ABSTRACT

HANDGRIP PATTERN RECOGNITION

by
Zong Chen

There are numerous tragic gun deaths each year. Making handguns safer by personalizing them could prevent most such tragedies. Personalized handguns, also called "smart" guns, are handguns that can only be fired by the authorized user. Handgrip pattern recognition holds great promise in the development of the smart gun.

Two algorithms, static analysis algorithm and dynamic analysis algorithm, were developed to find the patterns of a person about how to grasp a handgun. The static analysis algorithm measured 160 subjects' fingertip placements on the replica gun handle. The cluster analysis and discriminant analysis were applied to these fingertip placements, and a classification tree was built to find the fingertip pattern for each subject.

The dynamic analysis algorithm collected and measured 24 subjects' handgrip pressure waveforms during the trigger pulling stage. A handgrip recognition algorithm was developed to find the correct pattern. A DSP box was built to make the handgrip pattern recognition to be done in real time. A real gun was used to evaluate the handgrip recognition algorithm. The result was shown and it proves that such a handgrip recognition system works well as a prototype.

HANDGRIP PATTERN RECOGNITION

by
Zong Chen

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of Requirements for the Degree of
Doctor of Philosophy in Computer and Information Science

Department of Computer Science

January, 2003

APPROVAL PAGE

HANDGRIP PATTERN RECOGNITION

Zong Chen

Dr. Michael Recce, Dissertation Advisor — Date
Professor of Computer and Information Science, NJIT

Dr. James McHugh, Committee Member — Date
Professor and Acting Chairperson of Computer and Information Science, NJIT

Dr. Frank Shih, Committee Member — Date
Professor and Associate Chairperson of Computer and Information Science, NJIT

Dr. Timothy Chang, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Michael Beres, Committee Member — Date
President and CEO of BES System Company

BIOGRAPHICAL SKETCH

Author: Zong Chen

Degree: Doctor of Philosophy

Date: January, 2003

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science
  New Jersey Institute of Technology, Newark, NJ, USA, 2003

- Master of Science in Electrical Engineering
  Southeast University, Nanjing, China, 1997

- Bachelor of Science in Electrical Engineering
  East China Institute of Technology, Nanjing, 1992

Major: Computer Science

Publications:

Chen, Z., Perl, Y., Halper, M., Geller, J., and Gu, H., "Partitioning the UMLS Semantic Network," *IEEE Transactions on Information Technology in Biomedicine*, Vol. 6(2), pages102--108, 2002.

Geller, J., Perl, Y., Halper, M., Chen, Z., and Gu, H., "Evaluation and Application of a Semantic Network Partition," *IEEE Transactions on Information Technology in Biomedicine*, Vol. 6(2), pages109--115, 2002.

Halper, M., Chen, Z., Geller, J., and Perl. Y., "A metaschema of the UMLS based on a partition of its Semantic Network," In S. Bakken, editor, *Proceedings of the 2001 American Medical Informatics Association (AMIA) Annual Symposium*, pages 234-238, Washington, DC, November, 2001.

Chen, Z., Halper, M., Geller, J., and Perl. Y., "A structural partition of the Unified Medical Language System's Semantic Network," In S. Laxminarayan, editor, *Proceedings of the 2000 IEEE EMBS International Conference on Information*

Technology Applications in Biomedicine (*ITAB'00*), pages 296-301, Arlington, VA, November, 2000.

Chen, Z., Gu, H., Halper, M., Geller, J., and Perl, Y., "A metaschema for the UMLS semantic network," In F. Valafar, editor, *Proceedings of the Int'l Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences* (*METMBS'00*), pages 755-761, Las Vegas, NV, June 2000.

Chen, Z., and Zou, C., "An Improved Multiresolution Adaptive Image Smoothing Algorithm", *Journal of Southeast University*, Vol. 27, No.1, 1997.

This dissertation is dedicated to

my family for their love and support

ACKNOWLEDGMENT

The author would like to express his sincere gratitude to his dissertation advisor Professor Michael Recce, for his remarkable guidance, constant support and encouragement throughout this research. The author's deepest appreciation also goes to Professor Timothy Chang and his Ph.D. student, Biao Cheng, for their help to build the DSP box.

The author would like to thank Dr. James McHugh, Dr. Frank Shih, and Dr. Michael Beres for their valuable suggestions to this dissertation and for serving as members of the committee.

The author would also like to express his special appreciation to Dr. Ivanov Dentcho, Dr. Dmitri Shishkin, Mr. Mourad Touzani, Mr. David Jack, and the other members of the research group for their timely help and support. The author also appreciates the suggestions from Dr. Yehoshua Perl, Dr. James Geller, Dr. Michael Halper, and Dr. Huanying Gu.

Finally, the author would like to express his deepest appreciation to his family, especially his parents and his wife, for their love, faith, patience and many sacrifices.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation of Personalized Gun

There are numerous tragic gun deaths each year. Many of those are related to unintentional shooting by young children. Statistically, the rate of unintentional firearm deaths among the group aged 15-19 was nearly five times higher than the unintentional firearm death rate for all other age groups (National Center for Health Statistics, 1998). For each fatal unintentional shooting, there are an estimated 10 nonfatal unintentional shootings victims treated in emergency rooms each year (Sinauer and Annest, 1996). In most cases, accidental firearm discharges could be prevented by firearm a safety device (US General Accounting Office, 1991).

Stolen firearms are another problem. Evidence from the National Crime Victimization Survey (NCVS) indicates that there are nearly 350,000 incidents of firearm theft from private citizens annually. Most of the stolen guns are handguns. Surveys indicate that guns illegally sold to criminals on the street often have been stolen from homes. When a personalized handgun is stolen, it cannot be used because it will function only for the authorized user.

Police officers would also benefit from personalized guns. FBI statistics show that during the decade 1987-1996, dozens of law enforcement officers in the United States were killed by criminals who wrested their firearms away and then shot them or other officers. Personalized guns would allow officers to exchange their weapons among the team and would prevent criminals from using them.

Many products in the United States have been modified to make them safer, including drug packages and motor vehicles. Government mandated changes in drug packaging have resulted in reductions of the number of deaths from child poisonings. Studies show air bags have reduced the total number of driver fatalities by 20 to 25%. Making handguns safer by personalizing them could similarly save lives. Personalized handguns, also called "smart" guns, are handguns that can only be fired by the authorized user.

1.2 Personalized Gun Technologies

Many devices have been designed to make the gun smarter. Basically, they can be divided into two categories, "Electronic" and "Biometric".

1.2.1 Electronic Technologies

Electronic technology uses an electronic device that allows a gun to fire when the device receives a correct signal. Examples are electromechanical lock, magnetic lock, and radio lock. Electronic technology is a mature technology and its reliability has been proven.

Electromechanical Lock

A lock powered by a battery is installed into the grip and trigger system. A personalized three or four digits preset identification number, entered into the keypad, allows an authorized user to unlock the gun. By using the push buttons singularly or in combinations three to six times, there are over ten thousands combinations. Repeated entry of wrong combinations locks the gun down for hours. This lock is personalized for user who knows the combination. This device is waterproof and can be used in extreme

conditions. However, the disadvantage is obvious: the user might forget the PIN or do not have time to input the PIN in some emergency cases. Smith & Wesson is engineering a gun using this technology.

Magnetic Lock

A magnetic lock is installed inside the grip of the gun to prevent the gun from firing. The magnet inside the grip is positioned to match the magnetic ring worn by the authorized user. The user adjusts the location of the blocking magnet inside the grip to correspond to their hand holding pattern. When the gun detects the correct magnetic field emitted from the magnetic ring worn by the user, a blocking device inside the grip moves away from its blocking position and lets the trigger be pulled. The disadvantage is the gun can operate if an unauthorized person has the ring. Furthermore, a gun owner may keep the ring and gun together to make it easier to find the ring when he wants to use the gun – which reduces the safety provided by the ring. An inventor, Ken Pugh, has led the development of this technology.

Radio Lock

An alternative uses a radio transponder. The authorized user wears a transponder bearing a unique code. The transponder can be imbedded in a ring, wristwatch, wristband, lapel pin or badge. The firearm transmits low power radio signals to the transponder (the grip contains a battery and microchip), which in turn notifies the firearm of its presence. If the transponder code is the one that has previously been entered into the firearm and is in a specified distance from the gun, the firearm recognizes it, the trigger can be pulled and the gun can fire. The gun can be programmed to recognize many different transponder codes. However, signal interference is possible. If the signal can't be transmitted clearly,

the gun operation could malfunction. This technology also has the limitations described above for magnetic locks – that the owner might keep the wristband or ring with the gun. FN Manufacturing, Inc. is currently developing this technology into a handgun for police officers.

1.2.2 Biometric Technologies

Biometric technology is rapidly developing. It concerns identification and verification based of physiological characteristics (fingerprint, voice, face, skin, DNA, hand geometry), or behavioral characteristics (keystroke dynamics, handwritten signature, handgrip) of individuals. These human characteristics are universal, which means every person has such characteristics, and they are unique, which means no two persons should be exactly the same in terms of these characteristics. Therefore, a smart gun based on biometric technology provides better protection from unauthorized users than electronic technology. On another hand, unlike electronic technology which is mature, and whose reliability has been proven, biometric technology is a new, on-developing technology. There are many problems that need to be solved, such as, which algorithm is good for a smart gun, how to arrange the sensors, and what happens if the battery runs out of power. The reliability of biometric technology also needs to be proven. Three biometric technologies have been applied to smart guns. They are fingerprint identification, skin analysis, and handgrip technology.

Fingerprint Identification

Fingerprint technology is the most well known biometric technology. It has been applied to many security applications. Oxford Micro Devices, Inc., is developing low cost,

miniature fingerprint capture and verification systems that can be built into the gun handle to quickly capture and recognize a person's fingerprints. Only persons whose fingerprints match those of authorized users would be able to fire a gun. A single gun could be operated by more than one person if desired, and the valid fingerprints could be changed if needed at a gun shop or police station. The system combines Oxford's high-speed A336 image processor chip with a tiny fingerprint sensor, a small memory chip for storing the fingerprints, the software to operate the A336 chip and an intuitive human interface so that the gun could be used quickly and easily. However, the fingerprint technology requires a clear scan image of the fingerprints which in some cases is not appropriate. Guns are used in all types of weather and in many different situations. If the user has dirty hands or wears gloves, fingerprint detection is useless. Also it is possible to use tape to transfer fingerprint.

Skin Analysis

Lumidigm Inc. is developing a light print sensor measuring the absorption of a range of colored light signal through the skin. Skin layer thickness, capillaries, and other structures all affect the light, creating a distinctive pattern of changes. This system works on any skin surface and is unaffected by cuts and burns. Smith & Wesson is working with Lumidigm to build a smart gun using this technology. The disadvantages of skin analysis are similar to fingerprint technology. Dirty hands or wearing gloves will make this technology useless.

Handgrip Pattern Recognition

Handgrip pattern recognition is another biometric technology that can be used to build a smart gun. Handgrip pattern recognition looks at the fingertip position distribution and

the handgrip pressure variation during a short period just before the gun trigger is pulled. The complexity of the human hand and its environment make written signatures highly characteristic and difficult to forge precisely. The same factors that make a written signature unique are also exhibited in a user's handgrip pattern. By comparing the features of fingertip position and handgrip variation, an authorized user can be recognized. This technology will be used for this research project.

1.3 Issues of Handgrip Pattern Recognition

Handgrip pattern recognition concerns about the fingertip position distribution and the handgrip pressure variation during a short period just before the gun trigger is pulled. By observation, the pressure variation is more consistent while the time nearer to the trigger pulled time point. In this project, the pressure variation during 100 ms before trigger pulled is considered as the input signal.

Pressure sensor translates pressure into voltage. The sensitivity of the pressure sensor needs to be considered. First, the pressure sensor should be able to translate pressure into voltage in less than 1 ms. Second, the pressure sensor should be able to tell small pressure variations. At the same time, the size of pressure sensor also needs to be considered. Many sensors may be put around the gun handle. The sensor size can not be too big.

When one pressure sensor is used, it can tell the fact whether this gun is grasped by someone or not by measuring the pressure on the sensor. When two sensors used, it may tell the fact that whether an adult grasping the gun or a child grasping it, because the adult has bigger hand and longer finger, therefore can touch two sensors, while the child

has smaller hand and shorter finger, only can touch one sensor. By using more sensors, more characteristics can be discovered. To maximize the recognition ability, more than 1 less than 100 sensors will be considered to put around the gun handle.

Noise is an issue in the handgrip pattern recognition. When pressure sensor translates pressure into voltage, noise will be introduced too. To eliminate noise, digital signal processing methods such as filtering need to be developed.

To identify the user, pattern recognition algorithms need to be chosen carefully. The identifying power of a particular algorithm is measured by two terms: False Rejection Rate (FRR), or Type I Errors, and False Acceptance Rate (FAR), or Type II Errors. A type I error occurs when the gun does not function for an authorized user. A type II error occurs when the gun allows an unauthorized user to fire it. The aim is to minimize these two error rates. However, false rejection rate and false acceptance rate are complementary. For example, if the false acceptance rate threshold is increased to make it more difficult for impostors to gain access, it also will become harder for authorized people to gain access. As FAR goes down, FRR rises. On the other hand, if the false acceptance threshold is lowered as to make it very easy for authorized users to gain access, then it will be more likely that an impostor will slip through. Hence, as FRR goes down, FAR rises.

The perfect smart gun is such a gun that it only let the authorized user to fire it and the authorized user can operate the gun in all situations. However, this ideal is impossible right now. The experiments described here are preliminary research. The present goal is to keep the false rejection rate below 1/20,000 which is the average mechanical failure rate, and keep the false acceptance rate at about 1%.

1.4 Overview of the Thesis Presentation

In this chapter, the research motivations for smart guns are introduced. Some existing gun safety technologies are discussed. Some of the smart gun issues are presented too.

Biometric identification technologies have been used for many decades. Most biometric identification techniques use either the physiological characteristics or behavioral characteristics to do the recognition. Some biometric recognition techniques will be presented in Chapter 2.

In Chapter 3, some related topics in signal processing will be presented. Fourier transforms and wavelet analysis techniques also will be discussed. These techniques are used in the experiments described in later chapters.

Chapter 4 is devoted to the discussion of some of the pattern recognition algorithms used in this research. Emphasize is placed on non-parametric pattern recognition algorithms because there is no existing model for handgrip pressure variation, and the parameters for handgrip recognition are difficult to estimate.

Fingertip positions of 160 subjects were collected and stored in the computer. The analysis of these fingertip locations is called static analysis. Static analysis is described in Chapter 5. A density search algorithm is used to partition the fingertip data set into four natural groups. Canonical coefficients are calculated to indicate the importance of each variable. Classification trees are calculated and shown for each finger.

The analysis of handgrip pressure variation is called dynamic analysis and it is described in Chapter 6. Handgrip pattern recognition algorithms are given in time domain

and frequency domain. Evaluation results are presented, and subjects are divided into groups to improve the recognition performance.

In Chapter 7, a real handgun, Beretta 92FS, and a simulator were used to evaluate grip recognition in a realistic environment. A DSP box was built and programmed to do the pattern recognition in real time.

Finally, the conclusion with the main contributions of this research and possibilities for future work in this area are given in Chapter 8.

CHAPTER 2

BIOMETRIC IDENTIFICATIONS

Biometrics is the development of statistical and mathematical methods applicable to data analysis problems in the biological sciences. Biometrics measures and analyzes a person's physiological or behavioral characteristics for identification and verification purposes. It associates an individual with a previously determined identity/identities based on how one is or what one does (Jain, Hong, and Pankanti, 2000). Since many physiological or behavioral characteristics are distinctive to each person, biometric identifiers are inherently more reliable and more capable than password based techniques in differentiating between an authorized person and an impostor.

A biometric system is essentially a pattern recognition system that makes a personal identification by establishing the authenticity of a specific physiological or behavioral characteristic possessed by the user. Logically, a biometric system can be divided into the enrollment module and the identification module (Jain, Hong, and Pankanti, 2000). During the enrollment phase, the biometric characteristic of an individual is scanned by a biometric sensor to acquire a digital representation of this characteristic. In order to facilitate matching and to reduce the storage requirements, the digital representation is further processed by a feature extractor to generate a compact but expressive representation, called a "template." During the recognition phase, the biometric reader captures the characteristic of the individual to be identified and converts it to a digital format, which is further processed by the feature extractor to produce the

same representation as the template. The resulting representation is fed to the feature matcher that compares it against the template(s) to establish the identity of the individual.

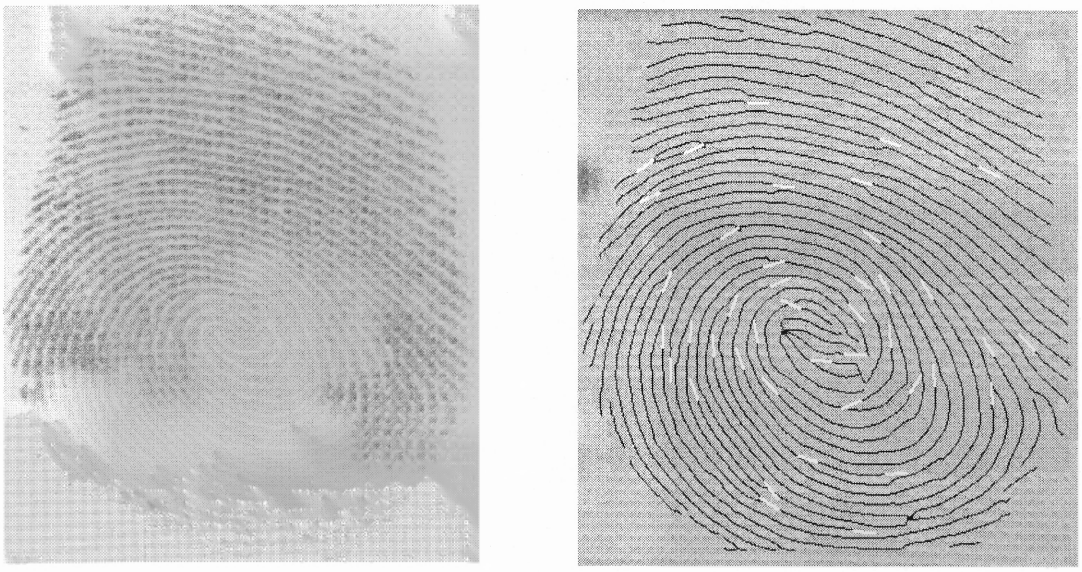
There are a number of biometric techniques widely used or under investigation. These techniques use either the physiological characteristics, like fingerprint, face imaging, hand and finger geometry, or behavioral characteristics, like handwritten signature and dynamic keystrokes.

2.1 Fingerprint Identification

Among all the biometric techniques, fingerprint based identification is the oldest method. Humans have used fingerprint for personal identification for centuries and the validity of fingerprint identification has been well established. A fingerprint is made of a series of ridges and valleys on the surface of the finger. The uniqueness of a fingerprint can be determined by the pattern of ridges and valleys as well as the minutia points. Minutia points are discontinuities in the ridge-valley pattern, which can be described as a combination of ridge endings and bifurcations. The types, positions and orientations of the minutiae are reliable features for fingerprint matching. Figure 2.1(a) shows the ridge-valley structure, where the ridges are represented by black lines, and the valleys are white. The minutia points are shown in Figure 2.1(b).

Most fingerprint recognition systems make use of minutiae. Before the minutiae being extracted and coded, the fingerprint image should be preprocessed. Initially, the original gray scale fingerprint is converted to binary format. A skeleton image is produced by thinning the lines within the binary image until the lines are one pixel wide (Lam, Lee, and

Suen, 1992). The advantage derived from using a skeleton image is that extraction of ridge features becomes a relatively straightforward procedure based on tracing line segments.



(a) Original fingerprint image. (b) Minutia points detected on a preprocessed image.

Figure 2.1 Fingerprint images.

Minutiae detection over the thinned image is a simple process. The ridge endings and bifurcations on the thinned image can be found by calculating the neighborhood of each pixel. An end point is a pixel with either one or no neighbors, and a bifurcation has exactly three neighbors. All other skeleton components have exactly two neighbors and are referred to as connecting points. Figure 2.1(b) shows some minutia points detected on the thinned image of Figure 2.1(a).

All available minutiae based fingerprint matching algorithms rely heavily on the quality of the input fingerprint images. However, a fingerprint image is one of the noisiest of image types. Dry skin tends to cause inconsistent contact of the finger ridges with the scanner's plate surface, causing broken ridges. For oily or wet skin, the valleys tend to fill up

with moisture, causing them to appear dark in the image similar to ridge structure. Wrinkles, scars, and excessively worn prints result in images containing many "false minutia" structures. Therefore, minutiae reduction needs to be performed (Xiao and Raafat, 1991). False minutiae can be eliminated using some distance criteria, e.g., minutiae which are too close to each other, or minutiae which are too close to the regions corresponding to the background or bad areas in the fingerprint, are discarded. Also, connectivity may be used for minutiae elimination. Minutiae that are connected by a short line segment are discarded.

The reduced minutiae list provides a matching module. In the matching stage, two sets of minutiae code using their locations are aligned (Jain and Minut, 2002) and the sum of similarity between the overlapping minutiae is calculated. Figure 2.2 shows the alignment and the matching of two sets of fingerprint images.

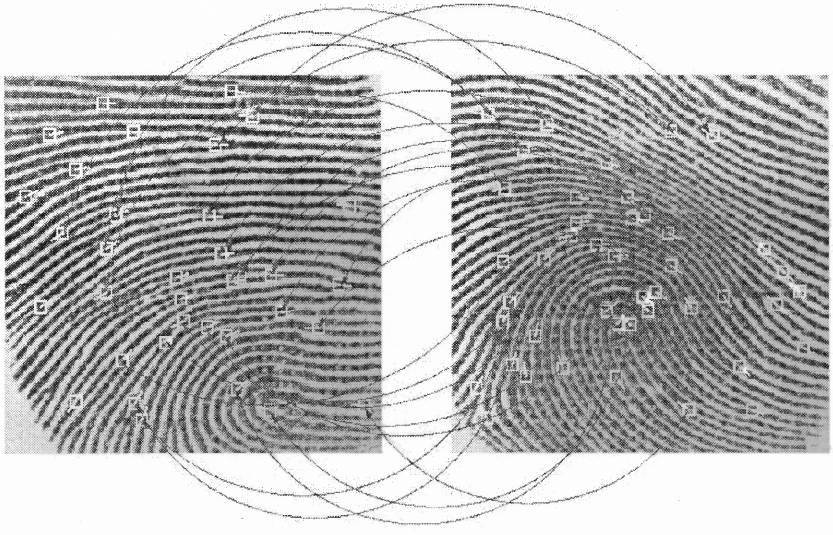


Figure 2.2 The alignment and matching of two sets of fingerprint images.

A match is made if the type and orientation of the two set minutiae are in the correct locations (Roddy and Stosz, 1999). Type matching is a binary decision (end point or

bifurcation), whereas location and orientation allow for some variation. The constraint on location is that a minutia point from the test fingerprint must be located within the image area corresponding to the template minutia point. The orientation constraint is that the detected minutia angle of rotation must be within 45° of the corresponding template minutia orientation value. The minutia matching score is the ratio of correctly matched minutia points to the total number store in the template file.

2.2 Dynamic Keystroke Recognition

As early as the turn of this century, psychology demonstrated that the mechanics of human actions are predictable in the performance of repetitive, routine tasks (Umphress and Williams, 1985). In 1895, observation of telegraph operators showed that each operator had a distinctive pattern of keying messages over telegraph lines (Bryan and Harter, 1973). The keystroke dynamics analyzes the way a user typing at a terminal by monitoring the keyboard inputs, and aims to identify users based on certain habitual typing rhythm patterns. When a person types, the latencies between successive keystrokes, keystroke duration, finger placement and applied pressure on the keys can be used to construct a unique signature for that individual (Monrose and Rubin, 1997). For regularly typed strings, such as username can be quite consistent.

A latency signature can be visualized by plotting characters typed versus latency times between successive keystrokes (Joyce and Gupta, 1990). A sample is shown in Figure 2.3. The latency time is the elapsed time from the keystroke of the previous letter to the next.

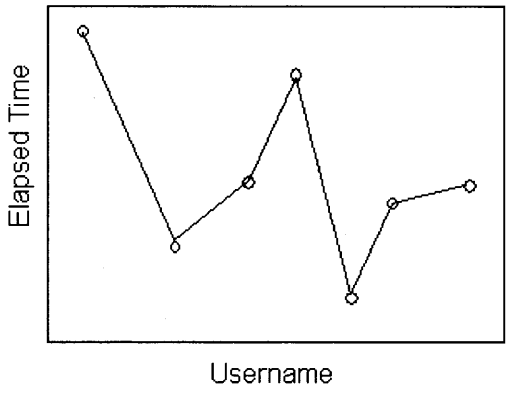


Figure 2.3 Latency signature.

The identity verifier compares a test keystroke signature with the reference keystroke signature. To carry out the comparison, a mean reference signature is first computed by calculating the mean and standard deviation. Then the Manhattan distance function or the Euclidean distance function can be used to calculate the similarity between the test signature with the reference signature. Once the distance has been computed, a suitable threshold should be decided. The threshold for each user should be set based on a measure of the variation of his/her signatures. A user that has little variation in his/her signatures would have a small threshold while another user with large variation should have larger threshold for accepting his/her test signatures. The standard deviation of the reference signatures can be used to decide such a threshold. Assuming the latency in normal distribution, success login rate would be about 84% if the threshold being set as mean plus one standard deviation. A threshold value based on two standard deviations should provide a false alarm rate of less than 3% although the imposter pass rate with a larger threshold would obviously be expect to be larger.

2.3 Handwritten Signature Verification

There is considerable interest in authentication based on handwritten signature verification. A lot of security and financial reasons justify the research in this field, like the verification of checks, transactions with credit cards and public documents. Researches have shown that a unique signature template can be built for an individual. Each time he signs, he traces out his template, not exactly identical but with small stochastic variations (Hastie, Kishon, Clark and Fan., 1991). These variations include the speed of writing, rotation, scale, and shear.

The signature data can be obtained from a digitizer, instrumented pen or similar device. Typically, the output of such a device gives the position in (X, Y) coordinates at equal spaced time intervals. Along with each location measurement, the digitizer also can record the downward pressure on the pen, or the force exerted on the tip of the pen. Given a new signature claiming to belong to subject s, the system compares it to the template signature for subject s. If the variations in the factors are beyond the threshold established for subject s, the verification fails.

Handwriting, much like speech, is time dependent. No two signatures will have exactly the same timing pattern, and these timing differences will not be linear. Dynamic Time Warping (Sankoff and Kruskal, 1983) allows one to get a point-to-point correspondence between two signals which is relatively insensitive to small difference in their timing patterns. In the work of Yasuara and Oka (1977), Dynamic Time Warping was used to match the (X_t, Y_t) coordinates of two signatures. In doing so, it introduces nonlinear distortions in the time domain to compensate for rotation, scaling and shear

which are linear transformation in the X, Y domain. The speed signal, on the other hand, is effectively invariant under these transformations.

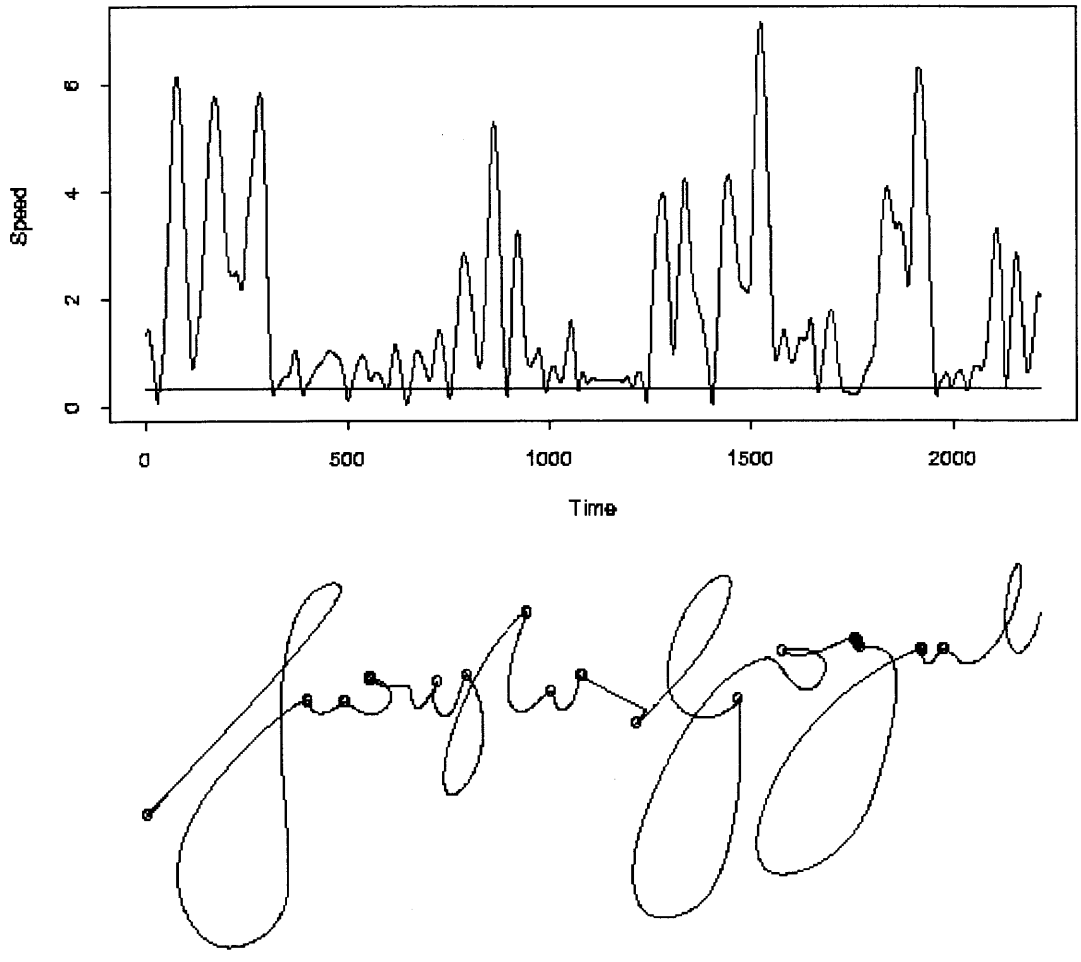


Figure 2.4 Signature segmentation based on the signal speed.

The basic idea of the handwritten signature recognition is to align and segment each signature into a series of corresponding sub-curves or *letters*, and then compare these letters with the template to find the whether the variations bigger than the threshold (Hastie, Kishon, Clark and Fan., 1991). This is achieved in several steps. The speed signal of the signature is first time-warped against a reference signature. By observation, handwriting speed is slow around points of high curvature, which are the areas of interest

for segmentation. Therefore, the speed signal of the signature can be used to segment it into distinct *letters*.

Figure 2.4 shows how a signature is segmented based on the speed signal. Each letter is then rotated and shifted to match the template letter as well as possible. At last, the distance between the input signature and the template is calculated at the letter level. The distance is compared to a threshold to verify whether this user is the authorized user.

2.4 Summary

Some commonly used biometric technologies are discussed in this chapter. Each biometric technology has its strengths and limitations. Fingerprint identification is the most well known biometric technology. It is easy to use, accurate and reliable. Fingerprint identification has been used in numerous applications. It also has been proposed for the smart gun. However, fingerprint identification requires a clearly scanned fingerprint image which is not always possible. User might have dirty hand or wear gloves. In such cases, fingerprint identification is not appropriate.

The behavioral biometric technology is based on the fact that human actions are predictable in the performance of repetitive, routine tasks. Keystroke dynamics and handwritten signature, discussed in this chapter, are two examples of behavioral biometric techniques. For an experienced firearm user, firing a gun is such a repetitive action. Like keystroke dynamics and handwritten signature, it is also possible to identify a subject by analyzing his handgrip dynamics. The analysis of how a person grasping the gun, and how the handgrip pressure varying during the firing process will be discussed in later chapters.

CHAPTER 3

HANDGRIP SIGNAL PROCESSING

Signal processing is used widely in many areas. Thousands of algorithms have been developed for different signal processing tasks. This chapter briefly discusses some of the signal processing algorithms related to sensor data processing. Initially, a general model for digital signal processing system is given. Then, stochastic signal processing algorithms in time domain and frequency domain are discussed. Wavelet analysis algorithms are also presented in this chapter. Wavelet analysis is widely used for feature extraction from noisy signals and is especially suitable for non-stationary signal processing.

A *signal* can be considered to be a function of independent variables (Mitra, 1998) such as time, distance, position, temperature, pressure, and etc. A signal may be continuous or discrete. Analog signals usually are continuous both in time and amplitude. Most signals that are acquired from the physical world are analog signals, like temperature, speed, acceleration, etc. Analog signals can be processed using analog circuits such as amplifiers and filters. These devices take analog signals as inputs and give analog signals as outputs.

Digital signals are discrete both in time and amplitude. Usually, they are the results of sampling and quantization from the corresponding analog signals. Due to the wide use of digital computers, it is natural that much of current signal processing is performed digitally.

In practice, sampling as well as quantization is done by electronic analogue-to-digital (A/D) converter circuits. Based on Shannon's sampling theorem (Shannon, 1948), if the input signal is bandwidth limited and the its maxim frequency component is f_x, then the signal can be restored without losing any information after the sampling if the sample rate is above $2f_x$.

The signal collected from the pressure sensors, which are put on the handgun handle, is the signal needs to be recognized in this research project. Each pressure sensor translates the pressure signals into voltage signals. The dynamic range of the pressure sensor needs to be selected carefully so that it can pick up the details of the pressure variation and does not get saturated for the big signal at the same time.

How many sensors need to be put on the gun handle is another issue. More sensors will give more accurate signal. However, more sensors require more computation time and make signal recognized slower. To achieve the goal that recognizing a user correctly in real time, a trade-off must be properly selected. The size of the pressure sensor is the issue related the number of sensors. If more sensors need to be put on the gun handle, the size of these sensors must be smaller.

Time constraint of the pressure signal needs to be considered because not every squeeze on the gun handle is the valid signal. Only the pressure variation during a short period before trigger pulled is the valid signal. Generally, people can click a button in 100ms or remove his/her hand from a button in 200ms when this person receives a "go" or "stop" signal. Therefore, 100ms is considered as the psychological moment. It means that if an action is done in 100ms, people will think this action is done instantaneously.

Therefore, the handgrip signal during the period of 100ms before the trigger pulled is the signal needs to be studied.

An A/D converter converts the analog voltage signals into digital signals by sampling and quantization. Since 100ms is considered as the psychological moment, each muscle probably moves in 10ms. If the sample rate is set to 1ms each sample, it will be fast enough to catch every muscle movement. Therefore, the sample rate is set to 1000 samples/second.

3.1 Digital Signal Processing System

A signal carries information, and the objective of signal processing is to extract the information carried by the signal (Mitra, 1998). The study of signals, their properties, their fundamental mathematical and physical limitations, belongs to signal analysis. A signal processing system is a device that processes input signals and produces output signals. Numerical or logical values can be considered as output signals too. In such case, various signals may be input to the system, and the system classifies them as they arrive as belonging to a particular class. By this view, handgrip pattern recognition system can be considered as such a DSP system that when a test handgrip signal is input into the system, several logic signals are output. The TRUE logic output signal indicates that an authorized user using this gun so that the gun should be operational, while the FALSE logic signal indicates that an unauthorized user is trying to use the gun, therefore the trigger should be blocked. The figure below shows a simplest model of digital signal processing system.

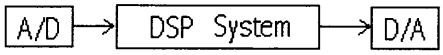


Figure 3.1 A general DSP system.

A simplest signal processing system is an amplifier, where the output signal is $y(t)=Ax(t)$ for analog signal or $y(n)=Ax(n)$ for digital signal. The ideal amplifier should be a linear system, i.e. $A[x_1(n)+x_2(n)]=Ax_1(n)+Ax_2(n)$, and $A[cx(n)]=cAx(n)$. Such ideal amplification can only be approximated in analog circuits but can be performed easily by digital processing system using a multiplication unit.

For more complex system, digital signal processing is more attractive because it reduces the system complexity significantly. In many situations, the output signal need to be considered as a function of present inputs, previous inputs and previous outputs, i.e., $y(n)=f[x(n), x(n-1), x(n-2), ..., y(n-1), y(n-2)]$. A small analog delay can be implemented by delay lines, which are appropriately chosen lengths of cable that allow the electric signal take a specific time to travel. By contrast, digital delays can be simply implemented by a FIFO buffer.

From the above examples, people have seen a lot of advantages using digital signal processing units instead of using analog circuits. To simplify the circuit design for the handgrip signal processing, a DSP unit (TMS320C31 DSP processor) is used to perform most of the signal processing.

In most cases, the desired signal $z(n)$ can not be directly observed. Only the signal $x(n)=z(n)+v(n)$ can be acquired, where $v(n)$ is an additive noise, with an average of zero.

Since the noise average is zero, the average of $x(n)$ should equal to $z(n)$. The method of how to reduce noise sufficiently and keep tracking of $z(n)$ at the same time is moving average (MA), which can be represented by $y(n)=1/M \sum x(n-m)$, where m=0, 1, ...M-1, and $y(n)$ is the approximation to the original $z(n)$. Using this equation will indeed remove the noise; however, it will also harm the signal. Notice in the above equation, all $x(n-m)$ are given equal weights. A more general MA filter will be like $y(n) = \sum a(m)\ x(n-m)$, m=0, 1, ... M-1, where the coefficients $a(m)$ need to be chosen to maximize the noise reduction and minimizing the signal distortion.

For the moving average equation $y(n)=1/M \sum x(n-m)$, if somebody wished to try M+1, there is no need to add up all x again. $y(M)= [M\, y(M-1) + x(M)]/(M+1)$ can be used instead. This manipulation has converted the MA model into a recursion $y(n)=(1-\beta)\ x(n) + \beta\ y(n-1)$, $0<\beta<1$, which is called autoregression (AR) model. More generally, AR model can be represented by

$$y(n)= x(n) + \sum_{m=0}^{M} b(m)\, y(n-m) \tag{3.1}$$

In MA model, the output signal is only related to current and previous input signals. In AR model, the output signal is only related to current input and previous output signals. More general model combines AR and MA model is called ARMA model, where the output signal depends on both previous input and previous output signals.

$$y(n) = \sum_{l=0}^{L} a(l)\, x(n-l) + \sum_{m=0}^{M} b(m)\, y(n-m) \tag{3.2}$$

For a system, if the relationship of its input signal and output signal can be represented by the ARMA model, then this system is a linear time-invariant system.

Time-invariant means the model parameters do not change over time. Linear means that if $x(t) \rightarrow y(t)$ and $x'(t) \rightarrow y'(t)$, then $cx(t) \rightarrow cy(t)$, and $x(t)+x'(t) \rightarrow y(t)+y'(t)$, where c is a constant.

Most real systems have complex properties and no simply characterization is possible. However, it is usually convenient to restrict the study of systems to some limited conditions where the system may be linear. Therefore, ARMA model is widely used in signal analysis.

The handgrip recognition system is a complicated digital signal processing system. It is time-invariant, but not linear. If $x(t) \rightarrow$ TRUE and $x'(t) \rightarrow$ FALSE, $x(t)+x'(t)$ can not make the output as TRUE+FALSE. However, it is convenient to assume the handgrip pattern recognition system as a linear time-invariant system in some stages such as denoising, data reduction and feature extraction. Therefore, an ARMA model can be applied to the task.

A consequence of the linear time-invariant property is that a LTI discrete-time system is completely specified by its impulse response $h(n)$.

$$y(n) = \sum_{m=-\infty}^{\infty} h(n-m)\, x(m) \tag{3.3}$$

This equation is also called convolution sum of $x(n)$ and $h(n)$, and represented compactly as $y(n) = h(n) \otimes x(n)$, where the notation $\otimes$ denoted the convolution sum. Knowing the impulse response, the output sequence for any given input sequence can be computed by using the convolution sum of above equation. Therefore, a DSP system can be represented as a figure as below.

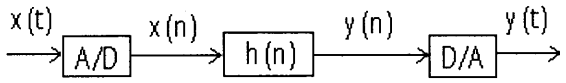


Figure 3.2 A linear time-invariant DSP system model.

If a LTI system is constructed by two cascaded LTI subsystem with impulse response function $h_1(n)$ and $h_2(n)$, then the whole system impulse response function is $h(n)= h_1(n)\otimes h_2(n)$. If a LTI system is constructed by two paralleled LTI subsystem with impulse response function $h_1(n)$ and $h_2(n)$, then the whole system impulse response function is $h(n)= h_1(n)+ h_2(n)$.

A realizable system should satisfy two conditions: $h(n)=0$, when $n<0$; and $\sum|h(n)|<\infty$ for $-\infty<n<\infty$. If the impulse response sequence of a system is finite length, then this system is called finite impulse response (FIR) system. If the impulse response sequence of a system is infinite length, then this system is called infinite impulse response (IIR) system. Usually, FIR system uses an MA model, and IIR system uses an AR or ARMA model. The finite impulse response implies $\sum|h(n)|<\infty$, therefore, FIR system design does not need to consider the system stability problem, which is the most important issue of an IIR system design. To simplify the algorithm design, an FIR is chosen for denoising task.

3.2 Frequency Domain Analysis

The Fourier transform is the most important tool in the signal analysis theory. There are a lot of advantages to Fourier transforms. First, in the analysis and design of systems, it is often useful to characterize signals in terms of frequency domain parameters such as bandwidth or spectral content. Second, the response of a linear time-invariant system to a sinusoid of a given frequency is itself a sinusoid of the same frequency, providing a means of solving for the response of such systems (Ziemer, Tranter, and Fannin, 1993). The Discrete Fourier transform (DFT) can be represented by

$$X(k) = X(f)\Big|_{f=\frac{k}{NT}} = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} \tag{3.4}$$

where T is the sampling interval. For a finite length N signal, its DFT can be simplified to

$$X(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nfT} \tag{3.5}$$

Its inverse discrete Fourier transform (IDFT) is given by

$$x(n) = \sum_{k=0}^{N-1} X(k) e^{j2\pi nk/N} \tag{3.6}$$

Applying the DFT to the impulse response function, $y(n)=h(n)\otimes x(n)$ turns to $Y(f)=H(f)X(f)$, where $Y(f), H(f)$ and $X(f)$, are the Fourier transforms of $y(n)$, $h(n)$ and $x(n)$, respectively. Here $H(f)$ is also called transfer function, which represents the system frequency properties. Another importance of this equation is that the convolution can be simplified to multiplication by using the Fourier transform.

One of the most widely used signal processing operations is filtering. Filtering is used to pass certain frequency components in a signal through the system without any

distortion and to block other frequency components. The system implementing this operation is called a filter (Mitra, 1998). The range of frequencies that is allowed to pass through the filter is called the pass-band, and the range of frequencies that is blocked by the filter is called the stop-band.

The key to the filtering process is the inverse discrete Fourier transform. By appropriately choosing the values of magnitude function of the LTI filter at frequencies corresponding to the input signal frequency components, some of these frequency components can be selectively heavily attenuated or filtered with respect to the others.

A low-pass filter passes all low frequency components below a certain specified frequency f_c. A high-pass filter passes all high frequency components above a certain frequency f_c and blocks all low frequency components below f_c. A band-pass filter passes all frequency components between two frequencies f_{c1} and f_{c2}, and blocks all frequency components below the frequency f_{c1} and above the frequency f_{c2}. A band-stop filter blocks all frequency components between two frequencies f_{c1} and f_{c2}, and passes all frequency components below the frequency f_{c1} and above the frequency f_{c2}.

In many applications, the input signal may get corrupted by noises, where the desired signal occupies a frequency band from f_{L1} to f_{H1}, and the noise occupies another frequency band from f_{L2} to f_{H2}. In such cases, filters can be used to recover the desired signal from the noise-corrupted signal.

The handgrip signals collected through the pressure sensors inevitably contain noise. This noise effects the recognition tasks and therefore needs to be eliminated. Since noise is distributed evenly across all frequency components, and most of the pressure signal usually is in the low frequencies, a low-pass filter can be used here to delete the

noise. Another source of noise is power line 60Hz sinusoidal signal corrupting the desired handgrip signal. A band-stop filter with center frequency at 60Hz can remove such noise.

3.3 Stochastic Signal Processing

The most significant characteristic of a signal is whether it is deterministic or stochastic. Deterministic signals are those that are generated by some none-probabilistic generator They are reproducible, predictable and mathematically well behaved. Stochastic signals are generated by systems that contain randomness. At any particular time, the stochastic signal is less than mathematically predictable. In fact, all signals collected from real world are stochastic signals because signals inevitably contain noise. Except noise, some signals are inherently stochastic. Many physiological signals do not have obvious regularity and are stochastic signals (Devasahayam, 2000). The handgrip signal is an example of a stochastic signal. Each person has his/her own handgrip pattern. Each handgrip sample has some random variations from its pattern. Therefore, the study of handgrip signal belongs to stochastic signal processing. The theory of stochastic process has been extensively developed and is discussed in many books (Parzen, 1962; Yaglom, 1962; Papoulis, 1965; Cox and Miller, 1968; Grimmett and Stirzaker, 1982).

It is often convenient to describe a random process in terms of its statistical behavior. A fundamental statistical description of a random process is the probability distribution function. The probability distribution function $P(X)$, of a random signal $x(n)$ may be defined as the probability that $x(n)$ has a value less than X, which can be written arithmetically as $P(X)=Probability(x<X)$. For example, if a random variable x whose maximum value is 10 and minimum value is 0, then $P(10)=1$, and $P(0)=0$.

The more often used probability function is probability density function, which is the probability that the random variable have a particular value, $p(x_1)$=*probability*$(x=x_1)$. For discrete random process,

$$P(x_1) = \sum_{x=-\infty}^{x_1} p(x) \tag{3.7}$$

And for continuous random process,

$$P(x_1) = \int_{-\infty}^{x_1} p(x)\, dx \tag{3.8}$$

Average or mean (expected value) and variance are two most important parameters of a stochastic signal. The mean of N observations of a process x at time t is defined as

$$u_x(t) = E[x(t)] = \sum_{s=1}^{N} x_s(t) p[x_s(t)] \tag{3.9}$$

And the variance is defined as

$$\sigma_x^{\,2}(t) = E[x - u_x(t)]^2 \tag{3.10}$$

Another important parameter of a stochastic signal is its autocorrelation function, which is defined as

$$R_{xx}(t_1, t_2) = E[x(t_1)\, x(t_2)] \tag{3.11}$$

If the essential properties (the probability distribution function) of a process are unchanging over time, then it is said to be a stationary process. In other words, if the statistical properties of the system are same for all points in time then the process is stationary. However, in many cases the properties may change very little or very slowly

over time, in such case it may be regarded as wide-sense stationary process. A process is considered as wide sense stationary or weakly stationary if its mean value and variation are constants as $u(t)=u$, $\sigma(t)=\sigma$, and its autocorrelation function is a function only of the difference $\tau = t_1 - t_2$, i.e. $R_{xx}(t_1, t_2)=R_{xx}(\tau)$.

The handgrip signal is found as a non-stationary signal by studying its $u(t)$, $\sigma(t)$ and $R_{xx}(\tau)$. Non-stationary signals are much more difficult to analyze than stationary signals. Some methods discussed below can be used to transform original handgrip signal to a wide-sense stationary signal and do signal processing based on the transformed signal.

A non-stationary process, which usually contains either a trend or some cyclic changes, can be transformed to a stationary process by remove the trend or cyclic changes. A method proposed by Box and Jenkins (1970), which is particularly useful for removing a trend, is simply to difference a given process until it becomes stationary

$$y_t = x_{t+1} - x_t = \nabla x_{t+1} \tag{3.12}$$

First-order differencing is now widely used in economics. Occasionally second-order differencing is required

$$\nabla^2 x_{t+2} = \nabla x_{t+2} - \nabla x_{t+1} = x_{t+2} - 2x_{t+1} + x_t \tag{3.13}$$

A cyclic effect can also be eliminated by differencing. For example, if there is a cyclic effect with period of T, it can be removed by letting $y(t)=x(t+T)-x(t)$. Another way is to let the non-stationary process through a high-pass filter to eliminate the trend and a band-stop filter to eliminate the cyclic effects.

Consider a wide sense stationary process *x(n)* passing through a linear time invariant system having impulse response *h(n)*. Similar to a deterministic signal passing this LTI system, the input-output signal relationship is $y(n)=h(n)\otimes x(n)$. Therefore,

$$E[y(n)] = E[\sum_{m=-\infty}^{\infty} h(m)x(n-m)] = \sum_{m=-\infty}^{\infty} h(m)E[x(n-m)] = u_x \sum_{m=-\infty}^{\infty} h(m) = u_x H(0) \tag{3.14}$$

and

$$R_{yy}(n+\tau, n) = E[y(n+\tau)y(n)] = E([\sum_{m=-\infty}^{\infty} h(m)x(n+\tau-m)][\sum_{m=-\infty}^{\infty} h(m)x(n-m)])$$

$$= \sum_{m=-\infty}^{\infty} h(m) \sum_{k=-\infty}^{\infty} h(k)E[x(n+\tau-m)x(n-k)] = \sum_{m=-\infty}^{\infty} h(m) \sum_{k=-\infty}^{\infty} h(k)R_{xx}(\tau+k-m) \tag{3.15}$$

Substituting $n=m-k$ in the above equation, it turns to

$$R_{yy}(\tau) = \sum_{m=-\infty}^{\infty} R_{xx}(\tau-n) \sum_{k=-\infty}^{\infty} h(k)h(k+n) = R_{xx}(\tau) \otimes h(n) \otimes h(-n) \tag{3.16}$$

The Fourier transform of autocorrelation is called power spectral density function.

$$G(f) = \sum_{\tau=-\infty}^{\infty} R(\tau)e^{-j2\pi f\tau} \tag{3.17}$$

Therefore, in frequency domain for a LTI system,

$$G_{yy}(f) = G_{xx}(f) \mid H(f) \mid^2 \tag{3.18}$$

3.4 Wavelet Analysis

Fourier analysis breaks down a signal into constituent sinusoids of different frequencies. It also can be considered as a mathematical technique for transforming the signal from time-based to frequency-based. For many signals, Fourier analysis is extremely useful because the signal's frequency content is of great importance. However, Fourier analysis

has a serious drawback. In transforming to the frequency domain, time information is lost. When looking at a Fourier transform of a signal, it is impossible to tell when a particular event took place.

If the signal properties do not change much over time --- that is, if it is a stationary signal --- this drawback isn't very important. However, most biometric signals are generally non-stationary with significant events characterized by both their time location and frequency content, and Fourier analysis is not suited to detecting them.

This situation can be improved by using Fourier analysis to analyze only a small section of the signal at a time --- a technique called windowing the signal. Using such windowing technique produces the short time Fourier transform (STFT). The result of the windowing process is that STFT is better able to locate events in time, but as an unavoidable trade-off, has poorer frequency resolution. Another drawback of STFT is that once a particular size time window is chosen, that window is the same for all frequencies. Many signals require a more flexible approach so that the window size can be changed to determine more accurately either time or frequency.

Wavelet analysis represents the next logical step: a windowing technique with variable sized regions. Fourier analysis breaks up a signal into sine waves of various frequencies. Similarly, wavelet analysis breaks up a signal into shifted and scaled versions of the original (or mother) wavelet. A long, low scale wavelet gives poorer time location but better frequency location. While a short, high scale wavelet gives better time location but poorer frequency location. In this view, wavelet analysis is a transform whose coefficients detail the magnitude of the basis wavelets required to reconstruct the original signal.

A wavelet is defined as a waveform of effectively limited duration that has an average value of zero. Compare wavelets with sine waves, which are the basis of Fourier analysis. Sinusoids do not have limited duration --- they extend from minus to plus infinity. While sinusoids are smooth and predictable, wavelets tend to be irregular and asymmetric. Signals with sharp changes might be better analyzed with an irregular wavelet than with a smooth sinusoid. It also makes sense that local features can be described better with wavelets that have local extent.

The continuous wavelet transform (CWT) is defined as

$$CWT_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t)\psi^* \left(\frac{t-\tau}{s}\right) dt \tag{3.19}$$

Here, $\psi(t)$ is the transforming function, called the mother wavelet, and the asterisk stands for complex conjugate. Figure 3.3 shows some examples for $\psi(t)$.

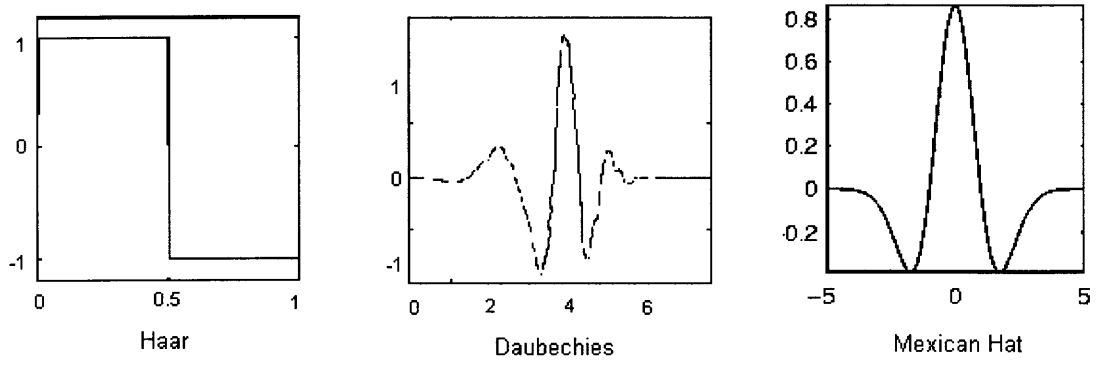


Figure 3.3 Mother wavelet examples.

In Figure 3.3, the first wavelet is called *Haar* wavelet, which is the simplest wavelet. The second one is a member of *Daubechies* wavelet family (Daubechies, 1988). The third one is a symmetric wavelet, called *Mexican Hat* wavelet.

As seen in the above equation, the transformed signal is a function of two variables, time variable τ and scale parameter s, both varying continuously. The scale parameter in the wavelet analysis is similar to the scale used in maps. In maps, a high scale corresponds to a global view, and a low scale corresponds to a detailed view. In wavelet, high scales correspond to low frequency components which indicate the trends of signals, and low scales corresponds to high frequency components which indicate the local details of signals. Figures 3.4 and 3.5 show a base wavelet after different s and τ applied.

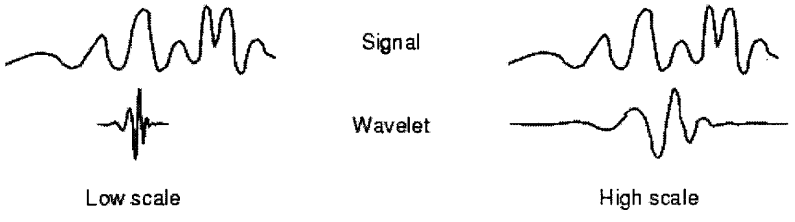


Figure 3.4 Scale aspects.

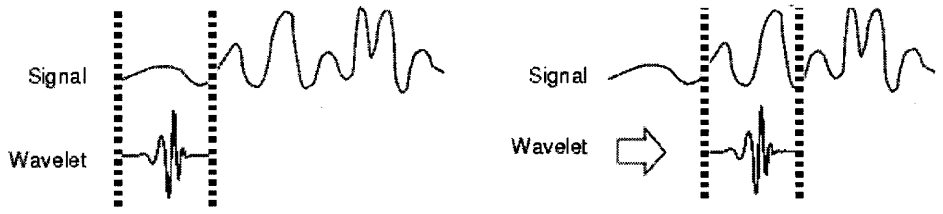


Figure 3.5 Time aspects.

The wavelet transform $CWT(\tau,s)$ gives the information of $x(t)$ at different levels of resolution and also measures the similarity between the signal $x(t)$ and each son wavelet $\psi(\tau,s)$ by convolution. This implies that a wavelet can be used for feature discovery if the wavelet used is similar to the feature components hidden in the signal. That is why the wavelet analysis is used to extract features from handgrip signals.

Calculating wavelet coefficients at every possible scale needs an amount of computation, and produces a lot of data, which are highly redundant to reconstruct the original data. To overcome this problem, Mallat (1989) developed a pyramidal algorithm that only do computation on scales and positions of powers of two, so-called dyadic scales and positions. This algorithm is same accurate with CWT while much more efficient. This is also called discrete wavelet transform (DWT).

The main idea is the same as it is in the CWT. The difference is that DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into approximations (high scale, low frequency information) and details (low scale, high frequency information). The procedure starts with passing the signal (sequence) through a half-band digital low-pass filter. After passing the signal through a half band low-pass filter, half of the samples can be eliminated according to the

Nyquist rule. Simply discarding every other sample will downsample (subsample) the signal by two, and double the scale of the signal. The decomposition process can be iterated, with successive approximations being decomposed in turn, so that one signal is broken down into many lower resolution components. This is called wavelet decomposition tree as shown in Figure 3.6.

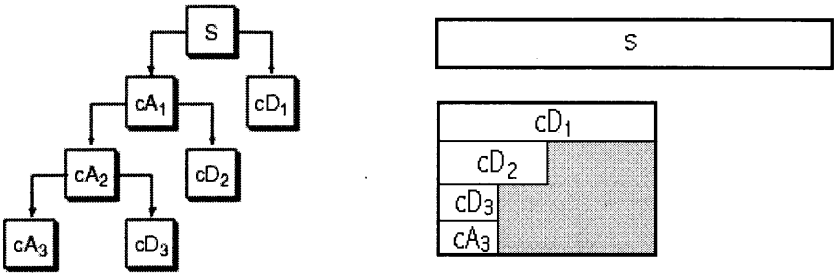


Figure 3.6 Wavelet tree.

Handgrip signal processing have benefited from wavelet transforms for denoising and data reduction (compression). Most important data is in low frequency area, therefore, noise can be eliminated by discarding high frequency (low scale) coefficients. Usually, the information that is most prominent in the original signal will appear as high amplitudes in that region of the DWT. By setting a threshold, and discarding all values in the DWT that are less than this threshold, can extremely reduce the data size and delete the noise.

3.5 Summary

In this chapter, a general DSP model is introduced. Some stochastic signal processing methods are also presented in both time domain and frequency domain. To overcome the disadvantage of losing time information in Fourier transform, wavelet analysis is developed and discussed. Signals with sharp changes can be better analyzed with an irregular wavelet than with a smooth sinusoid. It also makes sense that local features can be described better with wavelets that have local extent.

This chapter focuses on the general signal processing to the handgrip signal, such as denoising and feature extraction. However, this chapter does not cover the recognition part. How to recognize the user will be discussed in the next chapters.

CHAPTER 4

PATTERN RECOGNITION ALGORITHMS FOR HANDGRIP IDENTIFICATION

There are a number of publications on technologies for pattern recognition. However, not all of them are useful for the handgrip recognition. In this chapter, some algorithms related to handgrip recognition are briefly discussed. First, the concept of pattern recognition is introduced to help people to understand how a pattern recognition system works. Then two parametric pattern recognition algorithms are introduced. This research will focus on non-parametric algorithms because it is hard to estimate the parameters of the handgrip. Several non-parametric clustering algorithms are introduced, and these algorithms will be applied to the static analysis and dynamic analysis in the following chapters.

4.1 Pattern Recognition System

The goal of pattern recognition is to classify objects into a number of categories or classes (Theodoridis and Koutroumbas, 1999). A pattern is the description of an object.

Basically, a pattern recognition process can be divided into three phases: data acquisition, data preprocessing, and classification. In the data acquisition phase, a set of measured data is gathered from the physical world through transducers. The measured data are then used as the input to the second phase (data preprocessing) and grouped into a set of feature vectors. In the third phase, the classifier classifies the feature vectors into different classes based on some statistic criteria.

In the case of the handgrip pattern recognition system, the handgrip signal is sampled at discrete time point t_1, t_2, ... t_n. A pattern vector can be formed by as $x_1=f(t_1)$, $x_2=f(t_2)$, ..., $x_n=f(t_n)$. The pattern vectors can be denoted as $X=(x_1, x_2, ..., x_n)'$, where the prime(') indicates transposition. It is often useful to think of a pattern vector as a point in an n-dimensional Euclidean space. The set of patterns belonging to the same class corresponds to an ensemble of points scattered within some region of the measurement space. A 2D example is shown in the figure below. However, it is not always able to specify measurements that will result in neatly disjoint sets. In these cases, some statistic methods are needed to find the correct boundaries.

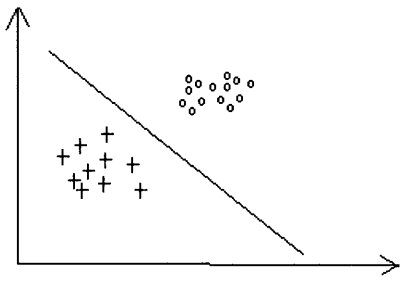


Figure 4.1 Two disjoint pattern classes.

In the above figure, the straight line is also known as decision line, which divides the feature space into either class A or class B. If a vector x falls into region A, then it is classified as class A. The classification may not be correct. In such a case, a misclassification error occurs. To minimize such errors, the decision line must be optimized.

Pattern recognition algorithms are often classified as either parametric or non-parametric. A parametric approach is to define the discriminant function by a class of probability densities, which are further defined by a relatively small number of

parameters. These parameters are known or can be estimated from the training data. Bayes classifier and linear discriminant function are two examples of parametric methods. Non-parametric methods are for the recognition tasks where no probability distribution are known or need to estimate. Nearest neighbor and clustering are two examples of non-parametric methods.

4.2 Data Preprocessing

In most cases, the input data need to be preprocessed. The first step is outlier removal. An outlier is defined as a point that lies very far from the mean of the corresponding random variable. This distance is measured with respect to a given threshold, usually a number of times the standard deviation. For a normally distributed random variable, a distance of two times the standard deviation covers 95% of the points, and a distance of three times the standard deviation covers 99% of the points. Points with values very different from the mean value produce large errors during training and may have disastrous effects. If the number of outliers is very small, they are usually discarded.

The second step of preprocessing is data normalization. In many practical situations a designer is confronted with features whose values lie within different dynamic ranges. Thus, features with large values may have a larger influence in the cost function than features with small values. The problem is overcome by normalizing the features so that their values lie within similar ranges. A straightforward technique is making all features to zero means and unit variances.

The absolute amplitude of handgrip signal may have big variations in different cases. For example, the hand pressure will be much weaker when a person is injured. If

the absolute value is used to do pattern recognition, the gun will malfunction in these situations. However, the relative pressure waveform will not change much in such cases. If there is an up trend at a particular time point in a regular handgrip signal, this trend will also appear at this time point even this person is injured, only the absolute value of the peak turns smaller. Normalizing the handgrip signal can adapt the handgrip pattern recognition system to overcome such problems.

The measurement space is usually of high dimensionality. These measurements inevitably contain information either redundant or irrelevant to the classification task. Moreover, the transducers are likely to introduce some distortion and noise. Direct use of all measurements not only extremely increases the computation complexity, but also may introduce some errors. The pattern recognition is usually performed based on a very few of the most important attributes. The process of selecting these most useful attributes from the input data is called feature selection or feature extraction.

The major task of feature selection is that: given a number of measurements, how to select the most important aspects so as to reduce the dimensionality and at the same time retain as much as possible of their class discriminatory information (Theodoridis and Koutroumbas, 1999). In a more quantitative description, features leading to large inter-class distance and small within-class distance should be selected.

There are three different methods of selecting a "best" set of m variables from n variables (James, 1985).

(1) Complete subsets: This is the most direct method by searching every possible subset of m variables and calculating the error rate on each to find the best one. Although

this procedure is guaranteed to find the best m variables, it costs too much computation.

(2) Stepwise forward: This is the most common and well-known method. First, find the single variable that minimizes the classification error rate. Then, find the variable which, paired with the previous selected variable, minimizes the error rate. Proceed in the same way to find the other variables.

(3) Stepwise backward: The stepwise backward method works like the stepwise forward method, except that it starts with all variables, and at each step, discard the variable which results in the highest error rate.

4.3 Linear Classifier

The linear classifier assumes that the patterns can be classified to different classes based on a linear function in the *n*-dimensional space

$$G(X)=a_1x_1+a_2x_2+\ldots+a_nx_n+a_0=0 \tag{4.1}$$

where $A=(a_1, a_2, \ldots a_n)$ is the weight vector and a_0 is the threshold. Let $G(X)>0$ if X belongs to class ω_1, let $G(X)<0$ if X belongs to class ω_2. In Figure 4.1, $G(X)$ can be referred as the line, which discriminates the two classes in a 2D-dimension space.

Now the major concern of linear discriminant function is the estimation of the weights. Many algorithms have been proposed to estimate the weights. Perceptron algorithm is the most widely used algorithm to estimate these parameters.

The perceptron algorithm (Rosenblatt, 1958; Minsky and Papert, 1988; Gallant, 1990) uses the reward-punishment concept. Given two training sets belonging to classes

ω_1 and ω_2, and let $A(1)$ be the initial weight vector which can be chosen randomly. Remember $G(X)>0$ when $X\in\omega_1$, and $G(X)<0$ when $X\in\omega_2$. At the kth training step:

If $X\in\omega_1$ while $G(X)<0$, let $A(k+1)=A(K)+c(k)X(k)$, where c is the correction increment.

If $X\in\omega_2$ while $G(X)>0$, let $A(k+1)=A(K)-c(k)X(k)$, where c is the correction increment.

Otherwise, let $A(k+1)=A(k)$.

Simply stated, the algorithm makes a change in the weight vector if and only if the pattern is misclassified. The simplest calculation of $c(k)$ is setting it as a constant. The gradient also can be used as the correction increment to faster the success of the above algorithm, because the gradient gives the rate of change of the function in the direction of that component: $c(k)=Grad\ G(X_k)=dG/dX_k=(\partial g/\partial x_1,\ \partial g/\partial x_2,\ ...,\ \partial g/\partial x_n)'$

Linear classifier is a simple pattern classification model. It is easy to understand, and the parameters can be estimated quickly. The best thing is that after the model is built completely, classification can be done very fast. Calculating a simple equation can save much valuable time for a real time work. However, many patterns are not linearly classifiable. Handgrip pattern is such a case. Therefore, linear classifier is not an appropriate algorithm for handgrip pattern recognition.

4.4 Bayes Decision Theory

Bayes classifier is another parametric classification algorithm. Let ω_1 and ω_2 be the two classes in which the patterns belong. If N is the total number of available training patterns, and N_1, N_2 of them belong to ω_1 and ω_2 respectively, then $P(\omega_1)=N_1/N$, $P(\omega_2)=N_2/N$. The other probability density functions required are $P(x|\omega_1)$ and $P(x|\omega_2)$. $P(x|\omega_i)$ where $i=1,2$ is referred as the likelihood function of ω_i with respect to x.

There is a Bayes rule in probability and statistic science. $P(B|A)=P(B,A)/P(A)$, and $P(B,A)=P(A,B)$, therefore $P(B|A)P(A)=P(A|B)P(B)$. By using Bayes rule to $P(x|\omega_i)$, $P(\omega_i|x)=P(x|\omega_i)P(\omega_i)/P(x)$ where $P(x)=\sum P(x|\omega_i)P(\omega_i)$. The Bayes classification rule can now be stated as (Fukunaga, 1990): if $P(\omega_1|x)>P(\omega_2|x)$, x is classified to ω_1, if $P(\omega_1|x)<P(\omega_2|x)$, x is classified to ω_2.

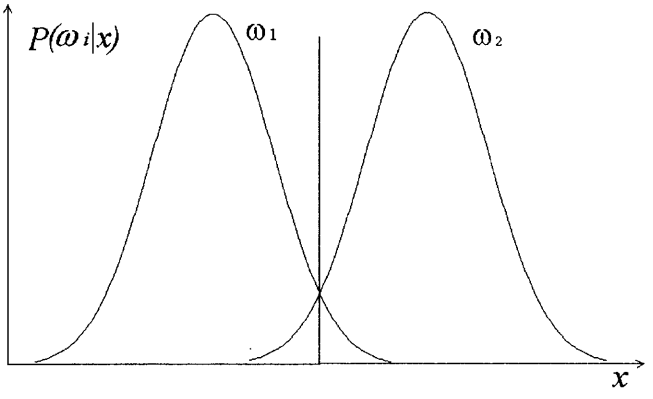


Figure 4.2 Bayes decision rule.

To extend to the tasks of classifying M classes, the M conditional probabilities $P(\omega_i|x)$ are calculated. Then the unknown pattern is assigned to the class corresponding to the maximum conditional probability.

Bayes classifier classifies the test sample to its most likely belonged classes. The classification error is the least in all pattern recognition algorithms. However, it requires the posteriori possibility distribution functions to be known in advance. In practice, there is no appropriate statistical model for the handgrip signal, and the posteriori possibility distribution function of handgrip signal is very difficult to estimate. Bayes classification is not suitable for handgrip pattern recognition.

4.5 Nearest Neighbor Rule

The nearest neighbor rule measures the similarity between samples by the distance among them. By nearest neighbor rule, sample x is considered to belong to class A if x is nearer to class A than to class B, as shown in figure below.

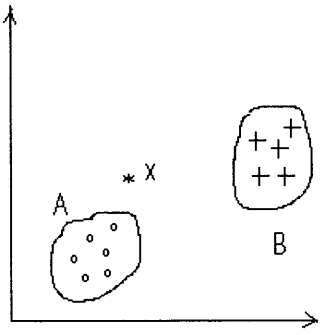


Figure 4.3 Nearest neighbor example.

The basic idea behind the nearest neighbor rule is that samples which fall close together in features space are likely either to belong to the same class or have the same posteriori possibility distributions of their respective classes.

Various distance functions have been proposed to measure the similarities among patterns. In the work of Hastie and Tibshirani (1996), an effective metric is suggested exploiting the local information at each point. Two simplest distance functions are the Manhattan distance and the Euclidean distance. If x and y are two feature vectors with n components, the Manhattan distance between x and y is given by

$$d(x,y) = \sum_{i=1}^{n} | x_i - y_i | \tag{4.2}$$

And the Euclidean distance is given by

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4.3}$$

Both distance functions are derived from norms. In general, the distance corresponding to the so-called L_p norm is given by

$$d(x,y) = (\sum_{i=1}^{n} | x_i - y_i |^p)^{1/p} \tag{4.4}$$

Thus the Manhattan distance corresponds to the L_1 norm and the Euclidean distance to the L_2 norm.

Nearest neighbor rule is the simplest non-parametric classification algorithm. The disadvantage of nearest neighbor rule is that it needs to calculate the distances between the test samples within all clusters to find the minimum one and this calculation costs some time. However, nearest neighbor rule is still the best choice as a non-parametric algorithm for the handgrip pattern classification. This rule will be applied to this research work in the following chapters.

K-nearest neighbor rule expands nearest neighbor rule to *K* predefined classes. It is described below:

Step 1. Arbitrarily choose *K* initial cluster centers $c_j(1), j=1, 2, .., K$.

Step 2. Distribute the samples in the data set to the *K* clusters based on the distance to each cluster center by nearest neighbor rule.

$x \in C_j(n)$ if $d[x, c_j(n)] < d[x, c_i(n)]$, for all $i=1, 2, ..., K, i \neq j$.

Step 3. From the result of Step 2, compute the new center such that the sum of the squared distances from all points in the cluster to the new center is minimized. The new center can be simply calculated by the mean of all samples in the cluster.

$c_j(n+1)=\sum x/n_j$ where n_j is the number of samples in the cluster C_j.

Repeat Step 2 and Step 3 until the cluster distribution keeps same, i.e., if $C_j(n+1)=C_j(n)$, then the procedure can be terminated.

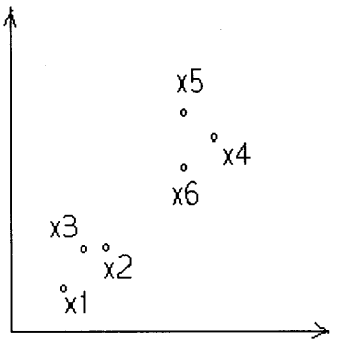


Figure 4.4 Sample patterns used in illustrating the K-nearest neighbor algorithm.

As a simple example, let us examine the samples shown in Figure 4.4, where x_1=(1,1), x_2=(2,2), x_3=(1.5,2), x_4=(5,5), x_5=(4,6), x_6=(4,4). In the first step, two clusters are predefined and x_1, x_2 are chosen as the initial centers, c_1=(1,1), c_2=(2,2).

In Step 2, x_3, x_4, x_5 and x_6 are distributed into x_2's cluster, i.e., $C_1=\{x_1\}$, $C_2=\{x_2, x_3, x_4, x_5, x_6\}$. The new center c_1=(1,1), c_2=(3.3,3.8) is calculated in Step 3. Repeat Step 2 to distribute the samples based on new centers, there are 2 clusters now, $C_1=\{x_1, x_2, x_3\}$, $C_2=\{x_4, x_5, x_6\}$. When repeating the above steps, the result is same. Therefore, the algorithm terminates here.

K-nearest neighbor rule does not require the posteriori possibility distribution function, and it is simple to implement. However, there are disadvantages. First the parameter *K* need to be pre-assumed. If *K* is not chosen properly, the algorithm will give wrong answers. Second, the order in selecting initial centers will affect the final classification result.

4.6 Clustering Analysis

In the last three sections, the algorithms of how to classify a test sample into pre-built classes have been discussed. From this section, some algorithms of how to build these classes will be discussed. This process is also called clustering. The objective of clustering analysis is to partition a set of objects into disjointed groups or clusters so that the objects in the same cluster are very similar, while objects is different clusters are dissimilar (Anderberg, 1973).

Many definitions of clustering have been proposed (Johnson, 1967; Wallace and Boulton, 1968; Everitt, 1993). Most of these definitions are based on proximity or dissimilarity. The simplest way to measure dissimilarity is the distance function. For the distance between two points, Manhattan distance and Euclidean distance are two mostly widely used distance functions. For the distance between two sets, four proximity measures, the max proximity function, the min proximity function, the mean proximity function and the average proximity function, are widely used (Duda and Hart, 1973). The max proximity function can be defined as $D_{max}(X,Y)=max\{d(x,y)\}$, where $x \in X$, and $y \in Y$. The min proximity function can be defined as $D_{min}(X,Y)=min\{d(x,y)\}$, where $x \in X$, and $y \in Y$. The mean proximity function can be defined as $D_{mean}(X,Y)=d(m_x,m_y)$, where m_x is mean value of all x in X, and m_y is the mean value of all y in Y. The average proximity function can be defined as

$$D_{avg}(X,Y) = \frac{1}{N_x N_y} \sum_{x \in X} \sum_{y \in Y} d(x,y) \tag{4.5}$$

where N_x is of number of samples in X, and N_y is of number of samples in Y.

Clustering algorithms can be divided into three main categories: sequential algorithms, agglomerative algorithms, and divisive algorithms. The latter two also can be called as hierarchical clustering algorithms.

The basic sequential clustering algorithm works as below (Hall, 1967):

Step 1. A cluster center c_1 is chosen arbitrarily.

Step 2. For the next sample x in the dataset, find C_k so that $d(x,C_k)=min\{d(x,C_i)\}$, if $d(x,C_k)$>threshold, then let x to be a new cluster, otherwise, merge x into C_k and update C_k's center.

Repeat Step 2 until all samples have been considered.

The drawback of the above algorithm is that the order of the samples presented plays an important role in this algorithm. Different presentation orders will lead to different results. For that reason, a modification algorithm has been proposed. This modified sequential clustering algorithm also can be called maxmin algorithm.

In the first step of maximin-distance algorithm, a cluster center c_1 is chosen arbitrarily. Then find the farthest sample from c_1, and make this sample as the second cluster center c_2. In the next step, compute the distance from each remaining sample to c_1 and c_2, save the minimum value for each pair of distances. If the maximum of these minimum value is greater than a fraction of distance $d(c_1,c_2)$, then choose it as the third center c_3. Repeat this step until no new centers found.

However, maximin-distance algorithm highly depends on the choice of the threshold, i.e., the fraction of distance $d(c_1,c_2)$, and the choice of the first center c_1,. If these two parameters are not chosen properly, this algorithm may fail.

The hierarchical algorithms work in another way. The general agglomerative algorithm works as below: in the first step, consider every sample in the dataset as a cluster; then find C_a and C_b so that $D(C_a,C_b)=min\{D(C_i,C_j)\}$, if $D(C_a,C_b)<$threshold, merge C_a and C_b to be a new cluster, and repeat above algorithm, otherwise, stop this algorithm.

Divisive algorithm follows the reverse strategy from the agglomerative algorithms. In the first step, it considers all samples in one cluster. Then find the maximum distance between two points in this cluster, and divide the original cluster into two clusters centering at these two points. In the meantime, assign all points into these two new clusters based on nearest neighbor rule. In the second step, calculate all distances $d(x,y)$ where x and y belongs to one same cluster C_i, $i=1, 2, \ldots n$, n is the existing cluster number in current step. Find a and b so that $d(a,b)=min\{d(x,y)\}$, $a\in C_k$ and $b\in C_k$. If $d(a,b)>$threshold, then divide C_k into two clusters centering at a and b, and assign all points in C_k into these two new clusters based on nearest neighbor rule. Repeat Step 2 until $d(a,b)<$threshold, terminate this process.

Since the divisive algorithm need calculate distances between any pair points in all clusters, its computation is very demanding. This is the main drawback of divisive algorithm compared with the agglomerative algorithm. Thus, some computation simplification is required for any use in practices (Gowda and Ravi, 1995; MacNaughton-Smith, 1964).

4.7 Density Search Techniques

If individuals are depicted as points in a metric space, a natural concept of clustering suggests that there should be parts of the space in which the points are very dense,

separated by parts of low density (Everitt, 1993). Each high dense region generally is taken to signify a different group.

Carmichael (1968 and, 1969) describes a method that attempts to search for continuous and relatively densely populated regions of the space, surrounded by continuous and relatively empty spaces. The distance between groups is defined as that of the closest pair of individuals, where only pairs consisting of one individual from each group are considered.

Each point is considered as a cluster at the first step, and distances between all clusters are calculated and given in a symmetric matrix like below.

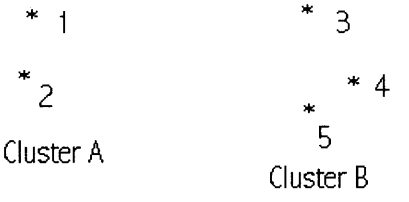


Figure 4.5 Sample patterns used in illustrating the density search algorithm.

$$D_1 = \begin{vmatrix} 0 & & & & \\ 1 & 0 & & & \\ 5 & 6 & 0 & & \\ 7 & 7.5 & 1.5 & 0 & \\ 6 & 5 & 2 & 1 & 0 \end{vmatrix} \qquad (4.6)$$

In D_1, the distance between node 1 and 2 is smallest, therefore, node 1 and 2 are merged into one cluster as cluster *A*. Then, the distances between cluster *A* with the other points are recalculated.

$$D_2 = \begin{vmatrix} 0 & & & \\ 5 & 0 & & \\ 7 & 1.5 & 0 & \\ 5 & 2 & 1 & 0 \end{vmatrix} \tag{4.7}$$

In D_2, the distance between node 4 and 5 is the minimum, therefore, node 4 and 5 are merged into cluster *B*. Continue the above algorithm and add one node into a cluster in each step until cluster *A* containing node 1 and 2, and cluster *B* containing node 3, 4 and 5. Now the distance matrix is

$$D = \begin{vmatrix} 0 & \\ 5 & 0 \end{vmatrix} \tag{4.8}$$

Here, the distance between cluster *A* and *B* is much bigger than the distance between node 3 and cluster *B* (containing node 4 and 5) in the last step, which indicates a discontinuity. Therefore, it stops here and gives two natural clusters *A* and *B*. The only parameter that needs to be preset is the stop distance.

Wishart (1969) proposed another density search algorithm called Mode analysis algorithm. Initially, the kernel density at each observation is estimated by

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x_i - x}{h}) \tag{4.9}$$

where $K(u)$ denotes a so-called kernel function, and h denotes the window size. For the uniform kernel, $K(u)=1/2\ I(|u|<1)$, and the density function is $f(x)=1/2nh$ {the number of observations fall in $[x-h,\ x+h]$}. The Uniform kernel implies that all observations in the neighborhood are given the same weight. It is reasonable to attribute more weights to observations that are near to the kernel and less weight to distant observations. The

triangle kernel density function has such property. The triangle kernel is defined as $K(u)=(1-|u|)\ I(|u|<1)$.

The initial window size is the smallest radius that containing a specific number of observations in a hyper-sphere centered at this observation. For each observation, find the nearest neighbor with a greater estimated density. If such a neighbor exists, join the cluster, which the observation belongs with, to the cluster, which the specified neighbor belongs with (Gitman, 1973; Huizinga, 1978). Cascaded density estimates are obtained by computing initial kernel density estimates and then, for each cluster, taking the arithmetic mean of the initial density estimates of the observations within the neighborhood. The cascaded density estimates can, in turn, be cascaded, and so on. Let f_i^k be the density estimate at x_i cascaded k times, then at times of k+1,

$$f_j^{k+1} = \sum_{j \in N_i} \frac{f_j^k}{n_i} \tag{4.10}$$

Keep the searching and emerging process, until the cluster density achieves its maximum value. At this time point, the whole dataset would be divided into some high dense areas, separated by some low dense areas. Each high dense area represents a natural group.

4.8 Canonical Analysis

The objective of canonical analysis is to derive a linear transformation that will emphasize the difference among the pattern samples belonging to different classes. Canonical analysis also can be used to determine the importance of each variable discriminating two or more groups of objects by comparing the magnitudes of the linear transformation's coefficients. The canonical analysis will be used to find which finger

variable give most contribution to the cluster classification. Then the sensor locations can be set based on the finger variable importance.

If there are p variables and q classes, there will be q-1 or p (usually $p>q$-1) canonical discriminant functions. Each canonical discriminant function is a linear combination of variables, i.e., a linear transformation of original variables: $f=u_0+u_1x_1+u_2x_2+...+u_px_p$, where u_i is the canonical coefficients. Bigger canonical coefficient also means more importance of the corresponding variable.

The coefficients for the first canonical discriminant function are derived so as to maximize the differences between the group means. The coefficients for the second canonical discriminant function are derived to maximize the difference between the group means, subject to the constraint that the values on the second canonical discriminant function are not correlated with the values on the first canonical discriminant function, and so on. In other words, the second canonical discriminant function is orthogonal to the first, and the third canonical discriminant function is orthogonal to the first and the second canonical discriminant function, and so on.

Let $\boldsymbol{X} = [x_{ij}]$ be the data matrix with n rows (individuals or observations) indexed by i and p columns (variables) indexed by j. The overall mean of variable j is written as: $x'_j=\sum x_{ij}/n$. Assume there are n_k samples in each group, $\sum n_k=n$. Let $x'_{kj}=\sum x_{ij}/n_k$. For every variable j, there is $x'_j=\sum x'_{kj}*n_k/n$. The total covariance between two variables a and b can be calculated as:

$$Covariance(a,b) = 1/n \sum_{i=1}^{n} (x_{ia} - x'_a)(x_{ib} - x'_b) \tag{4.11}$$

The above equation can be reduced to

$$Covariance(a,b)=1/n \sum_{k=1}^{q} \sum_{i \notin k} (x_{ia} - x'_{ka}) (x_{ib} - x'_{kb}) + \sum_{k=1}^{q} n_k/n (x'_{ka} - x'_{a}) (x'_{kb} - x'_{b}) \quad (4.12)$$

This means that the total covariance (T) equals to the sum of the covariance within the groups (W) and the covariance between the groups (B), i.e., T=B+W.

The canonical discriminant function with the desired properties can be derived, for which it would be necessary to solve the following simultaneous equations:

$$\sum b_{1i}v_i=\lambda\sum w_{1i}v_i, \sum b_{2i}v_i=\lambda\sum w_{2i}v_i, \ldots, \sum b_{pi}v_i=\lambda\sum w_{pi}v_i, \text{ and } \sum v_i^2=1, \quad (4.13)$$

where λ is a constant, called the eigenvalue, v's are a set of p coefficients.

Each solution, which yields its own λ and the set of v's, corresponds to one canonical discriminant function. The canonical discriminant function with the largest λ value is the most powerful discriminator, while the function with the smallest value is the weakest. Usually, the first two canonical discriminant functions can explain about 99% differences. Therefore, only the first two canonical discriminant functions are useful in reality.

4.9 Classification Tree

Similar to discriminant analysis, classification trees are also used to predict group membership of an object. A classification tree is an unbalanced binary tree structure that has one root node at its top and multiple leaf nodes at its base. Each node of the tree contains a test condition (or "split") for one of the object's variables. To classify an object, a path is followed from the root node to a particular leaf node which contains the predicted group of the object. The particular path through the tree is defined by the test conditions of the internal nodes. At any one node the test condition (e.g. Thumb X>50?)

is evaluated, if true the right child node is selected otherwise the left child is taken and so on until a leaf reached.

Splitting rule gives how the splits are decided. A widely used technique is to measure the impurity of the parent and its children. Consider that each sample in the training data set has been assigned to a leaf. At ith stage, assume n_i is the total sample size for this leaf, and n_{ik} is the size of samples belonging to class k. Then, at this leaf there is a probability distribution $p_{ik}=n_{ik}/n_i$. The impurity is defined as:

$$D = -2 \sum_i \sum_k n_{ik} \log(p_{ik}) \tag{4.14}$$

The splitting strategy is to choose the attributes which minimize the above deviance.

4.10 Recognition and Verification for Smart Gun

Sometimes verification and recognition are interpreted as similar terms but they have two distinct meanings. Recognition occurs when an individual's characteristic is being selected from a group of stored data. Usually, it deals with clustering the data set into several groups. Biometric devices that implement recognition techniques can be quite time consuming. Often it requires from 5 to 15 seconds or more to identify the appropriate individual.

In another hand, verification only deals with the comparison between the input data and the data stored for the specific user. It does not need to search the whole data set to find a matching. Therefore, it is much faster than recognition. The question to be answered in verification is, "Are you who you say you are?" instead of, "Do I know who you are?"

For the smart gun project, both recognition and verification are needed. In most cases, verification is enough because only one user (the owner) is allowed to fire the gun. However, a group of people may have authorization to use this gun in some cases. Such as, police officers or a team of soldiers need exchange their guns among the partners. In such cases, recognition technique is needed. All algorithms discussed in this chapter are for recognition. However, they can be used for verification too.

4.11 Summary

This chapter covers the pattern recognition part of the handgrip recognition project. A number of pattern recognition algorithms have been discussed. Since the parameter pattern recognition algorithms need the posteriori possibility function about pressure distribution, which is not provided and hard to estimate, the focus is put on non-parameter pattern recognition techniques.

It is believed that the handgrip signal is a recognizable biometric pattern. But this needs to be proven. To collect the accurate handgrip signal, a number of pressure sensors need to be put on the gun handle. How many sensors are needed? Which places are the best positions to put these sensors? These questions will be answered in the next two chapters.

CHAPTER 5

FINGERTIP POSITIONS ANALYSIS

The physical dimensions of a human hand contain information that is capable of authenticating the identity of an individual. In fact, each human hand is unique. Finger length, width, thickness, shapes and relative location of these features distinguish every human being from every other (Miller, 1971, Sidlauskas, 1988).

The fingertip placement on the gun handle is affected by human hand geometry and human grasping behavior simultaneously. The hand geometry uniqueness does not guarantee the fingertip placement recognizable for everybody. Therefore, an experiment was performed to investigate the placement of a person's fingertips on the handle of a replica gun. The experiment was conducted to answer the following questions:

1. Is the fingertip placement of one subject repeatable enough so that it can be used as a biometrics pattern?
2. Is the difference of the fingertip placement between two subjects more than the variation of one subject?
3. If the answer of above two questions are "Yes", where are the best places to locate sensors on the gun handle, i.e., which points can separate most subjects?

From the preliminary statistical analysis of 160 subjects, it appears that the fingertip placement of one subject is repeatable and there could be enough variation between subjects. The fingertip placement can be used as a biometrics pattern.

5.1 Fingertip Positions Acquisition

A digital camera was used to take the picture when a subject was asked to grasp and hold a replica handgun. To ensure pictures for all subjects have the same scene geometry, the camera and replica gun were both securely fixed to a flat board attached to a tripod. The tripod could be raised or lowered so the gun was held level to the shoulder of the subject.

Three points on the gun handle were defined as reference points. These three points can be used as coordinates to transform a distance measured by pixel number to a distance in mm. This computation is done by comparing the pixel number and corresponding scene distance in mm for the edges of the known triangle defined by points G1, G2 and G3 which are shown in Figure 5.1 and G1 is defined as the point where X=0, Y=0. The approximate image scaling for the plane of the gun handle is one pixel corresponding to an area of about 0.31×0.26 mm^2 in the real scene.

In order to find the center points of fingernails, colored circular markers were placed at the center of the fingernail for the middle, ring, pinkie and the thumb for each subject. Figure 5.2 shows the digital image of how a subject grasped the replica gun. The four markers as well as the detected fingertip position can be seen as shown by the crosshairs. Every subject was asked to grasp the replica gun, hold, release and re-grasp for a total of five times and five pictures were taken. Each picture was then sent to the image processing routing to produce the detected fingertip locations.

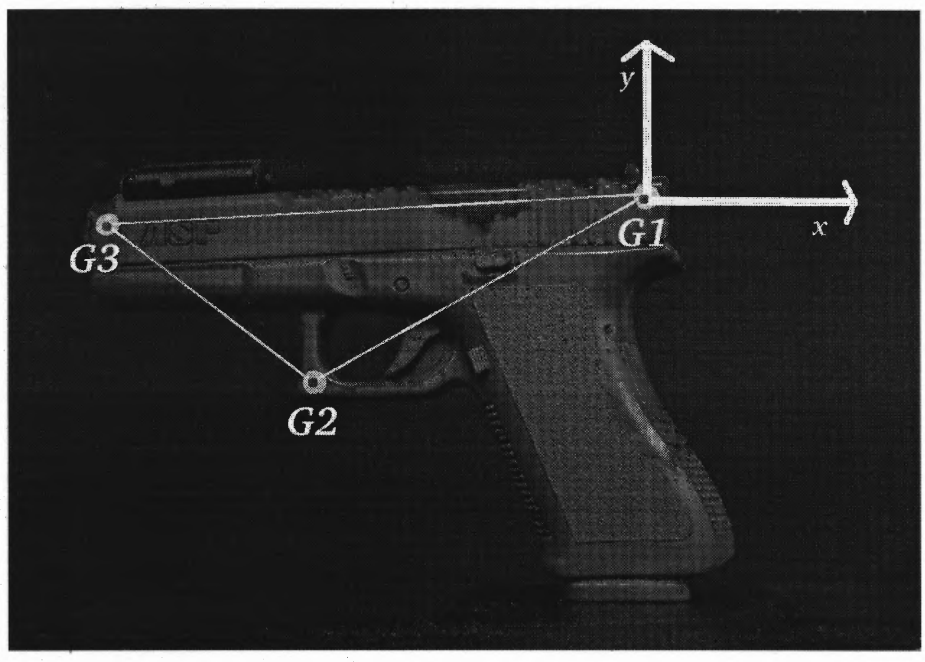


Figure 5.1 The reference frame used to define the fingertip coordinates.

Figure 5.2 The measured fingertip positions are shown by crosshairs.

There were 160 subjects (70% male, 30% female) that participated in this experiment. The age of these subjects range varies from 18 to 62 years, though the majority of subjects were NJIT students between the ages of 18 and 30 years. Most subjects (90%) were right handed. Every subject was asked if he/she had any previous experience with firearms. About 77% subjects claimed no firearm experience, while the remaining proportion of subjects did.

There are 16 police officers among the 160 subjects, 87.5% male versus 12.5% female, with age varying from 23 to 58 years old. Among the tested police officers, 80% officers are under 45 years of age. Most police officers (94%) are right handed.

The police officers in this study had between 1.5 and 30 years of firearm experience. About 37% of them have around 15 years firearm experience, while the remaining proportion have an approximate uniform distribution. When the police officers were asked about their prefer grip patterns, 75% of police personnel subjects claimed to use their both hands while firing a weapon (25% use only one). In addition, 43.8% of the officers stated they used the "cup and saucer" grip with the gun cradled in the opposite hand, while 25% stated they held the support hand level with the weapon.

(a) An inexperienced subject.

(b) A police officer.

Figure 5.3 Fingerprint positions for two different subjects for all five trials.

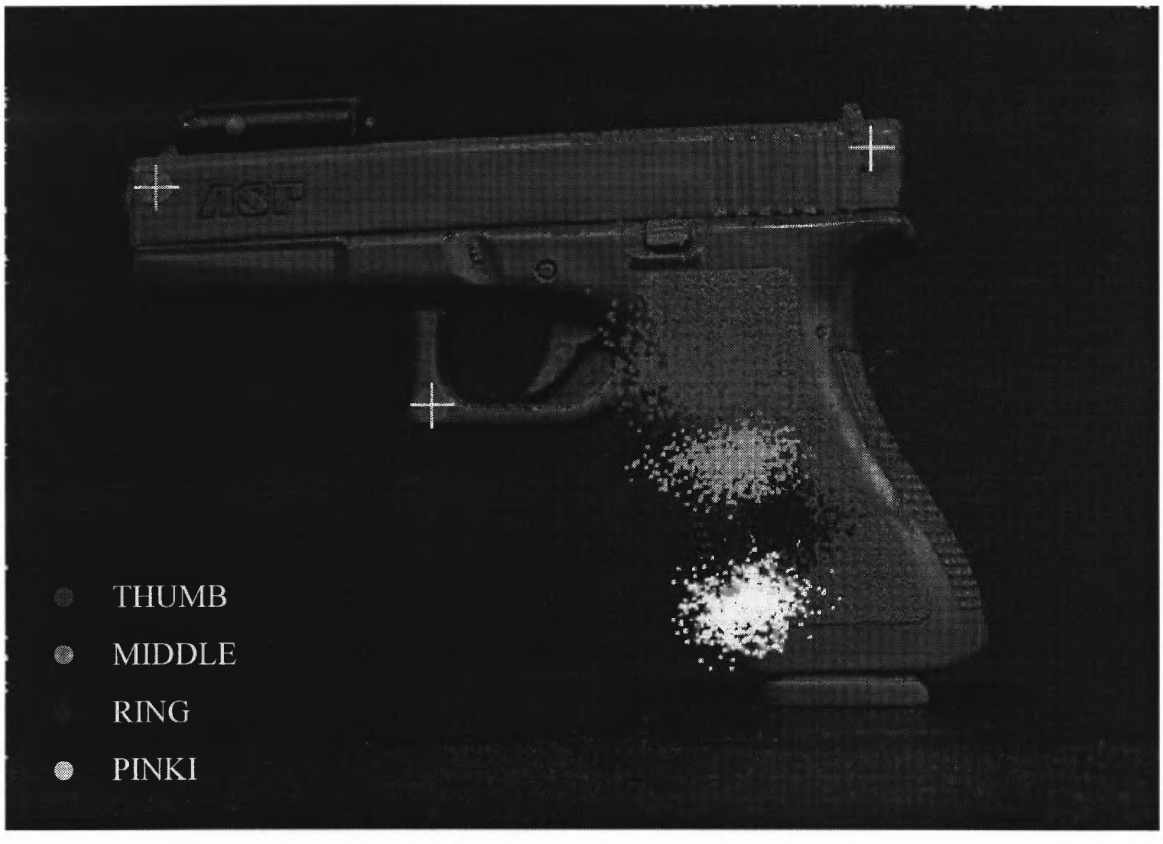


Figure 5.4 Fingertip positions for all trials of all subjects.

5.2 Fingertip Position Distribution

Example sets of fingertip positions for two subjects are shown in Figure 5.3. Figure 5.3(a) is the fingertip positions of an inexperienced subject drawn from the NJIT community. Figure 5.3(b) is the fingertip positions of a police officer. By comparing these two figures, it is obvious that an experienced user has tighter grouping of fingertip locations across the repeated trials. The complete data set showing all trials (818 in total) for all 160 subjects is shown in Figure 5.4. For a number of subjects with small hands, some of the fingertips were measured as being off the gun handle. These trials were removed from the data set to give a total of 798 different readings of fingertip positions for 160 different subjects. Table 5.1 summarized the means and standard deviations for all subjects.

Table 5.1 Means And Standard Deviations For The Coordinates Across All Subjects

FINGER	*X* (Mean)	*X* (Std Dev.)	*Y* (Mean)	*Y* (Std Dev.)
Thumb	-45.12	15.41	-52.75	9.33
Middle	-39.17	9.02	-73.06	4.46
Ring	-27.88	9.34	-88.64	4.66
Pinkie	-35.35	8.09	-106.98	4.91

Given that many subjects had very little previous experience with firearms, they had large variation fingertip positions across the five trials. Table 5.2 details the minimum and maximum standard deviations within each subject. The mean standard deviations are also given in Table 5.2. The small standard deviation suggests that the fingertip positions of the subject were very tightly grouped and could be easily

recognized. Though the large maximum and reasonably large mean values seems to highlight that the subjects did show large variations among the trials.

However, the standard deviations of the police officers, which are shown in Table 5.3, were much lower that those of the entire data set. The possible reason is that police officers had much more experience with firearms and therefore had relatively fixed handgun grasp patterns for themselves.

Table 5.2 The Minimum, Mean And Maximum Standard Deviations For All Subjects

Variable	Min (mm)	Mean (mm)	Max (mm)
Xt	0.29	4.11	17.36
Yt	0.29	4.07	22.99
Xm	0.41	3.12	12.94
Ym	0.26	2.04	5.67
Xr	0.25	3.18	15.07
Yr	0.22	2.08	5.91
Xp	0.29	2.93	10.67
Yp	0.14	2.11	6.16

Table 5.3 The Minimum, Mean And Maximum Standard Deviations For The Police Officers In The Study

Variable	Min (mm)	Mean (mm)	Max (mm)
Xt	0.46	2.82	5.47
Yt	0.86	2.52	6.54
Xm	1.04	2.33	5.83
Ym	0.56	1.27	2.05
Xr	1.10	2.36	4.82
Yr	0.38	1.41	2.43
Xp	0.72	1.99	4.16
Yp	0.18	1.55	2.93

By comparing the results reported in Tables 5.2 and 5.3, there is strong evidence that the subjects with no handgun experience tend to change their fingertip positions more often than those who have firearm experiences. Therefore, it is much harder to recognize these none-experience users.

In order to make the fingertips mathematically measurable, each set of fingertip positions is represented by a single point $(X_t, Y_t, X_m, Y_m, X_r, Y_r, X_p, Y_p)'$ in 8D space.

5.3 Static Analysis

Using K nearest neighbor clustering, the 798 samples of fingertip sets from 160 subjects could be distributed into 160 clusters. However, 74 out of the 160 subjects were not individually separable, i.e., class A does not only contain samples from class A but also contain samples from other classes. Therefore, the density search algorithm is used to partition this data set into natural groups and the canonical analysis is used to find the importance of each variable.

Four natural groups are produced when applying the Mode analysis algorithm to the fingertip data set. Table 5.4 shows these four natural clusters.

Table 5.4 Details For The Four Natural Clusters Produced By The Density Search Technique

Cluster	Frequency	Subjects	Proportions (%)
1	525	105	65.625
2	115	23	14.375
3	63	13	8.125
4	95	19	11.875
Totals	798	160	100

Table 5.5 shows the eight coefficients of the first two canonical discriminant functions. The magnitude of the coefficients of the first canonical variables is ordered and shown in Table 5.6.

Table 5.5 The Canonical Coefficients For The Eight Fingertip Variables

Variable	Can1	Can2
Thumb *X* (X_t)	-0.0118	0.0596
Thumb *Y* (Y_t)	0.0816	-0.0289
Middle *X* (X_m)	0.0205	-0.1093
Middle *Y* (Y_m)	0.0541	-0.0323
Ring *X* (X_r)	0.1660	0.1799
Ring *Y* (Y_r)	0.0042	-0.748
Pinkie *X* (X_p)	-0.0178	-0.578
Pinkie *Y* (Y_p)	-0.0926	0.0212

Table 5.6 The Eight Fingertip Variables Re-ordered In Terms Of The Contribution Each Makes In Discriminating Between The Cluster Set

Variable	Can1	Can2
Ring *X* (X_r)	0.1660	0.1799
Pinkie *Y* (Y_p)	-0.0926	0.0212
Thumb *Y* (Y_t)	0.0816	-0.0289
Middle *Y* (Y_m)	0.0541	-0.0323
Middle *X* (X_m)	0.0205	-0.1093
Thumb *X* (X_t)	-0.0118	0.0596
Pinkie *X* (X_p)	-0.0178	-0.578
Ring *Y* (Y_r)	0.0042	-0.748

From this analysis it would appear that the horizontal position of the ring finger is the most important in classifying the sample subjects into the four groups. Followed by the vertical position of the pinkie, then the vertical position of the thumb, and finally, the vertical position of the middle finger. Informally, this result can be explained by the

information of the grip each variable provides. The horizontal position of the ring finger provides information about the length of the hand and (or) fingers, while the vertical position within other fingers positions are informative about the relative dispersion of the hand as well as variations in hand placement on the gun handle.

5.4 Classification Tree

Classification trees not only provide a very simple classification method but they also provide information of what locations on the gun handle are important for separating subjects. Combined with the outcome of discriminant analysis both methods provide information of where on the handle are the best locations to place any sensors in order to measure the crucial variables that separate the subjects. Classification trees are capable of providing both the potential number of sensors and the locations they should be placed at in order to discriminate between the four natural groups.

The main purpose of the classification tree including all fingers is its ability in ranking the importance of the individual variables. A variable that splits the subjects at the root (the top node of the tree) would be the one most responsible in explaining most of the variation between the four groups. A classifier that splits the subjects at a lower node would explain less variation between those four groups and is of less importance. In other words, the best location of a sensor would be determined by the split criteria at the root. The next best locations of sensors would be determined by split criteria at the next lower nodes. Finally, the locations of the last sensors would be determined by the split criteria at the terminal nodes of the tree.

While the classification tree including all fingers helps in determining the order of importance of sensors' locations, classification trees for each finger at a time are helpful in determining the locations and order of importance of sensors corresponding to each finger. Again the best location of a sensor would correspond to the split criteria at the top node of the tree, while the next best locations of sensors would correspond to split criteria at lower nodes of the tree.

The classification tree including the thumb, and middle, ring and pinkie fingers (Figure 5.5) as predictors suggests that the horizontal change in the ring fingertip position accounts for most variation between the four groups. In addition to the vertical change of the thumb position, horizontal change in the middle fingertip position and the thumb position seem to account for the rest of the variation. The classification trees for the thumb, middle, ring and pinkie fingers are shown in the Appendix A.

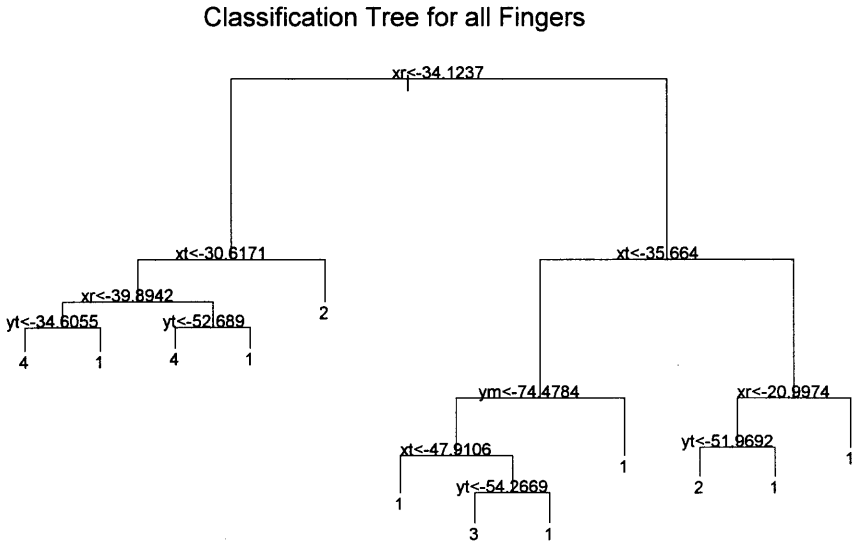


Figure 5.5 The classification tree for all fingers.

Instead of showing the actual tree, classification trees can be explained by showing the actual partitions on the gun handle for each of the clusters. The figures with partition in Cartesian space for the thumb, middle, ring and pinkie are shown in the Appendix A respectively.

5.5 Summary

The fingertip positions of 160 subjects were analyzed in this chapter. The static analysis shows that the fingertip placements of the firearm-experienced subjects such as police officers are more repeatable and have less variation than the fingertip placements of non-firearm-experienced subjects. Density search algorithm was applied to cluster the fingertip dataset into four natural groups. The cluster result is a best case scenario where there are no misclassified samples within these four natural groups at all. The static analysis also shows that the most important parameter to classify different users into four natural groups is the horizontal position of ring finger, followed by the vertical position of the pinkie, and the vertical position of the thumb. A classification tree was built based on the static analysis to find the possible sensor positions.

The static analysis of 160 subjects answered the questions which were asked at the beginning of this chapter:

1. The fingertip placement is repeatable enough to be used as a biometric pattern.
2. There is difference of the fingertip placement between two subjects more than the variation of one subject.
3. The best positions to put the sensors are given in the classification trees.

However, the static analysis is not enough for the handgrip recognition because it only divides the subjects into four natural groups. The goal of the handgrip recognition is to recognize each subject. This requires more analysis based on the dynamic pressure variations. The dynamic analysis will be discussed in the next chapter.

CHAPTER 6

PRESSURE PATTERN RECOGNITION

Society has relied on the written signature to verify the identity of an individual for hundreds of years. The complexity of the human hand and its environment make written signatures highly characteristic and difficult to forge precisely. Keystroke is also developed recently as an identity method based on the same reason. The handgrip has a parallel on the handwritten signature and keystroke. The same neurophysiological factors that make a written signature unique are also exhibited in a user's handgrip pattern.

Handgrip pattern recognition is a behavioral biometric rather than a physical biometric such as fingerprint and hand geometry. It seeks to analyze the dynamics inherent in grasping. How hard do people grasp the gun? How does the pressure vary from the beginning to the end of the firing process? How and when does the pressure cross a threshold? There are a number of such parameters inherent within the dynamic process.

Strictly speaking, authentication requires the person being identified to lay claim to an identity, so that the system may decide on either accepting or rejecting the claim. As with any security system, given that the claimant is, or is not, a true instance of the user, there are four possible outcomes: Acceptance of Authentic (AA), Acceptance of Imposter (IA), Rejection of Authentic (RA), and Rejection of Imposter (RI). Since the first and fourth outcomes are the desired results, one's main goals in designing an authentication system are to maximize the conditional probabilities of (AA) and (RI), while minimizing the likelihood of (IA) and (RA). The likelihood of (IA) is also called as false acceptance

rate, and the likelihood of (RA) is also called as false rejection rate. In practice, these goals are not always achievable and the level of acceptance for type I and type II errors is application specific.

The overall picture is slightly different for the case of profile recognition. The main focus here is to examine a set of stored data, and select the one that yields the best possible match to the unknown profile being presented. This chapter concentrates on the problem of recognition, as it is strictly more difficult than authentication. Any success in recognition can be directly transferable to authentication and therefore this chapter focuses the efforts on the problem of profile recognition.

6.1 Data Acquisition

A digital circuit has been designed and implemented for the Leybold-Inficon quartz sensor (Figure 6.1, left) that features a pressure-modulated resonance. The circuit has the advantage of simplicity and low cost. However, it does not offer a broad dynamic range and is now being redesigned for at least 1 MHz of coverage. As an alternative solution, a piezoelectric ceramic PZT disk (Figure 6.1, right) has been configured as a pressure sensor. Preliminary tests verified the feasibility of this approach. The PZT disk can also be partitioned into different zones, thus offering multi-sensing capability. A charge amplifier array has been developed to interface the sensors to the computer.

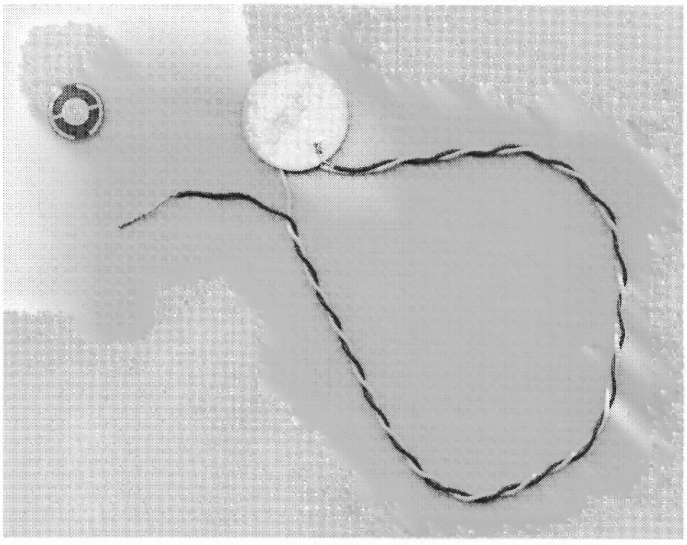


Figure 6.1 Quartz pressure sensor (left) and piezoelectric sensor (right).

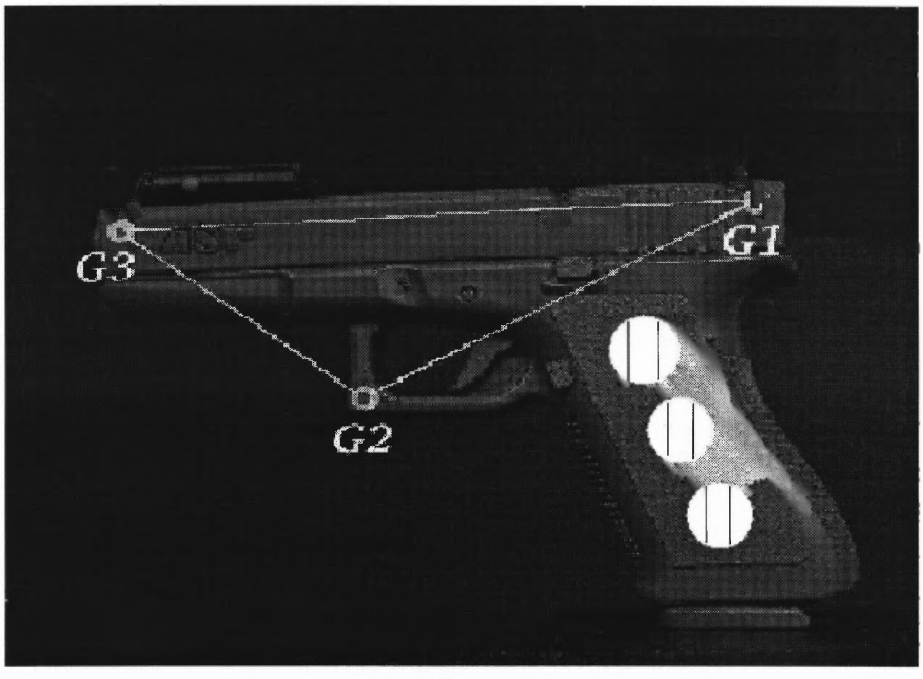


Figure 6.2 Sensor arrangement.

There are total five sensors being put around the replica gun handle. Since most users are using right hand to grasp the gun, three sensors are put at the front side of the

gun handle and two are put at the other side as shown in Figure 6.2. Each sensor is partitioned into three zones corresponding to different finger length. Longer fingers can reach more zones and therefore, provide more channels to read signal. This is especially useful to distinguish children from adults.

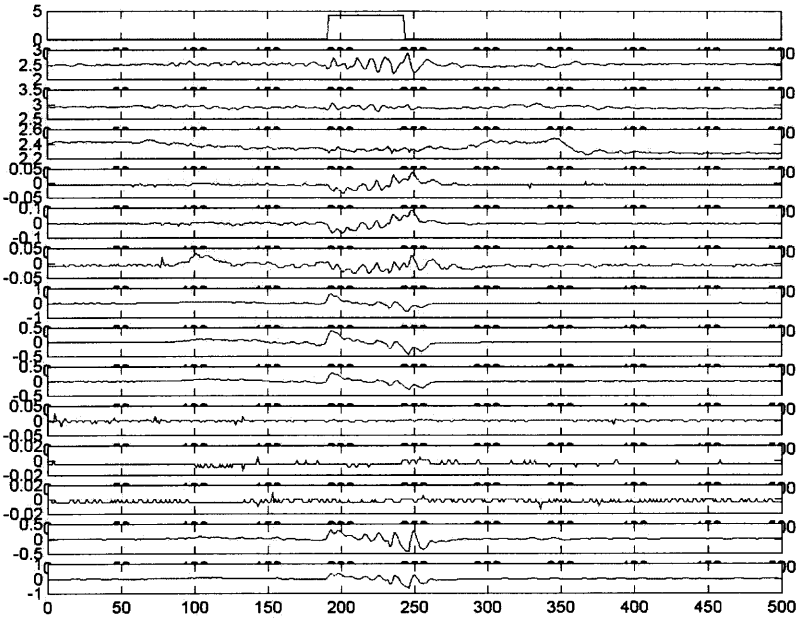


Figure 6.3 Data collected from 15 channels.

A DAQ PC-card with 15 channels (DAQ/112B from IoTech company) is used to collect data at the sample rate 1000 samples/second. Channel 1 is used for the trigger, therefore there are 14 channels used to collect the pressure data. Figure 6.3 shows what happened during five seconds of data collecting process. The trigger is pulled at the time of 1.9 seconds. The most important part is a short period just before the trigger pulled. In this period, the hand muscle repeats its pattern for gun firing. Out of this period, the

muscle may be affected by other factor such as user's emotion, pose, etc. The detail of what happens during the 100ms before the trigger being pulled is shown in Figure 6.4.

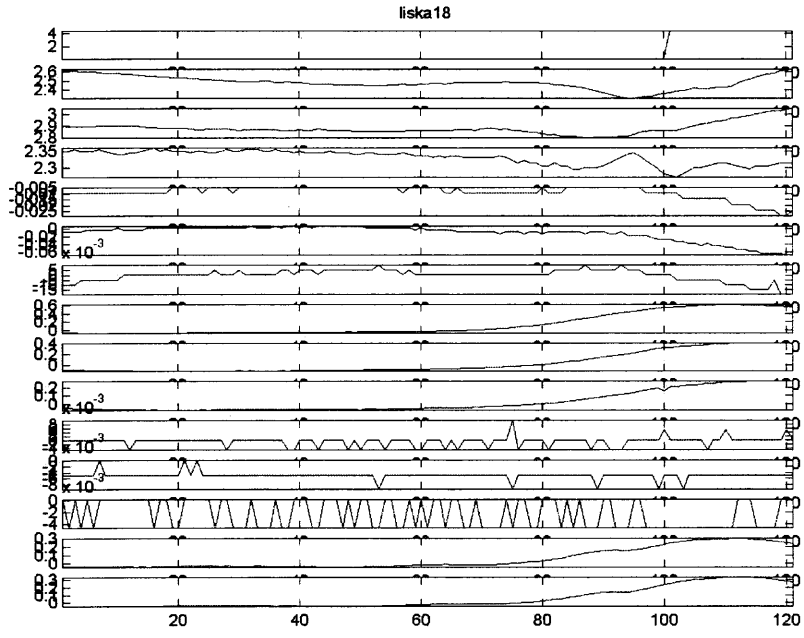


Figure 6.4 Details in the 100ms before trigger pulled.

Total 24 subjects, including 19 police officers from Montclair Police Department, participated in this experiment. Each subject was asked to grasp the gun, pull the trigger, and then release the gun for five times. One subject did this twenty times. Therefore, there are total 24×5+15=135 samples. Figures 6.5 and 6.6 show the data of channel 9 from five trials of two subjects. A down trend pattern for the first subject and a up trend pattern for the second subject can be found in these two pictures. Note in the next experiment, the data during 128ms before the trigger being pulled instead of 100ms will

be used because 128 is the nearest integer of 2^n to 100, and it is more convenient to use 2^n to do FFT or other transform in the future.

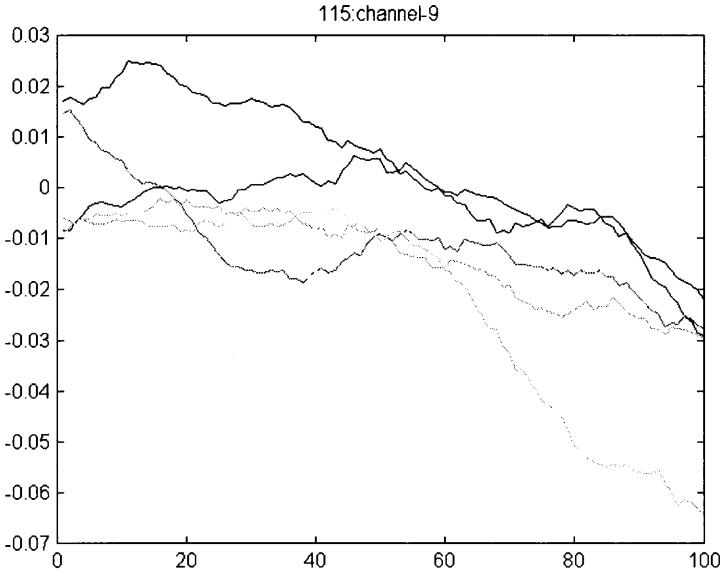


Figure 6.5 Pressure signal on channel 9 from subject 1.

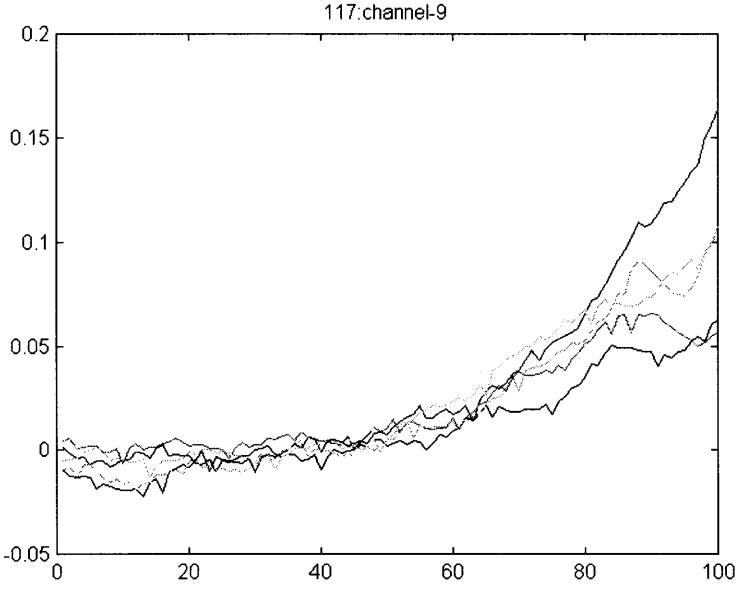


Figure 6.6 Pressure signal on channel 9 from subject 2.

6.2 Matching in Time Domain

In order to depict the similarity between two time series, it needs to define a similarity measurement during the matching process. Given two time series $x(t)=\{x_0, x_1, ..., x_n\}$ and $y(t)=\{y_0, y_1, ..., y_n\}$, a standard approach is to compute the distance $D(x, y)$ between time series $x(t)$ and $y(t)$.

Two distance measurements are widely used. One is the Manhattan distance function.

$$D(x,y)=\sum_{t=0}^{n}|x(t)-y(t)| \tag{6.1}$$

Another is the Euclidean distance function.

$$D(x,y)=\sum_{t=0}^{n}\sqrt{|x(t)-y(t)|^2} \tag{6.2}$$

In this case, Euclidean distance costs more computation than Manhattan distance. Remember the recognition needs to be completed in real time. Therefore, Manhattan distance is selected to measure the similarity.

Given an unknown time series $x(t)$ and the template data $\{y_1(t), y_2(t), ..., y_m(t)\}$, then $x(t)$ is most similar to $y_a(t)$ if $D(x, y_a)=min\{D(x, y_i)\}$, $i=1,2,..., m$, and $D(x, y_i)$ here represents the total distance between x and y_i over all channels. If $y_a(t)$ belongs to class Y_a, then $x(t)$ belongs to Y_a. The smallest five distances over 14 channels between a test sample from subject 1 and the data in template are shown in Figure 6.7. The vertical axis in the figure indicates the distance. The horizontal axis in the figure shows the channel number. It shows that the total distance between the test sample and the template data from subject 1 is the smallest. Therefore, this test sample can be classified as belonging to subject 1. Figure 6.8 shows another example for subject 2.

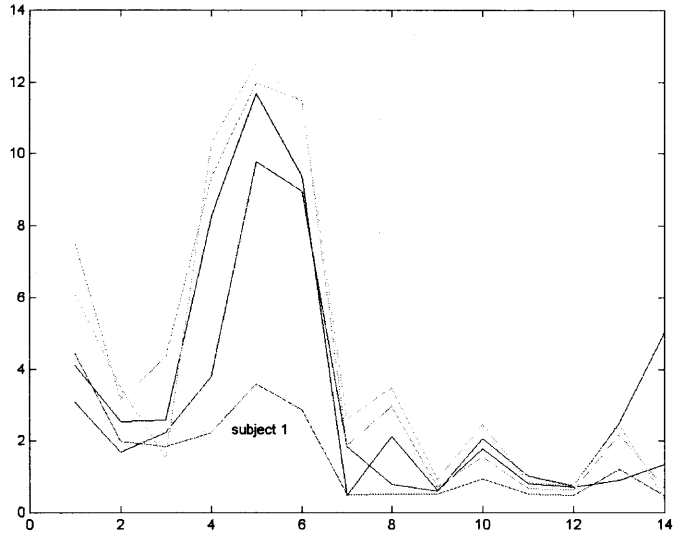


Figure 6.7 Distance distributions over 14 channels against subject 1.

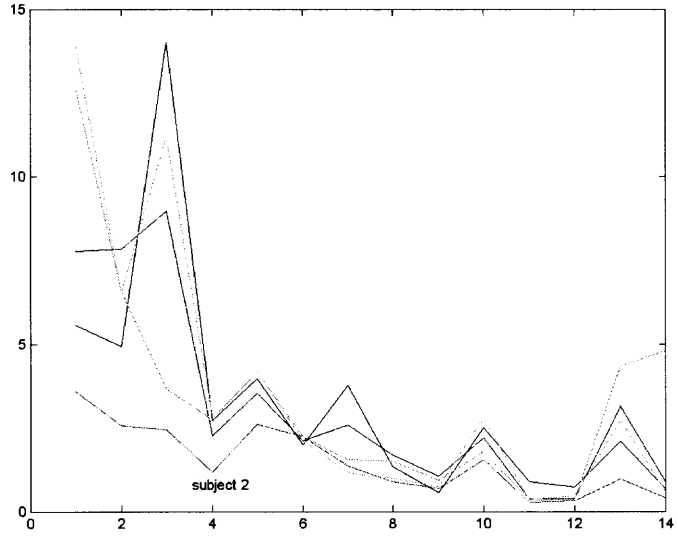


Figure 6.8 Distance distributions over 14 channels against subject 2.

The drawback to direct use of the Manhattan distance is the tendency of the largest-scaled distance from one channel will dominate the others. A vote algorithm is used to overcome this disadvantage.

The vote algorithm works as follows: First, the distances of the input data to the stored samples on each channel are calculated. If on channel 1, $D(x, y_j)$ is the minimum and $y_j \in Y_j$, then Y_j gets one vote. If Y_b finally gets more votes than the others, then x may belong to Y_b.

Both the Y_a from total distance measurement algorithm and the Y_b from vote algorithm are possible classes that x may belong to. Usually, Y_a and Y_b are same. But this is not always true. Sometimes, Y_a and Y_b may be different, and the vote algorithm may result in more than one possible class.

The first three samples from each subject were selected as the template and the other samples were used as test to evaluate this algorithm. Among all $24\times2+1\times15=63$ test data, the above algorithm can directly find the correct pattern for 33 samples. There are six samples can not be recognized correctly. For the other 24 samples, this algorithm gives more than one possible patterns and the correct pattern is among these possible patterns. In such cases, further recognition is required.

At the same time, using distance measure technique in each channel, it will show how many patterns can be recognized if the distance is measured for only one channel. Table 6.1 shows the result of that. Table 6.2 shows how important each channel is based on the Table 6.1. The most important channel is shown first.

From these two tables, it is clear that channel 3 is the most important which can discriminate 26 correct patterns. Channel 11 is the second most important which can

discriminate 20 correct patterns. Channel 2 is as important as channel 4 because they both can discriminate 12 patterns. Channel 9 is the least important among all 14 channels.

Table 6.1 Correct Patterns For Each Channel

Channel number	Number of patterns recognized correctly
1	4
2	12
3	26
4	12
5	4
6	14
7	7
8	9
9	3
10	17
11	20
12	9
13	10
14	13

Table 6.2 The Importance Of Each Channel

Importance	Channel number
1	3
2	11
3	10
4	6
5	14
6	2
6	4
8	13
9	8
9	12
11	7
12	1
12	5
14	9

By studying the importance of each channel, it is obvious that giving more weight to more important channel would produce better recognition result. For example, channel 3 can discriminate 26 correct patterns. Thus, channel 3 would be assigned a weight as 26. The weights for the other channel would be assigned in the same way. The final weights would be normalized so that the sum of all weights would be one.

Relative distance is another method to measure the similarity. To find the relative distance, the average waveform and average intra-group distance on each channel for each subject need to be calculated. On every time point for each channel, the average waveform is defined as $y'(t)=1/n \sum y_i(t)$. The average intra-class distance is defined as

$$d'=1/n \sum_{i=1}^{n} \sum_{t} | y_i(t)-y'(t)|. \tag{6.3}$$

And the relative distance between input signal $x(t)$ and the template $y_j(t)$ is defined as

$$d(x, y_j) = \sum_{channel} D(x, y_j)/d'_j \tag{6.4}$$

If $y_c \in Y_c$ and $d(x, y_c)=min\{d(x, y_j)\}$, then Y_c is considered as the possible class that $x(t)$ belongs to.

However, there is a big problem of directly using relative distance algorithm to do the pattern recognition. For example, if subject a has big average variation (big average distance d_a) and subject b has less variation d_b. For the input signal $x(t)$ and the average waveform from subject a and b, there are $d(x, a) = D(x, a)/d_a$ and $d(x, b) = D(x, b)/d_b$. Even if $D(x, a)$ is larger than $D(x, b)$, it is possible that $d(x, a)$ is smaller than $d(x, b)$, because d_a is larger than d_b. As a result, the relative distance algorithm will mistakenly classify the test data x belong to subject a.

In summary, directly using the relative distance algorithm does not give good result. It can be used to improve the answer of total distance and vote algorithm. This algorithm will be discussed in Section 6.4.

6.3 Matching in Frequency Domain

Linear transforms, especially Fourier transform, are widely used in solving problems in science and engineering. The Fourier transform, in essence, decomposes or separates a waveform or function into sinusoids of different frequency which sum to the original waveform. It identifies or distinguishes the different frequency components and their respective amplitudes. As stated in Chapter 3, the discrete Fourier transform is defined as

$$X(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nfT} \tag{6.5}$$

If $X(f)$ is the Fourier transform of $x(n)$ and τ is a real constant, then

$$\sum_{n=-\infty}^{\infty} x(n-\tau) e^{-j2\pi nfT} = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi(n+\tau)fT} = e^{-j2\pi\tau fT} \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nfT} = X(f) e^{-j2\pi\tau fT} \tag{6.6}$$

This time shifting property states that the Fourier transform of a shifted function is just the transform of the unshifted function multiplied by an exponential factor having a linear phase, and that the amplitude of $X(f)$ does not change.

Figures 6.9 and 6.10 show the amplitude of the frequency coefficients of channel 9 from five trials of two same subjects shown in Figures 6.5 and 6.6.

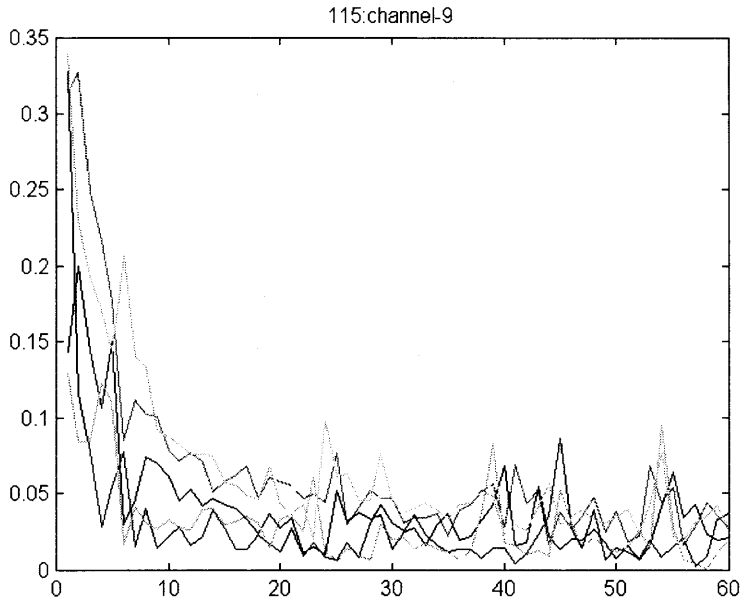


Figure 6.9 Data from Figure 6.5 represented in frequency domain.

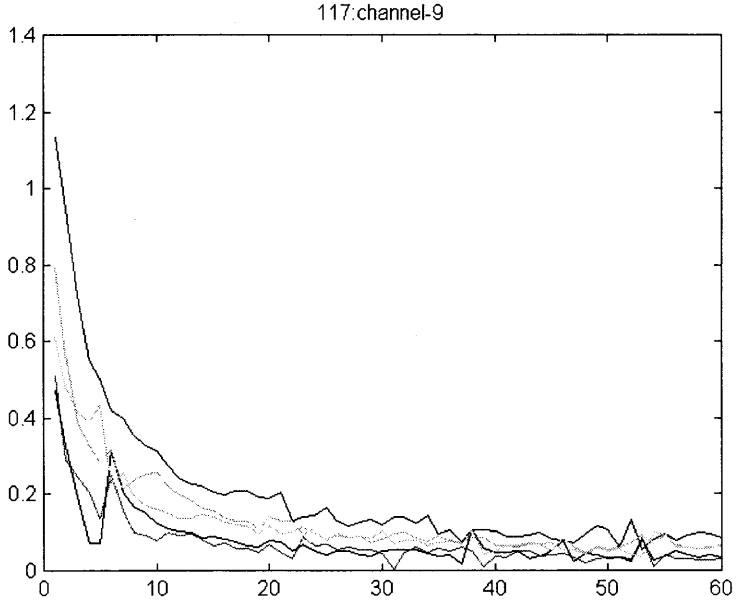


Figure 6.10 Data from Figure 6.6 represented in frequency domain.

Now, it is time to evaluate the algorithms discussed in the previous section in frequency domain. First, transform the original data to frequency coefficients. Then, use

the same distance measure function defined in the previous section to measure the distance between the coefficients in frequency domain. The first three samples from each subject are used as the template and the other samples are used as test to evaluate this algorithm. Among all 24×2+1×15=63 test data, the recognition algorithm can give the correct pattern for 43 test data, and for 15 test data, can find more than one possible patterns that contain the correct one. There are five recognition failed.

It is interesting that the failures of time domain matching and the failures of frequency domain matching are not same. By getting the union of the results from time domain matching and frequency domain matching, 28 test data can be recognized directly, and for 33 test data, it can find more than one possible patterns that contain the correct one. Only two recognition failed. It shows that getting union of time domain matching and frequency domain matching can produce more accurate candidates for the next step finer recognition.

6.4 Handgrip Pattern Recognition Algorithm

The distances in time domain and in frequency domain have been calculated. Neither one can give a perfect result. However, if they are combined together, the union of the results from total distance algorithm, vote algorithm and relative distance algorithm will cover most possible answers. The handgrip pattern recognition algorithm is presented as following:

1. Calculate the distances between the test data $x(t)$ and the data stored for each channel in time domain. Then use vote algorithm to find the possible pattern Y_b.

2. Calculate the total distance between the test data $x(t)$ and the data stored. Then find the minimum total distance to identify the possible pattern Y_a.
3. Apply the total distance algorithm and the vote algorithm in frequency domain to find the possible patterns Y_a' and Y_b'. Let Y=*union*(Y_a, Y_b, Y_a', Y_b').
4. Use Y as the candidates and apply the weight relative distance matching algorithm in time domain to choose one as the final answer. The weights are designed in the same way as defined in Section 6.2.

To evaluate this handgrip pattern recognition algorithm, three samples were randomly picked from each subject to create the data templates, and the others were used as the tests. Repeat this process 30 times, so that there are 1890 tests totally. Among these 1890 tests, 1688 tests give the correct answers, that produces 1688/1890=89.3% success rate. The mistakes done by the handgrip recognition algorithm are shown in Table 6.3.

In the first row of Table 6.3, it shows that there are eight test samples from subject #1 mistakenly classified as subject #2, one test sample from subject #1 mistakenly classified as subject #6, and one test sample from subject #1 mistakenly classified as subject #19. Since eight tests from subject #2 are mistakenly classified as subject #2, which means that handgrip pattern of subject #1 is very similar to subject #2, subject #1 and subject #2 can be put into group #1.

In the third and fourth row of Table 6.3, it shows that there are ten test samples from subject #3 mistakenly classified as subject #4, and nine test samples from subject #4 mistakenly classified as subject #3. Compared to the fact that three test samples from subject #4 mistakenly classified as subject #12, and two test samples from subject #4

mistakenly classified as subject #19, subject #3 and subject #4 are much more similar to each other. Therefore, subject #3 and subject #4 can be put into group #2.

Table 6.3 Mistakes While Trying To Recognize Each Subject

Group ID	Subject ID	classified as subject	mistakes	classified as subject	mistakes	classified as subject	mistakes
1	1	2	8	19	1	6	1
1	2	0	0	0	0	0	0
2	3	4	10	0	0	0	0
2	4	3	9	12	3	19	2
3	5	10	11	17	10	9	1
4	6	0	0	0	0	0	0
5	7	9	2	10	2	8	1
6	8	0	0	0	0	0	0
7	9	10	4	5	2	14	1
3	10	11	10	5	2	7	1
3	11	16	20	10	16	14	3
8	12	14	5	6	3	17	2
8	13	14	4	0	0	0	0
8	14	12	8	13	6	4	1
9	15	24	2	2	1	0	0
3	16	11	4	0	0	0	0
3	17	5	2	18	1	7	1
10	18	17	2	5	1	0	0
11	19	16	1	0	0	0	0
12	20	0	0	0	0	0	0
13	21	19	4	5	1	23	1
14	22	0	0	0	0	0	0
15	23	3	3	14	1	11	1
16	24	6	4	7	2	2	1

By studying Table 6.3, these 24 subjects can be partition into 16 groups, where one group contains five subjects, one group contains three subjects, two groups contain two subjects, and twelve groups contain only one subject. The criteria is that if more than four tests from one subject are mistakenly recognized as another specific subject, these two subjects should be grouped together.

After partitioning 24 subjects into 16 groups, the handgrip pattern recognition algorithm was evaluated based on groups again. Three samples from each subject were randomly picked to create the data templates, and the others were used as the tests. This process was repeated 30 times. In the end, there were 97 misclassifications. The average success rate was increased to 1793/1890=94.9%.

6.5 Summary

The dynamic analysis of handgrip signals from 24 subjects was discussed in this chapter. A handgrip pattern recognition algorithm was also proposed and evaluated. The average success recognition rate is 89.3%. Dividing the 24 subjects into 16 groups increases the success rate to 94.9%.

A plastic replica handgun was used this chapter. However, a fake handgun does not have same weight with the real handgun. The fake gun can not give the user the feel of a real firing either. This affects the handgrip recognition algorithm working properly. In addition, all pattern recognition in this chapter was done in a workstation. The decision was not made in real-time. To overcome these disadvantages, a real handgun will be used and a DSP based handgrip recognition system, which can recognize the user in real-time, will be presented in the next chapter.

CHPATER 7

DSP IMPLEMENTATION

Signal processing can be divided into two areas: non-realtime signal processing and realtime signal processing. Realtime signal processing means that the processing must keep pace with external event; non-realtime signal processing has no such timing constraint. It is obvious that this project needs signal processed in realtime. While conventional microprocessors can be used for both non-realtime signal processing and realtime signal processing, their architectures and instruction sets limit their usefulness for realtime signal processing as the specially designed signal processors (Bateman, 1989). DSP microprocessors are characterized by fast multiply instructions, reduced instruction sets, and specialized instructions to make DSP algorithms execute fast and efficiently (Chassaing and Horning, 1989).

This chapter presents a realtime handgrip pattern recognition system implemented by the TMS320C31 DSP Starter Kit (DSK). The C31 DSK is a relatively powerful, inexpensive stand-alone application development board. The DSK includes the TMS320C31 processor, the TLC32040 analog interface circuit (AIC), and comes with an assembler and debugger.

7.1 Overview of TMS320C31 DSK

The TMS320C31 is the low cost member of the third generation family of digital signal processor from Texas Instruments Incorporation. Compared to its price, $5 each, it is the best choice for the smart gun project.

The TMS320C31 integrates both system control and math-intensive functions on a single high-performance CMOS 32-bit floating-point device. This system integration allows fast, easy data movement and high-speed numeric processing performance. With the 40ns instruction cycle time, it provides capabilities for 50 million floating-point operations per second (MFLOPS) or 25 million instructions second (MIPS). The 24-bit address bus provides large address space (16 million 32bits words) for program, data and input/output.

Figure 7.1 shows the TMS320C31 block diagram. The CPU consists of the following components: Floating-point/integer multiplier, Arithmetic logic unit (ALU), Internal buses, Auxiliary register arithmetic units, and CPU register files.

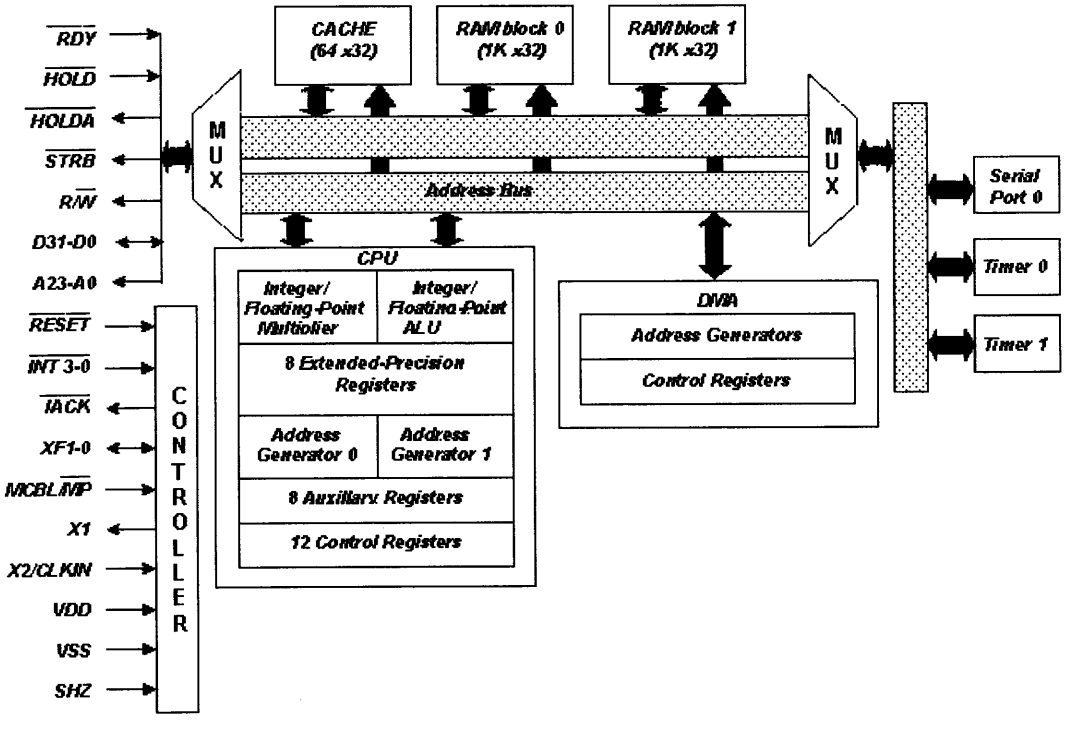


Figure 7.1 TMS320C31 block diagram.

The multiplier performs single-cycle multiplication on 24-bit integer and 32-bit floating-point values. The ALU performs single-cycle operations on integer, logical, and floating-point data, including single-cycle integer and floating-point conversions. Four internal buses, CPU1, CPU2, REG1, and REG2 carry two operands from memory and two operands from the register file, allowing parallel multiplies and adds/subtracts on four integer or floating-point operands in a single cycle. Two auxiliary register arithmetic units (ARAU0 and ARAU1) can generate two addresses in a single cycle. The ARAUs operate in parallel with the multiplier and ALU. There are 28 registers in a multi-port register file that is tightly coupled to the CPU. All of the primary registers can be operated upon by the multiplier and ALU and can be used as general-purpose registers. Since signal processing algorithms tend to involve repeated cycles of multiplication and accumulation of data elements with coefficients stored in different parts of memory, these features (ALU, multiplier, internal buses, and auxiliary register arithmetic units) extremely increase the processing speed.

The on-chip DMA controller can read from or write to any location in the memory map without interfering with the CPU operation. The DMA controller contains its own address generators, source and destination registers, and transfer counter. Dedicated DMA address and data buses minimize conflicts between the CPU and the DMA controller. Since the DMA and CPU have distinct buses, they can operate independently of each other. However, when the CPU and DMA access the same on-chip or external resources, there is a conflict and C31 assign highest priority to the CPU.

The total memory space of the C31 is 16 million 32-bit words to contain program code, data, and I/O mapped registers. The C31's separate program, data, and DMA buses

allow for parallel program fetches, data reads and writes, and DMA operations. A 64 × 32-bit instruction cache stores often-repeated sections of code, greatly reducing the number of off-chip accesses and allowing code to be stored off-chip in slower, lower-cost memories.

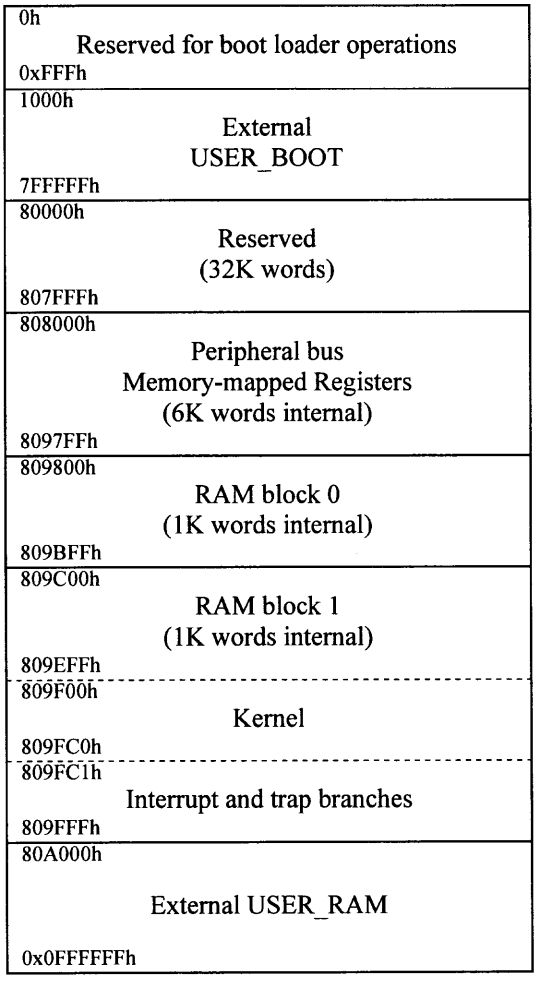


Figure 7.2 TMS320C31 DSK memory map.

The TMS320C31 DSK memory map is shown in Figure 7.2. Locations 0h–FFFh are reserved for boot loader operations. Locations 800000h–807FFFh are reserved too.

All of the memory-mapped peripheral bus registers are in locations 808000h–8097FFh. There are two on-chip 1K × 32-bit RAM blocks. RAM block 0 is located at addresses 809800h–809BFFh, and RAM block 1 is located at addresses 809C00h–809FFFh. While the C31 has 2K words of on-chip memory, the last 256 words of internal memory (809F00h–809FFFh) on the DSK are used for the communications kernel and vectors. The actual external RAM installed on the DSK that can be used by the program is 256K words, from location 880000h to location 8BFFFFh.

The DSK board communicates with the PC host through a standard parallel printer port interface connector cable. A program in C (or assembly) can be compiled and linked to create an executable common object file format (COFF) file and the resulting COFF file can be downloaded into the C31 on the DSK. An executable file can be loaded into the DSK using either the debugger or a boot loader.

7.2 DSP Implementation on a Real Gun

In the previous chapters, a plastic replica handgun was used to do the static analysis and dynamic analysis experiments. However, a plastic replica handgun does not give the subject feelings of a real firing. In addition, all pattern recognition works needed to be done after samples collected and processed by a PC. In reality, the smart gun is based on a real gun and the pattern recognition needs to be done in real time. In this chapter, a real handgun, Beretta 92FS, specially designed and manufactured for the military by Beretta Corporation is used as the prototype of a smart gun. There are 16 sensors being put on the gun handle. There is another sensor connected to the trigger to provide a pulse signal when trigger pulled.

The TLC32040 analog interface circuit (AIC) provides C31 the interface to these 16 sensors. The AIC converts analog input signals into digital signals for C31 analysis. After the signal being processed, the output data is returned to the AIC for conversion into an analog signal. The analog-to-digital converter (ADC) and digital-to-analog converter (DAC) with 14-bit dynamic range can work at variable rate. The TLC32040 A/D and D/A use offset binary for the digital representation of a number, where both positive and negative analog inputs are represented as a binary number ranging from 00...0 for the most negative input to 11...1 for the most positive input.

The AIC is controlled by C31 through the serial port. The AIC uses two types of communication: the primary communication is used to pass the 14 bits through the D/A and A/D converters, the secondary communication mode is used to initialize and control the AIC. There are four control registers, TA, RA, TB, and RB.

Sample rate = system frequency / (2 × TA/RA × TB/RB)

The A registers control the switched capacitor bandpass filters frequency (SCFF). TA/RA = system frequency / (SCFF × 2). The B registers control the sample rate of the A/D and D/A converters. TB/RB = SCFF / sample frequency.

The values of register A and B are determined by:

A-value: TA ← [*b13 – b9*] RA ← [*b6 – b2*] [*b1, b0*] = [*0, 0*]

B-value: TB ← [*b14 – b9*] RA ← [*b7 – b2*] [*b1, b0*] = [*1, 0*]

Therefore, the register A should be set as A_LOC=((TA<<9)+(RA<<2)+0). The register B should be set as B_LOC=((TA<<9)+(RA<<2)+2).

In this case, the serial timer port clock has been set to 6.25 MHz, and the sample rate is set to 1 KHz for each channel with total 16 channels. Therefore, the A/D sample

rate should be 1 KHz/channel × 16 channels = 16 KHz. The A registers should ensure the SCFF to be 288 KHz so that TA/RA = 11, and TB/RB = 18.

The C31 timer and serial port is initialized by following code:

```
long *T0_ctrl  = (long *) 0x808020;          /*  Timer 0 global control      */
long *T0_count = (long *) 0x808024;          /*  Timer 0 counter             */
long *T0_prd   = (long *) 0x808028;          /*  Timer 0 period              */
long *S0_gctrl = (long *) 0x808040;          /*  Serial 0 global control     */
long *S0_xctrl = (long *) 0x808042;  /*  Serial 0 FSX/DX/CLKX port control  */
long *S0_rctrl = (long *) 0x808043;  /*  Serial 0 FSR/DR/CLKR port control  */
long *S0_xdata = (long *) 0x808048;  /*  Serial 0 Data transmit          */
long *S0_rdata = (long *) 0x80804C;  /*  Serial 0 Data receive           */

void ST_STUB(void)                   /* Startup stub          */
{
  *T0_ctrl = 0;                      /* Halt Timer 0          */
  *T0_count= 0;                      /* Set counts to 0       */
  *T0_prd  = 2;                      /* Set Timer 0 period (2 is 6.25 MHz) */
                                     /* (1 is 12.5 MHz)       */
  *T0_ctrl = 0x2C1;                  /* Restart timer         */
  *S0_xctrl = 0x00000111;            /* Configure serial port transmit control */
  *S0_rctrl = 0x00000111;            /* Configure serial port receive control */
  *S0_xdata = 0;                     /* DXR data value        */
  *S0_gctrl = 0x0E973300;            /* Enable RINT & 16 bit transfers */
  AIC_INIT();
}
```

The function AIC_INIT() initializes the AIC.

```
void AIC_INIT(void)
{
 asm(" andn  034h,IF ");
 asm(" ldi   004h,IE ");      /* Enable only INT2              */
 *S0_xdata = 0;
 asm(" rpts  0040h   ");      /* Repeat 64 times               */
 asm(" ldi   34,IOF  ");      /* XF0=0 resets AIC              */
 asm(" ldi   38,IOF  ");      /* XF0=1 runs AIC                */
 asm(" rpts  040h    ");      /* Repeat 64 times               */
 asm(" nop           ");      /* No operation                  */
 asm(" andn  034h,IF ");      /* Set INT2,XINT0, and RINT0     */
 asm(" ldi   014h,IE ");      /* Enable EINT2 and EXINT0 interrupts */
                              /* EINT2 is Parallel Port        */
                              /* EXINT0 is Serial Port 0 xmit  */
 prog_AIC(3);                 /* program control register      */
```

```
 prog_AIC(0xFFFC  );     /* Program the AIC to be real slow       */
 prog_AIC(0xFFFC|2);     /* Program the AIC to be real slow       */
 prog_AIC(B_LOC   );     /* Program TB/RB                         */
 prog_AIC(A_LOC   );     /* Program TA/RA                         */
 asm("  or  080h,ST");   /* Use the overflow mode for fast saturate */
}
```

The function prog_AIC(int xmit2) transmit a value to a AIC control register.

```
void prog_AIC(int xmit2)
{
 *S0_xdata = 0;        asm(" idle "); /* Pretransmit a safe value      */
 *S0_xdata = 3;        asm(" idle "); /* Request the 2nd communication mode */
 *S0_xdata = xmit2;    asm(" idle "); /* Send register value    */
 *S0_xdata = 0;        asm(" idle "); /* Leave with a safe value       */
}
```

After AIC being initialized, ADC data can be read from A/D mapped address 0x80804C. The DAC output can be done by sending data to D/A mapped address 0xC00000.

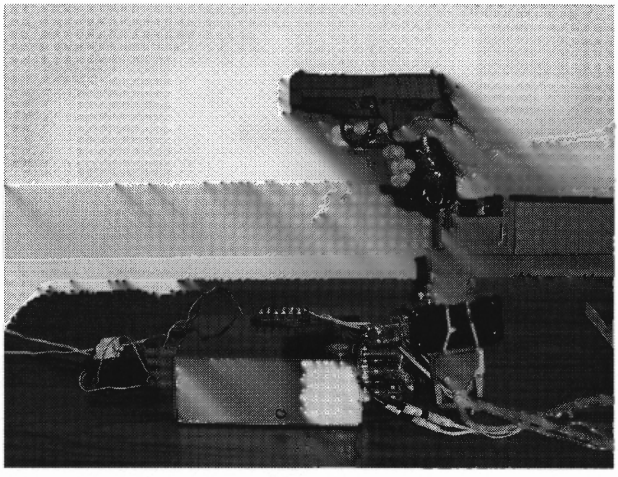


Figure 7.3 The DSP box.

The TMS320C31 DSK is incorporated into one box with 16 analog input channels and four analog output channels. The DSP box is shown in Figure 7.3. The four output

channels were connected to a red/green LCD light and two buzzes. If the test subject is an authorized user, green light and one buzz give the "go" signal. Otherwise, red light and another buzz give the "no-go" signal.

7.3 Wavelet Analysis of Handgrip Signal

The handgrip pattern recognition algorithms have been discussed before. However, the algorithm discussed in Chapter 6 is not totally appropriate for a real-time recognition work. The frequency analysis requires a lot of computation to do the Fourier transform. Even though DSP is specially designed for multiplication and addition, and FFT is developed for fast Fourier transform, it still costs about 10ms to do the computation for each channel. On the other hand, the recognition must be done in less than 100ms. Obviously, such frequency analysis is not acceptable here.

As discussed in Chapter 3, wavelet analysis is an alternative transform analysis tool. There are a wide variety of popular wavelet algorithms, including *Daubechies* wavelets, *Mexican Hat* wavelets and *Morlet* wavelets. These wavelet algorithms have the advantage of better resolution. But they have the disadvantage of being expensive to calculate. Due to the approximation quality and the simplicity, the *Haar* wavelet, which requires the least computation, becomes a good choice for the real-time work.

The *Haar* wavelet transform applies a rectangular window and a step function, $\psi(t)=1$ while $t\in[0, 1/2)$ and $\psi(t)=-1$ while $t\in[1/2, 1)$, to the input signal. After the transform, the signal is divided into two bands, low frequency band (smoothed values) and high frequency band (details). The smoothed values and the details are calculated as:

```
for (t = 0; t < n/2; t = t+1)
{
   s(t) = [v(2t) + v(2t+1)]/2;
   d(t) = [v(2t)  - v(2t+1)]/2;
}
```

where $v(t)$ is the input signal with length of n, $s(t)$ is the smoothed value (low frequency value), $d(t)$ is the detail (high frequency value). The inverse transform of *Haar* wavelet transform is

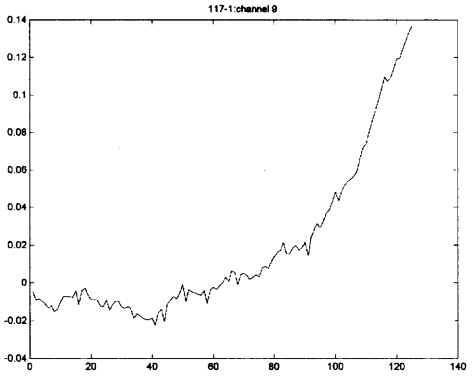
```
for (t = 0; t < n; t = t+1)
{
   v(2t) = s(t) + d(t);
   v(2t+1) = s(t) - d(t);
}
```

The first pass of the above algorithm over the input signal uses a window width of two. The window width is doubled at each step until the window encompasses the entire time series.

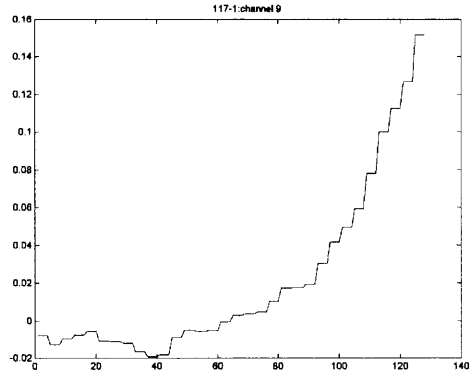
It will be easier to explain the discrete *Haar* wavelet transform by using an example. Assume the input signal to be x={4, 5, 1, 3, -3, 2, 7, -1}. Then the smoothed value of x is s={4+5, 1+3, -3+2, 7-1}/2 ={4.5, 2, -0.5, 3}; and the detail of x is d={4-5, 1-3, -3-2, 7+1}/2={-0.5, -1, -2.5, 4}. Apply *Haar* wavelet to the smoothed values again so that s'={4.5+2, -0.5+ 3}/2={3.25, 1.25} and d'={4.5-2, -0.5- 3}/2={1.25, -1.75}. Repeat this calculation again so that s''={3.25+1.25}/2=2.25 and d''={3.25-1.25}/2=1. The final answer of the discrete wavelet transform of x is y={s'', d'', d', d}={2.25, 1, 1.25, -1.75, -0.5, -1, -2.5, 4}.

In most cases, the important data is in low frequency area. Therefore, noise can be eliminated by discarding high frequency coefficients. In addition, the information that is most prominent in the original signal will appear as high amplitudes in that region of the

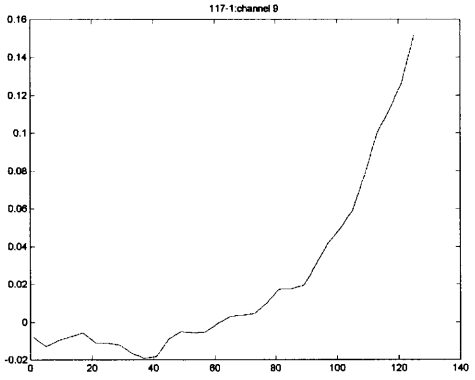
DWT. By setting a threshold, and discarding all values in the DWT that are less than this threshold, can extremely reduce the data size and delete the noise.



(a) Original data.



(b) Compressed data.



(c) Final filtered data.

Figure 7.4 Wavelet analysis of handgrip signal.

Figure 7.4a shows a handgrip signal collected from a subject. Figure 7.4b shows the inverse discrete *Haar* transform of this handgrip signal after all the discrete *Haar* coefficients except the first 16 coefficients are set to zero. By discarding the last 96 coefficients, both the feature size and the computation for distance comparisons are reduced. A filter must be used to improve this inverse transform in this case.

```
for (t = 0; t < 32; t = t+1)
{
   v(4t+2) = [v(4t) + v(4t+4)]/2;
   v(4t+1) = [v(4t) + v(4t+2)]/2;
   v(4t+3) = [v(4t+2) + v(4t+4)]/2;
}
```

Figure 7.4c shows the final result of the improved inverse transform. This result represents the pressure signal much better than the original noise corrupted data. From this example, it shows that discrete wavelet transform not only helps to reduce the data size (only 32 coefficients representing 128 points data), but also helps to eliminate the noise.

7.4 Sensor Arrangements

In order to find the best sensor positions, six subjects, who are professional firearm users, were asked to participate in an experiment. Initially, these six subjects were asked to grasp a Beretta 92FS handgun in order to get their fingertip position distribution. Then, the positions of the 16 pressure sensors on the gun handle were carefully designed based on their fingertip distribution to mostly distinguish these six subjects. The positions of the 16 pressure sensors are shown in Figures 7.5 and 7.6.

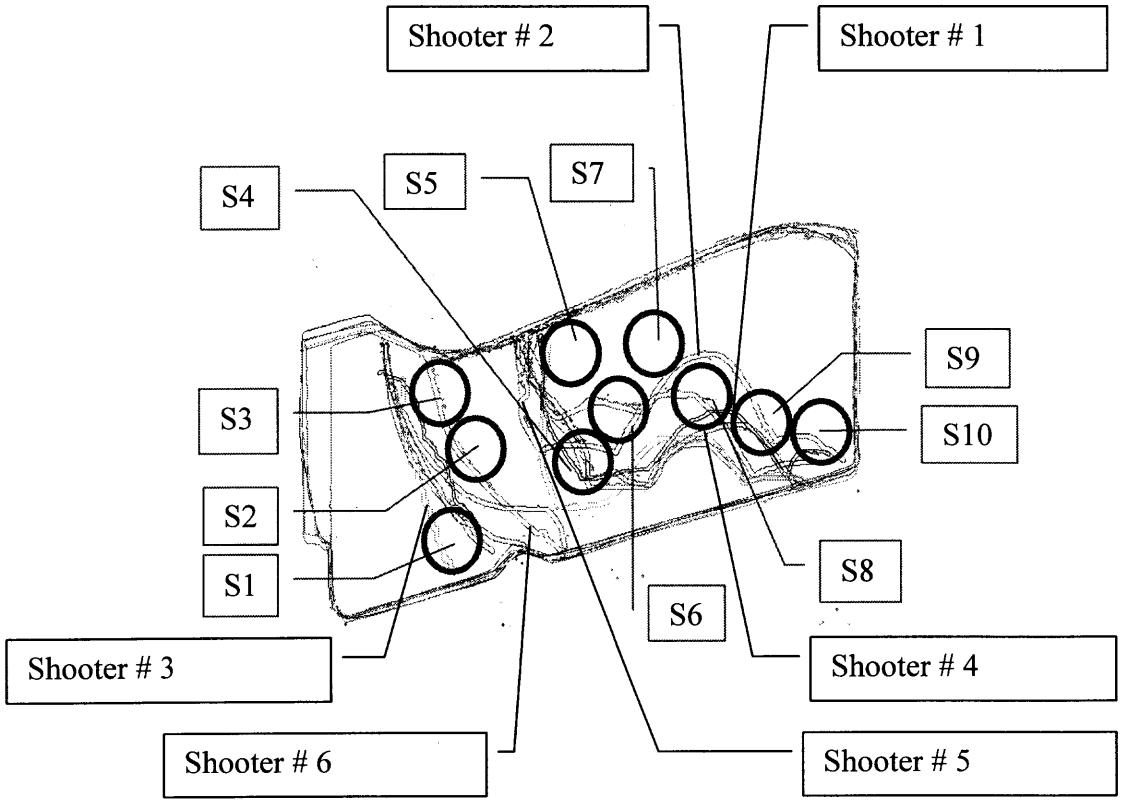


Figure 7.5 Sensor arrangements on the left side grip.

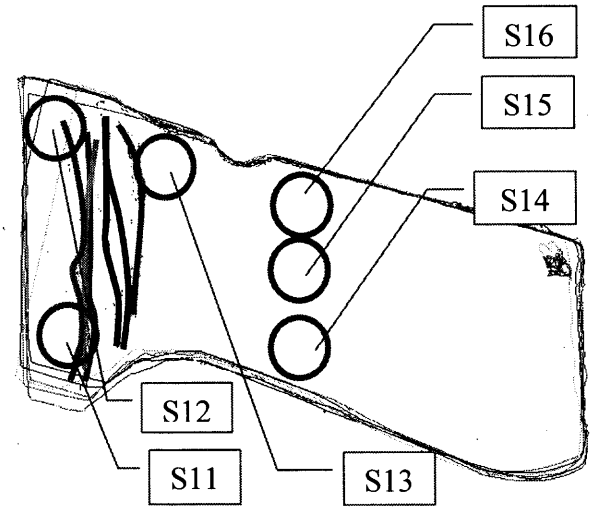


Figure 7.6 Sensor arrangements on the right side grip.

Tables 7.1 and 7.2 show the configuration of the sensors for each subject. In these two tables, 1 represents that the subject touches the sensor, 0 represents that the subject does not touch the sensor. By studying these tables, it shows that subject #6 can be discriminated from the other subjects by examining whether the pressure sensor S9 being touched. Subject #3 can be distinguished from the other subjects by examining the pressure sensor S4. In addition, subject #5 can be distinguished from the others by examining sensor S12. After excluding subject #5, subjects #1 and #2 can be classified by examining the sensors S13 and S3, respectively.

Table 7.1 Configurations For Sensors 1-10

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Blue [#1]	1	1	0	1	1	1	1	0	1	0
Red [#2]	1	0	1	1	1	1	1	1	1	1
Green [#3]	0	1	0	0	0	1	0	1	1	0
Black [#4]	0	1	0	1	0	1	1	1	1	0
Violet [#5]	1	1	1	1	1	0	0	1	1	1
Purple [#6]	0	0	0	1	0	1	0	0	0	0

Table 7.2 Configurations For Sensors 11-16

	S11	S12	S13	S14	S15	S16
Blue [# 1]	0	0	1	1	1	1
Red [#2]	0	0	1	1	1	1
Green [#3]	1	0	0	1	1	1
Black [# 4]	0	0	0	1	1	1
Violet [# 5]	1	1	0	1	1	1
Purple [# 6]	1	0	0	1	1	1

After the sensor positions were decided, pockets were machined on the gun grip. Sensors were put into the pockets and were wired to connect to the DSP box.

7.5 Realtime Handgrip Recognition System and Its Evaluation

The disadvantage of the handgrip recognition algorithm discussed in Chapter 6 is that it requires too much computation to complete. First, the computation of FFT is slow. The discrete *Haar* wavelet transform can be used to solve this problem. Second, it compares the input test with all stored data to find a most similar one. If there are N subject with n stored samples for each subject, there will be $n \times N$ comparisons. This is not necessary. To make the comparison faster, a pattern is calculated for each subject and the following recognition procedure will use this pattern to do the identification.

The handgrip pattern recognition focuses on the pressure variation over the time not the absolute amplitude of the handgrip signal, because the absolute amplitude of the handgrip signal may have big variations in different cases. The hand pressure will be much weaker when a person is sick or injured. However, the relative pressure waveform will not change much in such cases. Normalization needs to be done before to create patterns for each subject.

To normalize the data, all DC values of all samples in the template need to be removed. Then, the maximum pressure value on all sensors from each sample during the 100ms before trigger pulled is calculated and is set to +/- 1. The other values are normalized in the same ratio so that the pressure waveform keeps same.

The average waveforms (patterns) and the average intra-class distances are calculated in the same way as defined in Chapter 6. At the same time, apply the discrete *Haar* wavelet transform to the pattern $Y_a(t)$ and store it as $H_a(t)$. In the preprocessing stage, the weights for each channel are also calculated in the same way as defined in Chapter 6. By picking the first 50 samples and using the other 25 samples from each

subject (totally 125 samples) as the tests, Table 7.3 shows the correct patterns recognized as measuring similarity on only one channel. This result can be used for assigning weights to each channel.

Table 7.3 Correct Patterns For Each Channel

Channel number	Number of patterns recognized correctly
1	62
2	46
3	64
4	60
5	62
6	68
7	56
8	66
9	39
10	71
11	67
12	60
13	68
14	51
15	71

Given the test data x, the real time handgrip pattern recognition algorithm can be defined as following:

1. Calculate the total distances between x and the stored patterns. If $D(x, Y_a)$ is the minimum, then Y_a is identified as the possible pattern.
2. Apply the discrete *Haar* wavelet transform to x to get x_h. Calculate the distances between x_h and the stored patterns. If $D(x_h, H_b)$ is the minimum, then Y_b is identified as the possible pattern.
3. If $Y_a = Y_b$, then it is the answer. Let $Y = Y_a$ and terminate the algorithm.
4. Otherwise, calculate the weighted relative distances $d(x, Y_a)$ and $d(x, Y_b)$.

5. If $d(x, Y_a)$ is much greater than $d(x, Y_b)$, say $d(x, Y_a) / d(x, Y_b) > 1.1$, then $Y = Y_b$.

   If $d(x, Y_a)$ is much smaller than $d(x, Y_b)$, say $d(x, Y_a) / d(x, Y_b) < 0.9$, then $Y = Y_a$.

   Otherwise, calculate the ratios of $r_1 = D(x, Y_a)/D(x, Y_b)$ and $r_2 = D(x_h, H_b)/D(x_h, H_a)$.

   If $r_1 < r_2$, i.e., the difference between $D(x, Y_a)$ and $D(x, Y_b)$ is greater than the difference between $D(x_h, H_a)$ and $D(x_h, H_b)$, then $Y = Y_a$. Otherwise, $Y = Y_b$.

Among these six subjects, five subjects (subject 1-2 and 4-6) participated in the realtime dynamic analysis experiment. In order to give the subject as much same feeling as a true firing, a simulator was used to produce the firing force back. Each subject was tested for five rounds. In each round, the subject used a handgun simulator to aim at different targets and fired the handgun simulator for 15 times. Two subjects wore the gloves to test whether wearing gloves affects the analysis result. It shows that gloves would not affect the handgrip recognition.

To evaluate the realtime handgrip recognition algorithm, 50 samples were randomly picked from each subject as the templates, and the other samples were used as tests. This process was repeated 20 times, with totally 2500 tests. The recognition speed is 19ms per recognition, based on the TMS320C31 DSP system. The success rates are 100%, 80.2%, 88.8%, 92.8%, and 85.4% respectively for these five subjects. The average success rate is 89.44%. The patterns of five subjects are shown in Appendix B, and the recognition results are shown in Table 7.4.

From Table 7.4, it shows that the handgrip of subject #1 is the most consistent one. No test from subject #1 was mistakenly recognized as others. The second most consistent handgrip signal is from subject #4, whose success recognition rate is 92.8%. There are 2.4% tests being mistakenly recognized as subject #1, 1.8% tests being

mistakenly recognized as subject #2, 0.6% tests being mistakenly recognized as subject #3, and 2.4% tests being mistakenly recognized as subject #5.

Table 7.4 Recognition Results

Subject	1	2	3	4	5
1	100%	0	0	0	0
2	3.6%	80.2%	7.8%	4.2%	4.2%
3	1.6%	0.8%	88.8%	2.2%	6.6%
4	2.4%	1.8%	0.6%	92.8%	2.4%
5	2.0%	0.6%	10.8%	1.2%	85.4%

The recognition success rate is not as good as expected. One of the reasons could be the grip shear deformation during the test. The grip machining procedure involved the flattening and consequent pocketing of the grip, which eventually could undermine its structural integrity. The grip deflection caused an unwanted response from the sensors. An extremely sensitive sensor behavior to the shear and high frequency vibration was detected. Therefore, the grip has to be reinforced with copper or aluminum plate of small thickness in order to prevent the shear deflection of the grip during actual firing procedure. Sensor assembly would be mounted on the top of the plate. The plate would provide simultaneous shielding of the sensors and electronic boards from the 60 Hz industrial noises. Simultaneously, a layer of damping material (such as silicone) could be deposited on the plate surface for the sensor vibration protection.

The handgrip recognition algorithm can be used for verification too. In the preprocessing stage, the verification algorithm works similar to the recognition algorithm. The average waveforms in time domain and wavelet (*Haar*) domain would be calculated as the templates for each subject. The intra-class distances in time domain and wavelet

domain also would be calculated. Then, the mean intra-class distances and the standard deviations can be found. The thresholds would be decided in the preprocessing stage. Usually, they could be defined as fractions of the standard deviations of the intra-class distances.

During the verification stage, after the user claims his individuality, the input data would be compared to the template, and the distances in time domain and wavelet domain between the input data with the templates would be calculated. If either the distance in time domain, or the distance in wavelet domain, being greater than the threshold, this person would be rejected. Otherwise, the gun becomes operable.

To evaluate this verification algorithm, 50 samples were randomly picked from each subject as the template, and the other samples were used as tests. This process was repeated 20 times, with totally 12500 tests. Table 7.5 and Figure 7.7 give the false rejection rate (FRR) and false acceptance rate (FAR) versus different thresholds. The word "std" in Table 7.5 means the standard deviation of the distances.

Table 7.5 FRR And FAR Versus Different Thresholds

Threshold	FRR	FAR
0.25 × std	7.84%	13.12%
0.5 × std	5.60%	16.16%
0.75 × std	3.84%	22.24%
1.0 × std	3.20%	27.84%
1.25 × std	2.56%	31.04%
1.5 × std	1.92%	36.32%
1.75 × std	1.76%	43.20%
2.0 × std	1.44%	48.32%

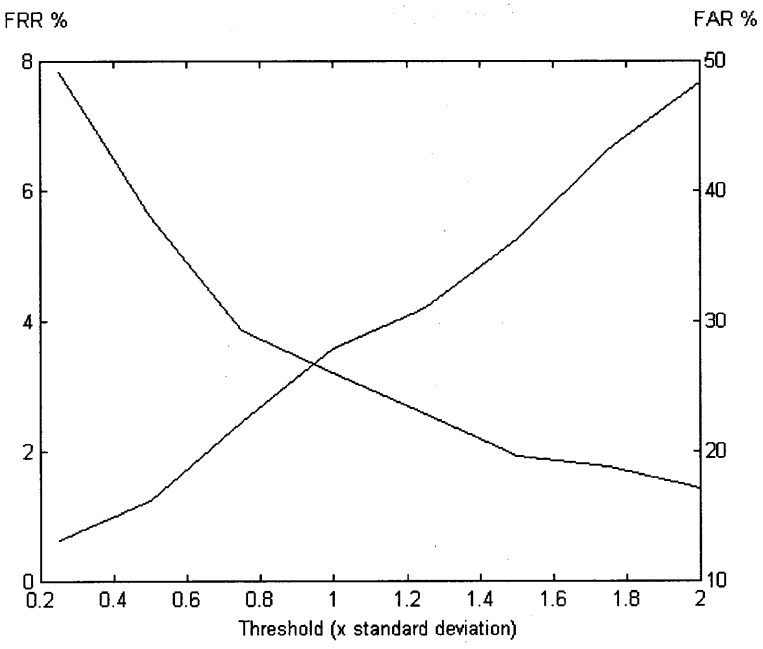


Figure 7.7 FRR and FAR versus different thresholds

7.6 Summary

A DSP box with 16 input channels and 4 output channels was built in this chapter so that the handgrip pattern recognition can be done in real time. A real handgun was used to do the handgrip pattern recognition. The handgrip pattern recognition algorithm was modified to be more suitable to the real time work.

To evaluate the performance of the realtime handgrip recognition system, five subjects participated into one experiment. The pressure sensors were put into the pockets, whose positions were specially designed to maximize the differences among these five subjects. A handgun simulator was used in this experiment to provide the force back. Each subject grasped the gun, aimed at different targets, and pulled the trigger 75 times. Among the 75 samples, 50 samples were randomly picked to build the pattern. The other 25 samples were used as the tests. This process was repeated 20 times. The average

success rate is about 90%. This result proves that the handgrip pattern recognition algorithm is a successful algorithm.

A modified handgrip verification algorithm was also presented in this chapter. Different false rejection rates and false acceptance rates were given versus different thresholds. Users can choose appropriate thresholds based on their application requirements.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

In this research, a new biometric identification technology is proposed to recognize the authorized users of handguns. This approach is based on the fingertip distributions and handgrip pressure dynamic variations during the firing process. Based on the study of numerous biometric identification technologies, handgrip pattern recognition is the most appropriate one for the smart gun.

Handgrip pattern recognition is a behavioral biometric technology. The behavioral biometric technology is based on the fact that human actions are predictable in the performance of repetitive, routine tasks. For an experienced firearm user, firing a gun is such a repetitive action. Like handwritten signature, a number of parameters inherent within the dynamic process make the handgrip unique for every person.

A prototype system was developed for this research. A digital camera was used to get the fingertip positions of 160 subjects. Then, these fingertip distributions were analyzed and divided into four natural groups. The static analysis shows that the fingertip placements of the firearm-experienced subjects such as police officers are more repeatable and have less variation than the fingertip placements of non-firearm-experienced subjects. The static analysis also shows that the most important parameter to classify different users into four natural groups is the horizontal position of ring finger, followed by the vertical position of the pinkie, and the vertical position of the thumb. A classification tree was built based on the static analysis to find the possible sensor positions.

The dynamic analysis of handgrip signals from 24 subjects was discussed. A handgrip pattern recognition algorithm was also proposed and evaluated. The average success recognition rate is 88.1%. Dividing the 24 subjects into 16 groups increases the success rate to 94.4%.

A plastic replica handgun was used for the preliminary analysis. However, a fake handgun does not have same weight with the real handgun. The fake gun can not give the user the feeling of a real firing either. To overcome these disadvantages, a real handgun was used. In addition, a DSP based handgrip recognition system was built to recognize the user in realtime. The handgrip pattern recognition algorithm was modified to be more suitable to the real time work. A figure of FRR and FAR versus threshold was given. User can choose the appropriate threshold based on their prefer FRR and FAR.

Some future works are expected. The realtime handgrip recognition system is based on a gun simulator. The next step would be collecting data from live firings. The time between the trigger being pulled and the hammer falling should be calculated. At the same time, the program running time of the handgrip recognition algorithm should be examined to make sure that there is enough time to make the decision. The mechanical lock should be carefully studied so that it can block the hammer in time if the user is recognized as an unauthorized user.

APPENDIX A

CLASSIFICATION TREES

The figures in Appendix A give the classification trees for each finger. Figure A.1 shows the classification tree for the thumb. Figure A.2 shows the classification tree for the middle finger. Figure A.3 shows the classification tree for the ring finger. Figure A.4 shows the classification tree for the pinkie finger.

Instead of showing the actual tree, classification trees can be explained by showing the actual partitions on the gun handle for each of the clusters. The figure with partition in Cartesian space for the thumb is shown in Figure A.5. The figure with partition in Cartesian space for the middle finger is shown in Figure A.6. The figure with partition in Cartesian space for the ring finger is shown in Figure A.7. The figure with partition in Cartesian space for the pinkie finger is shown in Figure A.8.

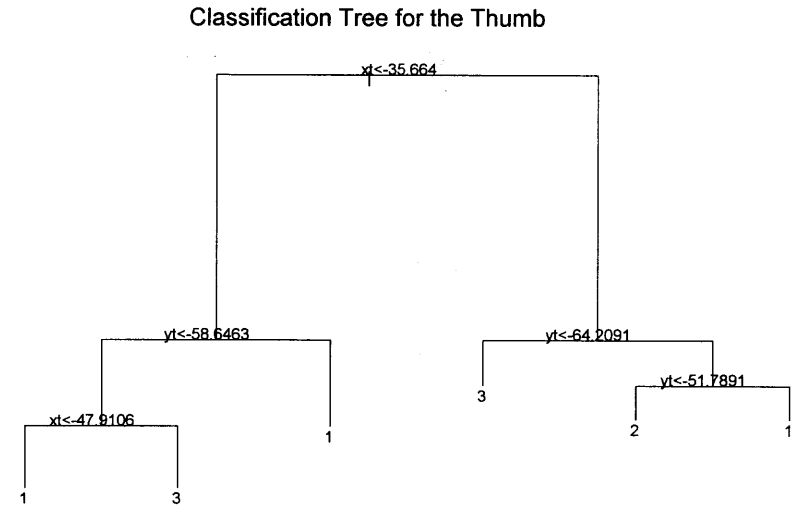


Figure A.1 Classification tree for the thumb.

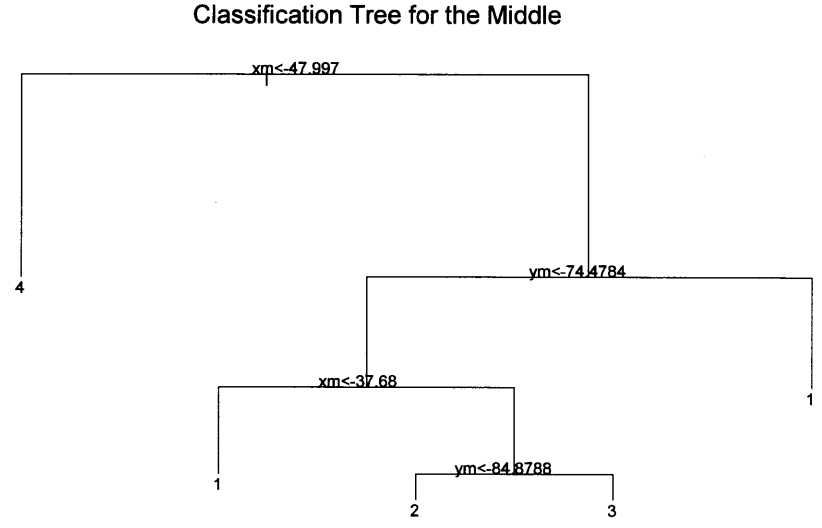


Figure A.2 Classification tree for the middle finger.

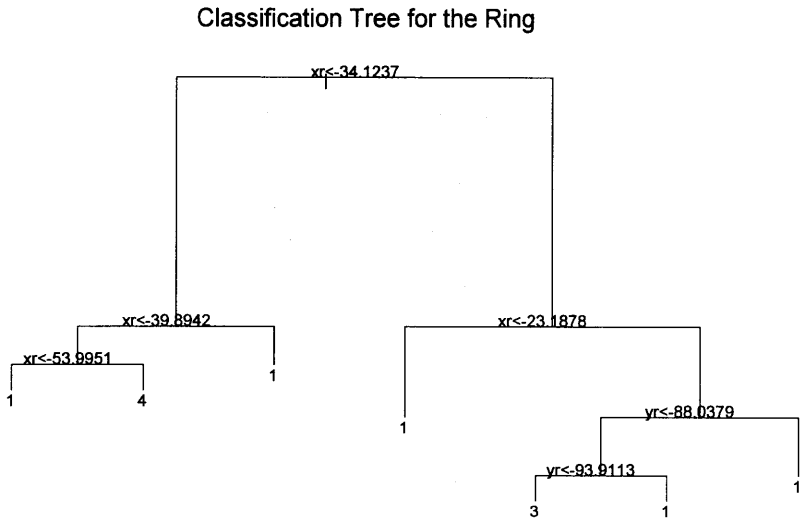


Figure A.3 Classification tree for the ring finger.

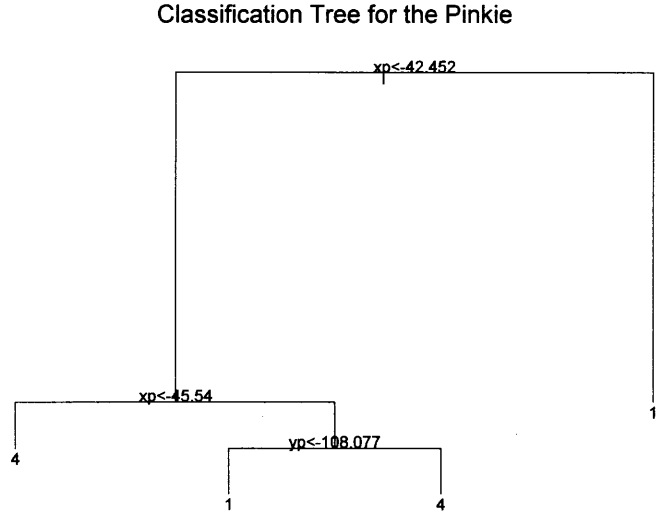


Figure A.4 Classification tree for the pinkie finger.

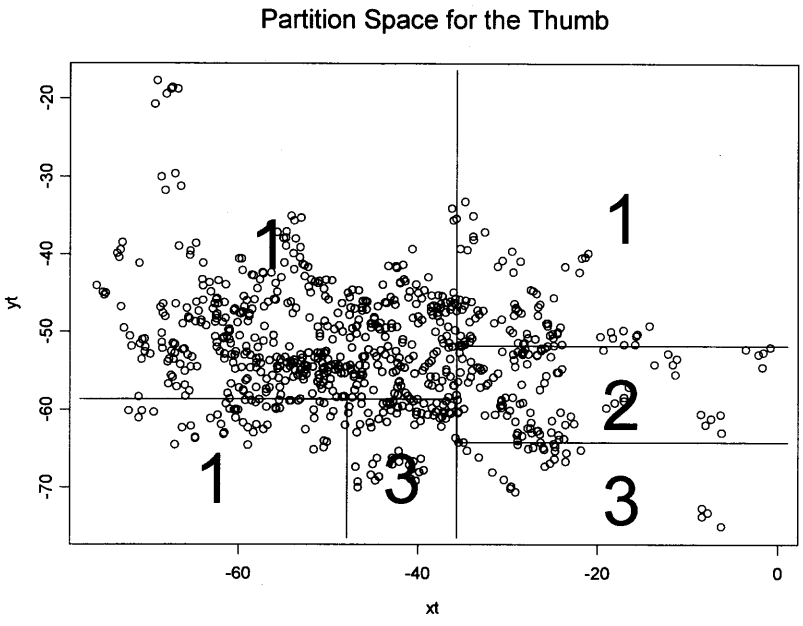


Figure A.5 The partition space for the thumb.

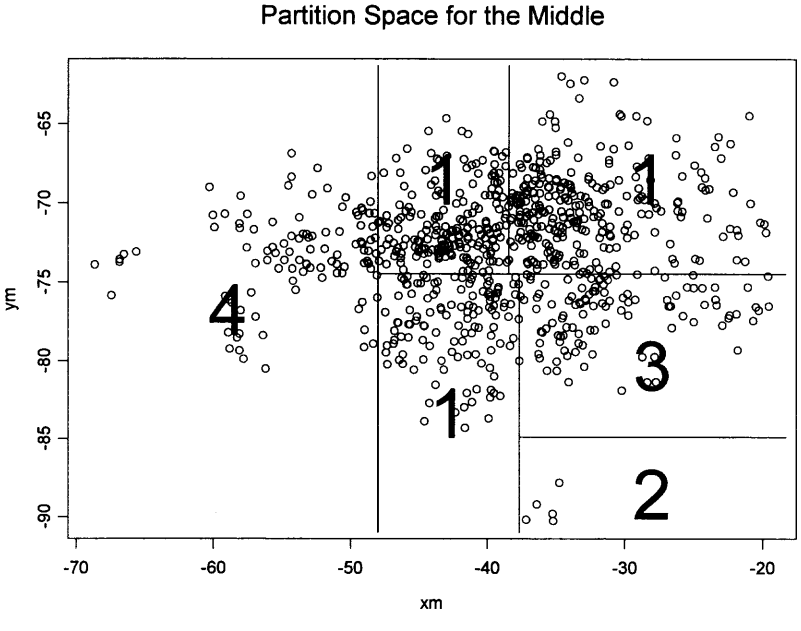


Figure A.6 The partition space for the middle finger.

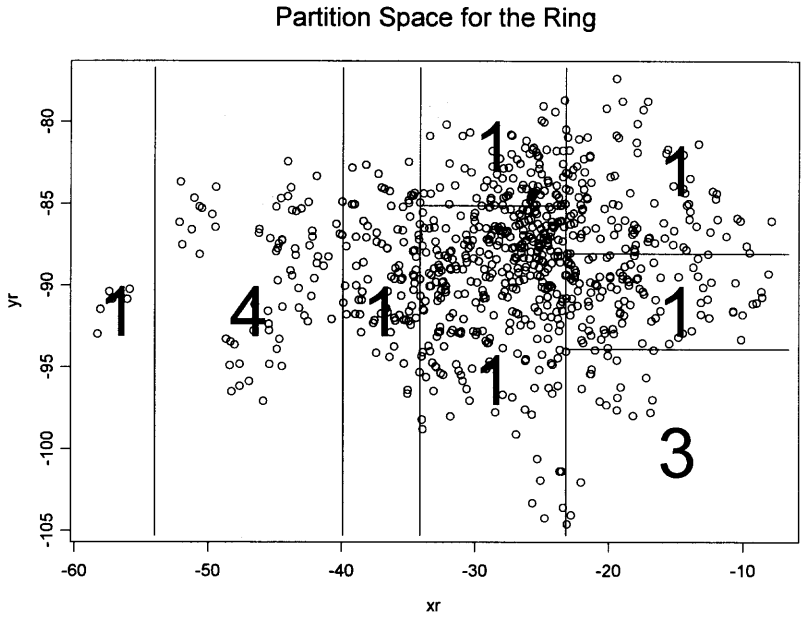


Figure A.7 The partition space for the ring finger.

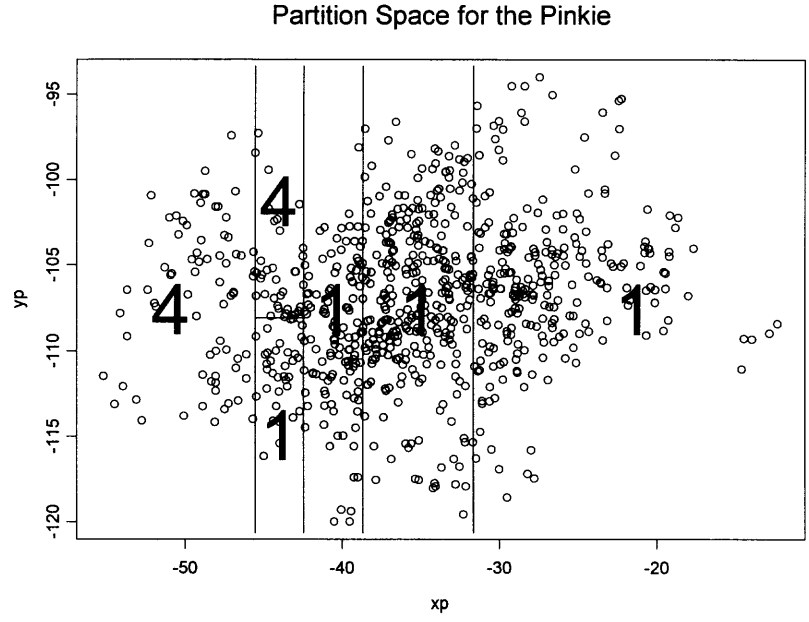


Figure A.8 The partition space for the pinkie finger.

APPENDIX B

HANDGRIP PATTERNS

The figures in Appendix B give the pressure patterns on all 15 channels for each subject who participated in the real-time handgrip pattern recognition experiment. Figure B.1 and Figure B.2 show patterns on all channels for subject 1. Figure B.3 and Figure B.4 show patterns on all channels for subject 2. Figure B.5 and Figure B.6 show patterns on all channels for subject 3. Figure B.7 and Figure B.8 show patterns on all channels for subject 4. Figure B.9 and Figure B.10 show patterns on all channels for subject 5.

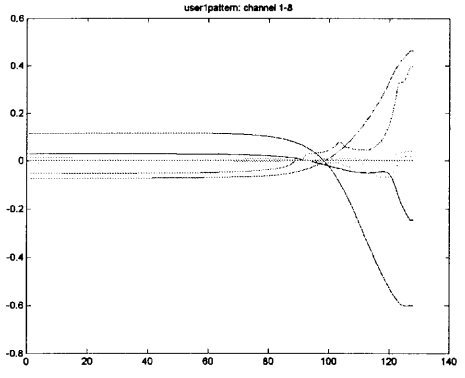


Figure B.1 User 1 pattern, channel 1-8.

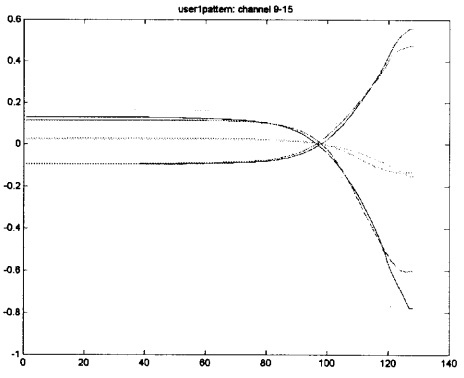


Figure B.2 User 1 pattern, channel 9-15.

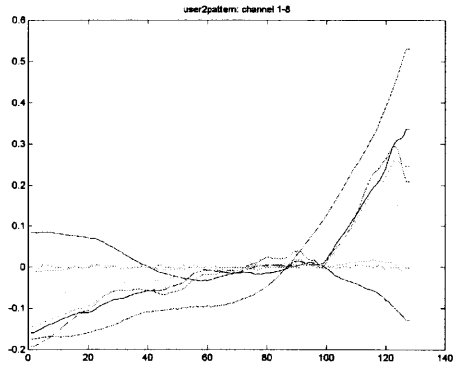


Figure B.3 User 2 pattern, channel 1-8.

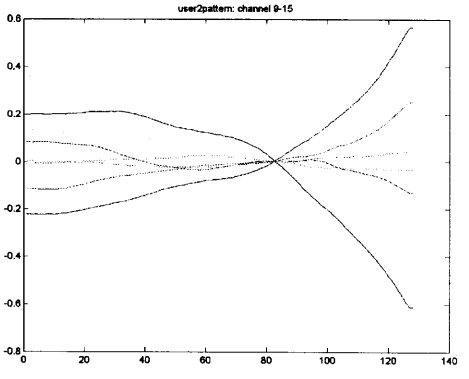


Figure B.4 User 2 pattern, channel 9-15.

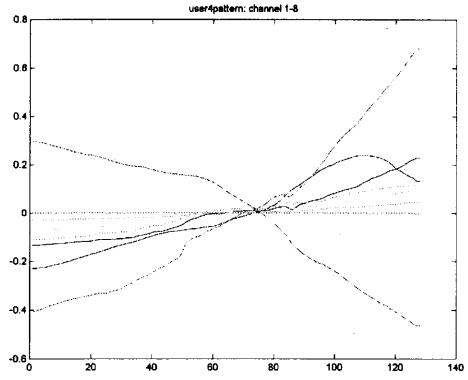


Figure B.5 User 3 pattern, channel 1-8.

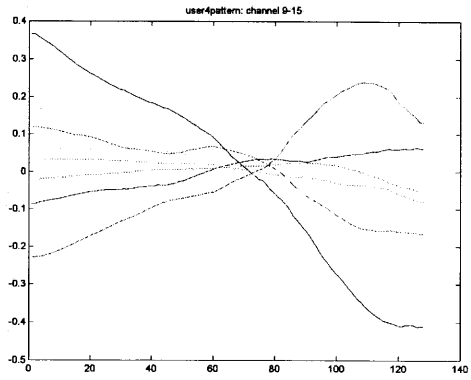


Figure B.6 User 3 pattern, channel 9-15.

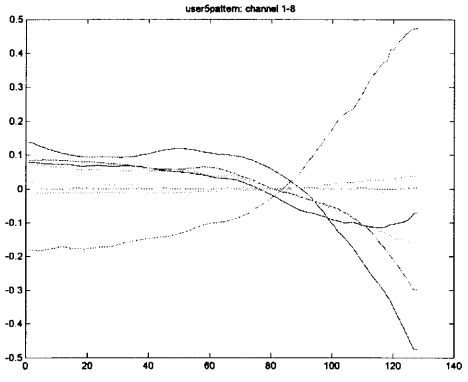


Figure B.7 User 4 pattern, channel 1-8.

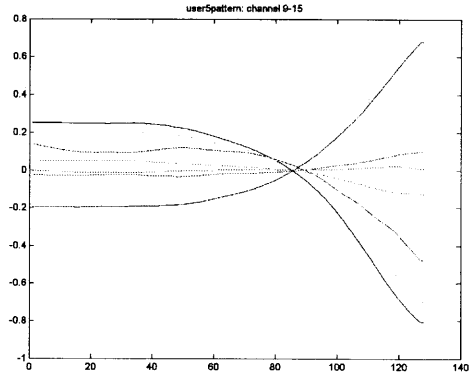


Figure B.8 User 4 pattern, channel 9-15.

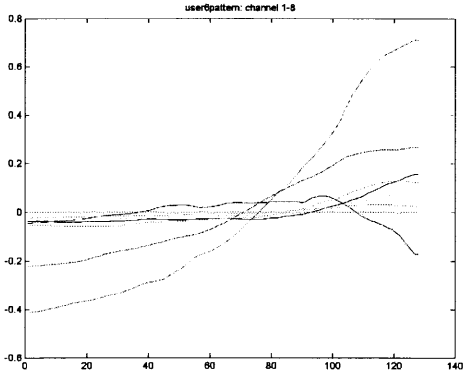


Figure B.9 User 5 pattern, channel 1-8.

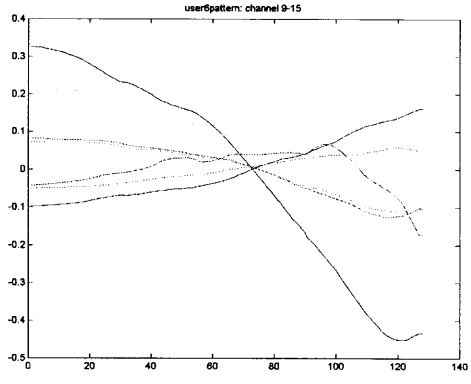


Figure B.10 User 5 pattern, channel 9-15.

REFERENCES

Anderberg, M. R. 1973. *Cluster Analysis for Applications*. Academic Press.

Bateman, A. 1989. *Digital Signal Processing Design*. Computer Science Press.

Box, G. E. P., and G. M. Jenkins. 1970. *Time Series Analysis, Forecasting, and Control*. Holden-Day Press.

Bryan, W. L., and N. Harter. 1973. "Studies in the Physiology and Psychology of the Telegraphic Language." *The Psychology of Skill: Three Studies*. New York Times Corporation.

Carmichael, J. W., J. A. George, and R. S. Julius. 1968. "Finding natural clusters." *Syst. Zool.* Vol. 17, pp. 144-150.

Carmichael, J. W., and P. H. A. Sneath. 1969. "Taxometric Maps." *Syst. Zool.* Vol. 18, pp. 402-415.

Chassaing, R., and D. W. Horning. 1989. *Digital Signal Processing with the TMS320C25*. A Wiley-Interscience Publication.

Cox, D. R., and H. D. Miller. 1968. *The Theory of Stochastic Processes*. Chapman and Hall Press.

Daubechies, I. 1988. "Orthonormal Bases of Compactly Supported Wavelets." *Comm. Pure and Applied Mathematics*. Vol. 41, pp. 909-996

Devasahayam, S. R. 2000. *Signals and Systems in Biomedical Engineering*. Kluwer Academic/Plenum Publisher.

Duda, R., and P. E. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley Press.

Everitt, B. S. 1993. *Cluster Analysis*, 3rd Edition. Halsted Press.

Fukunaga, F. 1990. *Introduction to Statistical Pattern Recognition*, 2nd Edition. Academic Press.

Gallant, S. I. 1990. "Perceptron based learning algorithms." *IEEE Transactions on Neural Networks*. Vol. 1(2), pp. 179-191.

Gitman, I. 1973. "An Algorithm for Nonsupervised Pattern Classification." *IEEE Transactions on Systems, Man, and Cybernetics*. SMC-3, pp. 66-74.

Gowda, C. K., and T. V. Ravi. 1995. "Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity." *Pattern Recognition.* Vol. 28(8), pp. 1277-1282.

Grimmett, G. R., and D. R. Stirzaker. 1982. *Probability and Random Process*. Clarendon Press.

Hall, A. V. 1967. "Methods for demonstrating resemblance in taxonomy and ecology." *Nature*. Vol. 214, pp. 830-831.

Hastie, T., E. Kishon, M. Clark, and J. Fan. 1991. "A Model for Signature Verification." *Proceeding of IEEE International Conference on Systems, Man, and Cybernetics.* pp. 191-196.

Hastie, T., and R. Tibshirani. 1996. "Discriminant adaptive nearest neighbor classification." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 18(6), pp. 607-616.

Huizinga, D. H. 1978. "A Natural or Mode Seeking Cluster Analysis Algorithm." *Technical Report 78-1.* Behavioral Research Institute, 2305 Canyon Blvd., Boulder, Colorado 80302.

Jain, A., L. Hong, and S. Pankanti. 2000. "Biometric Identification." *Communications of the ACM.* Vol. 43(2), pp. 91-98, 2000.

Jain, A., and S. Minut. 2002. "Hierarchical Kernel Fitting for Fingerprint Classification and Alignment." *Proceeding of International Conference on Pattern Recognition.* Quebec, August 11-15.

James, M. 1985. *Classification Algorithms*. John Wiley & Sons, Inc.

Johnson, S. C. 1967. "Hierarchical clustering schemes." *Psychometrika.* Vol. 32, pp. 830-831.

Joyce, R., and G. Gupta. 1990. "Identity Authentication Based on Keystroke Latencies." *Communications of the ACM.* Vol. 33, No. 2, pp. 168-176.

Lam, L., S. W. Lee, and C. Y. Suen. 1992. "Thinning methodologies: A comprehensive survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 14, No. 9, pp. 869-885.

MacNaughton-Smith, P., W. T. Williams, M. B. Dale, and L. G. Mockett. 1964. "Dissimilarity analysis." *Nature.* Vol. 202, pp. 1034-1035.

Mallat, S. 1989. "A theory for multiresolution signal decomposition: the wavelet representation." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 11, No. 7, pp. 674-693.

Miller, R. P. 1971. "Finger dimension comparison identification system." *U.S. Patent No. 3576538.*

Minsky, M. L., and S. A. Papert. 1988. *Perceptrons*, expanded edition. MIT Press, Cambridge, MA.

Mitra, S. K. 1998. *Digital Signal Processing, A Computer-Based Approach.* The McGraw-Hill Companies, Inc.

Monrose F., and A. Rubin. 1997. "Authentication via Keystroke Dynamics", *Proceedings of the 4th ACM conference on Computer and communications security*, pp. 48-56.

National Center for Health Statistics Mortality Data Tapes for number of deaths, 1998, U.S. Bureau of Census population estimates.

Papoulis, A. 1965. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Press.

Parzen, E. 1962. *Stochastic Processes*. Holden-Day Press.

Rosenblatt, F. 1958. "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review*. Vol. 65, pp. 386-408.

Roddy, A. R., and J. D. Stosz. 1999. "Fingerprint Feature Processing Techniques and Poroscopy." *Intelligent Biometric Techniques in Fingerprint and Face Recognition.* Edited by L. C. Jain, U. Halici, I. Hayashi, S. B. Lee, and S. Tsutsui. CRC Press.

Sankoff, D., and J.B. Kruskal. 1983. *Time warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison.* Addison-Wesley Publishing Company.

Shannon, C. E. A. 1948. "A mathematical theory of communication." *Bell Syst. Tech. Journal*. Vol 27, pp. 623-656.

Sidlauskas, D. P. 1988. "3D hand profile identification apparatus." *U.S. Patent No. 4736203.*

Sinauer, N., and J. L. Annest. 1996. "Unintentional, Nonfatal Firearm-Related Injuries: A Preventable Public Health Burden." *Journal of the American Medical Association*. Vol. 275, No. 22.

Theodoridis, S., and K. Koutroumbas. 1999. *Pattern Recognition.* Academic Press.

Umphress, D., and G. Williams. 1985. "Identity Verification Through Keyboard Characteristics." *International Journal Man-Machine Studies.* Vol. 23, pp. 263-273.

US General Accounting Office. 1991. "Accidental shootings: many deaths and injuries caused by firearms could be prevented -- report to the Chairman, Subcommittee on Antitrust, Monopolies, and Business Rights, Committee on the Judiciary, House of Representatives." Washington, DC: US General Accounting Office, 1991, report no. GAO/PEMD-91-9.

Wallace, C. S., and D. M. Boulton. 1968. "An elementary mathematical theory of classification and prediction." *Computer Journal.* Vol. 11, pp. 185-194.

Wishart, D. 1969. *Numerical Taxonomy.* Academic Press.

Xiao, Q., and H. Raafat. 1991. "Fingerprint Image Postprocessing: A Combined Statistical and Structural Approach." *Pattern Recognition.* Vol. 24, pp. 985-992.

Yaglom, A. M. 1962. *An Introduction to the Theory of Stationary Random Functions.* Prentice Hall Press.

Yasuara, M., and M. Oka. 1977. "Signature verfication experiment based on non-linear time alignment." *IEEE Trans. Sys. Man Cybernetics.* Vol. 17, pp. 212-216.

Ziemer, R. E., W. H. Tranter, and D. R. Fannin. 1993. *Signals and Systems: Continuous and Discrete.* Macmillan Publishing Company, NY.