

# **A Virtual Keyboard Implementation Based on Finger Recognition**

**Yang Zhang**

A thesis submitted to Auckland University of Technology  
in partial fulfillment of the requirements for the degree of  
Master of Computer and Information Sciences (MCIS)

2016

School of Engineering, Computer and Mathematical Sciences

# Abstract

A keyboard requires a great deal of resources and is restricted by its physical features. Additionally, discarded keyboards also inevitably contribute to environmental pollution. Consequently, the touch screen is designed to replace the physical keyboard and thus reduce these flaws. However, the internal digital keyboard on the touch screen takes up a substantial amount of space, which causes some content to be covered. Moreover, the touch screen can be dirtied by fingerprints and become worn over time by human fingernails through frequent use. Hence, it is necessary to develop a new type of environment-friendly virtual keyboard with fewer flaws.

In this thesis, the author develops a novel type of virtual keyboard that allows users to use fingers to type on a piece of paper at any fixed plane. Finger recognition is based on human skin tone and then the BWMORPH algorithm is utilized to recognize the user's fingertip. If the user's fingertip has remained on a key for a long time, the program will regard this key as an input. The experiments in this thesis adopt five kinds of customized paper keyboards on a wall to demonstrate the usability of the proposed virtual keyboard. Typing without touching the keyboard is fulfilled to ignore obstructions covering the paper keyboard. The experiment results indicate that the overall recognition rate of the proposed virtual keyboard is 94.62%.

The proposed virtual keyboard can be put to use for a smartphone in the future. Furthermore, as a blueprint, it can be applied to computers after it is improved to allow ten-finger recognition. Moreover, machine learning can be potentially embedded into our virtual keyboard so as to greatly improve its finger recognition performance.

**Keywords:** human-computer interface, virtual keyboard, finger recognition

# Table of Contents

Abstract .....	I
List of Figures .....	IV
List of Tables .....	V
Attestation of Authorship .....	VI
Acknowledgment .....	VII
Chapter 1 Introduction .....	1
1.1 Background and Motivation .....	1
1.2 Objective of This Thesis .....	3
1.3 Structure of This Thesis .....	7
1.4 Novelty of the Virtual Keyboard .....	8
Chapter 2 Literature Review .....	9
2.1 Introduction .....	9
2.2 Image Processing .....	10
2.2.1 Image Enhancement .....	11
2.2.2 Image Binarization .....	11
2.2.3 Image Smoothing .....	13
2.2.3.1 Neighborhood Averaging Method .....	13
2.2.3.2 Median Filtering Method .....	16
2.2.3.3 Multiple Images Averaging Method .....	18
2.2.3.4 Low-pass Filtering Method .....	18
2.3 Recognition Algorithms .....	21
2.3.1 Artificial Neural Networks .....	21
2.3.2 $k$ -Nearest Neighbors Algorithm .....	25
2.3.3 Principal Component Analysis .....	27
2.3.4 Hidden Markov Model .....	28
2.4 Existing Technologies and Applications .....	30
2.4.1 Gesture Recognition .....	30
2.4.2 Character Recognition .....	34
2.5 Novelty of Proposed Solution .....	35
Chapter 3 Research Methodology .....	36
3.1 Introduction .....	36
3.2 Related Study .....	37
3.3 Research Questions .....	38
3.4 Research Design and Experimental Environment .....	39
3.4.1 Virtual Keyboard Design .....	39
3.4.2 Architecture .....	41
3.4.3 Experimental Environment .....	45
3.5 Limitations of the Research .....	45
3.6 Expected Outcomes .....	46
Chapter 4 Research Findings .....	47
4.1 Introduction .....	47
4.2 Experimental Test Bed and Data Selection .....	47
4.3 Experiments .....	50
4.3.1 Experimental Algorithms .....	50

4.3.2	Experimental Output .....	51
4.4	Experimental Results.....	57
Chapter 5	Discussions .....	62
5.1	Introduction .....	62
5.2	Analysis and Evaluations .....	62
5.3	Justifications.....	64
Chapter 6	Conclusion and Future Work .....	66
6.1	Conclusion.....	66
6.2	Future Work.....	67
References	.....	68
Appendix	.....	78

## List of Figures

Figure 2.1	NAM schematic diagram.....	14
Figure 2.2	3×3 template.....	15
Figure 2.3	Overlapped sub-image.....	15
Figure 2.4	Image smoothing templates.....	15
Figure 2.5	The types of the sliding window.....	17
Figure 2.6	The flowchart of low-pass filtering method.....	19
Figure 2.7	An artificial neuron.....	22
Figure 2.8	The structure chart of the neural networks.....	24
Figure 2.9	An schematic diagram of $k$ -NN Algorithm.....	26
Figure 3.1	The flowchart of the fingertip recognition subsystem.....	42
Figure 3.2	The flowchart of the customized keyboard recognition subsystem.....	43
Figure 3.3	The flowchart of the virtual keyboard.....	44
Figure 4.1	Chosen virtual keyboards for tests.....	50
Figure 4.2	Example 1: customized keyboard recognition.....	52
Figure 4.3	Example 2: handwritten keyboard recognition.....	53
Figure 4.4	The binary frame.....	54
Figure 4.5	The skeleton image.....	54
Figure 4.6	Examples of successfully recognized typing.....	57

## List of Tables

Table 2.1	The comparison of low-pass filters.....	21
Table 4.1	Test results of keyboard (a) under lamp light.....	58
Table 4.2	Test results of keyboard (a) under natural light.....	58
Table 4.3	Test results of keyboard (b) under lamp light.....	59
Table 4.4	Test results of keyboard (b) under natural light.....	59
Table 4.5	Test results of keyboard (c) under lamp light.....	59
Table 4.6	Test results of keyboard (c) under natural light.....	60
Table 4.7	Test results of keyboard (d) under lamp light.....	60
Table 4.8	Test results of keyboard (d) under natural light.....	60
Table 4.9	Test results of keyboard (e) with occlusions.....	61

## Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted of the award of any other degree or diploma of a university or other institution of higher learning.

Signature:

Handwritten signature in Chinese characters, reading '张阳' (Zhang Yang).

Date: 25/06/2016

# Acknowledgment

This research was completed as the part of the Master of Computer and Information Sciences of the School of Computer and Mathematical Sciences in the Faculty of Design and Creative Technologies at the Auckland University of Technology (AUT) in New Zealand. I would like to thank my wife for her encouragement and also to deeply thank my parents for the financial support they provided during my entire period of academic study in Auckland.

My deepest thanks are to my primary supervisor, Dr. Wei Qi Yan. He provided me with much appreciated technological guidance when I lost direction and our discussions also gave me a lot of creative inspiration. I believe that I could not have been able to achieve my Master Degree without his invaluable help and supervision. In addition, I also want to appreciate my secondary supervisor, Professor Ajit Narayanan, who was also my instructor for the Artificial Intelligence course. He is an earnest and enthusiastic mentor who presented some issues in my research and also gave a lot of advice despite being very busy, and his Artificial Intelligence course laid the foundation for my research about finger recognition. Finally, I am also very grateful to my friends, Dr. Fan Liu, Mr. Guangyu Wang and Mr. Datong Gu for their great help of being testers in my experiments and for giving me their advice and suggestions regarding program and research.



# Chapter 1 Introduction

## 1.1 Background and Motivation

Computing machines have the ability to make our daily lives much more convenient. To control these machines, we generally need a controller equipped with a number of keys. A keyboard is an example of a controller that has a diverse range of keys. Nowadays, we are familiar with the various keyboards that control our digital machines such as computers, TV, phones and so on. Thus, as it is to be expected, the physical keyboard is currently the most common human-machine interaction device. However, these physical keyboards are not the ideal solution because an appropriate keyboard or controller must be designed for each specific machine. Additionally, the functions of each individual key must be learned and memorized. Furthermore, natural human communication is based on body language. In other words, we naturally transfer information through languages, poses, and gestures. However, machines do not have these biological characteristics and they also cannot understand our intentions. Thus, the question of how to forge a simple, natural and efficient way of human-machine interaction is, at present, a red hot spot for researchers.

In 1993, a remote controller called Charade was proposed to allow speakers to use free-hand gestures to control a slide or an overhead. This controller consisted of a Dataglove and a Dataglove Control. The system had a gestural command set so it could distinguish gestural commands from other casual gestures. This allowed the speaker to control the computer without a keyboard and mouse whilst interacting with the audience (Baudel & Beaudouin-Lafon, 1993). However, the disadvantages of this kind of controller are obvious: 1) the user needs to wear an uncomfortable extra device, 2) the user needs to learn the gestural command set and 3) the general recognition rate is roughly 94% but only around 78% for the first-time use.

Due to the inconvenience of an extra add-on such as a Dataglove, researchers considered using a camera to directly capture the shapes of hands. Segen and Kumar chose to focus on the shadow of the hand in an attempt to capture the 3D hand pose. The algorithm that Segen and Kumar proposed could extract the “peaks” and “valleys” of the fingers’ shadow and then compute the value to identify the hand pose (Segen & Kumar, 1999). Although their system got rid of the limitations of a mechanical input device, the system was restricted by the brightness, background and angle of the hand. For example, if the hand were too close to a table, the hand’s shadow would consequently become covered and so could not be captured.

The environment in real life is not as stable as the assumed experimental one. Thus, hand shadow recognition is actually difficult to apply in reality. Generally, a direct hand recognition method is necessary for most practical applications. A customizable gesture recognition system was developed to control a TV or TV set-top box by using a custom 3D depth camera. The 3D camera would capture hand images and convert them into greyscale images by using a combination of phase measurements and Time-of-Flight (TOF) measurements. The processed image would match the gesture conditions by checking the XML-based database. Then, an IR blaster would deliver the command signal from a computer to the TV or the set-top box. Moreover, the users could modify the XML file in the command database to fulfill the customization needs of the gesture commands. In the tests, the testers could turn the TV on/off, adjust the volume, mute and change channels by posing a gesture in front of the embedded 3D camera. Additionally, the breakthrough here was that the system was not sensitive to the environment and the delay was merely 0.5 second (Ionescu, Ionescu, Gadea & Islam, 2011).

Besides its application in people’s daily life, this kind of novel controlling method is also used in factories. As we all know, an increasing number of robot arms are used in the manufacturing industry. In the near future, there might be a lot of robots working for humans. It would be useful for these robots to have electronic eyes so that they can know our commands by our gestures. An automatic robot controlling system was

designed to recognize the gesture commands in front of a 3D depth camera. Their algorithm could convert captured images into black-and-white images and then crop the hand frame to exclude the wrist and arm by using boundary selection. After image standardization, the Principal Component Analysis (PCA) method was employed to compare the images with patterns recorded in the database to determine if it was a command. The accuracy of their system was 90% in environments with proper light arrangement but the accuracy decreased if the light was poor (Raheja, Shyam, Kumar & Prasad, 2010).

Although there are still many other gesture control methods and applications that cannot be introduced in this paper, it is obvious that gesture recognition is truly a very effective way to input a small command set. However, the accuracy of gesture recognition is sensitive to lighting and background. Moreover, complicated settings and gesture commands are not easy to learn for people who do not possess much computer knowledge. Overall, it seems that gesture recognition is not the perfect solution, especially for mass input or complex controls such as the typing of words or mathematical formula. This might be the reason why gesture recognition systems are still not widely used today.

As we all know, a keyboard takes up a large amount of space. In order to improve portability, an alternative to the physical keyboard must be found. A touch screen virtual keyboard is the most popular solution for portable devices such as iPads and smartphones. Although some computers also have a touchscreen, vertical typing is not comfortable due to human typing habits. Hence, researchers are still in the process of finding a better interface for human-computer interaction.

## **1.2 Objective of This Thesis**

Nowadays, a variety of computers, smartphones and mobile devices are produced to meet users' needs. Accordingly, how to improve human-computer interactions has

always been a hot topic in computer science. Smartphones and other mobile devices like iPads are portable and convenient because the screens are also the interfaces for human-computer interaction. However, the small size of their hardware also restricts heat emission so these portable devices generally do not have advanced CPU, GPU and huge RAM capacity (Rowley, Baluja & Kanade, 1998). Although a touch screen is a simple and effective human-computer interface, it also has limitations because the screen can be covered by the user's hand or dirtied by fingerprints, food and so on. Additionally, the screen will inevitably become worn out due to friction caused by touching the screen. Finally, it is very difficult for people to type by using multiple fingers on a touch screen. Thus, computers and laptops are still popular despite being quite big and heavy.

Unlike smartphones and iPads, desktops and laptops must make use of a keyboard to realize human-computer interaction. As a common and classic human-computer interface, though the keyboard is not very portable, users still generally accept it because it plays an important role. To solve the portability problem of the keyboard, typing on a touch screen might be a feasible solution but researchers are still trying to explore other ways to realize new input methods.

The keys of a traditional keyboard are the mechanical buttons. A projected virtual keyboard with a sensor module was developed to replace the traditional physical keyboard and thus greatly improve portability. This system, named Canesta Keyboard, was made up of an infrared light resource, a projector and a sensor module. The infrared light emits a plane slightly over the image of the keyboard and simultaneously the sensor module detects the point of intersection between the fingertips and the infrared light. Subsequently, algorithms will calculate the gathered data so a user can type on this virtual keyboard just like a traditional keyboard. In usability tests, the average typing rate was 38 words every minute and the average error rate was 3.7% (Roeber, Bacus & Tomasi, 2003). In addition, the user satisfaction rating was only 4 out of 5 because, due to the small size of the keyboard, users needed to frequently look at the keyboard when typing to make sure that they pressed the right key position.

A real-time gesture action detection system was proposed to realize an in-air typing interface. In the experiment, an add-on camera was fixed under the bottom of a smartphone to capture the fingertip as a cursor for use on a virtual keyboard displayed in the touch screen of a smartphone. A user could type words even by using a finger that was far away from the screen. The time-series scale finger data were cropped and used as training data. Subsequently, obtained eigenvectors were calculated by principal component analysis (PCA) and linear discriminant analysis (LDA) to detect the input actions (Ishijima, Ogawa, Higuchi & Komuro, 2014). When the motion is identified as an input action, the cursor position becomes the key that the user wants to input. Although this was a novel input method, the average accuracy of recognition was about 80% and testers spent 36 seconds on average attempting to type “good morning”. According to the feedback obtained, users felt unaccustomed to the in-air input interface.

In the same year, a multi-finger virtual input interface was also proposed for smartphones to allow a user to type in a space behind the mobile device by using augmented reality (AR) technology. The finger image was extracted by setting the HSV to skin color. Following this, the finger image would be drawn and superimposed on the virtual keyboard image (Sagara, Higuchi & Komuro, 2014). Calculating the vertical coordinate would identify the pressing motion. The pressing position is the key that a user wants to input. Overall, the system only recognizes key pressing actions and does not recognize hands and fingers. However, the user cannot see that superposition process so this system is actually still a blueprint that needs to be improved.

From the above, the projection keyboard used optical technology to provide an easy and effective solution to replace the traditional keyboard and is now already being sold on the internet. However, although this kind of virtual keyboard is portable and visually appealing, the infrared ray add-on costs a wealth of money and electric power so the vast majority of people still prefer to use traditional keyboards. As for the in-air virtual keyboard based on finger recognition, it has the potential to become a main trend in the future. However, current 3D projection technologies are not mature enough and the

limited computation speed of mobile devices is also an obstacle to hand recognition (Watson, 1993). Finally, the hand recognition algorithm is the most important issue that needs to be overcome because it has a substantial effect on the accuracy of typing recognition. The pre-existing research also does not provide an image of a keyboard under the user's hands, which made users feel unaccustomed and uncomfortable.

Enlightened by previous research, perhaps it will be beneficial to consider whether it will be possible to create a novel virtual keyboard that can assimilate some of the advantages and remove some of the disadvantages of previous virtual keyboard models. First of all, such a virtual keyboard must be visible and touchable to meet the user's need for comfort and habit. Secondly, it will be better if the virtual keyboard is flexible for different environments and requirements. For example, the virtual keyboard can be a keyboard image printed on a collapsible plastic board so that it can be portable and be put on rugged ground. The virtual keyboard can also be an image printed on paper so that it can be very light, replaceable, recyclable and can be pasted on a wall. Alternatively, the virtual keyboard can also be a projection keyboard that can be projected onto any surface. Furthermore, if a 3D projection technology is mature enough in the future, an in-air projected virtual keyboard also becomes a possibility. Thirdly, the traditional input method using mechanical keys will be replaced due to the nature of the virtual keyboard. Thus, finger detection and input recognition will be the key technologies for the virtual keyboard.

In this thesis, the author proposes a virtual keyboard based on finger recognition. The proposed virtual keyboard is flexible enough to meet different requirements. Usability experiments were carried out to demonstrate the feasibility of the virtual keyboard. In the experiments, the tester performed input motions on a paper keyboard and the computer was able to recognize the input keys in the video dataset.

## **1.3 Structure of This Thesis**

This thesis mainly consists of five chapters. The first part is to present a sound background for gesture recognition and the previous research concerned. The second part is related to research methodology, the research questions and the experimental design. The final part is to clearly expound the experimental process, in-depth result analysis and discussion.

Chapter 2 contains the comprehensive literature review of image processing and recognition technologies. In this chapter, the various types of widespread image processing and recognition algorithms are introduced in a highly-organised way. In addition, a number of existing applications and systems are also presented. This chapter also introduces a great number of existing virtual keyboard technologies and discusses their advantages and disadvantages.

Chapter 3 explains the research methodology of this thesis and the research questions clearly. Moreover, the main difficulty, the imaginative design and process of the experiment are disclosed.

Chapter 4 elaborates upon the results and outcomes of the experiments. The specific experimental environment, the customized paper keyboards, the chosen algorithms, the experimental implementation and experimental results are presented as well.

Chapter 5 is the in-depth discussions and analysis of the results and outcomes of the experiments. At last, the conclusion and future work of this research are presented in Chapter 6.

## 1.4 Novelty of the Virtual Keyboard

The most important novelty of the proposed virtual keyboard is that it is very environmentally friendly. It is necessary to spend massive amounts of resources to produce physical keyboards. Consequently, when these physical keyboards become old or broken they are discarded by users and thus generate significant amounts of waste, which is a heavy burden to our environment. However, the proposed virtual keyboard can be a keyboard printed on plain paper, a projection keyboard on a desk or a laser keyboard on a wall or on any oblique plane. Thus, it is unlikely that the proposed virtual keyboard will lead to environmental pollution.

Secondly, another advantage of the proposed virtual keyboard is that users can replace the virtual keyboard by themselves when they want a new one. It is simple for anyone to abandon the previous paper keyboard and print a new customized one.

Thirdly, thanks to the nature of the paper keyboard, users can affix it to any oblique plane and conveniently carry it anywhere. This kind of virtual keyboard might be used on a smartphone in future. Furthermore, as a blueprint, it can be applied to computers after further improvement to allow for ten-finger recognition.



# Chapter 2 Literature Review

## 2.1 Introduction

With the development of HCI technologies, the virtual keyboard inevitably became the direction towards which the future of human-computer interaction technologies would take. Mankind's demand for increased portability prompted keyboards to become multifunctional, lighter and smaller. Thus, similarly, the various demands of today's consumers are bound to drive the development of virtual keyboards until they inevitably surpass traditional mechanical keyboards in the future. However, currently there are still many problems and difficulties that hinder the implementation of a robust, all-round virtual keyboard system. Thus, to build a better virtual keyboard system, it is necessary to understand the existing challenges.

The fundamental technologies of a virtual keyboard include hand image acquisition, image processing and feature extraction, hand/fingers recognition, and input motion identification. This chapter introduces not only these main technologies but also the existing virtual keyboard systems and applications.

Up to now, a variety of intelligent recognition technologies have been implemented for the purpose of face or gesture recognition (Keskin, Erkan & Akarun, 2003). These technologies can also be applied to virtual keyboards since hand or finger recognition is the essential component of virtual keyboard systems. Hence, there are numerous references related to gesture recognition in the literature review. Moreover, as the core part of gesture recognition technologies, artificial intelligence algorithms are also of great value. Consequently, gesture recognition algorithms are also introduced in the literature review.

## 2.2 Image Processing

For human beings, vision is the primary means of obtaining information from nature. According to statistics, visual information accounts for about 60% of all information humanly obtained, whilst auditory information accounts for approximately 20% and other methods account for a total of around 20% (Pan & Yang, 2009). Under these circumstances, it becomes obvious that visual information holds the most importance for humans. Images serve as the primary means through which humans acquire visual information. Therefore, it can be said that images are the most important, richest, and largest course of information that humans generally tend to experience (Zelevnik & Forsberg, 1999).

In the past, objects were filmed by optical mechanical cameras and made directly into physical photographs that can hardly be processed again (Pang, Ozawa & Kasabov, 2005). Nowadays, three-dimensional objects are generally captured by digital cameras and then turned into two-dimension digital images or even three-dimensional images. In addition, they can be processed, analyzed and modified further by computers according to the needs of humans. Digital images are actually bitmaps obtained after continuous analog images are processed by discretization. Strictly speaking, a digital image is a two-dimensional function that executes equal interval quantizing on the amplitude after the equidistance rectangular grid sampling. Simply put, it is a quantified two-dimensional sampling array. Digital image processing is a course that creates images that meet the needs of human vision or other receiving systems through utilizing a certain algorithm to handle digital images (Roy, Govil & Miranda, 1995). The foundations of image processing are various mathematics-based algorithm designs and implementations.

### **2.2.1 Image Enhancement**

Image enhancement means emphasizing and improving an image's features such as edges, outlines, contrast and so on. Thus, changing the dynamic range of these details enhances the visual effect and image intelligibility. The two main method types of image enhancement are spatial domain methods and frequency domain methods (Chen & Koskela, 2013). The spatial domain refers to the image plane itself, which is based on the direct processing of the image pixels. Frequency domain processing technologies, such as Fourier transform, DCT transform, and Wavelet transform, are based on modifying the frequency of transformed images (Lee & Tanaka, 2013).

Image enhancement is generally able to stretch the gray value range to the 0~255 gray level, thereby boosting the contrast and quality of the images (Athitsos, Wang & Stefan , 2010). The analysis and process of the image enhancement is based on the gray level histogram. Histogram equalization can boost the contrast of the whole image and increase the discrimination degree of the image (Zhang & Feng, 2013).

### **2.2.2 Image Binarization**

Image binarization is an important technology in image pre-processing, which is widely utilized in pattern recognition, Optical Character Recognition (OCR), medical imaging and so on (Nawaz, Mian & Habib, 2008). It is a common method that takes advantage of gray level transformation to research grayscale images.

Image binarization is to divide all pixels of a grayscale image into two groups according to a certain threshold and then assign 0 and 255 to pixels in these two groups respectively (Billon, Nedelec & Tisseau, 2008). Suppose that the function of an input image is  $f(x, y)$  and the function of the output image is  $g(x, y)$ , then the equation can be shown as equation (2.1),

$$g(x, y) = \begin{cases} 0 & f(x, y) < Threshold \\ 255 & f(x, y) \geq Threshold \end{cases} \quad (2.1)$$

where *Threshold* is a scaleplate that separates the foreground and the background regions. Hence, a proper threshold is able to not only maintain the useful information of an image but also decrease the interference of background and noise as much as possible.

Image binarization plays a vital role in practical digital image processing because a vast majority of image processing systems are based on binary images (Megiddo, 1983). A binary image has only two grayscale levels so its unitary attribute is not involved in the multiple values of the pixels, which not only needs less memory space but also decreases the amount of image processing and compression. Hence, binary images make image processing much easier and faster. The important thing is that, based on binary images, further image processing can be implemented to acquire more geometrical features or other features.

Binarization of color images generally adopts three methods besides the threshold method. These methods are the maximum method, averaging method and weighted averaging method.

1) Maximum method: the RGB value of each pixel is equal to the maximum of its RGB values.

$$R := G := B := \text{Max}\{R, G, B\} \quad (2.2)$$

2) Averaging method: the RGB value of each pixel is equal to the average of its RGB values.

$$R := G := B := (R + G + B)/3 \quad (2.3)$$

3) Weighted averaging method: the RGB value of each pixel is equal to the average of the weighted RGB values of every pixel.

$$R := G := B := (\omega_r \cdot R + \omega_g \cdot G + \omega_b \cdot B)/3 \quad (2.4)$$

where  $\omega_r, \omega_g, \omega_b$  are the weighted number and are greater than 0.

### 2.2.3 Image Smoothing

There are many kinds of noises in images, and the influence of noises on the signal amplitude and phase position of images is very complicated (Lech & Kostek, 2010). Noises are usually intertwined with the signals, especially multiplicative noises. The main purpose of image smoothing is to weaken the noises. However, if an improper smoothing method is adopted, it would make the image more blurry. Thus, in order to decrease the noises in images, it is essential to adopt various methods for specific circumstances or it can be difficult to get a satisfying effect. No matter what, the image smoothing process always removes some of the details in the image. Hence, the main task of image smoothing is always to find a method that can keep as many details as possible while doing the best to filter out noises (Lawrence & Platt, 2004).

Image smoothing has two effects (Nandakumar, Wan, Chan, Wang & Yau, 2013). The first effect is to remove or decrease the noise in images to improve the quality of images; the second effect is to blur the images so the images look soft and natural. Image smoothing can operate in the spatial domain or frequency domain. The common methods in spatial domain are the Neighborhood Averaging Method (NAM), median filtering method, multiple images averaging method and so on. For the frequency domain, the common methods are Ideal Low-Pass Filter (ILPF), Butterworth low-pass filter and so on. The main propose of image smoothing is to remove additive noises. The additive noises are mainly the pulse noise, Gaussian noise and so on (Kumar & Segen, 1999).

#### 2.2.3.1 Neighborhood Averaging Method

The NAM is to select a neighborhood area  $R$  surrounding each pixel  $(x, y)$  in the original image, then calculate the average of the grayscale level of all pixels in area  $R$

and assign the average to the corresponding pixel  $(x, y)$ . The schematic diagram of the NAM is shown in Figure 2.1.

$$g(x, y) = [f(x, y) + \sum_{(i, j) \in R} f(i, j)] / (m + 1) \quad (2.5)$$

where  $g(x, y)$  is the smoothed value of the central pixel,  $f(x, y)$  is its original pixel,  $f(i, j)$  is the value of a neighbor pixel and  $m$  is the number of the neighbor pixels.

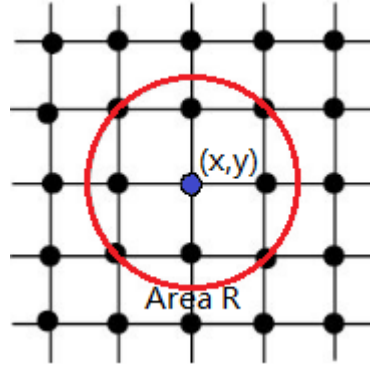


Figure 2.1 NAM schematic diagram

The NAM is generally implemented by means of the convolution of the template (Kurakin, Zhang & Liu, 2012). An image or a template can be regarded as a matrix so when the convolution is being calculated, the extension and rollover are done first and then the sum of products is calculated. However, it is quite complicated and the computation burden will be increased a lot if a image or a template is increased. Considering that the common templates are fully symmetric, the templates after rollover is similar to the original ones. Thus, the convolution of a template and an image can be proceeded as the following steps:

**Step 1.** While the template shown as Figure 2.2 is moving in an image from left to right and from top to bottom, the coefficients in the template overlaps some pixels in the sub-image shown as Figure 2.3;

**Step 2.** Each coefficient in the template multiplies the value of the gray level of each pixel;

**Step 3.** Calculate the sum of these products;

**Step 4.** Assign the sum to the pixel  $I$  that overlaps the central point in the template.

$$I(x, y) = \lambda_1 a_1 + \lambda_2 a_2 + \dots + \lambda_9 a_9 \quad (2.6)$$

where  $I(x, y)$  is the value of the pixel  $I$ .

$\lambda_1$	$\lambda_2$	$\lambda_3$
$\lambda_4$	$\lambda_5$	$\lambda_6$
$\lambda_7$	$\lambda_8$	$\lambda_9$

Figure 2.2 3×3 template

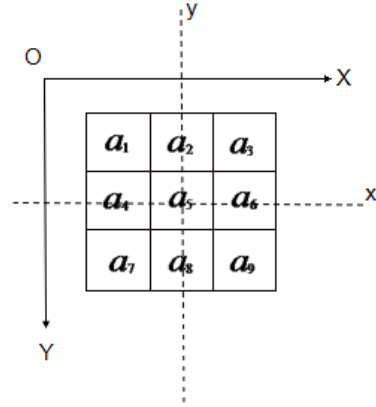


Figure 2.3 Overlapped sub-image

There are various image smoothing templates shown in Figure 2.4. After the image smoothing, the details and edges in the image become vague to different extents. Among the templates of the same size, the box template has the strongest image smoothing effect. For the same template, the larger the radius that the neighborhood area has, the stronger the image smoothing effect (Kumar, Mitchell & Yildirim, 2003).

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(a) Box Template

$$\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

(b) 4 Neighbors Averaging Template

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

(c) Gaussian Template

$$\frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(d) Weighted Averaging Template

Figure 2.4 Image smoothing templates

Smoothing template can filter some of the noise in an image. Generally, the neighborhood averaging method is combined with Threshold Method to not only

weaken the blurring effect but also abate the noise as well (Tsang, Kwok & Cheung, 2005). For example, when the difference value of the gray level value of a pixel and the gray level average of its neighborhood pixels is less than the threshold value  $T$ , the grayscale level value of this pixel stays the same. If the difference value greatly exceeds the threshold value  $T$ , it is likely to be noise. Thus, the gray level value of this pixel should be assigned to the grayscale level average of its neighborhood pixels.

In actual processing procedure, it is important to select a threshold value. If the threshold value is too large, it weakens the effect of the noise filtering. If the threshold value is too small, it decreases the inhibitory effect of the blurring effect. In addition, it is also very important to choose appropriate size, shape and weighted values in a template because that can overcome the simple local averaging effectively (Jeong, Jin, Song, Kwon & Jeon, 2012).

#### **2.2.3.2 Median Filtering Method**

Median filtering method is a nonlinear processing technology that need not know the statistic characteristics of the images in the actual operation process (Jie, Jinfang, Guangjin & Xiudong, 2011). It is known that the neighborhood averaging method can blur an image when smoothing the image but the median filtering method can overcome the detail blurring effect and maintain the edges of an image under certain conditions. Moreover, it is better used for filtering the impulse interference and image scanning noises. However, this method is not suitable for the images that contain too many effective tiny details such as points, small pinnacles or thin lines (Wu, Pan, Zhang, Qi & Li, 2009).

The operations of the median filtering method are sliding a window that can contain an odd number of pixels in an image and then assigning the median of the gray level value of the pixels in the window to the central pixel (Utsumi & Ohya, 1999). The examples of the window are shown as Figure 2.5.



$$f_i = Med\{f_{i-m}, f_{i-m+1}, \dots, f_{i+m}\} \quad (2.7)$$

where  $i$  is the number of the central pixel and  $m$  is the width of the array of the pixels in the window.

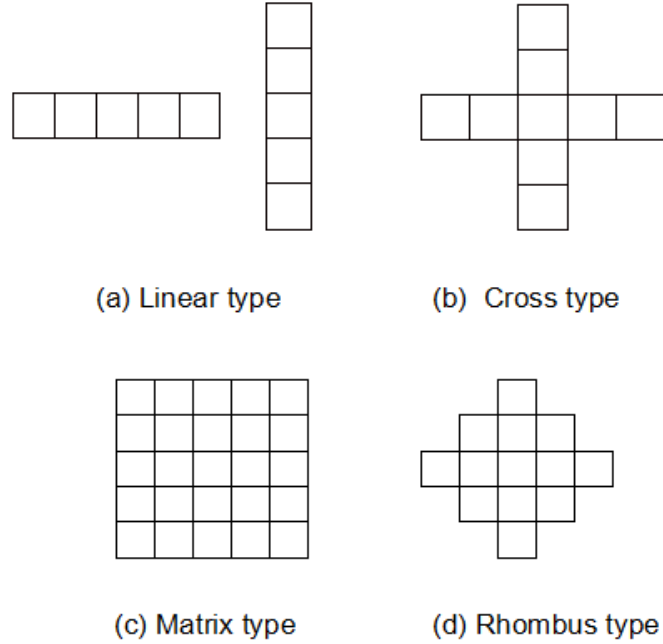


Figure 2.5 The types of the sliding window

The median filtering method can maintain the discrete transition signals and ramp signals and slightly smooth the peak of the discrete triangular signals. Different types of windows can produce different effects. If the window is centrosymmetric, the median filtering can maintain the jump edges in an arbitrary direction. For the images containing many long outlines, the matrix type or rhombus type is relatively appropriate. For the images containing a lot of pinnacles and peaks, the cross type is relatively suitable. Generally, it is difficult to ascertain the optimal size and type of the window at the beginning of the experiments (Youyi & Xiao, 2010).

All in all, the main function of the median filtering method is to smooth the gray level value of a pixel that is very different from its surrounding pixels to remove the independent noise point (Jonathan, 1995). It generates less vagueness than the neighborhood averaging method because it does not calculate the grayscale level

average.

### 2.2.3.3 Multiple Images Averaging Method

Multiple images averaging method is also referred to as interframe smoothing which averages the grayscale level values of the multiple images of the same thing to restrain the noises in the image acquisition (Yu, Schwaighofer, Tresp, Ma & Zhang, 2003).

When the images are being captured, transmitted and processed, the noise will come into being. Suppose that the noise is additive, an image can be present as

$$g_i(x, y) = f(x, y) + n_i(x, y) \quad (2.8)$$

where  $g_i(x, y)$  is the image acquisition containing noise,  $f(x, y)$  is the original image,  $n_i(x, y)$  is the noise.

Suppose the noise of each pixel is not relevant and the average is 0,

$$\bar{g}(x, y) = \frac{1}{m} \sum_{i=1}^m g_i(x, y) = f(x, y) + \frac{1}{m} \sum_{i=1}^m n_i(x, y) \quad (2.9)$$

where  $m$  is the number of the multiple images of the same thing,

$$E\{\bar{g}(x, y)\} = f(x, y) \quad (2.10)$$

$$\sigma_{\bar{g}(x, y)}^2 = \frac{1}{m} \sigma_{g_i(x, y)}^2 = \frac{1}{m} \sigma_{n_i(x, y)}^2 \quad (2.11)$$

It shows that the multiple images averaging method can reduce the variance  $\sigma$  of the noise to  $1/m$  of the original value. Thus, the more images of the same thing, the less noise after processing (Intrator & Edelman, 1998).

### 2.2.3.4 Low-pass Filtering Method

Generally, most noise concentrates in the high frequency band. Thus, the main effect of the low-pass filter is to remove the high frequency signals to fulfill the removal of the noise in an image (Hemayed, 2003). However, it also causes the loss of details in the

high frequency band. Suppose that an original noisy image is  $f(x, y)$  and  $F(u, v)$  is its Fast Fourier Transformation (FFT),  $G(u, v)$  is the FFT that is low-pass filtered by the transfer function  $H(u, v)$ . After the inverse Fourier transformation, the expected image is  $g(x, y)$ . The flowchart of Low-pass Filtering Method is shown as Figure 2.6.

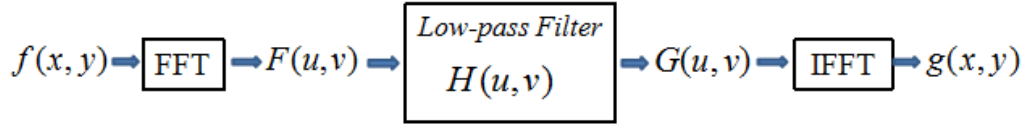


Figure 2.6 The flowchart of low-pass filtering method

According to the different chosen Low-pass filters, there are a few types of low-pass filter methods.

#### *Ideal Low-pass Filter (ILPF)*

The transfer function of ILPF is,

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases} \quad (2.12)$$

where  $D_0$  is a chosen non-negative number called cut-off frequency, and  $D(u, v)$  means the distance between the origin of the frequency coordinates and point  $(u, v)$ .

$$D(u, v) = \sqrt{u^2 + v^2} \quad (2.13)$$

Theoretically, ideal low-pass filter has the best effect of image smoothing. However, it also leads to serious blur and ringing phenomenon (Hu & Gao, 2011). That is because  $H(u, v)$  suddenly changes from 1 to 0 at  $D_0$ , which manifests as the concentric ring in the spatial domain. In addition, the smaller  $D_0$ , the more concentric rings. The most important problem is that the ideal low-pass filter cannot be implemented by physical

devices (Xue, Liao & Carin, 2007).

#### *Butterworth Low-pass Filter (BLPF)*

The transfer function of  $n$ -order BLPF is,

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{-2n}} \quad (2.14)$$

BLPF is also known as maximally flat filter. Compared with ILPF, there is no obvious discontinuity between pass band and stop band so that it does not lead to serious blur and ringing phenomenon. However, its image smoothing effect is worse than ILPF because it keeps more high frequency than ILPF (Hsieh, Liou & Lee, 2010).

#### *Exponential Low-pass Filter (ELPF)*

The transfer function of  $n$ -order ELPF is

$$H(u, v) = e^{-[D_0 / D(u, v)]^{-n}} \quad (2.15)$$

When  $D(u, v) = D_0$  and  $n = 1$ ,  $H(u, v) = \frac{1}{e}$ . ELPF has quite a smooth transition so it does not result in the ringing phenomenon. Compared with BLPF, it has stronger attenuation characteristic so the images processed by ELPF are slightly more vague than the ones processed by BLPF (Cheng, Luo, & Chen, 2014).

#### *Trapezoidal Low-pass Filter (TLPF)*

The transfer function of TLPF is,

$$H(u, v) = \begin{cases} 1 & D(u, v) < D_1 \\ [D(u, v) - D_1] / [D_0 - D_1] & D_1 \leq D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases} \quad (2.16)$$

where  $D_1$  is another cut-off frequency and  $D_0 < D_1$ . Generally, the first turning point of

$H(u, v)$  is chosen as  $D_1$ .

In summary, the comparison of the performances of these four types of low-pass filters are shown as Table 2.1.

Table 2.1 The comparison of low-pass filters

<i>Low-pass Filter</i>	<i>Ringing</i>	<i>Image blur degree</i>	<i>Smoothing effect</i>
ILPF	Serious	Serious	Best
TLPF	Slight	Slight	Good
ELPF	No	Slighter	Average
BLPF	No	Slightest	Average

## 2.3 Recognition Algorithms

### 2.3.1 Artificial Neural Networks

Artificial neural networks are a kind of mathematical model that simulate animals' neural structure of the brain's synapses and connections to implement distributed parallel information processing (Chapelle, Scholkopf & Zien, 2006). This algorithm model is made up of a great number of interconnected nodes, each of which is a specific output function, known as the activation function. Between every two nodes, there is a weighted value for every input signal, which is like a memory in the neural network. An artificial neuron is shown as Figure 2.7. Thus, the outputs of the artificial neural networks are based on the connection methods, weighted values and activation functions (Cortes & Vapnik, 1995).

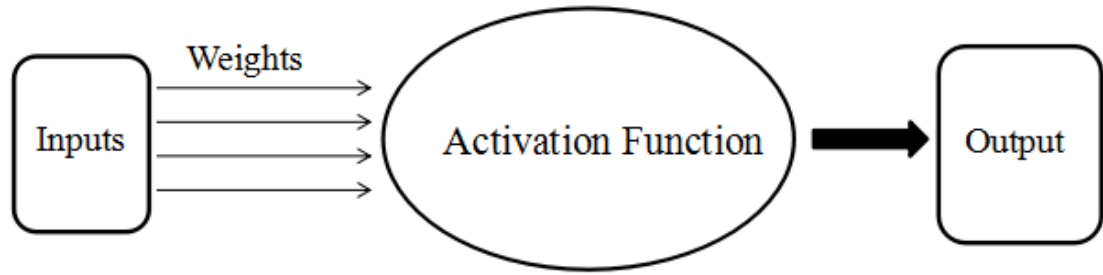


Figure 2.7 An artificial neuron

Artificial neural networks are nonlinear, adaptive information processing systems that have four essential characteristics (Adachi, Furuya, Greene & Mikuriya, 1991). The first characteristic is non-linearity. The nonlinear relationship is a common characteristic of nature. The wisdom of the brain is a kind of nonlinear phenomenon. A neuron has two statuses, activated or inhibited, which is a kind of nonlinear relationship in mathematics. A neural network that contains thresholds can improve fault tolerance and storage capacity (Silver & Mercer, 2001). The second is unlimitedness. An artificial neural network consists of many interconnected neurons. The overall behavior of a system does not depend on the feature of a single neuron but is mainly determined by the interaction and interconnection of the neurons (Eaton, 2006). Artificial neural networks simulate the unlimitedness of the brain through a great number of interconnections between neurons. Associative memory is a typical example of unlimitedness. The third is undeterminedness. Artificial neural networks are self-adaptive, self-organizing, self-learning. The information an artificial neural network processes has a variety of changes. In addition, the nonlinear dynamic system itself is also changing when processing information. The iterative process is often used to describe the evolution of the dynamical system. The last one is non-convexity. The evolution direction of a system depends on a particular state function under a certain condition. Non-convexity means that a function has multiple extrema so that the system has a number of relatively stable equilibriums, which leads to the diversity of the system evolution (Abu-Mostafa, 1989).

In artificial neural networks, neurons can represent different objects such as letters, features, concepts or meaningful abstract patterns (Silver & Poirier, 2004). Generally, a standard artificial neural network has three layers shown as Figure 2.8. The input layer contains a large number of neurons that receive a mass of input information known as input vector. The output layer produces the outcomes after the information is transmitted, analyzed, and weighted in the links of neurons (Baxter, 2000). The outcome is known as output vector. The hidden layer is the one that contains a great number of neurons and interconnections between the input layer and output layer. The hidden layer can also have multiple layers but it usually has only one layer. The larger the number of the neurons in the hidden layer, the more significant the non-linearity of the artificial neural network (Argyriou, Maurer & Pontil, 2008).

The architecture of neural networks and the type of training algorithm play a vital role in the performance of the neural networks for classification. The artificial neural network has great learning and memory ability. Learning is achieved by adjusting the weight value of the massive neurons after dataset training. Memory is embodied in producing the corresponding output based on the input and the weighted neural structure. Thanks to the robustness and strong learning ability, the neural network possesses a promising ability to solve the interacting features (RongQing, WenHui, Duo & Hua, 2008).

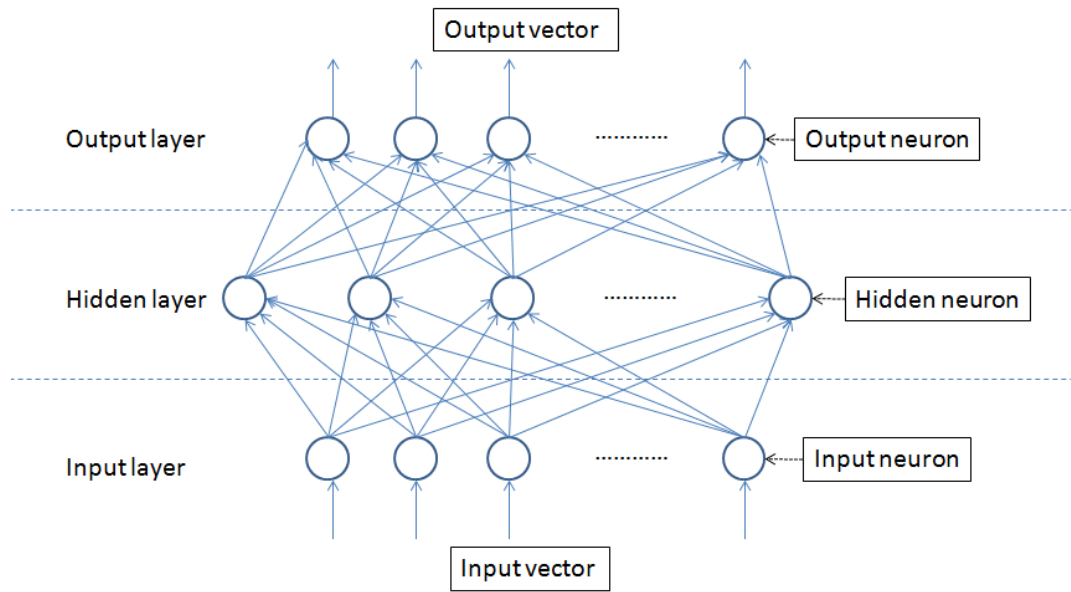


Figure 2.8 The structure chart of the neural networks

To understand performances of the neural networks on face recognition, Badal investigated human face recognition using neural networks based on a training database. He found that the rate of face recognition is reasonable and acceptable if the neural networks is trained based on an existing data set that only contains the front and profile images. In addition, different positions of the face images have a great impact on the accuracy of face recognition. In addition, the accuracy of face recognition can be improved by 4%~8% even if the training set is enlarged by only a few additional face views (Badal, 1993).

According to Xu (2006) , hand gesture recognition becomes especially challenging when the hand gestures involve high degrees of freedom. This added difficulty in developing a functional hand gesture recognition system is especially relevant when it comes to virtual reality driving training systems since the hand gestures involved in driving all have high degrees of freedom. In confronting the task of developing a hand gesture recognition system that would be suitable for a virtual reality driving training system, Xu opted to use a neural network approach. The reason this approach was chosen was because neural networks made it possible to directly map raw data to the



required hand gestures. In his paper, Xu utilized BP neural networks to recognize hand patterns contained within the raw sensor data captured by the DataGlove. To prove that this system could function properly and effectively 300 hand gestures were then used to simultaneously test and train the neural network. The results generated from these sessions demonstrated a recognition accuracy of 98.0% and proved that the neural network could be effectively used for a virtual reality driving training system.

### 2.3.2 *k*-Nearest Neighbors Algorithm

*k*-nearest neighbors algorithm (*k*-NN for short) is a non-parametric machine learning method that is widely used for pattern recognition, data mining, signal processing and regression (Chan, 2000). In *k*-NN classification, *k* is a positive value that means the *k* closest training examples were selected to train the sample set. The core concept of *k*-NN algorithm is that a sample  $x_i$  in the feature space is assigned to the class that most of its *k* closest training examples belong to. For example as Figure 2.9, the sample  $x_i$  is in a data set containing 3 classes  $\omega_1, \omega_2, \omega_3$ . Assumed  $k = 7$ , the algorithm calculated all distances between the sample  $x_i$  and other training examples and then 7 nearest neighbors are determined. There are 4 class  $\omega_1$  neighbors, 1 class  $\omega_2$  neighbor and 2 class  $\omega_3$  neighbors. It is obvious that the class  $\omega_3$  examples took a majority of 7 nearest neighbors. Thus, the sample  $x_i$  belongs to class  $\omega_3$ . As for *k*-NN regression, the value of a sample is the average of its *k* closest training examples. A better way is that *k*-NN is weighted according to the influence of different distances (Suarez & Murphy, 2012).

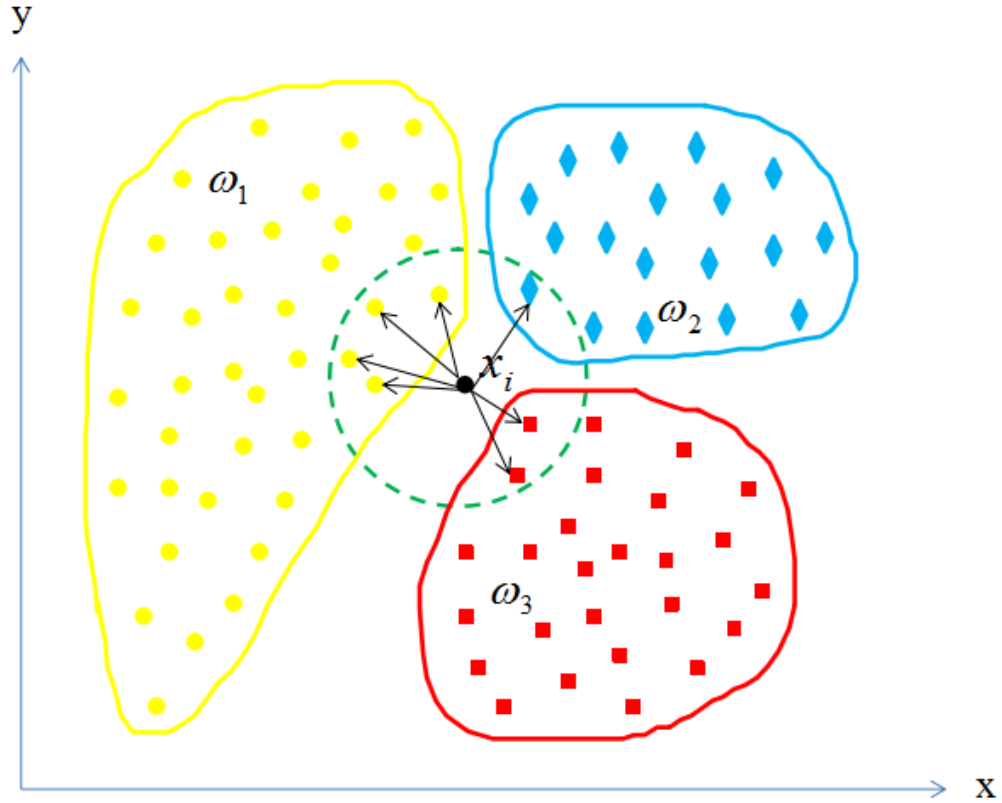


Figure 2.9 An schematic diagram of  $k$ -NN Algorithm

$k$ -NN algorithm is one of the simplest machine learning algorithms that can be used directly to classify the unknown data (Shi, Taib & Lichman, 2006). The training sample is classified into the classes that contain a majority vote of its  $k$  nearest neighbors. However,  $k$ -NN algorithm also has two main disadvantages (Bérci & Szolgay, 2009). Firstly, the outcome can be affected when the training examples are not balanced. For example, if the capacity of a class is very large compared with other classes, they might already occupy most of  $k$  nearest neighbors. Secondly, the calculation burden is too heavy because the algorithm must compute all distances between an unknown sample and all known training examples to obtain its  $k$  nearest neighbors (Cao, Lu, Gu, Peng & Wang, 2004).

Building upon the concept of a natural HCI, Yang et al. (2011) designed a real-time command system based on hand gesture recognition technology. The system was

designed so that a camera connected to the computer on which the system operated could continuously capture images in real-time. Subsequently, the computer would then use a Hue-Saturation-Value color model to determine the presence of any hand area in the captured image. Next, Fourier descriptors as well as K-NN classification would be utilized to distinguish between four different types of hand gestures. Hypothetically, the identified hand gesture would then be able to control Microsoft Office PowerPoint through the corresponding command connected to that particular hand gesture. Through experiments in hand gesture recognition Yang et al. (2011) were able to find evidence that suggested that the system could identify hand gestures with a satisfactory degree of accuracy. However, it was also noted that the rate of accuracy of the systems differed depending on the type of gesture with the OK gesture proving to be the most problematic. Yang et al. (2011) reasoned that strong lighting, shadows, or the thumb and index finger not closing enough could have produced this lower accuracy rate for the OK gesture. Other limitations of the system included the fact that it could recognize only four different types of gestures and failed to achieve satisfactory results when placed at an angle or in front of complex backgrounds.

### **2.3.3 Principal Component Analysis**

In actual research, in order to comprehensively and systematically analyze the problems, it is essential to consider a large number of factors. These involved factors are generally referred to as the indexes, also known as a variable in the multivariate statistical analysis (Thrun, 1996). Each variable reflects certain information regarding the research problem to a different extent and there is also a certain correlation with each other so some information that the statistical data reflects overlaps to a certain extent. In the research of multivariate problems, too many variables increase the amount of calculation and the complexity of the problem analysis. Hence, a method that can involve less variables but still keep the important information as much as possible in the process of quantitative analysis is needed (Boser, Guyon & Vapnik, 1992). As mentioned above, the principal component analysis is the ideal tool to solve this

problem.

Principal components analysis is a dimension reduction method that translates the multi-indexes into a few comprehensive indexes (Megiddo, 1983). In statistics, it is actually a linear orthogonal transformation that transforms the data into a new coordinate system to make the first largest variance of any data projection on the first coordinate (called the first principal component), the second largest variance on the second coordinate (called the second principal component) and so on. Each principal component is a linear combination of the original variables and each is unrelated. Therefore, these principal components can reflect most of the information of the original variables but they do not overlap each other. Principal component analysis is often used to reduce the dimensions of a data set, while keeping the characteristics that have great contributions to the variance in a data set (Nikhil, Mohan, Ramya & Kadambi, 2010).

### **2.3.4 Hidden Markov Model**

Hidden Markov Model is a statistical analysis model that is used to describe a Markov process that contains implicit unknown parameters (Kim, Kim & Lee, 2002). The difficulty is to determine the hidden parameters of the process from the observable parameters and then apply these parameters to make a further analysis such as pattern recognition, speech recognition, behavior recognition and so on (Corradini & Gross, 2000).

Research carried out by Min et al. (1997) delved into the use of HMM for the purpose of hand gesture recognition. The objective of their research was to develop a human computer interface that would allow graphics editor systems to be controlled by human hand gestures. The use of HMM was decided based upon the need for a means by which to recognize the continuous motion of human gestures. The results of their research

demonstrated a good overall recognition rate, thus proving that HMM can be effectively used for the purpose of hand gesture recognition.

Mitra and Acharya (2007) conducted a study pertaining to the design of a human-computer interface that could efficiently be put to use in the recognition of meaningful expression in human motion. The tools used to achieve gesture recognition included HMM, particle filtering and condensation algorithms, the FSM approach, as well as soft computing and the connectionist approach. HMM, condensation algorithms, the connectionist approach, and FSM were also specifically used for hand and arm gestures. Additionally, face and head gestures were identified using HMMs, PCA, FACs, contour models, facial feature extraction for gesture recognition, Gabor filtering, optical flow, skin color modeling, and the connectionist approach to facial gesture recognition. Of these tools, Mitra and Acharya (2007) found HMM, FSM, and particle filtering and condensation algorithms to be the major contributors. Thus, HMM was utilized due to its ability to efficiently model spatio-temporal information in a natural way. Particle filtering and condensation algorithms were also an important factor of gesture recognition because they allowed real-time estimates of nonlinear, non-Gaussian dynamic systems. Additionally, the FSM approach also played a role in analyzing and interpreting hand motions involved in hand gesture. Adapting to cross-cultural gestures was also accomplished through simple adjustments to the FSM by redefining it according to the relevant rules.

According to Yoon et al. (2001) there is much appeal in the idea of using hand gestures instead of interface devices to interact with computers since interface devices can prove to be inconvenient, awkward and clumsy whilst hand gestures are natural and more convenient. Consequently, many hand gesture recognition models have been proposed and of these endeavors most have made use of visual analysis tools such as syntactical analysis, neural networks, and the hidden Markov model. Similarly, the hand gesture recognition model proposed by Yoon et al. (2001) also utilizes a HMM for hand gesture recognition. In their approach, the preprocessing stage was comprised of hand localization, hand tracking, and gesture spotting procedures. Hand localization involved

the detection of the user's hand region based on motion and skin color whilst hand tracking involved finding and connecting the centroids of the moving hand regions in order to produce a hand trajectory. Lastly, the trajectories would be separated into real or meaningless segments by the gesture spotting algorithms. Through experimentation, Yoon et al. (2001) obtained results that were able to prove that their proposed HMM approach yielded a satisfactory recognition success rate even when hand gestures were viewed from an angle and when the system was tested on different hand sizes and differently shaped hands. Yoon et al. (2001) also suggested that future research should concentrate on enhancing hand location algorithms using color analysis and developing new feature detection algorithms for the HMM. These future research paths were deemed necessary because color information remains largely dependent upon the image-capturing environment and current feature detection algorithms do not fully consider the full range of characteristics inherent in dynamic gestures.

## **2.4 Existing Technologies and Applications**

### **2.4.1 Gesture Recognition**

In a review article written by Wu and Huang (2001) it was reported that, in the context of Human Computer Interfaces, we are far from perfecting a system that can be used for general-purpose gesture recognition. Instead, even the best vision-based gesture recognition and tracking systems that have been developed can be said to still be in their infancy. At this time, 3D hand models are used to completely capture the complex motion of human hand gesture whilst hand gesture recognition is accomplished mainly through static hand posture recognition and temporal gesture recognition. However, there is still much room for improvement upon current hand gesture recognition systems. In closing, Wu and Huang (2001) suggested that future research should delve deeper into computer vision, graphics, machine learning, and psychology so that a natural and reliable hand gesture interface can be developed.

A continuous posture recognition system was developed based on a recurrent neural network to deal with dynamic gesture processes in real time. The proposed gesture recognition system can recognize a kind of Japanese finger alphabet including 42 characters (Murakami & Taguchi, 1991). In their experiments, the recognition rate increased by more learning patterns and the recognition accuracy of the trainers was higher than the untrained users' accuracy. However, the data was sampled by Data Gloves so it is not convenient in reality.

When it comes to the recurrent neural network, a novel variant called Continuous Time Recurrent Neural Network (CTRNN) is developed to classify dynamic gestures from inexpensive sensors. Their gesture recognition system is made up of a memory block, a signal predictor block (CTRNN) and an error block. The objective of this system is to provide a fast, easy and modular way to recognize the dynamic gesture signals on the wearable devices in the future. According to their experiments, for the isolated gestures, the recognition rate of 8 gestures is about 94% for the testing set (Bailador, Roggen, Tröster & Triviño, 2007). However, when the gestures were gathered in a realistic environment, the recognition rate was only around 64% for the testing instances.

Su et al. (1996) used composite neural networks to develop a system that would be capable of recognizing static hand gestures. SDDL algorithms were implemented so that the HRCNNs could learn the intricate mappings of finger-flex angles involved in human hand gesture. The system was evaluated based on its ability to correctly classify a set of 51 static hand gestures. Results produced a recognition accuracy rate of 93.9%.

Stergiopoulou and Papamarkos (2009) devised a novel approach to hand gesture recognition by proposing a new procedure involving new Self-Growing and Self-Organized Neural Gas (SGONG) networks. Firstly, the detection of the region of the user's hand would occur through color segmentation techniques. Next, the SGONG network would determine the shape of the user's hand. This would include the extraction of the hand's morphologic characteristics as well as the identification of raised fingers. Hand gesture recognition is then finally achieved via a probability-based

classification technique. Under extensive testing, the proposed system was found to be a success in regards to hand gesture recognition due to its promising recognition success rate of 90.45%.

Murthy and Jadon (2010) designed a human-computer interface that could detect the presence of a human hand, count the number of fingers, and finally identify the direction being indicated. This type of hand gesture recognition was accomplished through the use of neural network functions. Experiments achieved a recognition rate of 89%, thereby demonstrating a satisfactory degree of accuracy.

Sohn et al. (2013) focused on hand gesture recognition systems with practical applicability, especially in regards to controlling household appliances. The method they examined utilized a depth camera to capture 3D hand motion trajectory. This trajectory would then be normalized so that feature extraction could occur. Their method also made use of Dynamic Time Warping algorithms combined with K-NN classification. This combination was chosen because it could be implemented with ease and could adapt to different users and varying gesture types. Sohn et al. (2013) referenced one example for experimentation and obtained results that showed a significantly increased recognition rate. Sohn et al. (2013) claimed that this suggested that their modification of DTW recognition performed better than conventional DTW recognition.

Research conducted by Panwar (2012) attempted to improve hand gesture recognition success rates by using shape parameters to identify hand gestures. This approach was unique in that it depended entirely upon shape parameters and did not consider additional factors such as skin color and texture and other features that could vary under different conditions. Despite its simplicity, this shape-based approach was capable of identifying 45 different hand gestures (Panwar, 2012). In accordance with the researcher's expectations, the shape-based approach also produced a significant increase in the success rate of hand gesture recognition.

A proposal written by Tam and Li (2012) proposed an interactive discussion system that



made use of the Microsoft Kinect depth camera. Tam and Li (2012) claimed that the Microsoft Kinect depth camera could be combined with smartphones to facilitate interactive discussion or presentation that could potentially be used for e-learning systems in the future. A prototype of their proposed gesture recognition and command specification software that would integrate the Microsoft Kinect depth camera with smartphones was developed to demonstrate the feasibility of such a project. The prototype presented in their proposal utilized the Microsoft Kinect depth camera because it could efficiently recognize human gestures. Meanwhile, the QR code related abilities of smartphones made it the ideal control device. Tam and Li (2012) asserted that their working prototype was readily usable with minimal learning required on the part of the user. Additionally, they claimed that demonstrations of their prototype were met with strong interest and that their proposal opened up many potential directions for future investigation.

Another study compared various learning algorithms commonly utilized in hand gesture recognition (Trigueiros, Ribeiro & Reis, 2012). The four learning algorithms upon which the study focused were k-NN, NB, ANN, and SVM. All four are classification algorithms that can be applied in static hand gesture recognition. These algorithms were compared based on hand pixel identification, feature extraction, and the identification of the specific hand pose based on the extracted hand features. Ultimately, ANN was found to be the best learning algorithm for hand gesture recognition (Trigueiros et al., 2012). Additionally, the study results suggested that the feature selection and data preparation phase was the most important step in hand gesture recognition, especially when low-resolution images are involved.

Wu and Lin (2013) developed a depth based hand gesture recognition system that could be used to control household appliances. Their proposed system made use of a depth camera and achieved recognition of dynamic hand gestures by using static posture recognition and hand trajectories. In static gesture recognition, the features of the hand are extracted and used to classify the hand posture. Experiments demonstrated that the system was capable of identifying seven common hand gestures and was effective in

controlling household appliances with accuracy rates ranging from 82.7% to 97.2%.

## **2.4.2 Character Recognition**

Generally, feature extraction and the neural network are widely used to recognize various characters. For different languages, the corresponding structure knowledge of the corresponding language is also very helpful in improving the recognition accuracy of the neural network. For example, Singh and Amin (1998) developed a neural network system based on feature extraction and structure knowledge to recognize handwritten Latin characters. The digital sample characters that contain some Gaussian noise were pre-processed by the feature extraction. For Latin characters, the two main structural features are straight lines and curves. The skeletons of the input Latin characters were traced and extracted based on several criteria. The neural network system then recognized the handwritten Latin characters using ten fold cross-validation (Singh & Amin, 1998). This produced an overall average recognition rate of about 87% (Singh & Amin, 1998).

It is a great challenge to recognize handwritten Chinese characters due to the huge number of characters, the great deviation of handwriting, the pattern complexity and similarity. Thus, the recognition rate of handwritten Chinese is quite low, especially for off-line recognition. An off-line recognition system that combined the neural network and the structural knowledge of Chinese characters was developed to improve the recognition performance. The main theory of this recognition system was the stroke detection, the stroke extraction and layer structure recognition. The experiments showed that the recognition rate of 17 characters was about 52% on average but the recognition system was fault-tolerant to the shift, slight rotation and light scaling of the input handwritten characters (Yeung & Fong, 1994).

## **2.5 Novelty of Proposed Solution**

In contrast to the existing work, the proposed virtual keyboard does not require any special device like a Kinect camera (Tam & Li, 2012), or a specific 3D camera (Hu & Gao, 2011). In addition, the camera for this virtual keyboard only shoots the typing motions on the keyboard area so that complex scenes need not be a concern, compared with other gesture recognition systems (Stefan, Athitsos, Alon & Sclaroff, 2008).

Compared with existing studies of other virtual keyboards, the proposed virtual keyboard is visible and flexible. It can be printed on a plain paper so it can be placed on a desk or stuck on a wall or on any oblique plane. Users can have the feeling and view of normal typing. Another advantage of this virtual keyboard is that users can replace the paper keyboard by themselves when they want a new one for any reason. It is simple for anyone to abandon the previous paper keyboard and print a new customized one. For the paper keyboard, users can stick it on any oblique plane and conveniently carry it anywhere.

# Chapter 3 Research Methodology

## 3.1 Introduction

Nowadays, many researchers are working towards making the virtual keyboards portrayed in movies and science fiction a reality. A robust virtual keyboard, although not mechanical, must have all the basic functions of a traditional keyboard. In addition, it would be an advantage if the virtual keyboard were very environment-friendly. Given that it is currently impossible for a computer to achieve 100% accuracy in regards to pattern and motion recognition no matter which existing artificial intelligence algorithm is chosen, the virtual keyboard still has a major disadvantage when compared with the mechanical keyboard. The performance of the virtual keyboard must be improved if users are to accept it.

This thesis aims to design a virtual keyboard that possesses the previously mentioned characteristics. To achieve this objective, the explicit research methodology that follows the in-depth comprehensive literature review is of immense importance. The proposed virtual keyboard is a practical application, which requires usability and robustness of the real system. Besides discussion regarding the studies of related theories, this thesis also includes experimentation in its methodology. In this chapter, Section 3.2 introduces the related studies regarding previous virtual keyboard systems that provided the inspiration for this thesis. Following this, Section 3.3 presents the research problems in detail. The research design and experiment design are explained in Section 3.4. The limitations of my research and experiment are delineated in Section 3.5 and the expected outcomes are depicted in Section 3.6.

## 3.2 Related Study

The keyboard is an important human-machine interface. Generally, a traditional mechanical keyboard is made up of keys and a Single Chip Microcomputer (SCM). The working principle is that the SCM constantly detects where a key is pressed. Subsequently, the SCM outputs the transmission codes of the corresponding keys to the computer through electrical signals (Evgeniou & Pontil, 2004). The CPU then transforms the transmission codes to ASCII codes and save them in the keyboard buffer.

The concept of a virtual keyboard is that it has the same input and output functions as a traditional mechanical keyboard but is neither mechanical nor material. Several technologies such as digital image processing, pattern recognition, finger recognition, and tracking, are required to complete a virtual keyboard. Currently, there are still a variety of challenges that must be overcome in order to develop a robust virtual keyboard. These challenges include variable conditions such as differences in skin tones between users, complex environments, inconvenient add-on devices, and the limitations of various algorithms (Gao, Fan, Jiang & Han, 2008).

Terajima et al. (2009) developed a system that was capable of using the images captured by a high-frame-rate camera to track the 3D motion of the human hand and fingers so that in-air typing motion could be recognized. Overall, the objective of the research was to design a fast finger tracking system that could be used as an in-air typing interface. By using a template-matching-based method Terajima et al. were able to significantly reduce background effect and make tracking possible even without markers. The high-frame-rate cameras were used because they were capable of tracking and recognizing rapid typing motion. Furthermore, so that real-time recognition could be accomplished, Terajima et al. built new hardware for the purpose of parallelizing and accelerating the image processing speed. Ultimately, Terajima et al. were able to complete their objective of achieving real-time recognition of in-air typing motion. Additionally, the template-matching-based method that they implemented proved

functional even when faced with environments of varying illumination and cluttered backgrounds.

Operation areas on touch screens are becoming smaller with the miniaturization of mobile devices. Consequently, the much narrower screen means that the built-in virtual keyboard occupies a larger portion of the screen. Furthermore, the overall smaller size of the screen can make it uncomfortable for users to type on. Thus, an in-air typing interface was developed by Niikura et al. (2012) to remove these disadvantages of the traditional mobile interface. Their interface required add-on devices including a high-frame-rate camera, an IR filter, four IR LEDs and a vibration motor. A Lucas-Kanade algorithm was combined with a band pass filter and used to recognize and track the user's fingertip. Two experiments were designed to test the usability of their interface based on three aspects - frame rate, latency and tactile feedback. Based on the results of their research, Niikura et al. found that reducing several milliseconds of latency can shorten task completion time. In order to control the latency, the frame rate should not be lower than 15 fps. Unexpectedly, mistypes decreased significantly thanks to vibration feedback.

### 3.3 Research Questions

The main purpose of this thesis is to design a new virtual keyboard based on finger recognition. Taking the previous related work into account, it appears that existing virtual keyboard systems still have many challenges and issues left to resolve. Thus, this thesis aims to create a new virtual keyboard that has better performance but uses fewer materials. Hence, the main research question of this thesis is: **Whether a novel virtual keyboard can be built based on finger recognition?**

Developed from the main question, there are three important sub-questions concerned as shown below,

**Question 1.** Is the virtual keyboard portable, replaceable and environmentally friendly?

**Question 2.** Where can the virtual keyboard be placed?

**Question 3.** How is the usability of the virtual keyboard?

The experiment in this thesis is to create a virtual keyboard that can recognize what a finger types on a virtual keyboard through a normal camera. Each of these sub-questions will be answered by the experimental results.

## **3.4 Research Design and Experimental Environment**

### **3.4.1 Virtual Keyboard Design**

Theoretical research is important for a virtual keyboard but the actual performance of the system is the main feature that users really care about. The point of the experiment was to make an actual virtual keyboard and then test it to see if it could work satisfactorily to recognize what the user typed on the virtual keyboard.

The first important problem is to ascertain what the imagined virtual keyboard will look like. When the virtual keyboard was designed, it was expected to have the following advantages,

- a) The virtual keyboard is not mechanical, heavy and complex;
- b) The virtual keyboard can be placed on any oblique plane or even on the wall;
- c) The virtual keyboard is environmentally friendly;
- d) The virtual keyboard should be easy to get and the cost is acceptable.

To sum up, in order to satisfy the above conditions, the options for the virtual keyboard are limited to a paper keyboard or a projection keyboard. Taking cost, portability and simplicity into account, a paper keyboard would be the preferred choice for most users although the projection keyboard would also be a decent choice in fixed locations like

offices and classrooms.

In regards to the camera, most people do not want to buy specific and expensive add-ons like 3D depth cameras. Nowadays, there are generally at least one or two internal cameras on every mobile device. Additionally, an external camera is common for computers and easy for most people to acquire. Thus, a single normal camera was selected for use for the proposed virtual keyboard in order to recognize what the user's finger typed. In the experiment, a smartphone was consistently used to film all the typing videos.

This virtual keyboard is just a prototype that tests the usability of this design. Since the system only uses a common camera, situations where ten fingers are typing are too complex for fingertip tracking and the recognition of input actions because the overlaps and crossovers among the fingers will cause a high number of recognition errors. Furthermore, recognizing and tracking ten fingers are a heavy burden to computers and smartphones due to the many calculations involved. This leads to serious delays. Mobile devices in particular do not usually have CPUs that are fast enough to deal with massive calculations. Therefore, this virtual keyboard was designed to recognize only what a single finger typed on the virtual keyboard. Accordingly, this virtual keyboard's application platform at the present stage is limited to mainly small mobile devices such as iPads and smartphones.

If the proposed virtual keyboard can be customized, it will greatly improve the user experience. However, a replaceable virtual keyboard requires that the recognition system be able to recognize a new virtual paper keyboard every time the user decides to make a change. This means that the system has to be capable of machine learning. Key recognition is actually the recognition of standard English characters. Since Optical Character Recognition (OCR) is already mature enough to recognize the characters in an image, the OCR tool in Matlab was chosen to extract the information of each of the characters and numbers on the virtual keyboard.



The proposed virtual keyboard is designed to detect a fingertip through a normal single camera that can easily be obtained by all users. Given that the camera and the paper keyboard can both be properly fixed in place, a great influence factor known as complex background can be removed by adjusting the distance between the camera and the paper keyboard to ascertain that the paper keyboard completely fills the whole screen.

### **3.4.2 Architecture**

The aforementioned sections presented the basic design and expected performance of the proposed virtual keyboard. In this section, the program architecture that was implemented in the proposed virtual keyboard is discussed.

In the system, there are two subsystems that are responsible for the processing of the input custom keyboard and videos. They are the fingertip recognition subsystem and the recognition subsystem of the customized keyboard.

In the fingertip recognition subsystem, the Gaussian filter will smooth each frame of the input video before the image becomes dilated and eroded. The filtered image is divided into R,G,B channels and then binarized. The BWMORP algorithm is utilized to extract the skeleton of the human hand and then the furthest endpoint of the skeleton is selected to represent the fingertip. When this fingertip is detected, the coordinates of the pixel are known as well. The flowchart of the fingertip recognition subsystem is shown in Figure 3.1.

To meet the virtual keyboard customization needs, the OCR tool in Matlab is utilized to extract all the characters and numbers in the image of the paper keyboard. OCR converts the input keyboard image to a binaryzation image and smooths it to decrease the noises. Slant is then rectified and characters are segmented. At last, characters are recognized by using feature extraction. A 2D array then stores all the characters and numbers with their coordinates relative to the four corners of the rectangle on which

they are located. As a matter of fact, this 2D array is the virtual keyboard that the program will read. The coordinates of the fingertip are given by the fingertip recognition subsystem so that when the fingertip locates in the area of a key, the subsystem records this key as a candidate that the user might want to type. The flowchart of the customized keyboard recognition subsystem is shown in Figure 3.2.

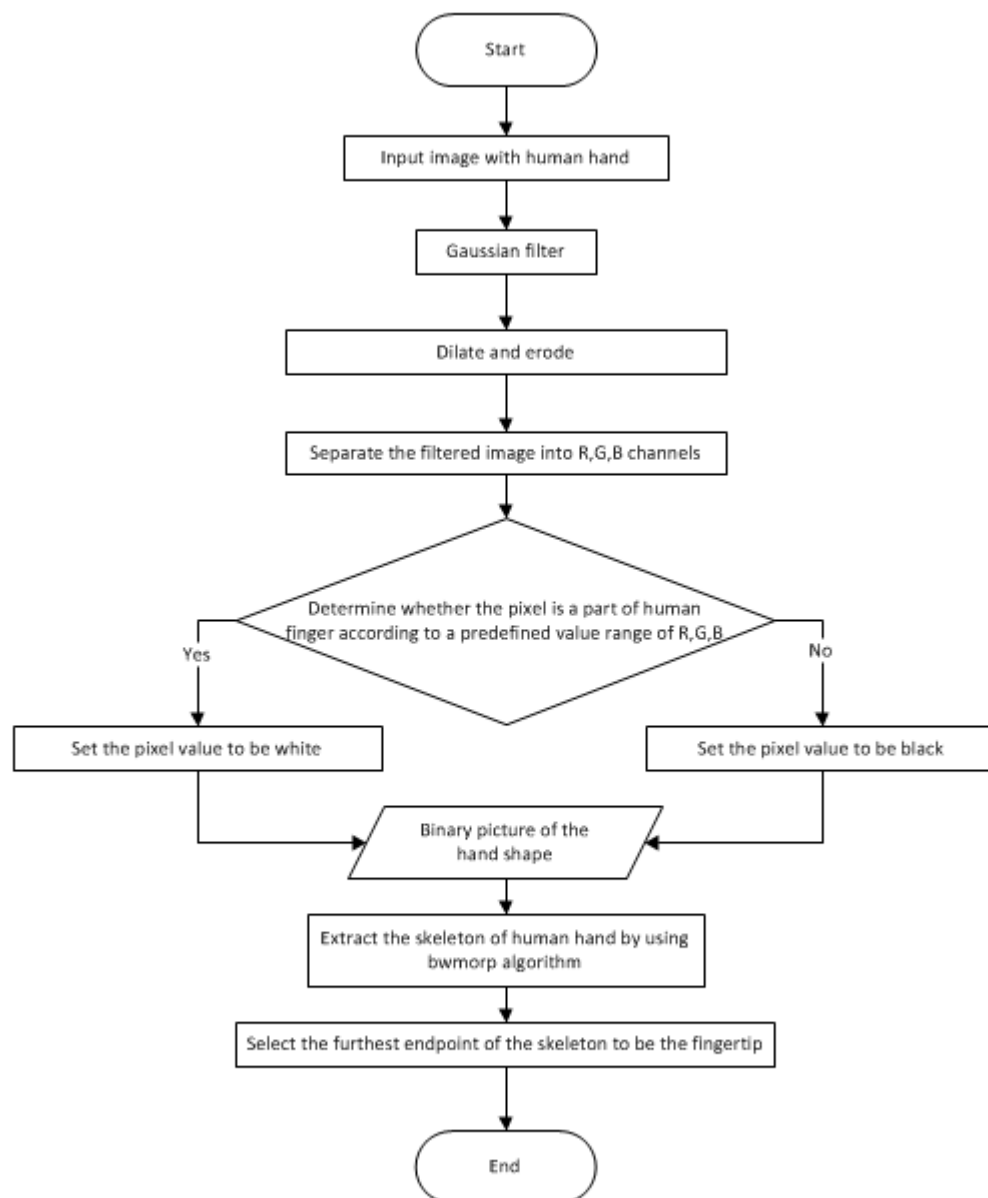


Figure 3.1 The flowchart of the fingertip recognition subsystem

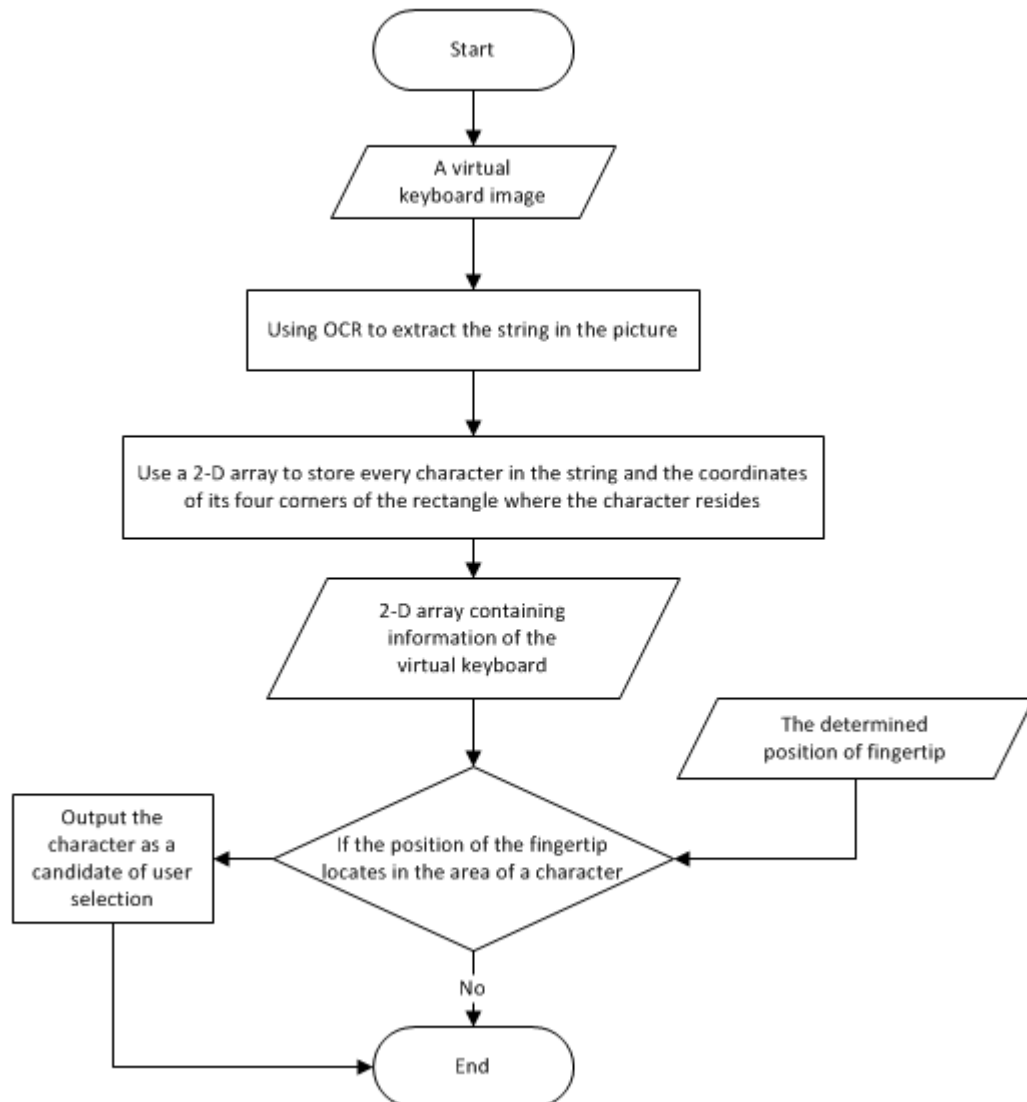


Figure 3.2 The flowchart of the customized keyboard recognition subsystem

The customized keyboard recognition subsystem provides computer-readable keyboard information to the entire system no matter what the customized virtual keyboard looks like in reality. The fingertip recognition subsystem will keep detecting the position of the fingertip in every frame, which means that the program can always track the fingertip on the virtual keyboard. The fingertip moves above the virtual keyboard and when the fingertip stays on a key for more than four frames, the system regards this as a

typing action executed by the user. According to the coordinates of where the fingertip had stayed, the corresponding key is found in the virtual keyboard. If there is no corresponding key for where the fingertip stayed, the virtual keyboard will carry on. The flowchart of our virtual keyboard is shown in Figure 3.3.

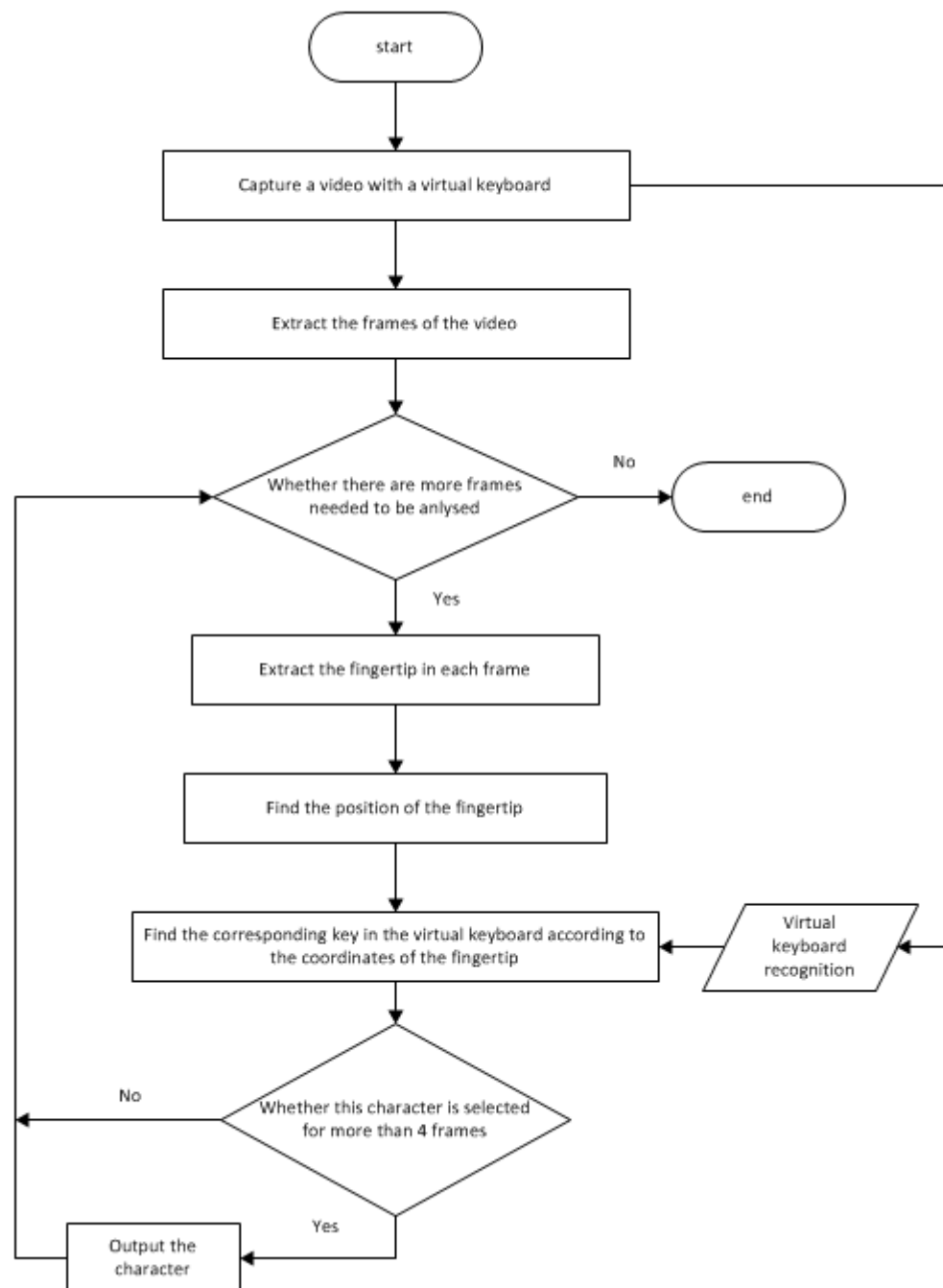


Figure 3.3 The flowchart of the virtual keyboard

### **3.4.3 Experimental Environment**

For the convenience of experimentation, the experimental data comes from a set of video footages of various typing motions that are filmed by the camera in advance. Three testers are invited to participate in the experiments to test if the virtual keyboard can recognize different hands. Five kinds of paper keyboards are selected to test if the virtual keyboard is customizable and replaceable. To test the usability in a broad spectrum of environments, the testing is conducted in common places such as classrooms and houses. The types of lighting that are used in the experiments are sunlight and lamplight. All paper keyboards are placed on a wall. The recognition results of this virtual keyboard are recorded, analyzed and discussed in Chapter 4 and Chapter 5.

## **3.5 Limitations of the Research**

Essentially, there are three limitations to this research. First of all, the voluntary testers cannot participate in the experiments and conduct tests at any random time or place because the experiments require special environments. Thus, tests can be conveniently repeated, the gathering of data in the experiments is based on pre-recorded videos and not real-time ones.

Secondly, the environment of the experiments is assumed to be stable, this means it is assumed that the lighting or background will not change and there will be no other interferences. Thus, the experiments do not take into consideration complex situations such as the possibility of using two or more fingers, and other factors that are deemed irrelevant.

Thirdly, the tests concerning different skin tones are not sufficient. In addition, there are not a great number of tests conducted for different types of virtual keyboards. Thus, the

compatibility of different skin tones and various customized virtual keyboards can not be completely tested.

### **3.6 Expected Outcomes**

The main expected outcome is that the program can recognize the majority of the letters that users type on the virtual keyboard. Hopefully, the accuracy can reach 90%. Secondly, the virtual keyboard is expected to possess all the advantages that are listed in Section 3.4.1. For example, it should be replaceable to fulfill customization needs. Finally, the author hopes that the testers are interested in using the completed version of this kind of virtual keyboard on mobile devices in the future.

# **Chapter 4    Research Findings**

## **4.1    Introduction**

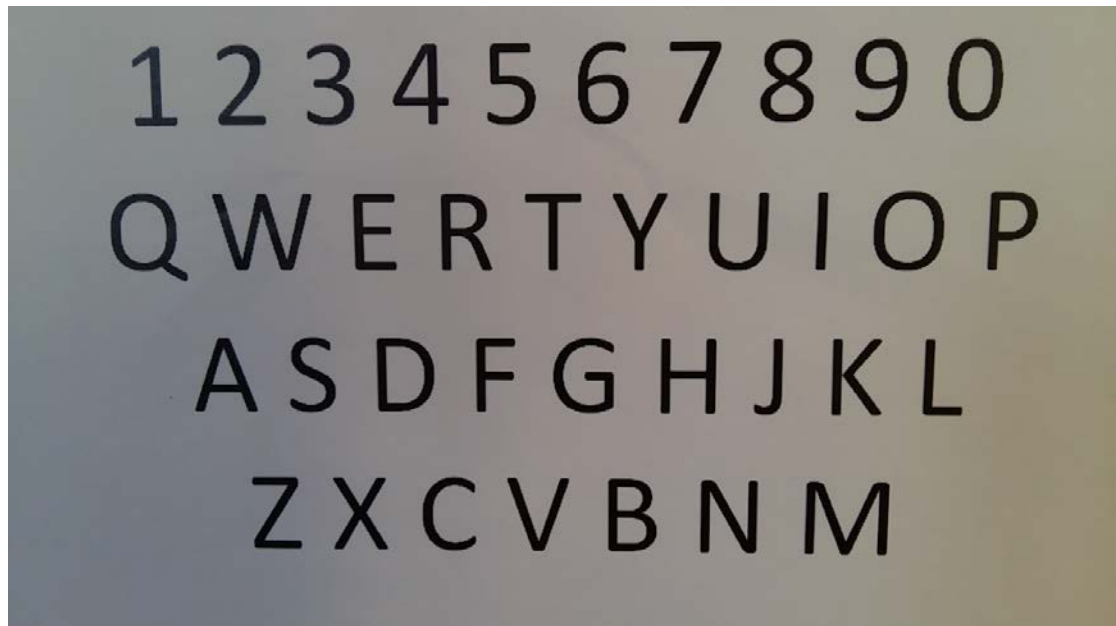
In this thesis, a new virtual keyboard is designed that possesses the multiple features mentioned in Section 3.4.1. In this chapter, the experiments are designed to test the practical performances of these features. The goal of the experiments is to test the virtual keyboard's performance under three main aspects: 1) normal typing; 2) differently customized virtual keyboards; 3) different environments and obstructions. Each test is described in detail and the results of the experiments are presented in Section 4.4.

## **4.2    Experimental Test Bed and Data Selection**

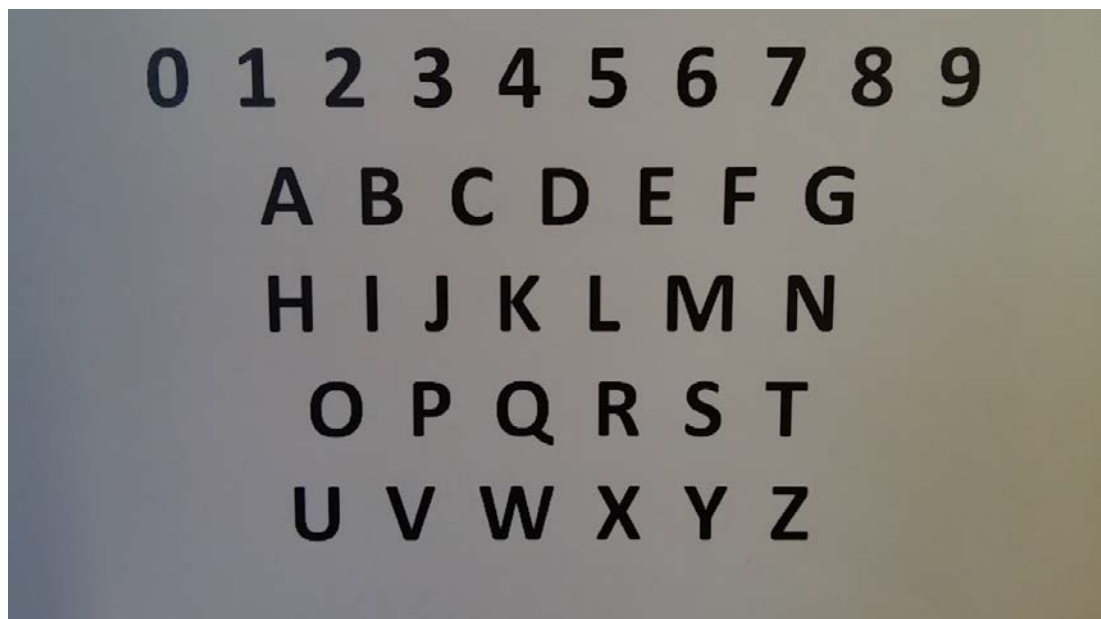
The experimental environments are based on an average laptop and the videos are captured by the built-in camera of a SAMSUNG A5 (smartphone). In addition, the videos are recorded in off-line mode and then transferred to the laptop. The prototype of the virtual keyboard is developed and implemented in Matlab R2014a.

Data is collected from three participants. Five kinds of customized virtual keyboards are selected to test the adaptability of this virtual keyboard. Five chosen customized paper keyboards are shown in Figure 4.1. For simplicity, the author refers to these keyboards in this thesis as keyboard (a), keyboard (b), keyboard (c), keyboard (d) and keyboard (e).

For the experimental environment, the two light sources are natural light and lamplight. The locations where the experiments are carried out are in a bedroom in a house as well as in an office at our university. Moreover, all paper keyboards are stuck on a cream colored wall.



(a) Keyboard with QWERTY order and customized font size

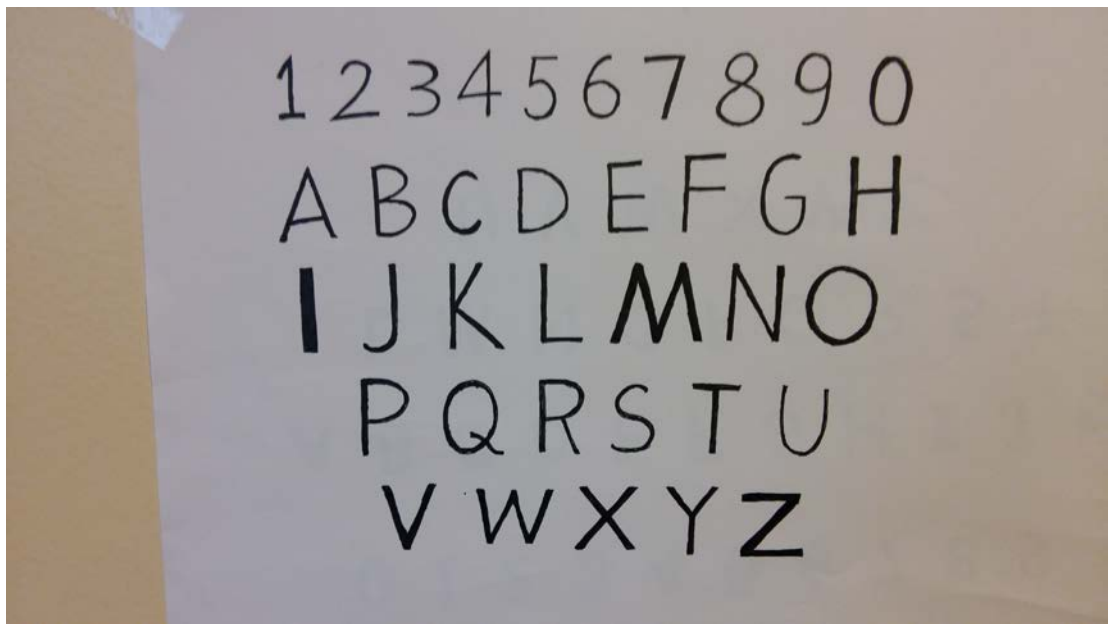


(b) Keyboard with customized font size and letter order

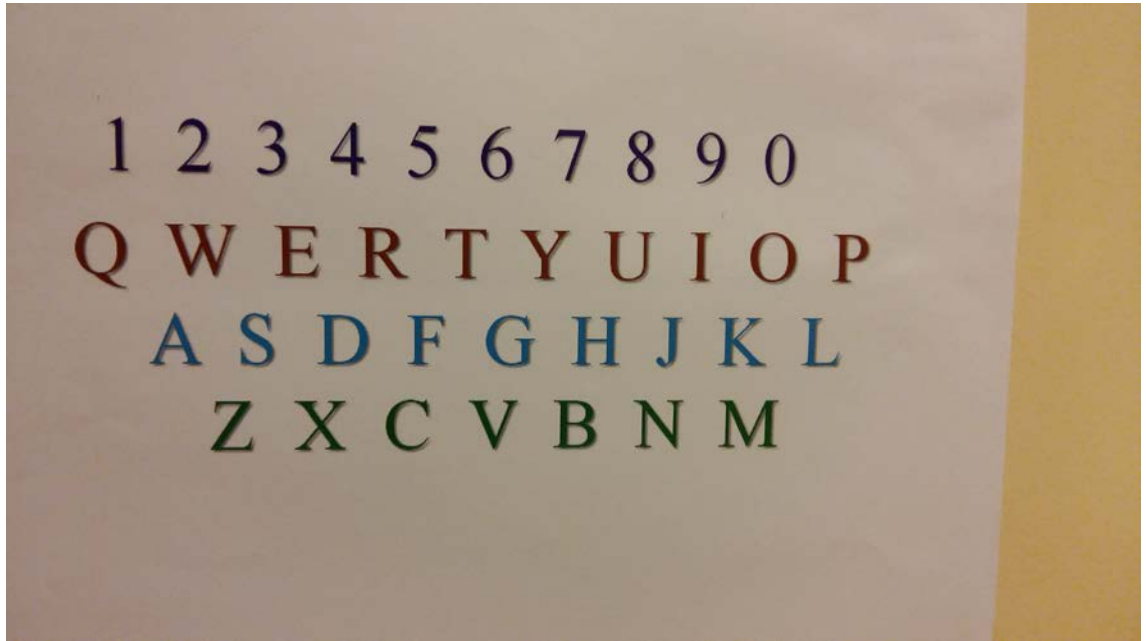




(c) Keyboard with shadows, customized font and letter order



(d) Handwritten keyboard with customized font, letter order and skin tone background



(e) A color keyboard (a) with shadows and skin tone background

Figure 4.1 Chosen virtual keyboards for testing

## 4.3 Experiments

### 4.3.1 Experimental Algorithms

The proposed algorithms used in the virtual keyboard are to locate the virtual keyboard (Algorithm 4.1) and to extract the fingertip position in each frame (Algorithm 4.2). These two algorithms were already introduced in detail in Section 3.4.2.

#### **Algorithm 4.1: Locate the virtual keyboard**

##### **Input:**

I = Keyboard Background

##### **Output:**

Box = areas of all characters in the background

##### **Procedure:**

##### **Begin:**

results = Apply OCR on the image of I

str = the string which is extracted from results without whitespace and irrelevant characters

len = length of str

box = Array with length of len to store the characters in the order according to

```

their occurrence in the keyboard image
    v = horizontal coordinate of fingertip
    m = vertical coordinate of fingertip
    h1 = left coordinate of the area of a character
    h2 = right coordinate of the area of a character
    h3 = top coordinate of the area of a character
    h4 = bottom coordinate of the area of a character
    if(v>h1 && v<h2 && m >h3&&m<h4){
        output this character as a candidate of user selection
    }
End

```

#### **Algorithm 4.2: Extract the fingertip position in a frame**

##### **Input:**

Img = An input video frame

##### **Output:**

Img\_out = the video frame with hint for fingertip

v = horizontal position of fingertip

m = vertical position of fingertip

##### **Procedure:**

Begin:

Set ImgPre=Apply an Gaussian filter on Img to reduce noise

Set ImgPrer = YCbCr Image of ImgPre

Set lx = horizontal coordinate of Img

Set ly = vertical coordinate of Img

Set I = Empty Array with same size of Img

Set Filter = filter of selecting hand by setting range limit of YCbCr color value

Set I = Apply Filter on ImgPrer

Set I = dilate I

Set I = erode I to avoid noise and refining result of color filte

Set I' = choosing largest selected part to be human hand in I

Set BW3 = using bwmorph to extract skeleton of I by using parameter of thin

Set endPoints = all endpoints of skeleton

Set endpoint = the furthest endpoint of all endpoints as fingertip

Set v = horizontal coordinate of endpoint

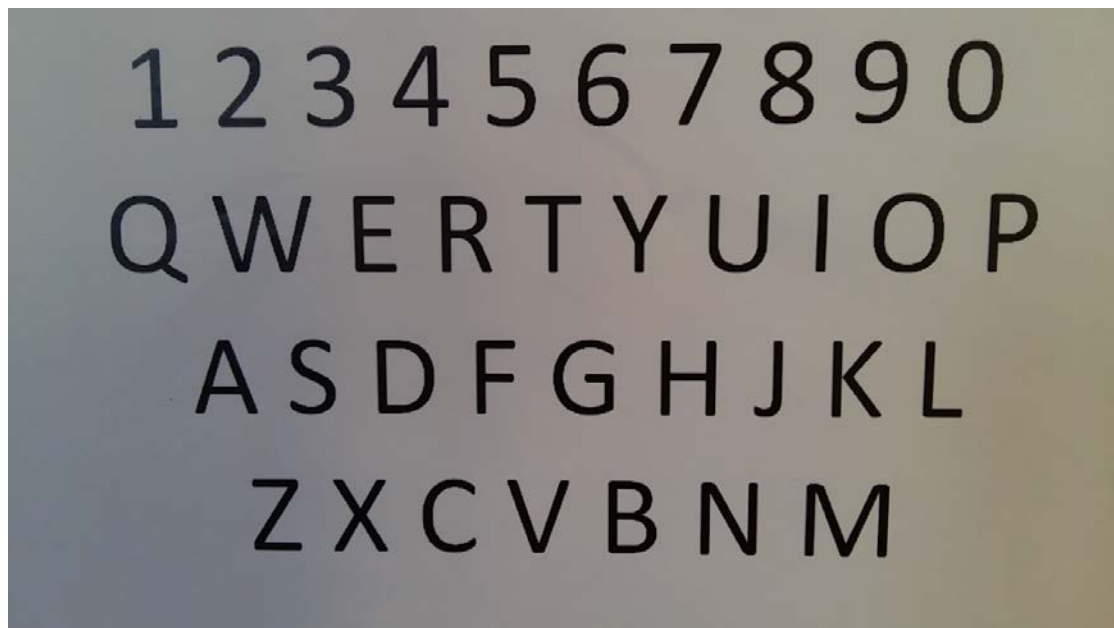
Set m = vertical coordinate of endpoint

End

### **4.3.2 Experimental Output**

The virtual keyboard can be customized to some extent but its design cannot be completely random. If a user creates a very random keyboard, the recognition accuracy

will partially decrease as a result. Thus, to solve this problem, the virtual keyboard can also output the recognition result of a customized keyboard and therefore allow users a chance to modify their work until it is acceptable. Figure 4.2 and Figure 4.3 show the results of two successful examples of customized keyboards. One paper keyboard is printed and the other is handwritten.

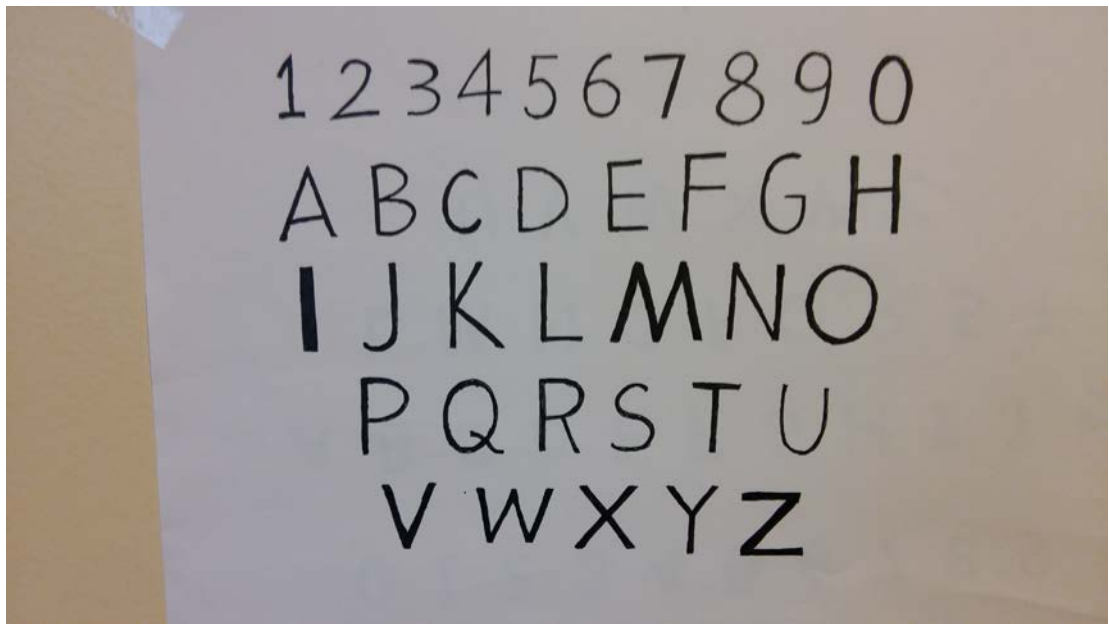


(a) Printed keyboard

```
>> chara  
  
ans =  
  
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

(b) Matlab result of the printed keyboard using Algorithm 4.1

Figure 4.2 Example 1: printed keyboard recognition



(a) Handwritten keyboard

```
>> chara

ans =

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

(b) Matlab result of the handwritten keyboard using Algorithm 4.1

Figure 4.3 Example 2: handwritten keyboard recognition

In the fingertip recognition subsystem, each frame of the input video is smoothed, dilated and eroded. One of the binary images is shown in Figure 4.4.

The BWMORP algorithm extracts the skeleton image from the processed binary frame shown above. The furthest endpoint of the skeleton is identified as the fingertip. One of the acquired skeleton frames is shown in Figure 4.5.

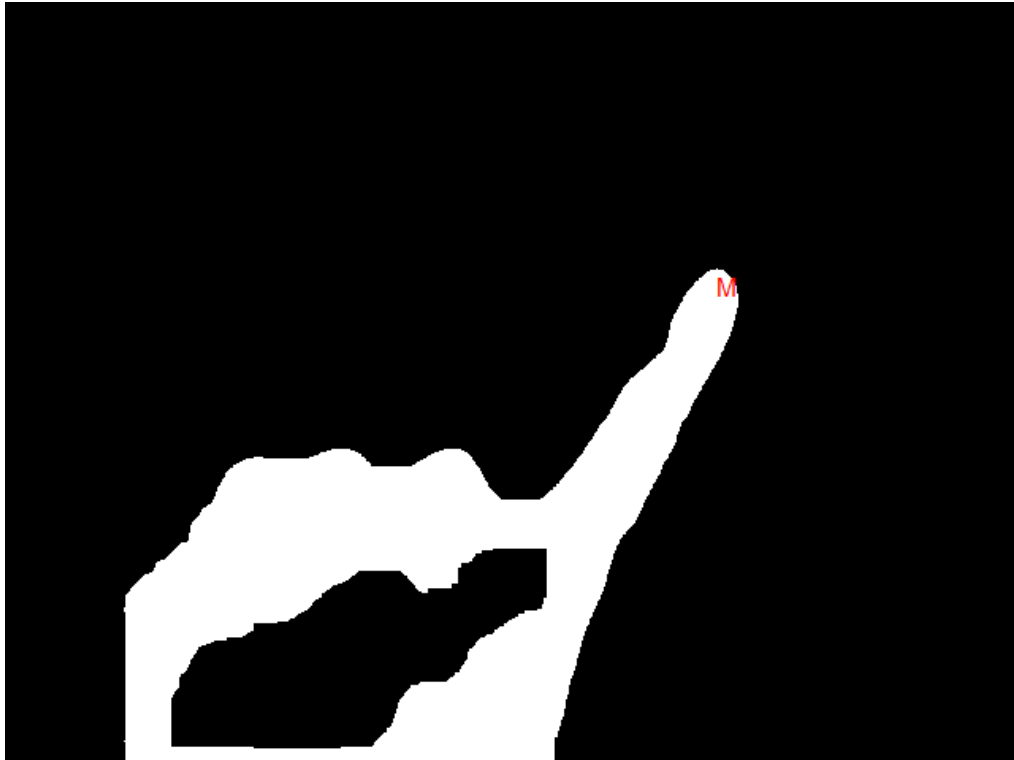


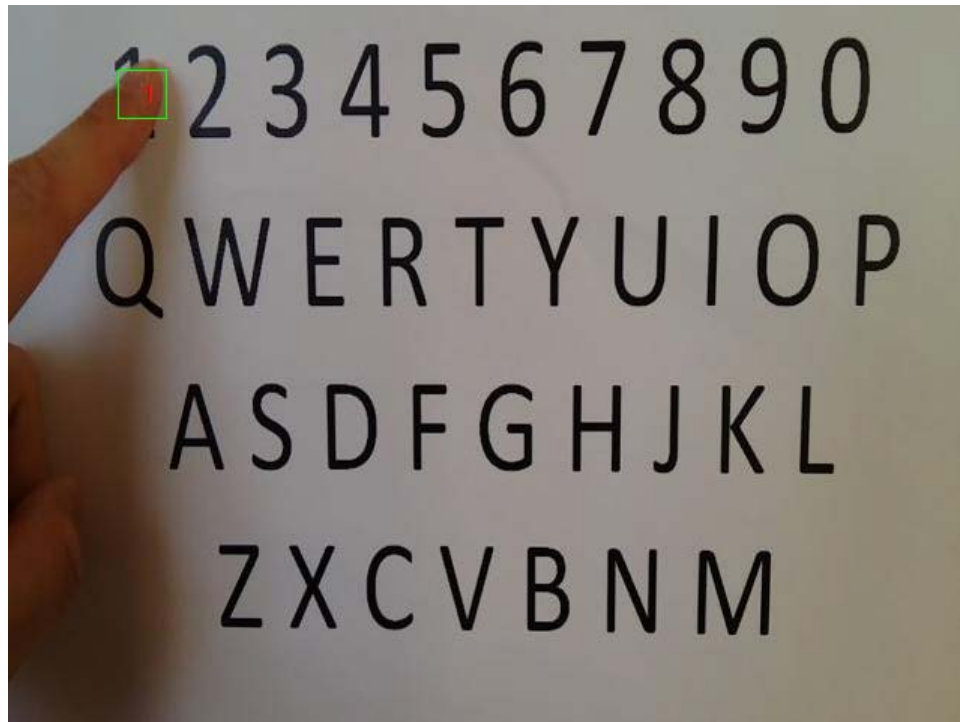
Figure 4.4 The binary frame



Figure 4.5 The skeleton image

After the fingertip is detected, the motion of the finger is tracked and judged by the

program to determine whether any typing actions occur. When any typing action occurs, the program outputs the recognized keys. One successful example for each type of customized paper keyboard adopted in the experiments is shown in Figure 4.6. Other examples are provided in the Appendix.



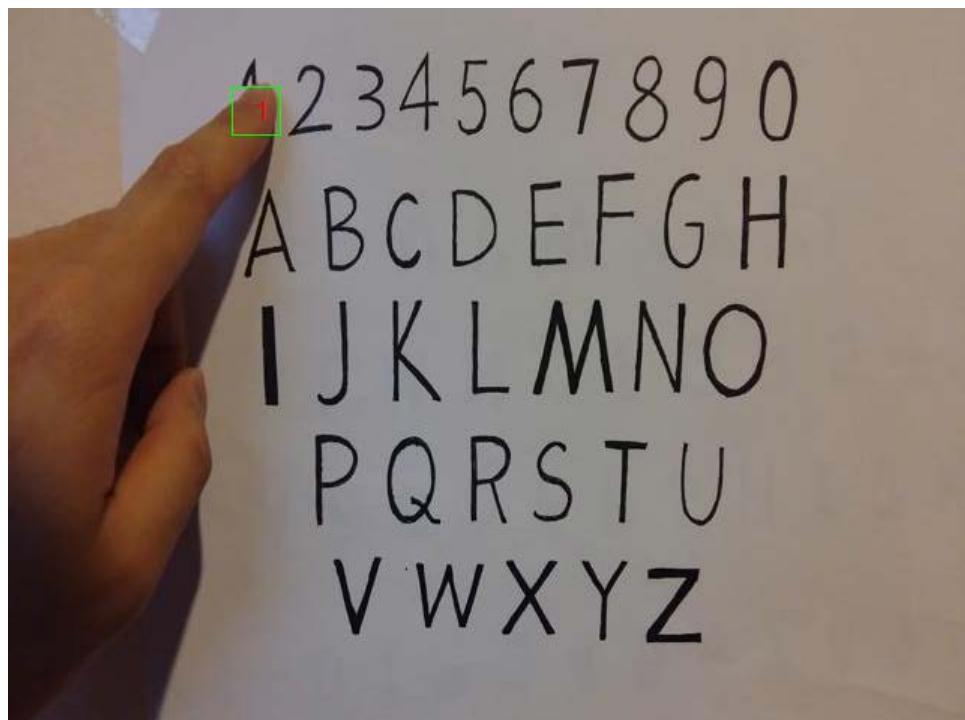
(a) The example for keyboard (a)



(b) The example for keyboard (b)

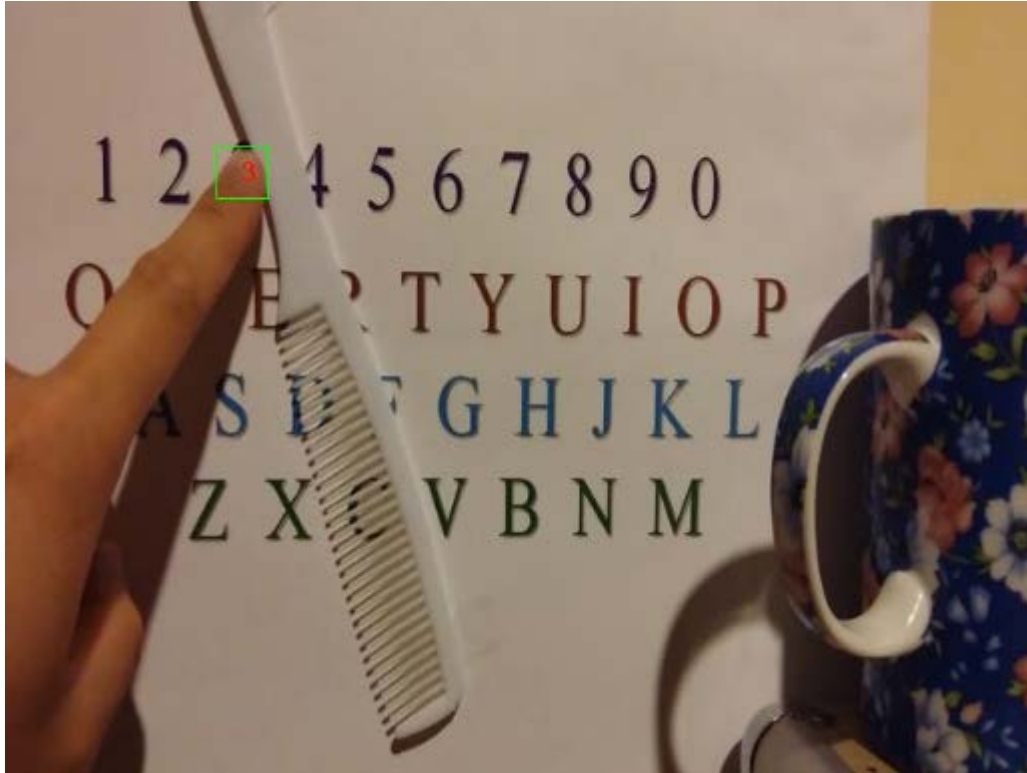


(c) The example for keyboard (c)



(d) The example for keyboard (d)





(e) The example for keyboard (e)

Figure 4.6 Examples of successfully recognized typing

## 4.4 Experimental Results

The experiments are designed to test the performances of the proposed virtual keyboard in various situations. The content of all tests is to ask the testers to type 10 Arabic numbers and 26 English letters on the chosen paper keyboards adding up to a total of 36 keys. Testers are required to type the 36 keys in 20 seconds, which is the average speed of normal typing. This is done for each type of keyboard. Every combination of keyboard, lighting, and tester is tested. A total of six tests are carried out for each unique keyboard and lighting combination and each of the three testers participates in two of these six tests. As mentioned above, all paper keyboards are stuck on a cream colored wall.

First of all, to test the adaptability of the virtual keyboard to various environments, two typical environments are selected for the experiments. One is under lamplight in a bedroom in a house and another is under natural light in an office at our university.

Secondly, five types of customized paper keyboards are selected to test whether the virtual keyboard can adapt to various customized keyboards. Keyboard (a) stands for the standard keyboard and uses the Calibri font. Compared with keyboard (a), keyboard (b) is selected for natural number and letter order with a smaller font size. Furthermore, keyboard (c) is utilized to test for adaptability to a different font such as Comic Sans MS. Keyboard (d) is adopted to test for adaptability to handwriting. Finally, keyboard (e) is chosen to test if the proposed virtual keyboard can process multiple colors and ignore obstructions. The results of all the tests are shown in the Tables below.

Table 4.1 Test results of keyboard (a) under lamplight

Test No.	Tester	Mismatched Keys	Accuracy
1	A	I	97.2%
2	A	V	97.2%
3	B	1, I	94.4%
4	B	M	97.2%
5	C	I, P, L, B, N	86.1%
6	C	I	97.2%
The average accuracy is 94.9%			

Table 4.2 Test results of keyboard (a) under natural light

Test No.	Tester	Mismatched Keys	Accuracy
1	A	1, I	94.4%
2	A	N/A	100%
3	B	I	97.2%
4	B	N/A	100%
5	C	5	97.2%
6	C	I	97.2%
The average accuracy is 97.7%			

Table 4.3 Test results of keyboard (b) under lamplight

Test No.	Tester	Mismatched Keys	Accuracy
1	A	9, L	94.4%
2	A	O, R	94.4%
3	B	9, A, G, J, O, P, Y, Z	77.7%
4	B	H, I, J, L	88.8%
5	C	I, U, T	91.6%
6	C	I	97.2%
The average accuracy is 90.7%			

Table 4.4 Test results of keyboard (b) under natural light

Test No.	Tester	Mismatched Keys	Accuracy
1	A	3, 9, I, U	88.8%
2	A	N/A	100%
3	B	9, B, I, T	88.8%
4	B	I	97.2%
5	C	3, I, L, X	88.8%
6	C	H	97.2%
The average accuracy is 93.5%			

Table 4.5 Test results of keyboard (c) under lamplight

Test No.	Tester	Mismatched Keys	Accuracy
1	A	0, K	94.4%
2	A	L, Y	94.4%
3	B	0, Y	94.4%
4	B	0, 1, A, Y	88.8%
5	C	0, 9, L	91.6%
6	C	N/A	100%
The average accuracy is 94%			

Table 4.6 Test results of keyboard (c) under natural light

Test No.	Tester	Mismatched Keys	Accuracy
1	A	2, F, U	91.6%
2	A	1	97.2%
3	B	L	97.2%
4	B	N/A	100%
5	C	0, 1, A, I	88.8%
6	C	0	97.2%
The average accuracy is 95.3%			

Table 4.7 Test results of keyboard (d) under lamplight

Test No.	Tester	Mismatched Keys	Accuracy
1	A	P, Z	94.4%
2	A	N/A	100%
3	B	I, L, R, Z	88.8%
4	B	G, O, Z	91.6%
5	C	I, L	94.4%
6	C	1, K	94.4%
The average accuracy is 93.9%			

Table 4.8 Test results of keyboard (d) under natural light

Test No.	Tester	Mismatched Keys	Accuracy
1	A	3, A, I	91.6%
2	A	1	97.2%
3	B	I	97.2%
4	B	N/A	100%
5	C	J	97.2%
6	C	N/A	100%
The average accuracy is 97.2%			

Table 4.9 Test results of keyboard (e) with obstructions

Test No.	Obstructions	Mismatched Keys	Accuracy
1	Pen and bottle	0, 1	94.4%
2	Pen and bottle	1, Q	94.4%
3	Pen and bottle	0, 1, 2, S	88.9%
4	Comb and cup	N/A	100%
5	Comb and cup	1, 3, I	91.6%
6	Comb and cup	2	97.2%
The average accuracy is 94.4%			

# **Chapter 5 Discussions**

## **5.1 Introduction**

The experimental details and the results of usability tests involving the proposed virtual keyboard are presented in Chapter 4. According to these tests, the outcomes are analyzed to evaluate the performance of the proposed virtual keyboard and the main experimental findings are discussed to answer the research questions raised in Chapter 3.

Chapter 5 is divided into two parts. Section 5.2 is the analysis and evaluation of the virtual keyboard based on test results. In Section 5.3 justifications are proposed to answer the research questions raised in Chapter 3.

## **5.2 Analysis and Evaluations**

The overall recognition rate of the proposed virtual keyboard is 94.62%. The highest recognition rate in all tests is 100% and the lowest is 77.7%. For all types of customized paper keyboards, the best average recognition rate is 97.7% and the worst is 90.7%. The results indicate that the recognition rate is related to both the accuracy of each typing position and the type of light source. The test results indicate that the virtual keyboard can process multiple colors and ignore some obstructions because the program recognizes and records the whole paper keyboard (e) before the partial obstructions cover the keyboard.

The average recognition rate under lamplight is 93.6% and the average recognition rate under natural light is 95.9%. In other words, the average recognition rate under natural light is 2.3% higher than the average recognition rate under lamplight. This can be because the algorithms recognize the fingertip and the 36 keys based on their colors in each frame. Thus, the lamplight can make the image of the paper keyboards dimmer

than they appear under natural light. In addition, lamplight also causes more shadows than natural light. These shadows are also darker under lamplight so the algorithms might regard these shadows as a part of a hand. In contrast, natural light contributes to fewer shadows, which leads to less interference with the recognition rate. Thus, bright lighting has a slightly positive influence on the performance of the virtual keyboard.

Moreover, the author notices that the recognition rate of the letter 'I' is the lowest when compared with the recognition rate of other keys. One reason for this is that 'I' is similar to '1', '|' and '/' so it sometimes confuses the character recognition algorithm. Another reason is that 'I' is quite a thin letter, which means that the area of its coordinates is very small. When a user is trying to type 'I', he or she might not be able to precisely place their fingertip within the area coordinates of 'I'. Therefore, it might not be recognized as an effective input. To cope with this problem, the program might have to handle 'I' in a special way to enlarge the area of its coordinates in the future work.

The author also finds that the recognition rate of the first test is often worse than the recognition rate of the second test that is carried out by the same tester under the identical conditions. The author speculates that this is because testers are initially unaccustomed to typing on the paper keyboard but are able to become accustomed to it quickly. The feedback that testers provided at the end of the experiment proves that the author's speculation is correct. The testers feel that they are unaccustomed to typing on a paper keyboard in the beginning but following a few attempts they become able to properly type on the paper keyboard as if it were a normal keyboard. The testers can also type over the paper keyboard in a manner similar to in-air typing, which means the proposed virtual keyboard can also be combined with 3D projection technology in the future.

## 5.3 Justifications

In this thesis, the main research question is whether a novel virtual keyboard based on finger recognition can be developed. The author presents a novel virtual keyboard that utilizes BWMORP algorithm to recognize the user's fingertip and adopts an OCR tool to recognize characters on various virtual keyboards customized by the users. To answer the main research question and the three sub-questions, a prototype of the virtual keyboard is developed in Matlab 2014Ra. As shown by the results of the experiments, the average recognition rate of the prototype is 94.62%. Two groups of tests show that the brightness of lighting in the environment can slightly influence the performance of the virtual keyboard but the recognition rate still remains quite stable. Thus, the usability of the proposed virtual keyboard is proved.

Thanks to the fact that the paper keyboard can be recognized, the virtual keyboard is both customizable and replaceable. Moreover, according to the experiments, both printed and handwritten keyboards are successfully recognized. When the paper keyboard becomes broken, the user can rapidly and conveniently replace it by printing or writing a new one. Another advantage is that the paper keyboard can be stuck on any oblique plane as well as on a wall.

Nowadays, the continuous deterioration of the earth's environment has caused green consumption and green marketing to develop worldwide. The proposed virtual keyboard caters to the demand for environment-friendly practices and products. Compared with traditional mechanical keyboards, this kind of virtual keyboard conserves more resources and does not create huge amounts of waste. Therefore, it is very environmentally friendly.

Although the proposed virtual keyboard has a number of advantages, the limitations are also undeniable. The recognition algorithms are based on grayscale levels of digital images. Consequently, lighting is able to influence the recognition rate. Unlike



mechanical keyboards, our virtual keyboard cannot work properly in very dim environments. Aside from the shadows, some colors are also regarded as a part of the user's hand so very colorful customized paper keyboards are not suitable for this virtual keyboard.

# **Chapter 6 Conclusion and Future Work**

## **6.1 Conclusion**

Previous research presents a great number of gesture recognition systems that were applied to relevant realistic applications. However, there is not much research done in the field of virtual keyboards since touch screens are currently still mainstream. The existing virtual keyboard systems have various limitations such as add-on, high costs, limited portability, and uncomfortable designs. In this thesis, the virtual keyboard is developed to identify typing by single finger on a paper keyboard under a fixed camera in a stable environment.

The novelties of the proposed virtual keyboard are expressed in four aspects. Firstly, the keyboard is merely a paper as the media, which means it preserves many resources and does not generate much rubbish. Thus, it is very environmentally friendly. Secondly, as a paper keyboard, it can be placed on any plane. Thirdly, users can expediently customize their virtual keyboards, especially since the proposed virtual keyboard also accepts handwritten keyboards. Finally, it is simple and portable because a normal camera and a paper keyboard can implement it.

The experiments are designed to test the adaptability of the virtual keyboard to various environments and customized keyboards. The two experimental environments are under lamplight in a bedroom in a house and under natural light in an office at our university. Five types of customized paper keyboards are used and one of them is handwritten. The overall recognition rate from testing is 94.62%. Another novel feature of this virtual keyboard is that it allows in-air typing to ignore some obstructions. Thus, the usability of the proposed virtual keyboard is proved. Moreover, the testers think that they can type properly on the virtual keyboard and are interested in future developments.

## 6.2 Future Work

Future work will focus on improving the performances of the proposed virtual keyboard. The anticipated future improvements are as follows: 1) the virtual keyboard can work in real time, which means this virtual keyboard is deployed into practice; 2) it can identify two hands typing simultaneously so it can be utilized on computers; 3) it can work in complex backgrounds or dim environments; 4) other function keys such as enter, delete, space, shift, etc, can be added on the virtual keyboard; 5) the diversity in participants should be considered in the future since the previous testers are only three university students of similar age. Children and elderly people can also be invited to participate in the tests. If more testers can be invited to take part in the experiments in future, more advice and feedback can also be gathered to improve the virtual keyboard.

In addition, if some other machine learning technologies like  $k$ -NN, HMM or ANN can be incorporated into this virtual keyboard, the recognition performance will surely be greatly improved. The author believes that ANNs offer significant advantages for dynamic gesture recognition and character recognition.

Finally, 3D projection technology will likely be more popular and less costly in the near future. At that time, the virtual keyboard could combine with 3D projection technology to project a fantastic projected virtual keyboard in mid-air to replace the previous paper keyboard. This will be an amazing experience for users.

# References

- Abu-Mostafa, Y. S. (1990). Learning from hints in neural networks. *Journal of complexity*, 6(2), 192-198.
- Adachi, T., Furuya, R., Greene, S., & Mikuriya, K. (1991). Feature selection for neural network recognition. In *IEEE International Joint Conference on Neural Networks*, pp. 696-701.
- Argyriou, A., Maurer, A., & Pontil, M. (2008). An algorithm for transfer learning in a heterogeneous environment. In *Machine Learning and Knowledge Discovery in Databases*, pp. 71-85.
- Athitsos, V., Wang, H., & Stefan, A. (2010). A database-based framework for gesture recognition. *Personal and Ubiquitous Computing*, 14(6), 511-526.
- Badal, D. Z. (1993). Neural network recognition of human face images stored in the database. In *Twelfth Annual International Phoenix Conference on Computers and Communications*, pp. 552-558.
- Bailador, G., Roggen, D., Tröster, G., & Triviño, G. (2007). Real time gesture recognition using continuous time recurrent neural networks. In *Proceedings of the ICST 2nd international conference on Body area networks*, pp. 15.
- Baudel, T., & Beaudouin-Lafon, M. (1993). Charade: remote control of objects using free-hand gestures. *Communications of the ACM*, 36(7), 28-35.
- Baxter, J. (1995). Learning internal representations. In *Proceedings of the eighth annual conference on Computational learning theory*, pp. 311-320.
- Baxter, J. (2000). A model of inductive bias learning. *J. Artif. Intell. Res.(JAIR)*, 12, 149-198.
- Bérci, N., & Szolgay, P. (2009). Towards a gesture based human-machine interface: Fast 3D tracking of the human fingers on high speed smart camera computers. In *IEEE International Symposium on Circuits and Systems*, pp. 1217-1220.
- Billon, R., Nedelec, A., & Tisseau, J. (2008). Gesture recognition in flow based on PCA analysis using multiagent system. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology* (pp. 139-146).

- Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144-152.
- Cao, W. M., Lu, F., Gu, Y. B., Peng, H., & Wang, S. (2004). Study of human face recognition based on principal component analysis (PCA) and direction basis function neural networks. In *Fifth World Congress on Intelligent Control and Automation*, Vol. 5, pp. 4195-4198.
- Chan, T. M. (2000). Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pp. 300-309.
- Chapelle, O., Scholkopf, B., & Zien, A. (2006). Semi-supervised learning. *Cambridge, MA: MIT Press*.
- Chen, X., & Koskela, M. (2013). Online RGB-D gesture recognition with extreme learning machines. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 467-474.
- Cheng, H., Luo, J., & Chen, X. (2014). A windowed dynamic time warping approach for 3D continuous hand gesture recognition. In *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-6.
- Corradini, A., & Gross, H. M. (2000). Camera-based gesture recognition for robot control. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Vol. 4, pp. 133-138.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.
- Eaton, E. (2006). Multi-Resolution Learning for Knowledge Transfer. In *AAAI*, pp. 1908-1909.
- Evgeniou, T., & Pontil, M. (2004). Regularized multi--task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109-117.
- Gadea, C., Ionescu, B., Ionescu, D., Islam, S., & Solomon, B. (2012). Finger-based gesture control of a collaborative online workspace. In *7th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp.

41-46.

- Gao, J., Fan, W., Jiang, J., & Han, J. (2008). Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 283-291.
- Hemayed, E. E. (2003). A survey of camera self-calibration. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 351-357.
- Hsieh, C. C., Liou, D. H., & Lee, D. (2010). A real time hand gesture recognition system using motion history image. In *2nd International Conference on Signal Processing Systems (ICSPS)*, Vol. 2, pp. V2-394.
- Hu, G., & Gao, Q. (2011). Gesture Analysis Using 3D Camera, Shape Features and Particle Filters. In *Canadian Conference on Computer and Robot Vision (CRV)*, pp. 204-211.
- Intrator, N. (1996). Making a low-dimensional representation suitable for diverse tasks. *Connection Science*, 8(2), 205-224.
- Ionescu, D., Ionescu, B., Gadea, C., & Islam, S. (2011). An intelligent gesture interface for controlling TV sets and set-top boxes. In *6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 159-164.
- Ishijima, R., Ogawa, K., Higuchi, M., & Komuro, T. (2014). Real-time typing action detection in a 3D pointing gesture interface. In *Proceedings of the 5th Augmented Human International Conference*, pp. 20.
- Jeong, S., Jin, J., Song, T., Kwon, K., & Jeon, J. W. (2012). Single-camera dedicated television control system using gesture drawing. *IEEE Transactions on Consumer Electronics*, 58(4), 1129-1137.
- Jie, X., Jinfang, C., Guangjin, H., & Xiudong, Y. (2011). A neural network recognition model based on ship acoustic-magnetic field. In *Fourth International Symposium on Computational Intelligence and Design (ISCID)*, Vol. 1, pp. 135-138.
- Jusman, Y., Sulaiman, S. N., Isa, N. A. M., Yusoff, I. A., Adnan, R., Othman, N. H., & Zaki, A. (2009). Capability of new features from FTIR spectral of cervical cells for cervical precancerous diagnostic system using MLP networks. In *IEEE Region 10 Conference TENCN*, pp. 1-6.
- Keskin, C., Erkan, A., & Akarun, L. (2003). Real time hand tracking and 3d gesture

- recognition for interactive interfaces using hmm. *ICANN/ICONIPP,2003*, 26-29.
- Kim, M. S., Kim, D., & Lee, S. Y. (2003). Face recognition using the embedded HMM with second-order block-specific observations. *Pattern Recognition*, 36 (11), 2723-2735.
- Kristensson, P. O., Nicholson, T., & Quigley, A. (2012). Continuous recognition of one-handed and two-handed gestures using 3D full-body motion tracking sensors. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pp. 89-92.
- Kumar, P., Mitchell, J. S., & Yildirim, E. A. (2003). Approximate minimum enclosing balls in high dimensions using core-sets. *Journal of Experimental Algorithmics (JEA)*, 8, 1-1.
- Kumar, S., & Segen, J. (1999). Gesture based 3d man-machine interaction using a single camera. In *IEEE International Conference on Multimedia Computing and Systems*, Vol. 1, pp. 630-635.
- Kurakin, A., Zhang, Z., & Liu, Z. (2012). A real time system for dynamic hand gesture recognition with a depth sensor. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pp. 1975-1979.
- Lawrence, N. D., & Platt, J. C. (2004). Learning to learn with the informative vector machine. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 65.
- Lech, M., & Kostek, B. (2010). Gesture-based computer control system applied to the interactive whiteboard. In *2nd International Conference on Information Technology (ICIT)*, pp. 75-78.
- Lee, U., & Tanaka, J. (2013). Finger identification and hand gesture recognition techniques for natural user interface. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, pp. 274-279.
- Megiddo, N. (1982). Linear-time algorithms for linear programming in R3 and related problems. In *23rd Annual Symposium on Foundations of Computer Science*, pp. 329-338.
- Min, B. W., Yoon, H. S., Soh, J., Yang, Y. M., & Ejima, T. (1997). Hand gesture recognition using hidden Markov models. In *IEEE International Conference on*

- Systems, Man, and Cybernetics*, Vol. 5, pp. 4232-4235.
- Mitchell, T. M. (1980). *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ.
- Mitchell, T. M. (1999). Machine learning and data mining. *Communications of the ACM*, 42(11), 30-36.
- Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3), 311-324.
- Murakami, K., & Taguchi, H. (1991). Gesture recognition using recurrent neural networks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 237-242.
- Murthy, R., & Jadon, R. S. (2010). Hand gesture recognition using neural networks. In *International Advance Computing Conference (IACC)*, pp. 134-138.
- Nandakumar, K., Wan, K. W., Chan, S. M. A., Ng, W. Z. T., Wang, J. G., & Yau, W. Y. (2013). A multi-modal gesture recognition system using audio, video, and skeletal joint data. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 475-482.
- Nawaz, T., Mian, M. S., & Habib, H. A. (2008). Infotainment devices control by eye gaze and gesture recognition fusion. *Consumer Electronics, IEEE Transactions on*, 54(2), 277-282.
- Niikura, T., Watanabe, Y., Komuro, T., & Ishikawa, M. (2012). In-air Typing Interface: Realizing 3D operation for mobile devices. In *2012 IEEE 1st Global Conference on Consumer Electronics (GCCE)*, pp. 223-227.
- Nikhil, S., Mohan, S., Ramya, B., & Kadambi, G. R. (2010). Design and development of a dsp processor based reconfigurable hand gesture recognition system for real time applications. In *2010 International Conference on Signal and Image Processing (ICSIP)*, pp. 39-44.
- Ozawa, S., Pang, S., & Kasabov, N. (2006). Incremental learning of feature space and classifier for on-line pattern recognition. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 10(1), 57-65.
- Ozawa, S., Pang, S., & Kasabov, N. (2006). On-line feature selection for adaptive evolving connectionist systems. *International Journal of Innovative Computing*,



*Information and Control*, 2(1), 181-192.

- Ozawa, S., Pang, S., & Kasabov, N. (2008). Incremental learning of chunk data for online pattern classification systems. *IEEE Transactions on Neural Networks*, 19(6), 1061-1074.
- Ozawa, S., Roy, A., & Roussinov, D. (2009). A multitask learning model for online pattern recognition. *IEEE Transactions on Neural Networks*, 20(3), 430-445.
- Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359.
- Pang, S., Ozawa, S., & Kasabov, N. (2005). Incremental linear discriminant analysis for classification of data streams. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(5), 905-914.
- Panwar, M. (2012). Hand gesture recognition based on shape parameters. In *International Conference on Computing Communication and Applications (ICCCA)*, pp. 1-6.
- Prasad, J. S., Saxena, A., Javar, N., Kaushik, K. B., Chakraborty, P., & Nandi, G. C. (2010, December). Gesture recognition by stereo vision. In *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia*, pp. 155-162.
- Raheja, J. L., Shyam, R., Kumar, U., & Prasad, P. B. (2010). Real-time robotic hand control using hand gestures. In *Second International Conference on Machine Learning and Computing (ICMLC)*, pp. 12-16.
- Roeber, H., Bacus, J., & Tomasi, C. (2003). Typing in thin air: the canesta projection keyboard-a new method of interaction with electronic devices. In *CHI'03 extended abstracts on Human factors in computing systems*, pp. 712-713.
- RongQing, Y., WenHui, L., Duo, W., & Hua, Y. (2008). Feature Recognition Based on Graph Decomposition and Neural Network. In *Third International Conference on Convergence and Hybrid Information Technology*, Vol. 2, pp. 864-868.
- Rowley, H., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 23-38.
- Roy, A., Govil, S., & Miranda, R. (1995). An algorithm to generate radial basis function

- (RBF)-like nets for classification problems. *Neural networks*, 8(2), 179-201.
- Roy, A., Govil, S., & Miranda, R. (1997). A neural-network learning theory and a polynomial time RBF algorithm. *IEEE Transactions on Neural Networks*, 8(6), 1301-1313.
- Roy, A., Kim, L. S., & Mukhopadhyay, S. (1993). A polynomial time algorithm for the construction and training of a class of multilayer perceptrons. *Neural networks*, 6(4), 535-545.
- Roy, A., & Mukhopadhyay, S. (1996). Iterative generation of higher-order nets in polynomial time using linear programming. *IEEE Transactions on neural networks/a publication of the IEEE Neural Networks Council*, 8(2), 402-412.
- Sagara, S., Higuchi, M., & Komuro, T. (2014). Multi-finger AR typing interface for mobile devices. In Proceedings of the companion publication of the 19th international conference on Intelligent User Interfaces, pp. 13-16.
- Segen, J., & Kumar, S. (1998). Fast and accurate 3D gesture recognition interface. In *Fourteenth International Conference on Pattern Recognition*, Vol. 1, pp. 86-91.
- Segen, J., & Kumar, S. (1999). Shadow gestures: 3D hand pose estimation using a single camera. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1.
- Shi, Y., Taib, R., & Lichman, S. (2006). GestureCam: a smart camera for gesture recognition and gesture-controlled web navigation. In *9th International Conference on Control, Automation, Robotics and Vision*, pp. 1-6.
- Silver, D., & Mercer, R. (2001). Selective functional transfer: Inductive bias from related tasks. In *IASTED International Conference on Artificial Intelligence and Soft Computing (ASC2001)*, pp. 182-189.
- Silver, D. L., & Poirier, R. (2004). *Sequential consolidation of learned task knowledge*, pp. 217-232.
- Singh, S., & Amin, A. (1998). Neural network recognition and analysis of hand-printed characters. In *Proc. IEEE International Joint Conference on Neural Networks*, Vol. 98, pp. 4-9.
- Sohn, M. K., Lee, S. H., Kim, D. J., Kim, B., & Kim, H. (2013). 3D hand gesture recognition from one example. In *IEEE International Conference on Consumer*

*Electronics (ICCE)*, pp. 171-172.

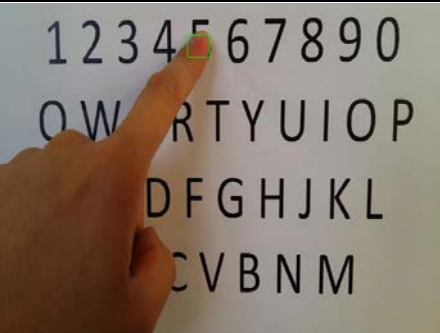
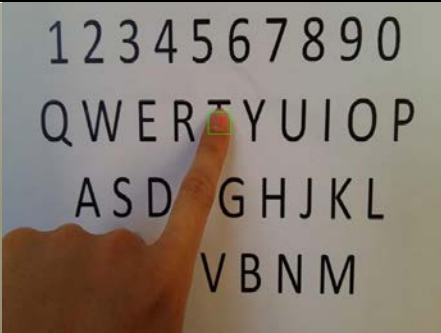

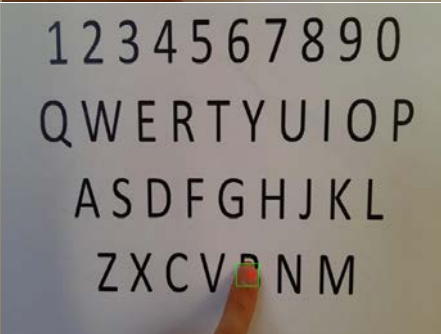




- Song, Y., Demirdjian, D., & Davis, R. (2012). Continuous body and hand gesture recognition for natural human-computer interaction. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(1), 5.
- Stefan, A., Athitsos, V., Alon, J., & Sclaroff, S. (2008). Translation and scale-invariant gesture recognition in complex scenes. In *Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*, pp. 7.
- Stergiopoulou, E., & Papamarkos, N. (2009). Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence*, 22(8), 1141-1158.
- Su, M. C., Jean, W. F., & Chang, H. T. (1996). A static hand gesture recognition system using a composite neural network. In *Fifth IEEE International Conference on Fuzzy Systems*, Vol. 2, pp. 786-792.
- Suarez, J., & Murphy, R. R. (2012). Hand gesture recognition with depth images: A review. In *RO-MAN*, pp. 411-417.
- Tam, V., & Li, L. S. (2012). Integrating the Kinect camera, gesture recognition and mobile devices for interactive discussion. In *2012 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pp. H4C-11.
- Terajima, K., Komuro, T., & Ishikawa, M. (2009). Fast finger tracking system for in-air typing interface. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pp. 3739-3744.
- Thrun, S. (1996). Is learning the n-th thing any easier than learning the first?. *Advances in neural information processing systems*, 640-646.
- Trigueiros, P., Ribeiro, F., & Reis, L. P. (2012). A comparison of machine learning algorithms applied to hand gesture recognition. In *7th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1-6.
- Tsang, I. W., Kwok, J. T., & Cheung, P. M. (2005). Core vector machines: Fast SVM training on very large data sets. In *Journal of Machine Learning Research*, pp. 363-392.
- Utsumi, A., & Ohya, J. (1999). Multiple-hand-gesture tracking using multiple cameras. In *IEEE Computer Society Conference on Computer Vision and Pattern*

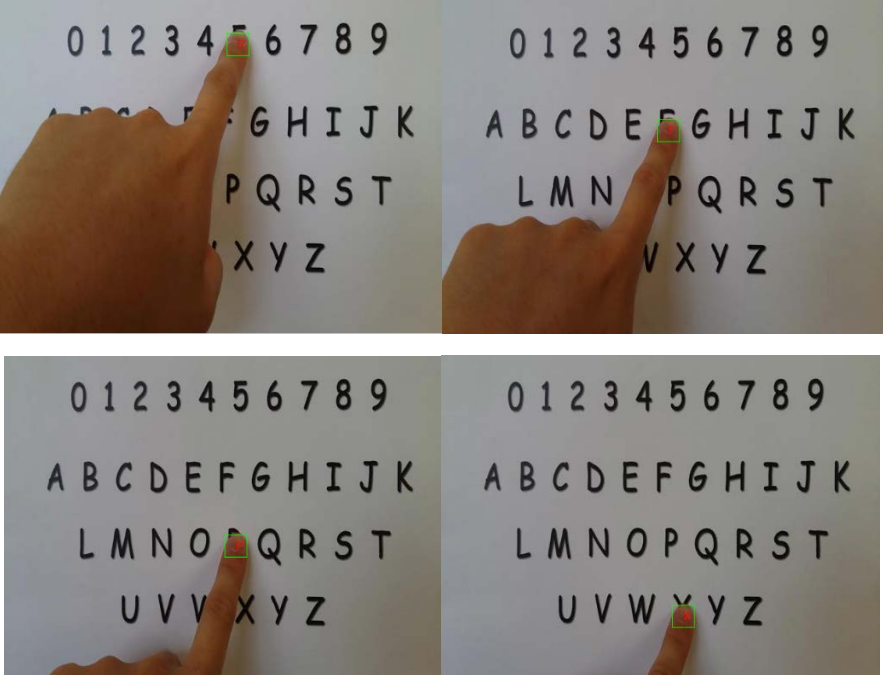
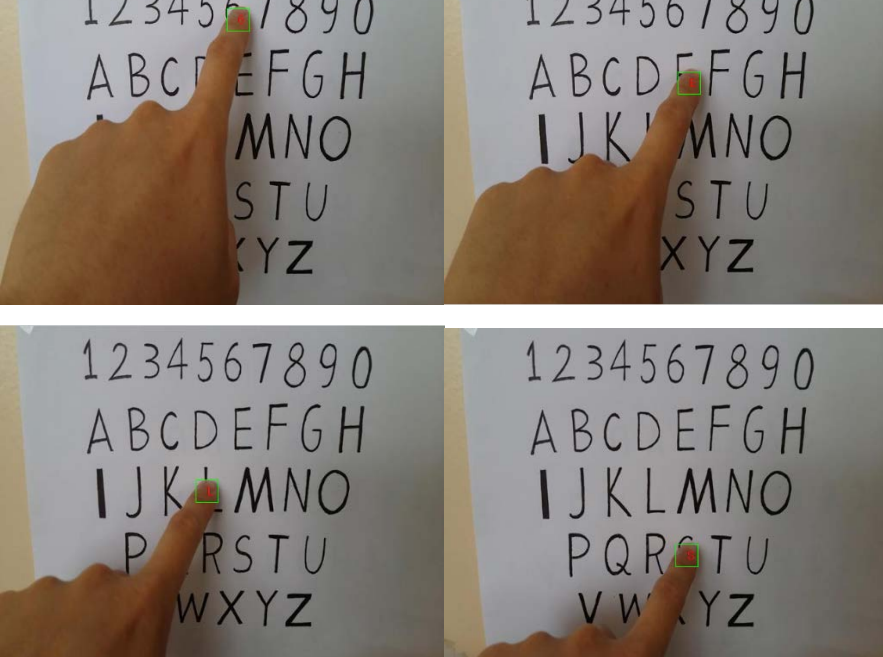
- Watson, R. (1993). *A survey of gesture recognition techniques*. Trinity College Dublin, Department of Computer Science.
- Wu, C. H., & Lin, C. H. (2013). Depth-based hand gesture recognition for home appliance control. In *IEEE 17th international symposium on consumer electronics (ISCE)*, pp. 279-280.
- Wu, J., Cheng, J., Zhao, C., & Lu, H. (2013). Fusing multi-modal features for gesture recognition. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 453-460.
- Wu, J., Pan, G., Zhang, D., Qi, G., & Li, S. (2009). Gesture recognition with a 3D accelerometer. In *Ubiquitous intelligence and computing*, pp. 25-38.
- Wu, Y., & Huang, T. S. (2001). Hand modeling, analysis and recognition. *Signal Processing Magazine, IEEE*, 18(3), 51-60.
- Xu, D. (2006). A neural network approach for hand gesture recognition in virtual reality driving training system of SPG. In *18th International Conference on Pattern Recognition*, Vol. 3, pp. 519-522.
- Xue, Y., Liao, X., Carin, L., & Krishnapuram, B. (2007). Multi-task learning for classification with Dirichlet process priors. *The Journal of Machine Learning Research*, 8, 35-63.
- Yang, J., Xu, J., Li, M., Zhang, D., & Wang, C. (2011). A real-time command system based on hand gesture recognition. In *Seventh International Conference on Natural Computation (ICNC)*, Vol. 3, pp. 1588-1592.
- Yeung, D. S., & Fong, H. S. (1994). A neural network recognition system for handwritten Chinese character using structured approach. In *IEEE International Conference on Neural Networks*, Vol. 7, pp. 4353-4358.
- Yoon, H. S., Soh, J., Bae, Y. J., & Yang, H. S. (2001). Hand gesture recognition using combined features of location, angle and velocity. *Pattern recognition*, 34(7), 1491-1501.
- Youyi, J., & Xiao, L. (2010). A method for face recognition based on wavelet neural network. In *Second WRI Global Congress on Intelligent Systems (GCIS)*, Vol. 3, pp. 133-136.

- Yu, K., Schwaighofer, A., & Tresp, V. (2002). Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical Bayes. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pp. 616-623.
- Zelevnik, R., & Forsberg, A. (1999). UniCam—2D gestural camera controls for 3D environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pp. 169-173.
- Zeng, B., Wang, G., & Lin, X. (2012). A hand gesture based interactive presentation system utilizing heterogeneous cameras. *Tsinghua Science and Technology*, 17(3), 329-336.
- Zhang, T., & Feng, Z. (2013). Dynamic Gesture Recognition Based on Fusing Frame Images. In *Fourth International Conference on Intelligent Systems Design and Engineering Applications*, pp. 280-283.





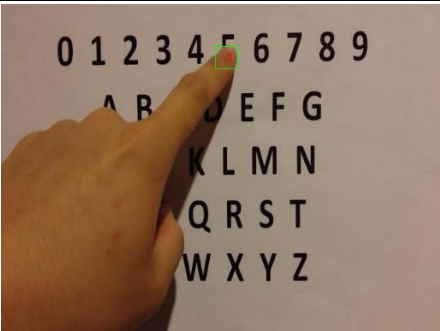
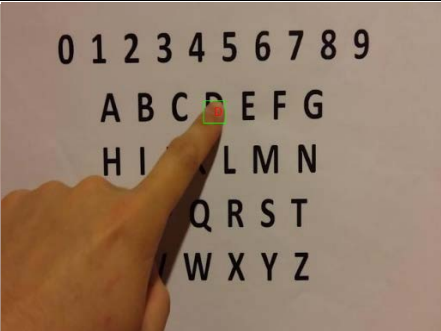
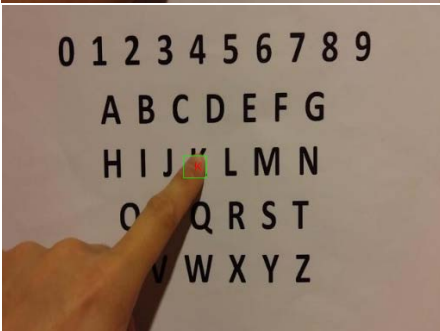

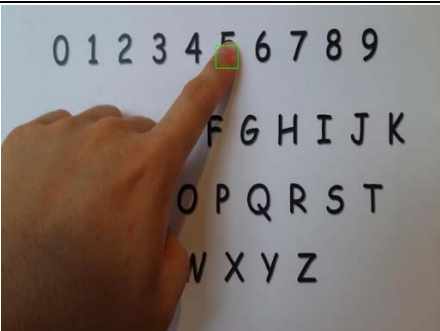

# Appendix

## Condition 1: Examples of testing under natural light



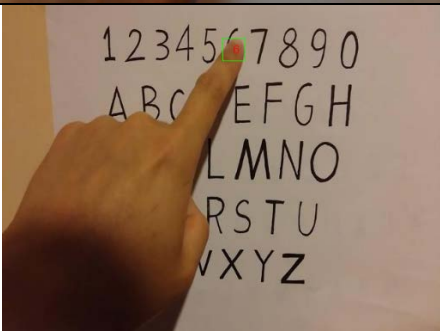
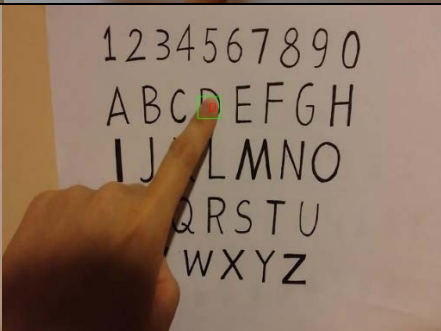
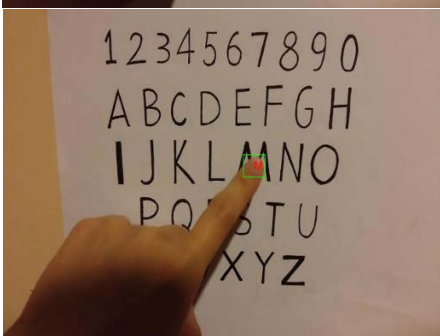
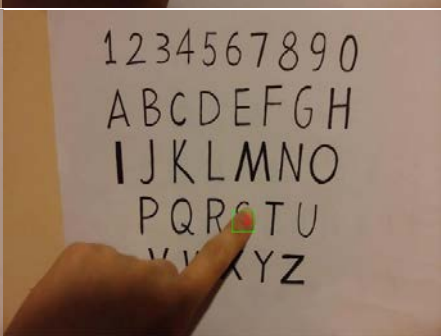
Keyboard type	Examples	
Keyboard (a)		
		
Keyboard (b)		
		

<p><b>Keyboard (c)</b></p>	
<p><b>Keyboard (d)</b></p>	

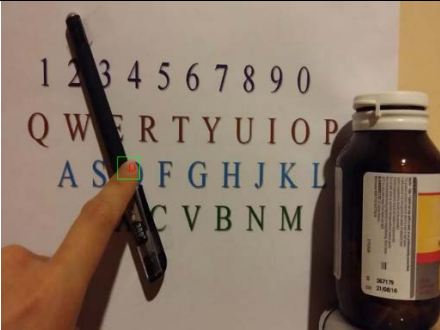
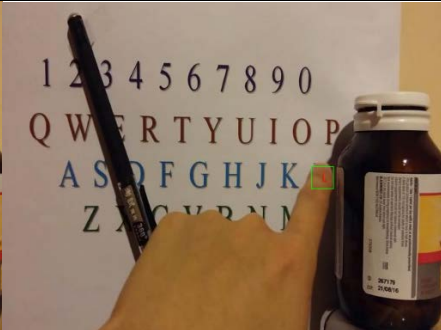
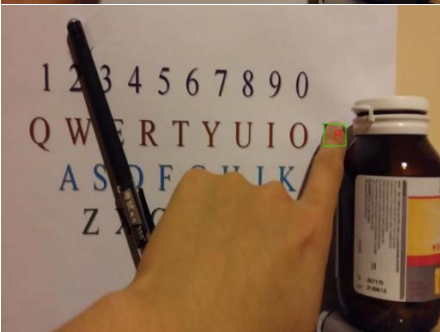
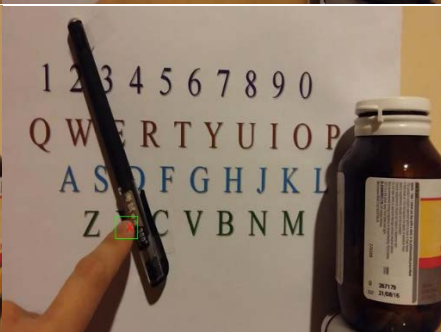
## Condition 2: Examples of testing under lamplight

Keyboard type	Examples	
Keyboard (a)		
		
Keyboard (b)		
		
Keyboard (c)		



	 
<b>Keyboard (d)</b>	   

### Condition 3: Examples of testing with occlusions

Occlusions	Examples
<b>Pen and bottle</b>	   

**Comb and cup**

