_sig IS
WHEN IdwtCtrlVector(TCtrl_Dst_Start_Offset) =>dst_start_offset <= unsigned(ctrl_data);
WHEN IdwtCtrlVector(TCtrl_Dst_Step) =>dst_step <= unsigned(ctrl_data);
WHEN IdwtCtrlVector(TCtrl_Src_start_offset_LO) =>
        src_start_offset_lo <= unsigned(ctrl_data);
WHEN IdwtCtrlVector(TCtrl_Src_start_offset_HI) =>
        src_start_offset_hi <= unsigned(ctrl_data);
WHEN IdwtCtrlVector(TCtrl_Src_Step_LO) =>src_step_lo <= unsigned(ctrl_data);
WHEN IdwtCtrlVector(TCtrl_Src_Step_HI) =>src_step_hi <= unsigned(ctrl_data);
WHEN IdwtCtrlVector(TCtrl_Elem_Count) =>elem_count <= unsigned(ctrl_data);
        state <= Ts_Calculate;
        ready <= '0';
WHEN OTHERS =>END CASE;

WHEN Ts_Calculate =>
        i <= 0;
        srcAdrHi := src_start_offset_hi;
        srcAdrLo := src_start_offset_lo;
        dstAdr := dst_start_offset;
        ReadMemReq(enable, rw, m_adr, srcAdrLo);
        CheckReadClkCyc(counter, enable);
        IF (IsReadCycOver(counter) = '1') THEN
                state <= TS_GET_HI;
        END IF;
        lo <= signed(mem_data);
```

```
WHEN TS_GET_HI =>
        ReadMemReq(enable, rw, m_adr, srcAdrHi);
        CheckReadClkCyc(counter, enable);
        IF (IsReadCycOver(counter) = '1') THEN
                state <= Ts_Cyc_Start;
        END IF;
        hi <= signed(mem_data);
        hi_a <= signed(mem_data);
        x_even <= filter_5_3_even(lo, signed(mem_data), signed(mem_data));

WHEN Ts_Cyc_Start =>
        WriteMemReq(enable, rw, m_adr, dstAdr, data, TYPE_DATA(x_even), access_bus);
        CheckWriteClkCyc(counter, enable, access_bus);
        IF (IsWriteCycOver(counter) = '1') THEN
                state <= TS_CHECK_TERMINATION;
        END IF;
        hi_a <= hi;

WHEN TS_CHECK_TERMINATION =>
        IF i >= TO_INTEGER(elem_count - 2) THEN
                state <= TS_FINAL_PASS;
        ELSE
                state <= TS_READ_HI;
        END IF;
WHEN TS_READ_HI =>
        adr := srcAdrHi + src_step_hi;
        ReadMemReq(enable, rw, m_adr, adr);
        CheckReadClkCyc(counter, enable);
        IF (IsReadCycOver(counter) = '1') THEN
                state <= TS_READ_LO;
        END IF;
        hi <= signed(mem_data);

WHEN TS_READ_LO =>
        adr := srcAdrLo + src_step_lo;
        ReadMemReq(enable, rw, m_adr, adr);
        CheckReadClkCyc(counter, enable);
        IF (IsReadCycOver(counter) = '1') THEN
                state <= TS_INCREMENT;
        END IF;
        lo <= signed(mem_data);
        x_even_b <= filter_5_3_even(signed(mem_data), hi_a, hi);

WHEN TS_INCREMENT =>
        srcAdrHi := srcAdrHi + src_step_hi;
```

```
            srcAdrLo := srcAdrLo + src_step_lo;
            x_odd <= filter_5_3_odd(hi_a, x_even, x_even_b);
            i <= i + 2;
            dstAdr := dstAdr + Shift_Left(dst_step, 1);
            state <= TS_STORE_X_ODD;

WHEN TS_STORE_X_ODD =>
            adr := dstAdr - dst_step;
            WriteMemReq(enable, rw, m_adr, adr, data, TYPE_DATA(x_odd), access_bus);
            CheckWriteClkCyc(counter, enable, access_bus);
            IF (IsWriteCycOver(counter) = '1') THEN
                        state <= Ts_Cyc_Start;
            END IF;
            x_even <= x_even_b;

WHEN TS_FINAL_PASS =>
            IF (elem_count(0) = '1') THEN
                        state <= TS_TERMINATE;
            ELSE
                        adr := dstAdr + dst_step;
                        x := filter_5_3_odd(hi, x_even, x_even);
                        x_odd <= x;
                        WriteMemReq(enable, rw, m_adr, adr, data, TYPE_DATA(x), access_bus);
                        CheckWriteClkCyc(counter, enable, access_bus);
                        IF (IsWriteCycOver(counter) = '1') THEN
                                    state <= TS_TERMINATE;
                        END IF;
            END IF;
WHEN TS_TERMINATE =>
            ready <= '1';
            state <= Ts_Ready;
END CASE;
END IF;
END IF;

END PROCESS;
```