Waseda University Master Thesis

# Realizing 3D Multi-Touch Interactions Using Depth Camera

44151560-8: ZHOU Li

Master (Engineering)

Supervisor: Professor Jiro TANAKA

Interactive Programming
Information Architecture
The Graduate School of Information, Production and Systems

July 2017

ABSTRACT

We all know that the emergence of multi-touch changes the traditional human-computer interaction. And with the development of 3D technology, we can not just stay in the 2D stage. 3D interactions become more and more popular. Some gestures are gradually applied in 3D space. And also 2D gestures became a little weak in interacting with machine. For example 2D multi-touch interaction detects points rather than fingers. But in 3D space, we can easily get users' finger information. That means there will be more types of gestures in 3D. And 3D gestures can be more natural to interact with computers. 3D interactions become more and more popular and users all like natural gestures. Therefor, we present a 3D multi-touch interaction using depth camera.

In this research, we generated a 3D multi-touch interaction by using Leap Motion. User can interact with computer by 3D multi-touch gestures over Leap Motion. Our approach is combine static data classification and dynamic data detect. Our system can gestures in 3D space using a depth camera Leap Motion. And also we give a sample application just like a photo viewer. It can detect 5 kinds of different gestures and give feedback. And if we need to detect more gestures, we can change the model file at any time. And if we need to detect more gestures, we can change the model file at any time. Moreover, because of that there is no touching action in 3D space, so we define a concept of "touch" state just like tap or click. It is a very important part for multi-touch gestures. We design a method “click down” and “click up”. When the fingers “click down” means activate the fingers and get into “touch” state, “click up” means release the fingers and back to “nature” state. We can describe the “touch” state as activating some fingers by “click down” condition and release by "click up" condition.

Finally we implement the system. Our system can interact with users very friendly. For every part of our system, its accuracy of recognition is more than 90%. Although in this work we only test our gesture interaction in a sample application as photo viewer, it also suitable for other possible application and cooperative work if gesture interaction is needed.

Keywords: Multi-Touch, 3D Gestures, Depth Camera, Leap Motion

# ACKNOWLEDGMENTS

It is a significant and unforgettable experience for me to spend such a long time on research and writing this thesis. Many kind-hearted people give me a great amount of help, professional advice and encouragement. Thus, here I would like to express my sincere gratitude to them.

First of all, I will show my sincere thanks to my supervisor Professor Jiro TANAKA has given me invaluable guidance and constant support and encouragement throughout the whole processes. During my research of Human Computer Interaction in this two years and the witting of this thesis, he has always been generous to offer me useful suggestions, answer my question and discuss research questions with me. His detail guidance and helpful comments on my thesis encourage me to overcome difficulties and think more critically during the processes of writing.

Secondly, I will also express my sincere gratitude to other professor and teachers who had given me constructive suggestion to me and thanks to them for instructing me to construct my knowledge in Computer Science in the past two years.

Thirdly, I want to thank all my affectionate laboratory members and friends. During the two years' research and the writing of this thesis, they have given me a lot of support and encouragement in their own ways.

At last, I will give the deepest gratitude to my family for their selfless support and love to me.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Research Background

Multi-Touch[1][2] is technology that enables a surface (a touchpad or touchscreen) to recognize the presence of more than one points of contact with the surface. Multi-touch is an innovation in the field of human-computer interaction after the mouse and keyboard[3].

Gesture-based interaction has been already a research topic for more than 30 years. It has shown that gesture can be a good choice for controlling the computer[4][5]. Wachs et al.[6] pointed out that gestures are useful for computer interaction because they are the most important and manifestations of human communication. They also indicate that the gesture should be designed intuitively and used by the user; that is, a gesture should be related to the operation. Gesture data extraction is an important process in gesture recognition, mainly divided into two categories[7]: glove-based, mainly to computer vision. Data gloves can be used to accurately obtain gesture data, but they usually hinder the user's tether and set the time. However, recognition based on computer vision is non-invasive and unconstrained, allowing the user to perform any special operating hardware or device so that he can interact with the system to produce more natural interactions[8]. This is consistent with the "interpersonal interaction" (HCI) design[9].

## 1.2 Multi-Touch

Touch technology people are no stranger to the bank ATM machines are mostly touch screen function, many hospitals, libraries and other hall has this touch technology computer, support touch screen mobile phone, MP3, digital camera is also a lot[10]. But these existing touch screens are single touch, can only identify and support each time a finger touch, click, if more than two points at the same time be touched, you can not make the right response, and Multi-touch technology can break down the task into two aspects of the work, one is to collect multi-point signal, the second is the meaning of each signal to judge, that is, the so-called gesture recognition, in order to achieve the screen recognition of five Finger at the same time do the click, touch action[11].

Through the observation of the previous study, found that there are many shortcomings and limitations before the paper, for example, 3, the user had to hover their hands on the screen button waiting for 1 second, to trigger the action, Longer response times may cause disgust and inconvenience to the user, and do not allow the user to cancel the operation after the trigger time has passed, but also to the user a lot of inconvenience[12][13]. Using crossover techniques, crossover techniques refer to cross events when the cursor intersects the boundary of the graphic object. While this technique is effective, these crossover techniques are limited for both a basic choice of operation[14]. Use the "index pointing, thumb up" gesture to perform the selection, which is not an intuitive choice gesture, since it is independent of its associated operation[15]. The posture used is defined by the lifting or folding of the fingers, and whether the thumb is stretched, aligned with the palm, or caught in the palm of the hand, these are not simple operations. With the further increase in vocabulary, the required gestures also increase, which requires the system to improve the input accuracy[16] The menu item is numbered, and the user must extend the corresponding number of fingers to select the given item. This forces the finger to count, so it can not be used for other operations[17]. Only the index finger of the hand is clicked, and the click gesture is limited to the selection mechanism.

Multi-touch is an innovation in the field of human-computer interaction after the mouse and keyboard. After our investigation[18], we found some kinds of multi-touch gestures are used in our daily life. Here are 14 kinds of multi-touch gestures that we found in the Internet, we did a survey of their use and function. These 14 kinds of gestures are very popular in our daily life, but also some other gestures that not so popular in our life. So choose these 14 kinds of gestures to do this survey[1].

**Table 1.1** Survey of some 2D multi-touch gestures.

| Action name | How to use | Function |
|---|---|---|
| a. Secondary click | Click or tap with two fingers on the touchpad | To perform the equivalent of Control-click or right-click. |
| b. Smart zoom | Double-tap with two fingers on the touchpad | To zoom in and back out of a webpage or PDF. |
| c. Scroll, slide, swipe | Slide two fingers up or down on the touchpad | To scroll. |
| d. Zoom in/out | Pinch with two fingers on the touchpad | To zoom in or out. |
| e. Rotate | Move two fingers around each other on the touchpad. | To rotate a photo or other item. |
| f. Swipe between pages | Swipe left or right with two fingers on the touchpad | To show the previous or next page. |

[1]https://support.apple.com/en-us/HT204895

**Table 1.2** Survey of some 2D multi-touch gestures.

| Action name | How to use | Function |
|---|---|---|
| g. Open Notification Center | Swipe left from the right edge with two fingers on the touch-pad | To show Notification Center. |
| h. Three finger drag | Use three fingers drag on the touchpad | To drag items on your screen, then click or tap to drop. |
| i. Look up and data detectors | Tap with three fingers on the touchpad | To look up a word or take other actions with dates, addresses, phone numbers, and other data. |
| j. Show desktop | Spread your thumb and three fingers apart | to show the desktop. |
| k. Launchpad | Pinch your thumb and three fingers together | To display Launchpad. |
| l. Mission Control | Swipe up with four fingers on the touchpad | To open Mission control. |
| m. App Exposé | Swipe down with four fingers on the touchpad | to see all windows of the app you're using. |
| n. Swipe between full-screen apps | Swipe left or right with four fingers on the touchpad | To move between desktops and full-screen apps. |

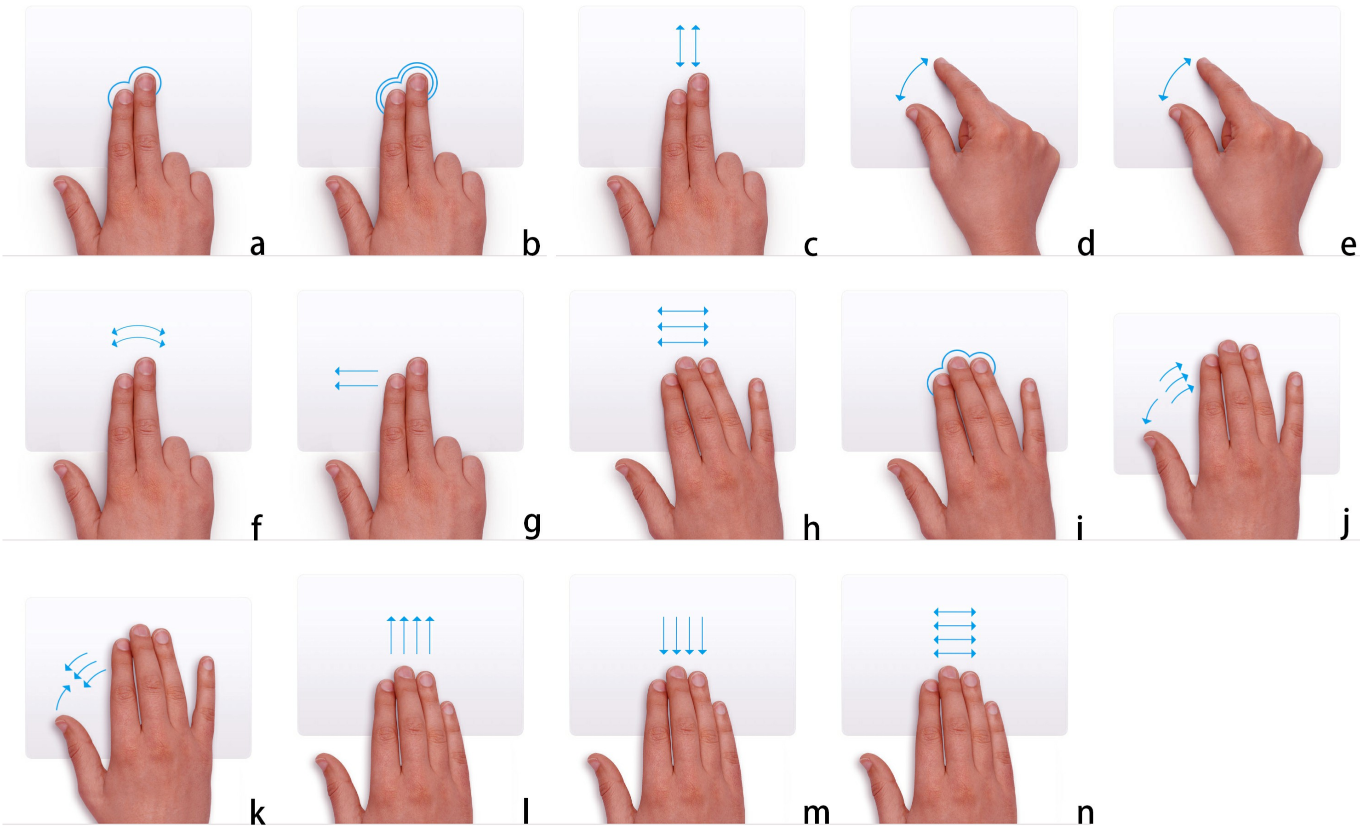



**Figure 1.1** 14 kinds of multi-touch gestures in 2D.

With the development of 3D technology, we can not just stay in the 2D stage. 2D multi-touch interaction detects points rather than fingers. But in 3D space, we can easily get users' finger information. That means there will be more types of gestures in 3D. And 3D gestures can be more natural to interact with computers. That is why we need 3D multi-touch.

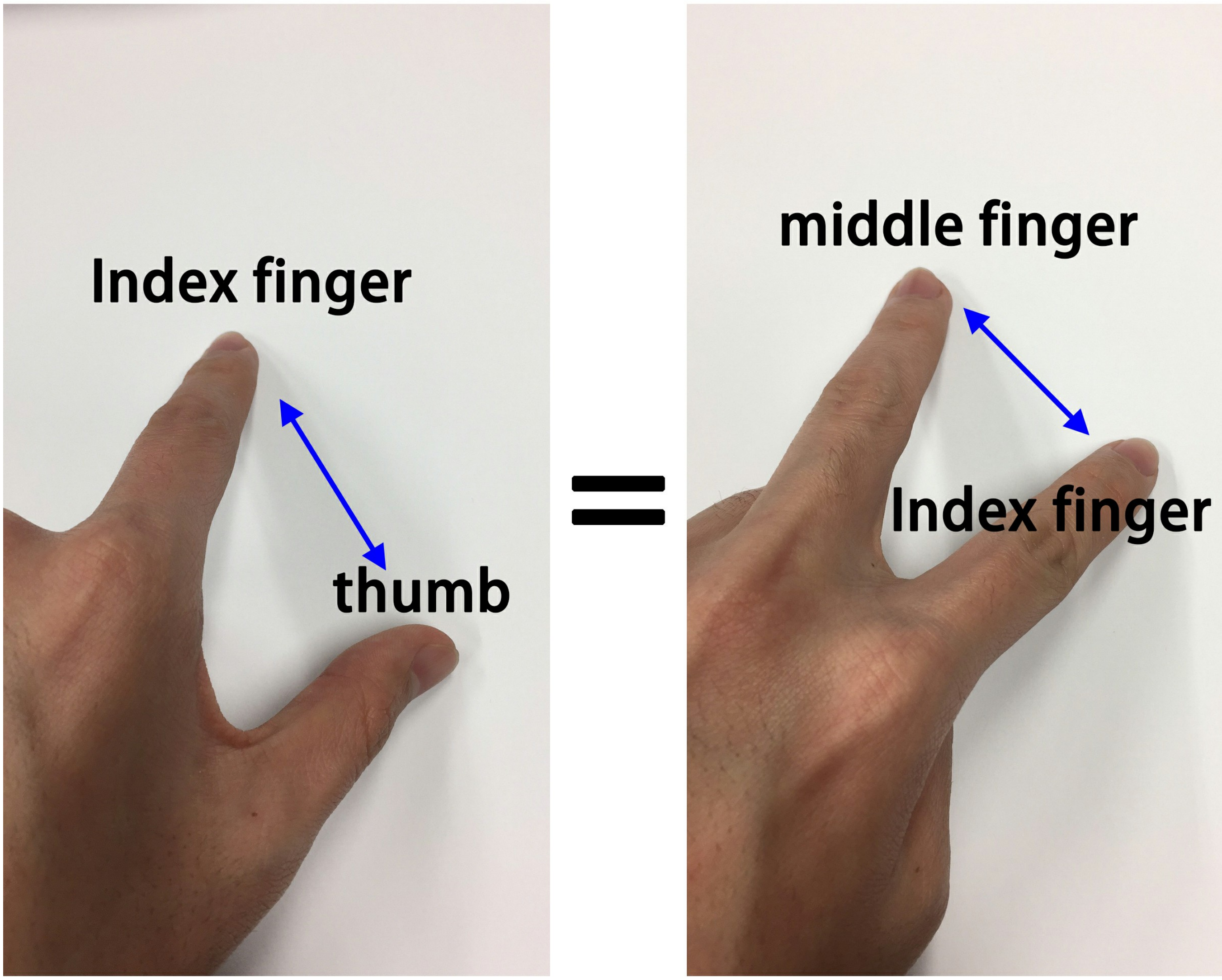



**Figure 1.2** In the case of 2D, these 2 kinds of gestures have the same meaning.

## 1.3 Target Problem

With the development of 3D technology, everyone enjoys 3D things and 3D interactions. 3D interactions become more and more popular. Some gestures are gradually applied in 3D space. And also 2D interactions became a little weak in interacting with machine. For example 2D multi-touch interaction detects points rather than fingers. But in 3D space, we can easily get users' finger information. That means there will be more types of gestures in 3D. And 3D gestures can be more natural to interact with computers. 3D interactions become more and more popular and users all like natural gestures. Therefor, we need 3D multi-touch interactions and we need use gestures in 3D.

In nowadays daily life, only the 2D field of operation has been unable to meet the needs of todays people, so we need 2d in the multi-touch to expand into 3D field, which is our paper to solve the problem, That is, to realize a 3D multi-touch interaction using depth camera.

## 1.4 Structure of this paper

The structure of this paper is designed as follows: Chapter 1 introduces some basic background and some definition on this field. Chapter 2 given the goal and approach of our research. In Chapter 3, we describe our system give the implementation of our system with such methods. And also give some evaluation of each part of the system. Chapter 4 gives the conclusion and discussion of our future work.

# Chapter 2

# Research Goal and Approach

Our research goal is to realize a Natural 3D Multi-touch Interaction, to connect the real world and the virtual world.

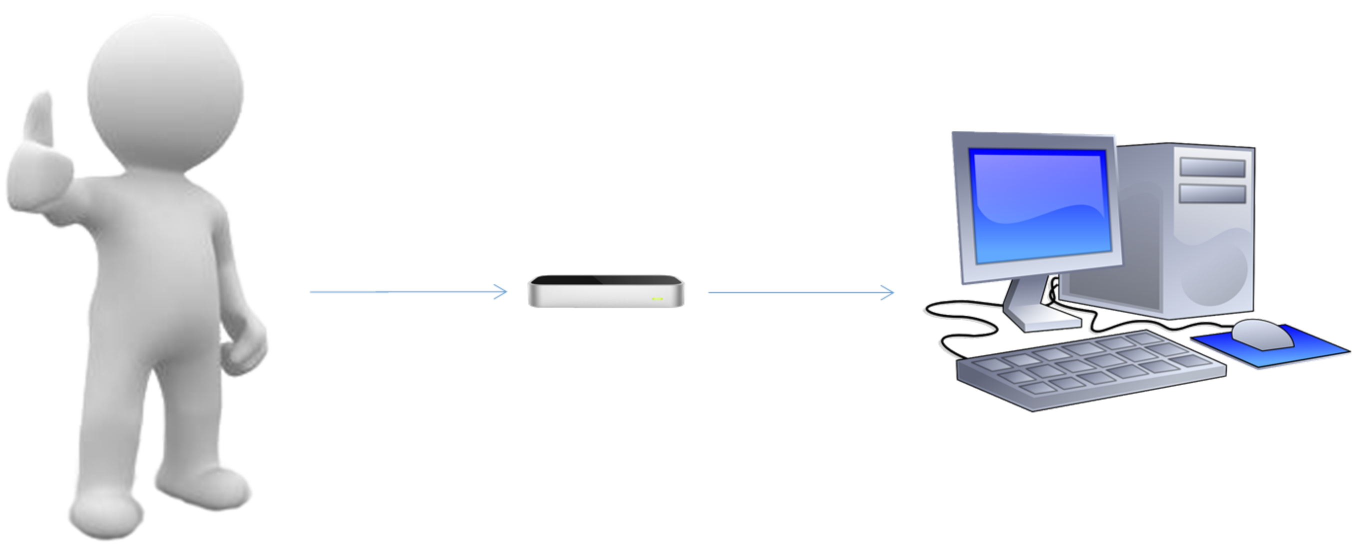


**Figure 2.1** Our research goal is to realize a Natural 3D Multi-touch Interaction, to connect the real world and the virtual world.

We all know that the emergence of multi-touch changes the traditional human-computer interaction. But with the development of 3D technology, we can not just stay in the 2D stage. 3D interactions become more and more popular. Some gestures are gradually applied in 3D space. And also 2D gestures became a little weak in interacting with machine. For example 2D multi-touch interaction detects points rather than fingers. But in 3D space, we can easily get users' finger information. That means there will be more types of gestures in 3D. And 3D gestures can be more natural to interact with computers. 3D interactions become more and more popular and users all like natural gestures. Therefor, we need 3D multi-touch interactions and we need use gestures in 3D.

But there are also some gaps to realize 3D multi-touch. For example in the case of 2D multi-touch gestures, when we want to do some gestures our fingers have to get into a state, touching the touchpad or screen. But in 3D we could not touch anything. So we need to define a "touch" state of 3D multi-touch, it is the most important thing. We design a method "click down" and "click up".It just like tap or click in 3D space but not real touch. When the fingers "click down" means activate the fingers and get into "touch" state, "click up" means release the fingers and back to "relax" state. We can describe the "touch" state as activating some fingers by "click down" condition. And release by "click up" condition. Our approach is combine static data classification and dynamic movement trajectory detect. Our system can gestures in 3D space using a depth camera Leap Motion. And also we give a sample application just like a photo viewer. It can detect 5 kinds of different gestures and give feedback. And if we need to detect more gestures, we can change the model file at any time.

Our system is designed by the following steps:

1. For every finger, recognizing the "click down" frame by frame. If any finger is detected as "click down" then set these fingers into "touch" state.

2. Count how many fingers in "touch" state and classify the finger shape. This is the initialization of gestures.

3. Detect the motion information of fingers to judge which gesture users perform and give feedback.

4. For every finger in “touch” state, if we recognize any finger with “click up” action or any other fingers "click down", then finish the current gesture or move to another multi-touch gesture.

# Chapter 3

# System Design and Implementation

In this section, we will introduce about the implementation process about my research. And also include some tools introduce and some experiments and evaluation.

## 3.1 Development tools

We mainly used several tools to implement our research. For hardware device we used a Leap Motion[19][20] sensor and a Windows PC, for software we used C++ development environment and Qt. And we also used libSVM[21] package in the machine learning part.

### 3.1.1 Hardware Overview

For our hardware devices we used a DELL notebook computer (Intel(R) Core(TM) i7-3612QM CPU @ 2.10 GHz, RAM 8GB, 64bit Windows 10 operation system), and a Leap Motion sensor[19] [20].

We use leap motion as our depth device to get the information of gestures. The leap motion system recognizes and tracks hands and fingers. The device operates in an intimate proximity with high precision and tracking frame rate and reports discrete positions and motion[20].

The Leap Motion controller uses optical sensors and infrared light[6]. And the Leap Motion system employs a right-handed Cartesian coordinate system just like the figure shows. The origin is the center of the top of the Leap Motion device. The x-axes and z-axes is a horizontal plane parallel to the Leap Motion device, and the x-axis is parallel to the long edge of the Leap Motion, the x-axis is parallel to the short edge of the Leap Motion. The y-axis is vertical with the plane just like the figure shows[22].

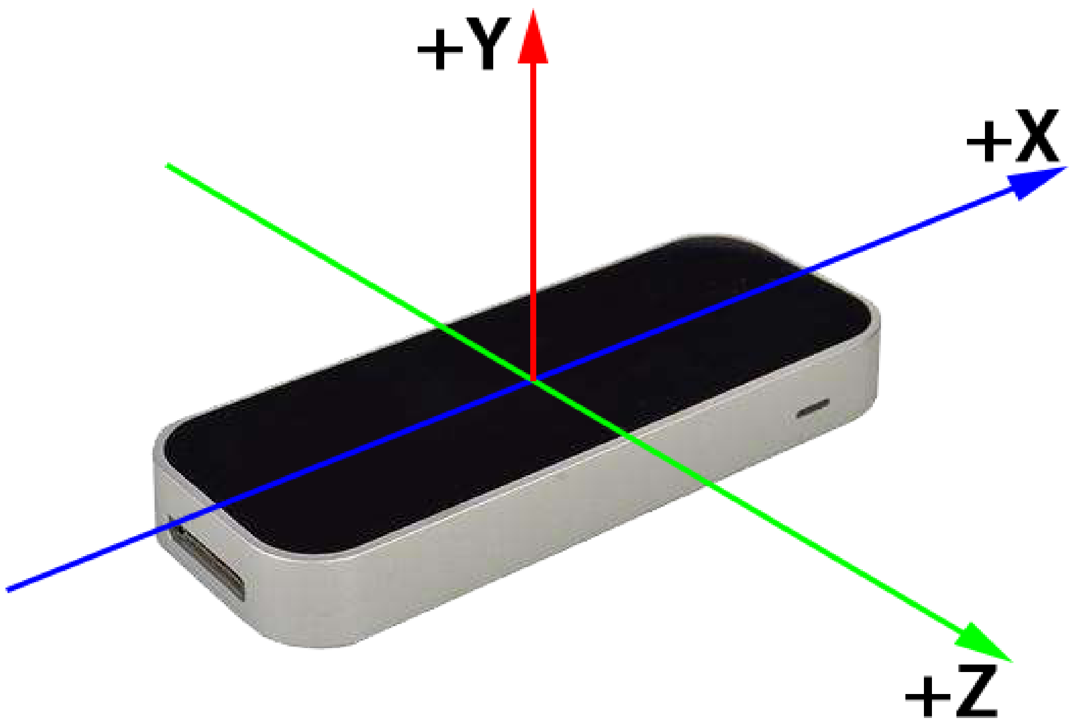



**Figure 3.1** The Leap Motion right-handed coordinate system.

### Leap Motion API Overview[20]

As the Leap Motion controller tracks hands and fingers in its field of view, it provides updates as a set frames of data. Each Frame object representing a frame contains any tracked hands, detailing their properties at a single moment in time. The Frame object is essentially the root of the Leap Motion data model.[20]

The hand model provides information about the identity, position, some vectors and other characteristics of a detected hand, and lists of the fingers associated with the hand. Hands are represented by the Hand class. Just like the following figure shows, we can get the Hand palmNormal() and direction() vectors which define the orientation of a hand.

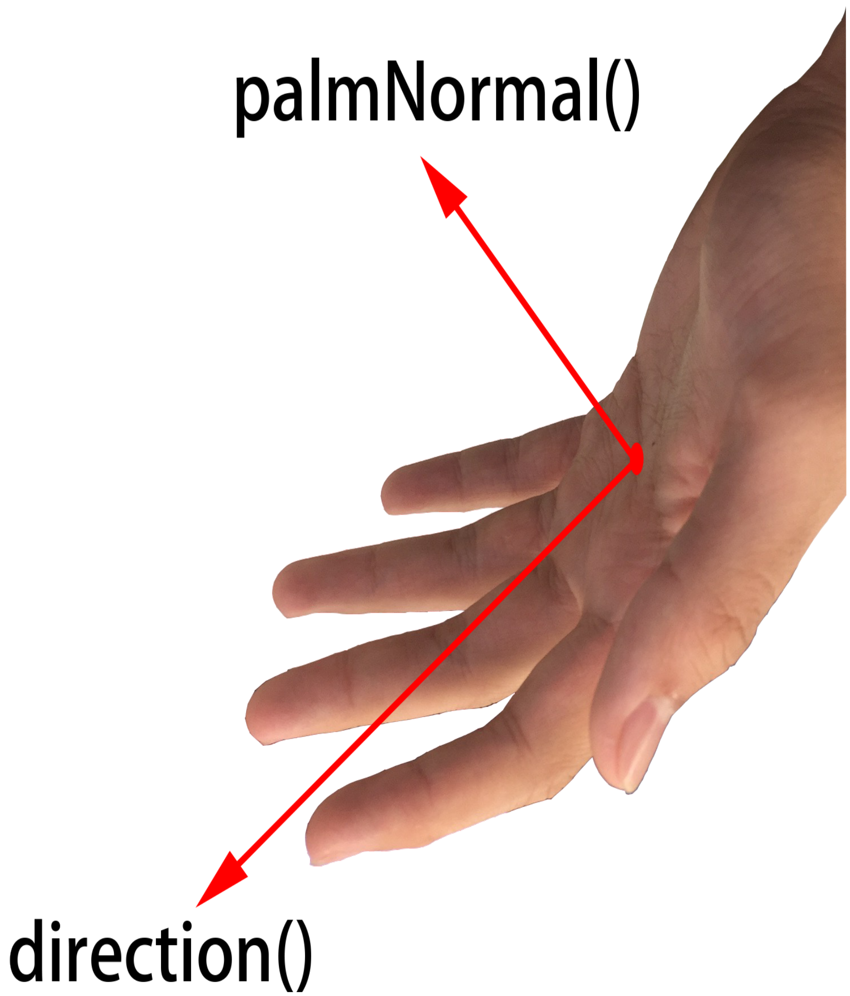



**Figure 3.2** The Hand palmNormal() and direction() vectors define the orientation of the hand.

The Leap Motion controller provides information about each finger on a hand. If all or part of a finger is not visible, the finger characteristics are estimated based on recent observations and the anatomical model of the hand. Fingers are identified by type name, i.e. thumb, index, middle, ring, and pinky[20]. Fingers are represented by the Finger class. Just like the figure shows, tipPosition() and direction() are saved as vector type, and we can get these of every finger directly from the Leap Motion API.

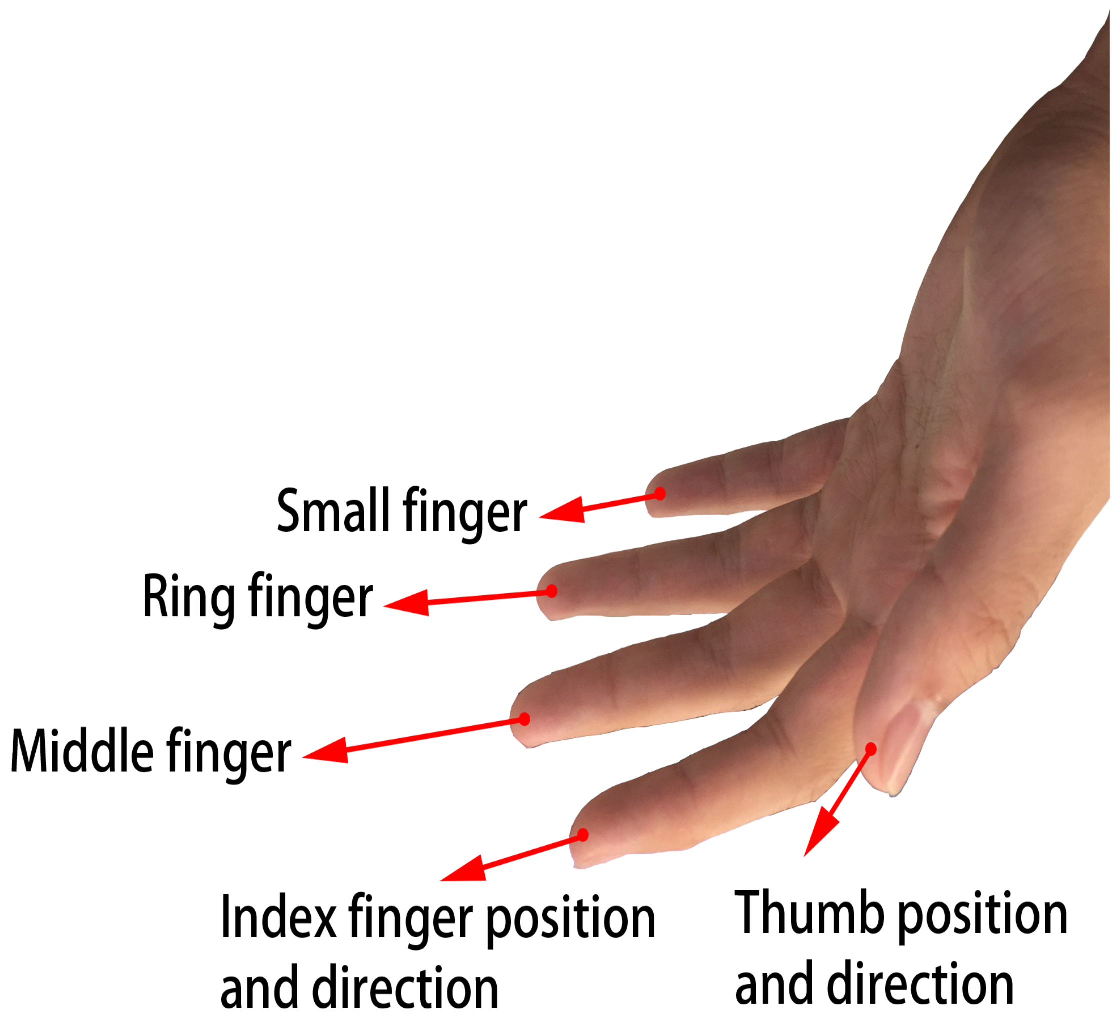



**Figure 3.3** Finger tipPosition() and direction() vectors provide the position of a finger tip and the general direction in which a finger is pointing.

A Finger object also provides a Bone object describing the position and orientation of each anatomical finger bone. All fingers contain four bones ordered from base to tip. We will use these kinds of informations in the classification part of this system. The Leap Motion API provides a model of fingers' bone object. And this model for the thumb does not quite match the standard anatomical naming system because a real thumb has one less bone than the other fingers[20].

### 3.1.2 Software Overview

We use Qt[23] for windows as our development software tool. Qt is a cross-platform application framework that is used for developing application software that can be run on various software and hardware platforms with little or no change in the underlying codebase, while still being a native application with native capabilities and speed.

Qt is a very good software to make GUI.[14]

LIBSVM[21] is a popular open source machine learning libraries, developed at the National Taiwan University and written in C++ though with a C API. LIBSVM implements the SMO algorithm for kernelized support vector machines (SVMs), supporting classification and regression.

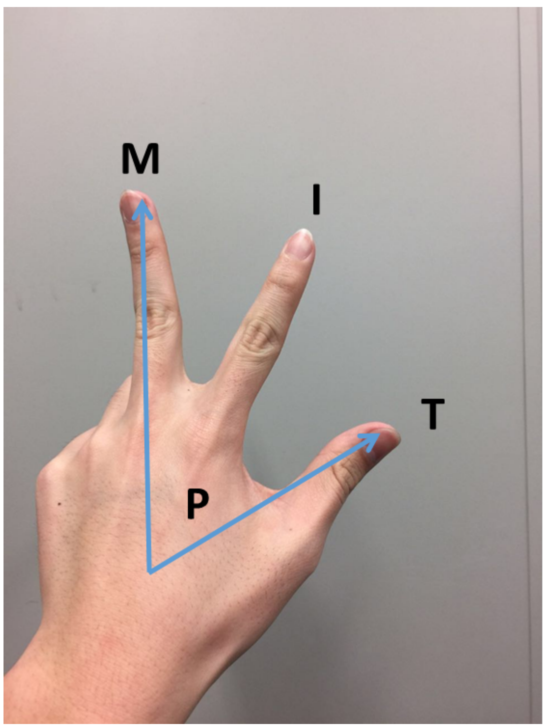
## 3.2 "Touch" State Implementation

To realize Multi-touch gestures in 3D space, the first question we need to solve is how to get in "touch" state. That is how to realize "click down" and "click up". In this section, we realize and compare some previous work.

### 3.2.1 Three Fingers Click

To realize Multi-touch gestures in 3D space, the first question we need to solve is how to get in "touch" state. That is how to realize "click down" and "click up". In this section, we realize and compare some previous work.

In this approach[24], we assume that a finger clicking event occurs when the index finger passes beyond a given threshold. First we can get the fingertip position of thumb, index finger, and middle finger and we noted these three points as T, I, and M. and we can also get the palm center of the hand we noted as P. As we know three fingers can make a plane, so we describe a plane [M, P, T] and we can also get the normal vector of the plane NP. And then we can get the angle between the vector NP and the vector PI. And the Residual angle of this angle is the angle between the plane [M, P, T] and the vector PI, we noted as $\theta$.



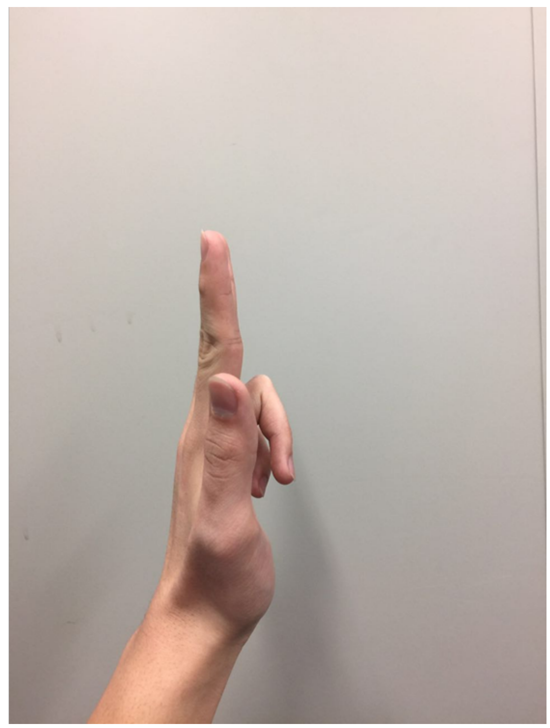


**Figure 3.4** Three fingers click sample a

To define the threshold, we did an experiment to get the angle $\theta$ and select a threshold of 12 degrees empirically.

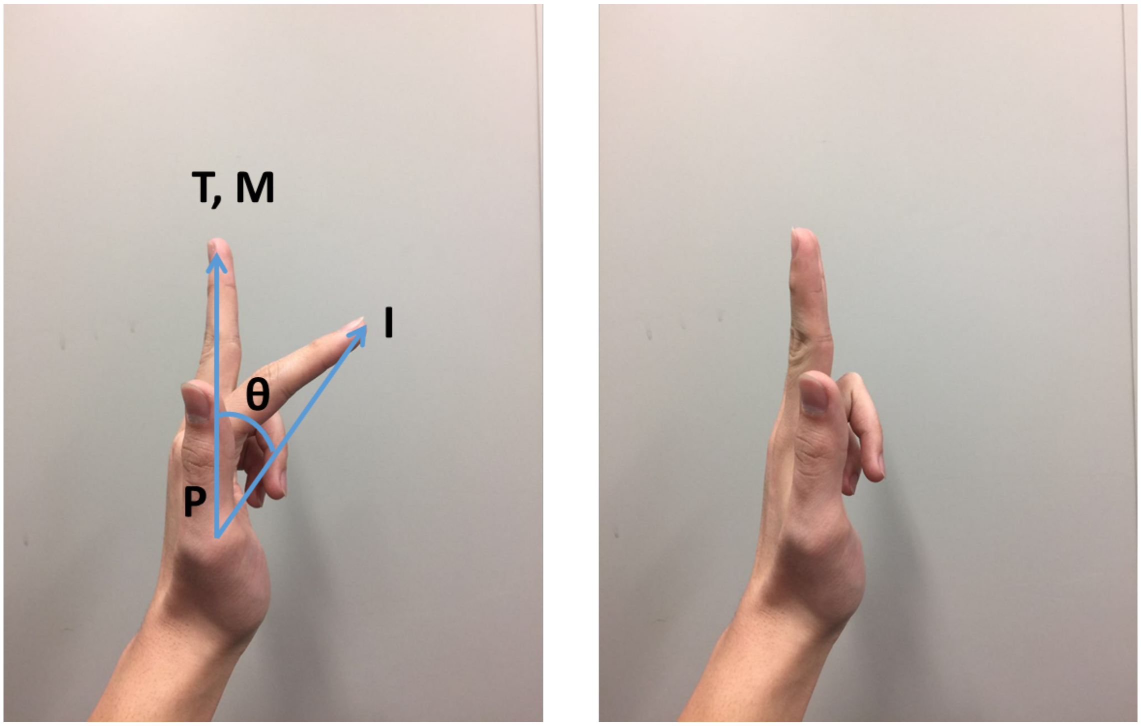



**Figure 3.5** Three fingers click sample b

We also test different directions to three fingers click, and in all of these case a, b, c, and d it can also works very well. But in the case of b and d, the recognition rate will be a little lower because of the perspective of device. Maybe in this case the camera has a blind spot of different fingers.

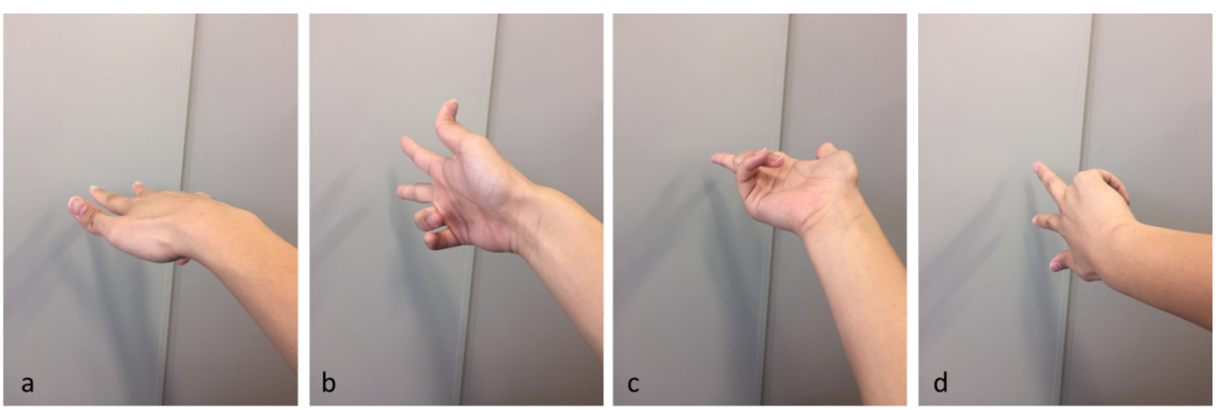



**Figure 3.6** Different direction of three fingers click

### 3.2.2 Depth Click

In this approach[24], we concentrate not only the index finger but all of five fingers. First we can get the fingertip position of thumb, index finger, middle finger, ring finger, and small finger, and we noted these five points as T, I, M, R, and S. and we can also get the palm center of the hand P and the normal vector of palm NP. So we can get the angle between the vector NP and any finger vector. And the Residual angle of this angle is the angle between palm and the finger vector, we noted as $\theta$[i].

**Table 3.1** The biggest and smallest $\theta$[i].

| | Thumb | Index finger | Middle finger | Ring finger | Small finger |
|---|---|---|---|---|---|
| Smallest angle | 18.3 | 21.1 | 21.7 | 17.5 | 16.7 |
| Biggest angle | 50.2 | 73.6 | 69.3 | 58.9 | 51.0 |

Because different finger has different smallest $\theta$, we choose an angle that smaller 5 degrees than $\theta$ as the threshold. If a finger crosses this threshold, we suppose that this finger is now in a click position.

**Table 3.2** The threshold of every finger.

| | Thumb | Index finger | Middle finger | Ring finger | Small finger |
|---|---|---|---|---|---|
| Threshold | 13.3 | 16.1 | 16.7 | 12.5 | 11.7 |

The Three Fingers Clicking gesture was limited, both in terms of ergonomics and functionality, the latter being that it only offered the users the possibility of performing a simple click; any other selection/gesture was impossible. But in depth click, we present "click down" and "click up" for every finger, that means every finger can make a depth click.

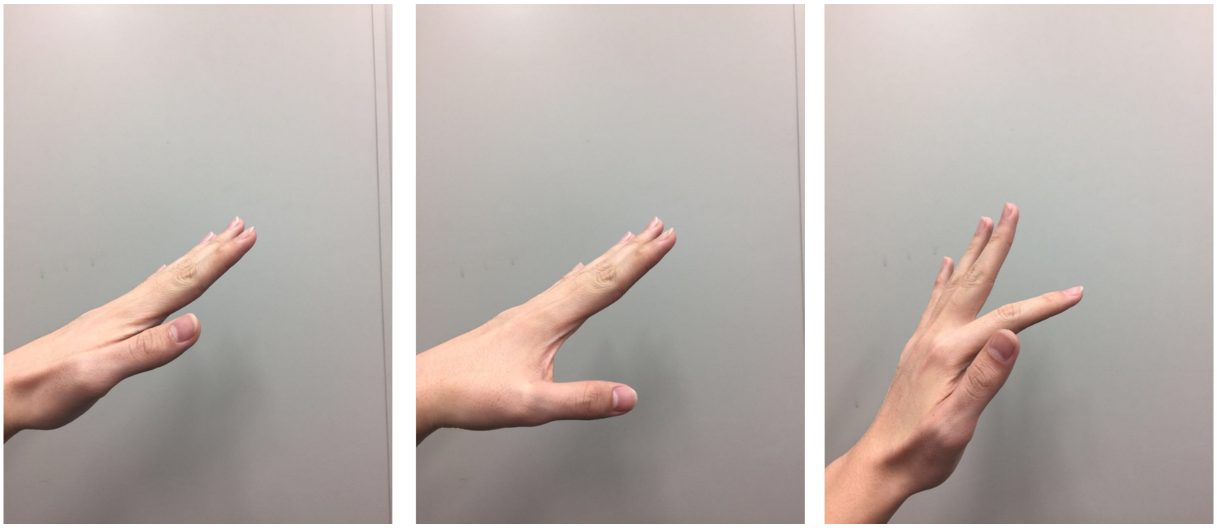


**Figure 3.7** Depth click with thumb and index finger

### 3.2.3 Xpli Click

Even though we were able to detect the click of any finger in hand in Depth Click, the clicking was also limited to a generic click detection, without being able to take advantage of the fingers position for further possible operations. To be able to explicitly detect clicks, we present an amelioration over Depth Click, “Xpli Click”.

Xpli click[24] mainly has two differences with depth click. First is that our hand can keep a relaxed neutral position. second is that we compute Y-Travel to determine the click finger.

- **Relaxed Neutral Position**

Neutral position is the posture that users have to keep their hands when they do not wish to generate any click. In both depth click and three fingers click, users have to keep their fingers fully extended to stay in the neutral position. Just like the following figure a shows is the fully extended neutral position. And in the case of depth click and three fingers click, the same threshold was used for every finger without taking into consideration of the different fingers. But the truth is that for each finger, the threshold can be different. Not only did holding their fingers in this posture caused some discomfort for the users, it also limited the possible interactions. So we present xpli click, in Xpli Click, we introduce a new neutral position where the fingers are open but instead of being fully extended, they are held in the most relaxed position for the users. Just like the following figure b. We took into account the fact that every finger had a different threshold. To be able to accomplish this new neutral position, we asked the participants to keep their fingers still, and in a posture that felt relaxed and easy to maintain as a neutral position. Finally we calculate the average of the angle of every finger, and then we determine the threshold for every finger.

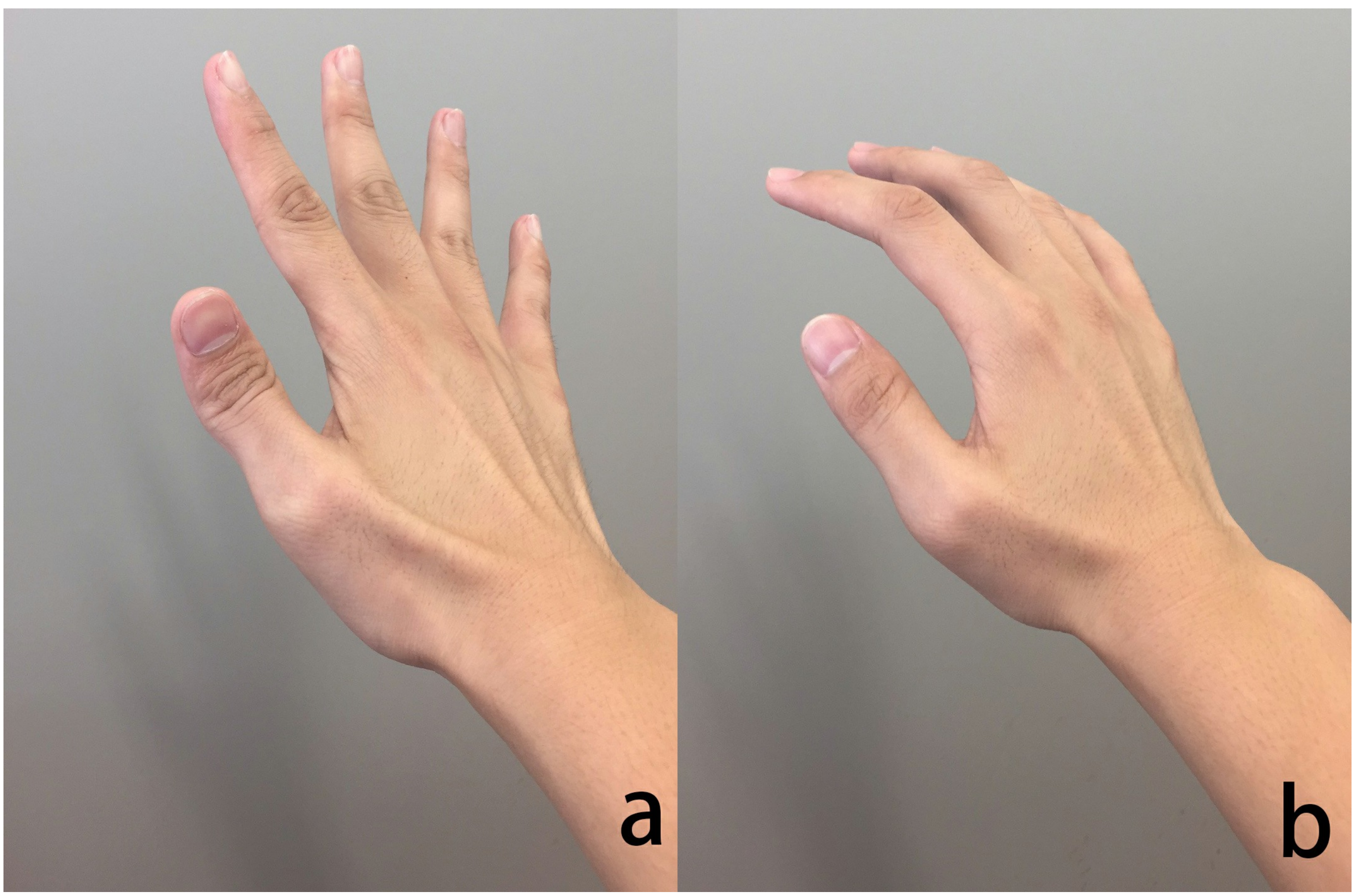



**Figure 3.8** figure a in neutral postures of Depth Click, and figure b is in neutral postures of Xpli Click.

First we keep the hand as relaxed neutral position for about 200 frame. And then compute the angle between finger vector and palm normal vector. And we use 5 degree more than the residual of neutral angle as the threshold for every finger.

**Table 3.3** The threshold of every finger in xpli click.

| | Thumb | Index finger | Middle finger | Ring finger | Small finger |
|---|---|---|---|---|---|
| Neutral angle | 82.3 | 84.7 | 85.2 | 84.6 | 83.2 |
| Residual angle | 7.7 | 5.3 | 4.8 | 5.4 | 6.8 |
| Threshold | 12.7 | 10.3 | 9.8 | 10.4 | 11.8 |

- **Y-travel**

Y-Travel describes the size of finger movement. And I also use angle to describe this. Every time we find the biggest Y-Travel and if current finger has the biggest Y-Travel, then current finger is the click finger.

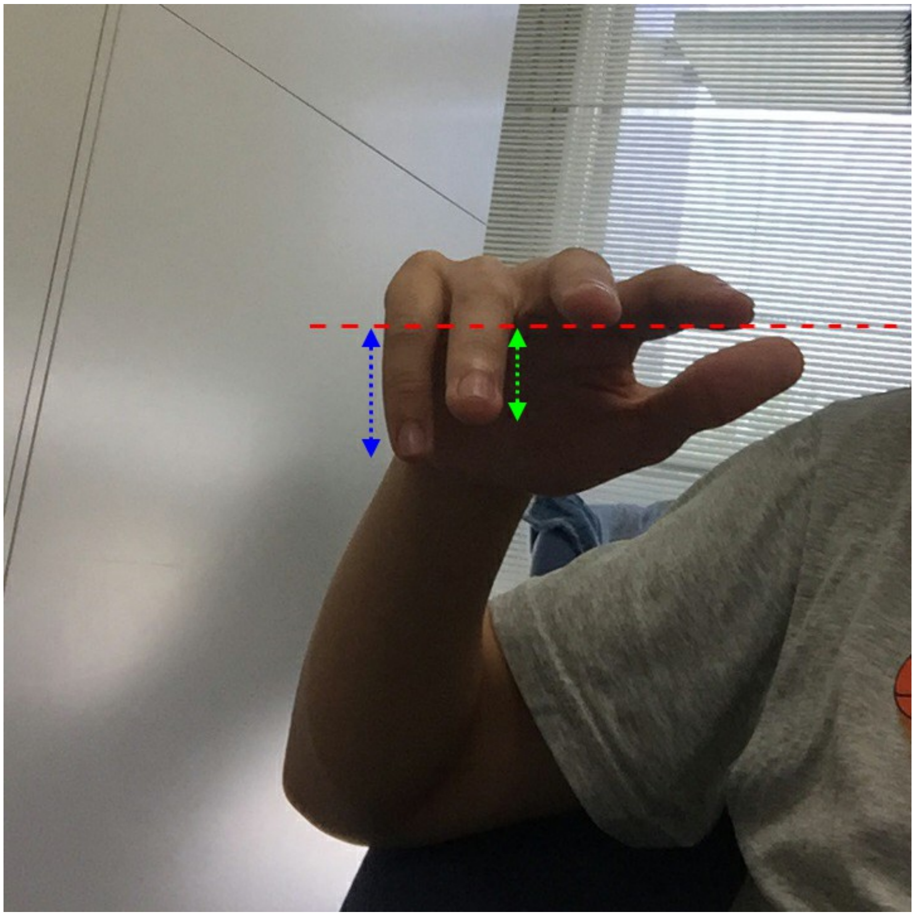


**Figure 3.9** Both the ring and small fingers have crossed their thresholds, however, the Y-Travel of the small finger is bigger than that of the ring finger, so the small finger is detected as the clicking finger.

### 3.2.4 Evaluation of Click

We test the recognition rate of this click in 3 different speed of three fingers click.

**Table 3.4** The recognition rate of three fingers click.

| Slow (10 clicks per min) | Normal (60 clicks per min) | Fast (120 clicks per min) |
|---|---|---|
| 95% | 95% | 90% |

We test the recognition rate of this click in 3 different speed of depth click.

**Table 3.5** The recognition rate of depth click.

| | Slow (10 clicks per min) | Normal (60 clicks per min) | Fast (120 clicks per min) |
|---|---|---|---|
| Thumb | 90% | 90% | 85% |
| Index | 95% | 100% | 90% |
| Middle | 95% | 95% | 90% |
| Ring | 95% | 90% | 90% |
| Small | 90% | 90% | 85% |

We test the recognition rate of this click in 3 different speed of xpli click.

**Table 3.6** The recognition rate of xpli click.

| | Slow (10 clicks per min) | Normal (60 clicks per min) | Fast (120 clicks per min) |
|---|---|---|---|
| Thumb | 90% | 91% | 90% |
| Index | 95% | 94% | 91% |
| Middle | 94% | 94% | 92% |
| Ring | 92% | 90% | 90% |
| Small | 93% | 91% | 91% |

The result shows that xpli click maybe the best one, so finally we choose xpli click as our click method.

Then we tested the recognize rate of "click down" and "click up" separately.

**Table 3.7** The recognition rate of click down.

| | Slow (10 clicks per min) | Normal (60 clicks per min) | Fast (120 clicks per min) |
|---|---|---|---|
| Thumb | 93% | 92% | 92% |
| Index | 96% | 94% | 95% |
| Middle | 95% | 94% | 93% |
| Ring | 94% | 92% | 92% |
| Small | 95% | 93% | 93% |

**Table 3.8** The recognition rate of click up.

| | Slow (10 clicks per min) | Normal (60 clicks per min) | Fast (120 clicks per min) |
|---|---|---|---|
| Thumb | 90% | 91% | 90% |
| Index | 95% | 94% | 94% |
| Middle | 94% | 94% | 92% |
| Ring | 93% | 91% | 91% |
| Small | 93% | 92% | 91% |

Here are some screenshots of our click system.

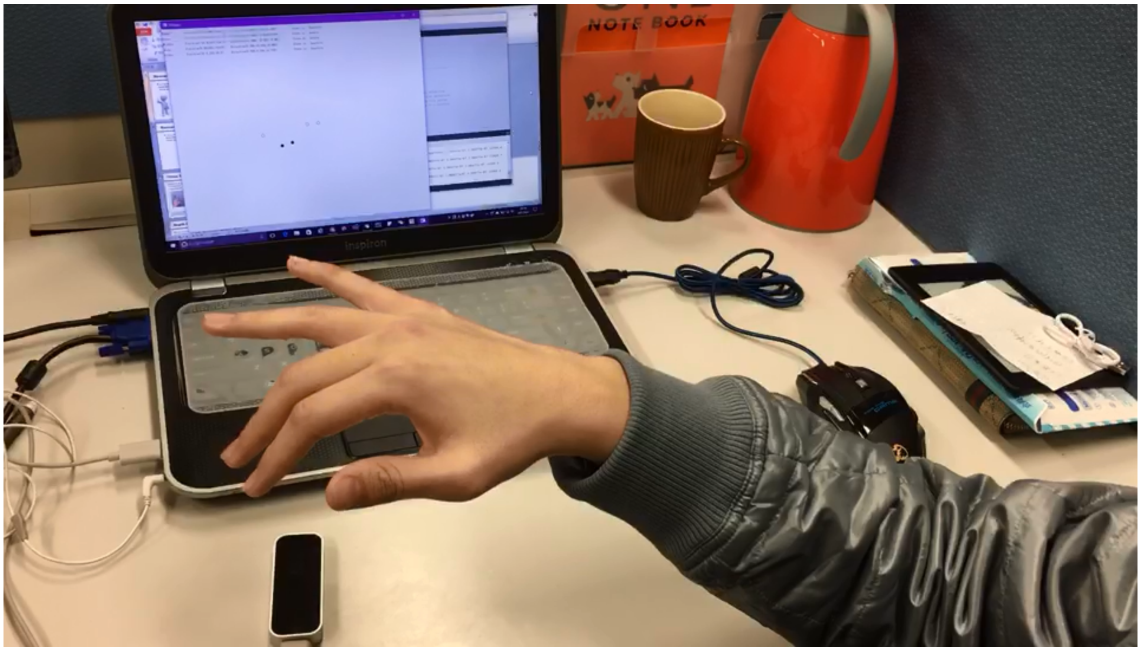



**Figure 3.10** Demonstration effect of click down.

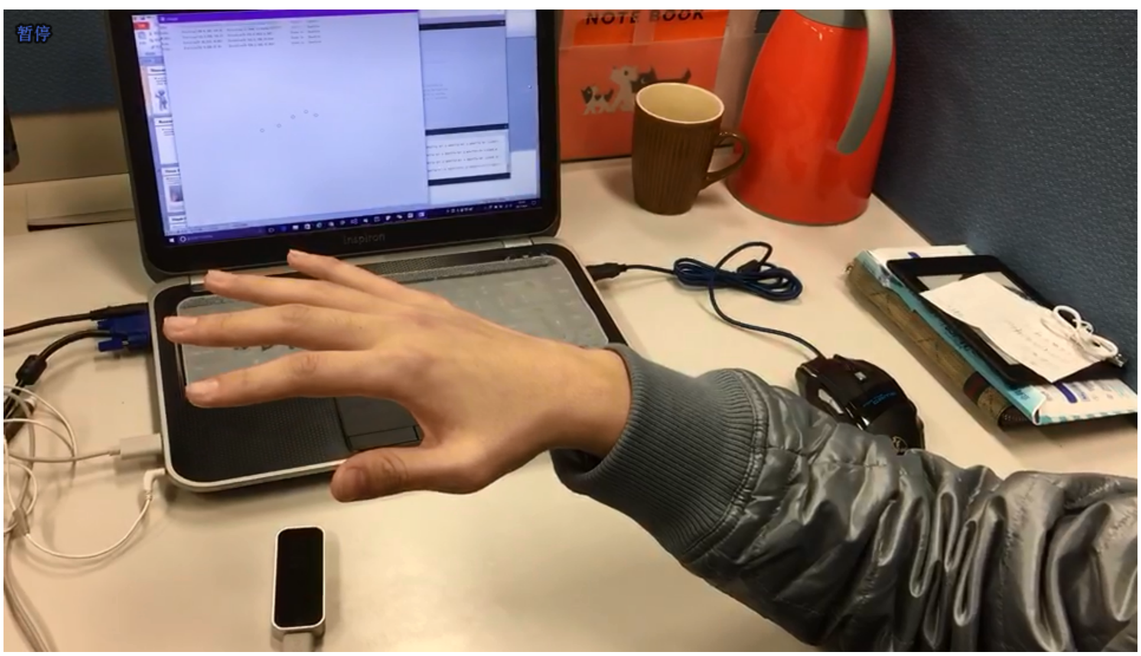



**Figure 3.11** Demonstration effect of click up.

## 3.3 Hand Shape Classification

The 3D hand model can be divided into body model, grid model, geometric model and skeleton model. In contrast, the three-dimensional skeleton model is easy to implement and can reflect the three-dimensional characteristics, can be reconstructed through a certain method to meet the real requirements of the training with virtual hands. So we use three-dimensional skeleton model to describe a hand shape.

### 3.3.1 Data Collection

We choose 26 key points to describe a hand shape.

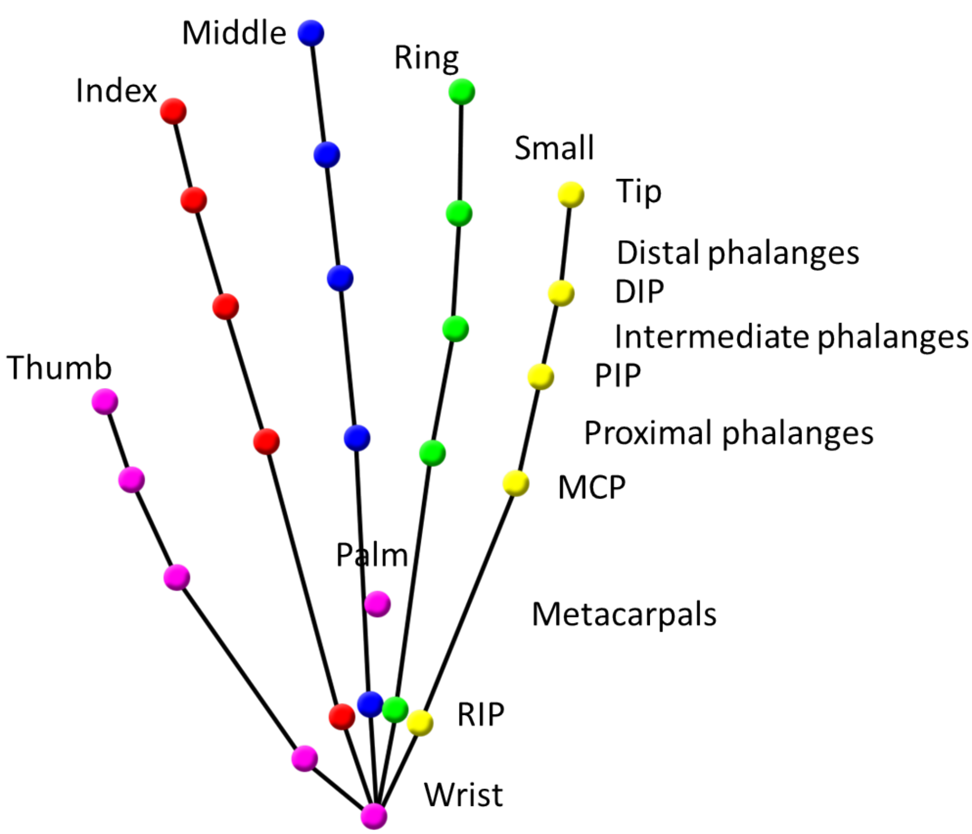



**Figure 3.12** 26 key points of a hand.

- Metacarpal – the bone inside the hand connecting the finger to the wrist (except the thumb);
- Proximal Phalanx – the bone at the base of the finger, connected to the palm.
- Intermediate Phalanx – the middle bone of the finger, between the tip and the base.
- Distal Phalanx – the terminal bone at the end of the finger.
- DIP, Distal Interphalangeal.
- PIP, Proximal Interphalangeal.
- MCP, Metacarpophalangeal.
- RIP, Root Interphalangeal.

This model for the thumb does not quite match the standard anatomical naming system. A real thumb has one less bone than the other fingers. So thumb does not have RIP and Metacarpal.

But not all of these points we can get directly from leap motion. So we need make a model to calculate these points. For the index finger, middle finger, ring finger, small finger, only consider the case of independent distortion, the state of the fingers are mainly straight state and bending 2 states. Regardless of the state of the finger, in the case of neglecting the thickness of the fingers, we can think of its TIP, DIP, PIP and MCP in the same plane; for the thumb, you can think that the fingers of the palm joint points coincide with the wrist, TIP, DIP and PIP is in the same plane. According to the position relation, the position of the key point in the local plane coordinate system can be constructed. The constructed local coordinate system of the finger can be expressed in the $L_p$ coordinate system, and it can be expressed by the coordinate transformation. And obtains the coordinates of the key points in the $L_p$ coordinate system to obtain the three-dimensional position of the key points.The $L_p$ coordinate system is the Leap Motion-based coordinate system in the real space, the three-dimensional coordinates of the key points in the coordinate system are completed Attitude estimation.

To sum up, according to the number of key points and modeling methods, the key points of the model is divided into two categories: index finger, middle finger, ring finger and little finger as a class, there are four key points of these fingers, constitute a four vertices of the quadrilateral are modeled using a planar quadrilateral model; the thumb is a class, and the thumb has three key points, forming three vertices of a triangle, using a planar triangular model.

For the plane quadrilateral model. Index finger, middle finger, ring finger, small finger four fingers similar structure, similar to the movement state, and have a TIP, DIP, PIP and MCP of the four key points, modeling methods and solution quadrilateral similar. To the finger of the palm joint point as the origin, that the palm joint point to the fingertip point of the direction of the x-axis, perpendicular to the x-axis and the far point of the joint, near the joint y value of non-negative direction of the y-axis , To construct the local plane coordinate system of the finger, as shown in the figure. Among them, the dots represent the positions of four key points, and $\theta_1$ and $\theta_2$ indicate the angle between the proximal and distal joints.

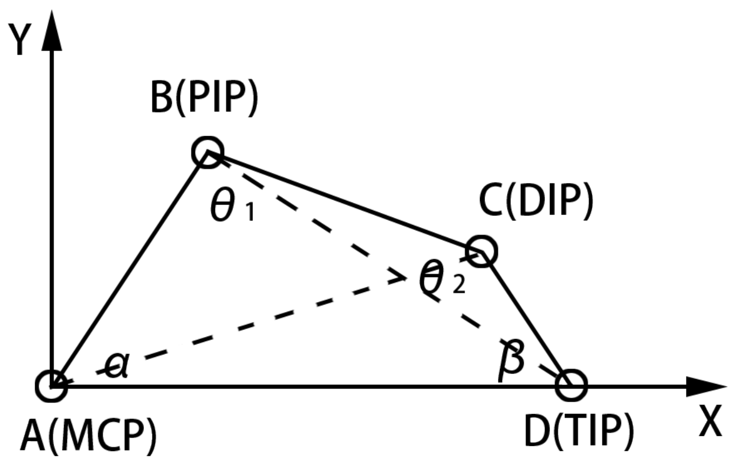



**Figure 3.13** Parallel deformation model.

Thus, the construction of a finger of the local plane coordinate system quadrilateral model is as follows:

$$\begin{cases} l_{AD} = d \\ l_{AB} + l_{BC} + l_{CD} = l \\ l_{AB} : l_{BC} : l_{CD} = a : b : c \end{cases} \tag{3.1}$$

d corresponds to the length of the finger's fingers in the actual space (ie, the distance from the palm joint point to the fingertip), l corresponds to the total length of the finger. The angle constraint is

$$\begin{cases} \alpha + \beta + \theta_1 + \theta_2 = 2\pi \\ \theta_2 = \gamma \cdot \theta_1, \gamma \geq 1 \\ \frac{\pi}{2} \leq \theta_1 \leq \pi, \frac{\pi}{2} \leq \theta_2 \leq \pi \end{cases} \tag{3.2}$$

In order to meet the physiological results of human hand, according to the general proportion of the length of the finger segment and the bending angle of the joint, through the comparison experiment with the real hand, this paper defines the relationship between the length of the segment and the angle ratio as the empirical estimate.

$$l_{AB} : l_{BC} : l_{CD} = 1.5 : 1.2 : 1 \theta_2 = 1.2\theta_1 \tag{3.3}$$

Solve the coordinates of 3 points B, C, D. For AC and BD 2 auxiliary lines, according to the cosine theorem to solve the model.

$$f(x, y, \phi) = x^2 + y^2 - 2xy\cos\phi \tag{3.4}$$

So,

$$\begin{cases} f(l_{AB}, l_{AD}, \alpha) = f(l_{BC}, l_{CD}, \theta_2) \\ f(l_{AD}, l_{CD}, \beta) = f(l_{AB}, l_{BC}, \theta_1) \end{cases} \tag{3.5}$$

B point is DIP, C point is PIP and D point is TIP, so,

$$\begin{cases} B(l_{AB}\cos\alpha, l_{AB}\sin\alpha) \\ C(l_{AD} - l_{CD}\cos\beta, l_{CD}\sin\beta) \\ D(l_{AD}, 0) \end{cases} \tag{3.6}$$

Thumb finger palm joint can be considered coincident with the wrist point, the thumb has a TIP, DIP and PIP 3 key points, modeling methods and solution triangle similar.

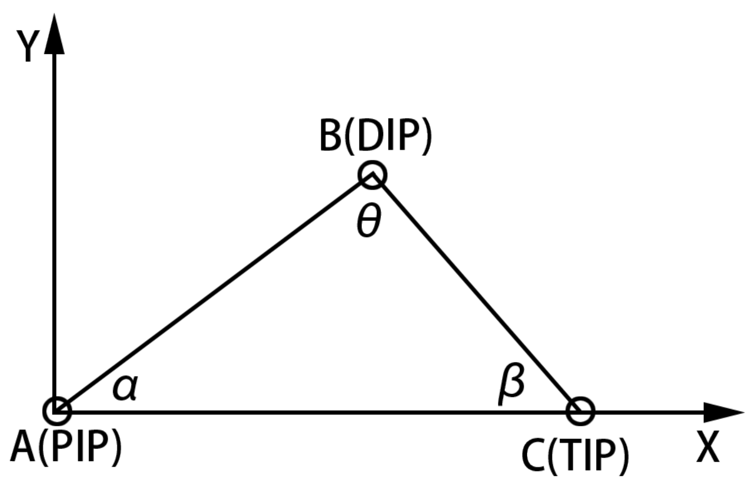



**Figure 3.14** Triangle model.

With the PIP of the thumb as the origin, the direction of the DIP to the TIP is the x-axis, the y-axis is the direction perpendicular to the x-axis and the y-axis of the far-off joint y is non-negative, As shown in the figure, where the dots represent the positions of the four key points and $\alpha$ is the angle at which the DIP are bent.

Thumb Local Plane Coordinate System Triangle model is as follows:

$$\begin{cases} l_{AC} = d \\ l_{AB} + l_{BC} = l \\ l_{AB} : l_{BC} = a : b \end{cases} \tag{3.7}$$

Solve the coordinates of B and C. Among them, the finger length proportional relationship is defined as the empirical value $l_A B : lBC = 1.2 : 1$.

According to the cosine theorem

$$f(l_{AB}, l_{AC}, \alpha) = l_{BC}^2 \tag{3.8}$$

So B is DIP and C is PIP, and so,

$$\begin{cases} B(l_{AB}\cos\alpha, l_{AB}\sin\alpha) \\ C(l_{AC}, 0) \end{cases} \tag{3.9}$$

The key model of this paper has the characteristics of self-adaptation for different real hands.It can be seen from the model description that the model calculates the adaptability of the model with the length of known and length adaptive.The model calculation uses the length of the finger The length of the extension d and the total length of the finger l, the extension length d of the finger is obtained by Leap Motion, the total length of the finger is the attribute of the finger, the total length of the different fingers is different, before the experiment needs to be calibrated by the length of the finger Method of collection, thus ensuring the model of different hands and different fingers of the adaptability.

The coordinates of the key points in the local coordinate system of the finger are obtained by solving the plane quadrilateral model and the plane triangular model.Obviously, the coordinates in the local plane coordinate system can not be used as the final attitude estimation results. For each frame data obtained, Change the coordinates of each key from the finger local coordinate system to the $L_p$ coordinate system.

As shown in the figure, the $L_p$ coordinate system is O-xyz, which is the coordinate system that reflects the position of the hand in the real world coordinates, and the finger plane coordinate system is directly expanded into the three-dimensional coordinate system $O' - x'y'z'$ according to the right-hand rule. Using the Leap Motion to obtain the finger length $L_{finger}$ and the fingertip coordinate $(x_T, y_T, z_T)$, the finger direction $u_f$ calculates the origin position of the finger coordinate system $O' - x'y'z'$ in the $L_p$ coordinate system as $(x_0, y_0, z_0) = (x_T, y_T, z_T) - u_f \cdot L_{finger}$. Using the finger direction $u_f$, the palm normal vector $u_p$ to solve the finger coordinate system $O' - x'y'z'$ coordinate axis vector in the $L_p$ coordinate system is expressed as $u'_x = u_f, u'_z = u'_x \times u_p$ and $u'_y = u'_z \times u'_x$.

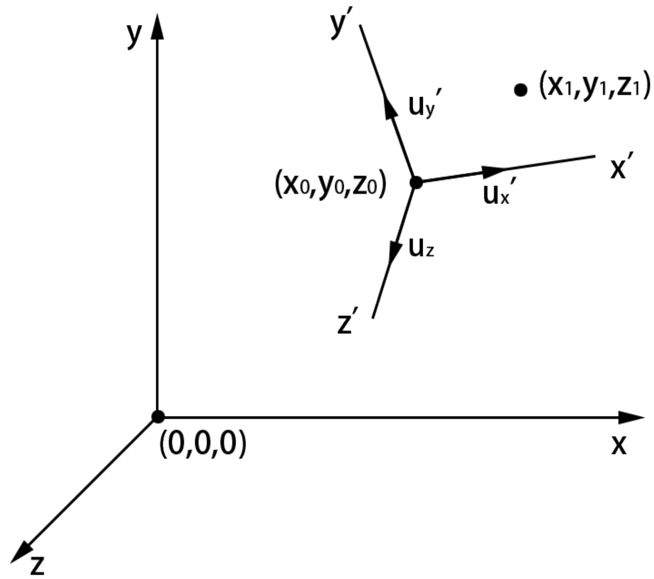



**Figure 3.15** The coordinate transformation of the finger plane coordinate system to the $L_p$ coordinate system.

Based on the obtained coordinate system coordinate $(x_0, y_0, z_0)$, $u'_x, u'_y, u'_z$ and the three-dimensional coordinate transformation formula, we can obtain the coordinate $(x', y', z')$ conversion formula of one point (x1, y1, z1) in the $L_p$ coordinate system O-xyz to the local coordinate system $O' - x'y'z'$ of the finger $(x', y', z', 1) = (x_1, y_1, z_1, 1) \cdot T \cdot R$. Through the matrix inverse transformation process, obtained within the finger local coordinate system $O' - x'y'z'$ $(x', y', z')$ to $L_p$ coordinate system O-xyz coordinates $(x_1, y_1, z_1)$ conversion formula

$$(x_1, y_1, z_1, 1) = (x', y', z', 1) \cdot T^{-1} \cdot R^{-1} \tag{3.10}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & -z_0 & 1 \end{bmatrix} \tag{3.11}$$

$$R = \begin{bmatrix} u'_{x_1} & u'_{y_1} & u'_{z_1} & 0 \\ u'_{x_2} & u'_{y_2} & u'_{z_2} & 0 \\ u'_{x_3} & u'_{y_3} & u'_{z_3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

For each frame of data, since the origin of the finger coordinate system and the coordinate axis direction are unique in the $L_p$ coordinate system, the transformation of the key points into the $L_p$ coordinate system is also unique.

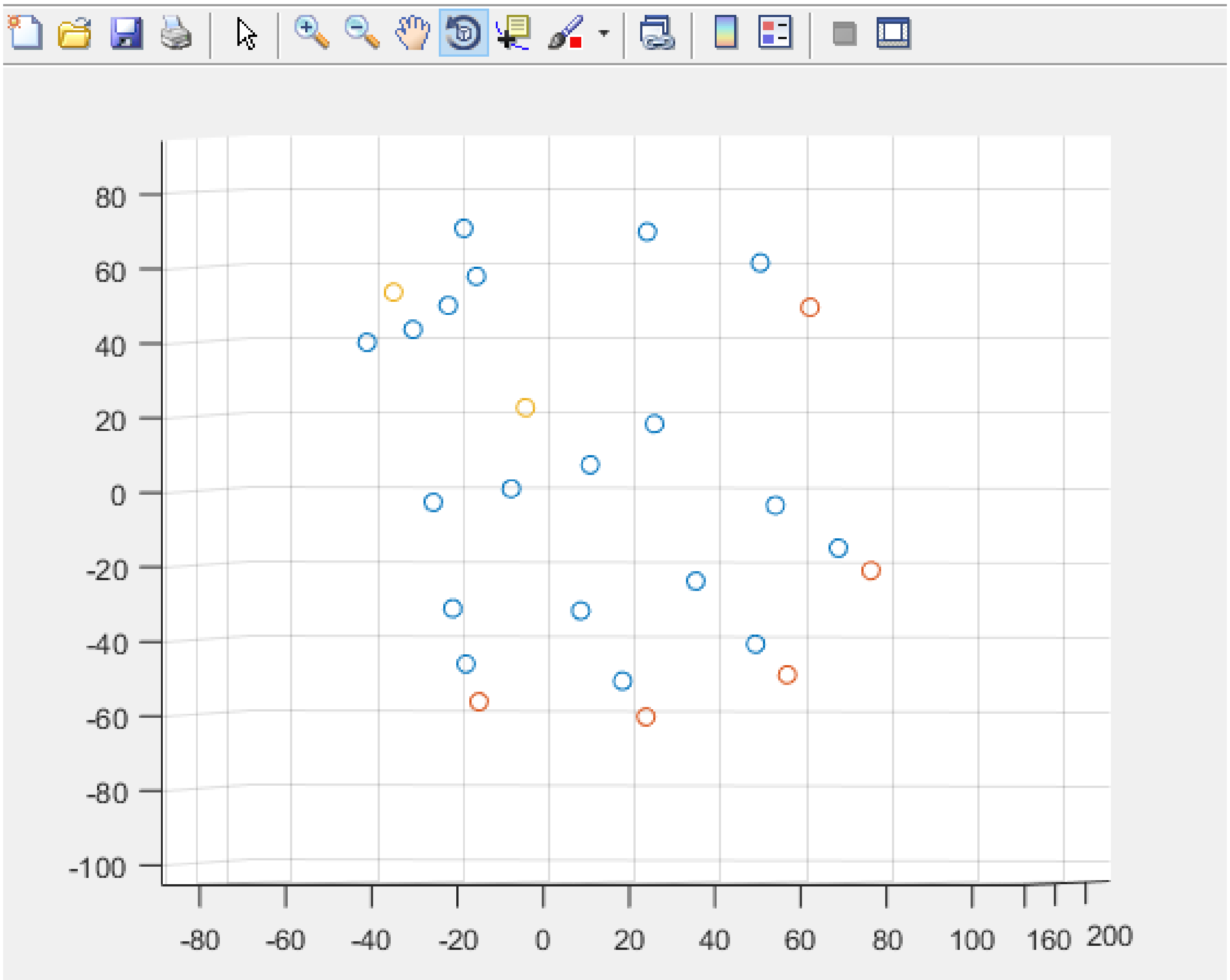



**Figure 3.16** We use matlab to show the 26 points of a hand from different perspective.

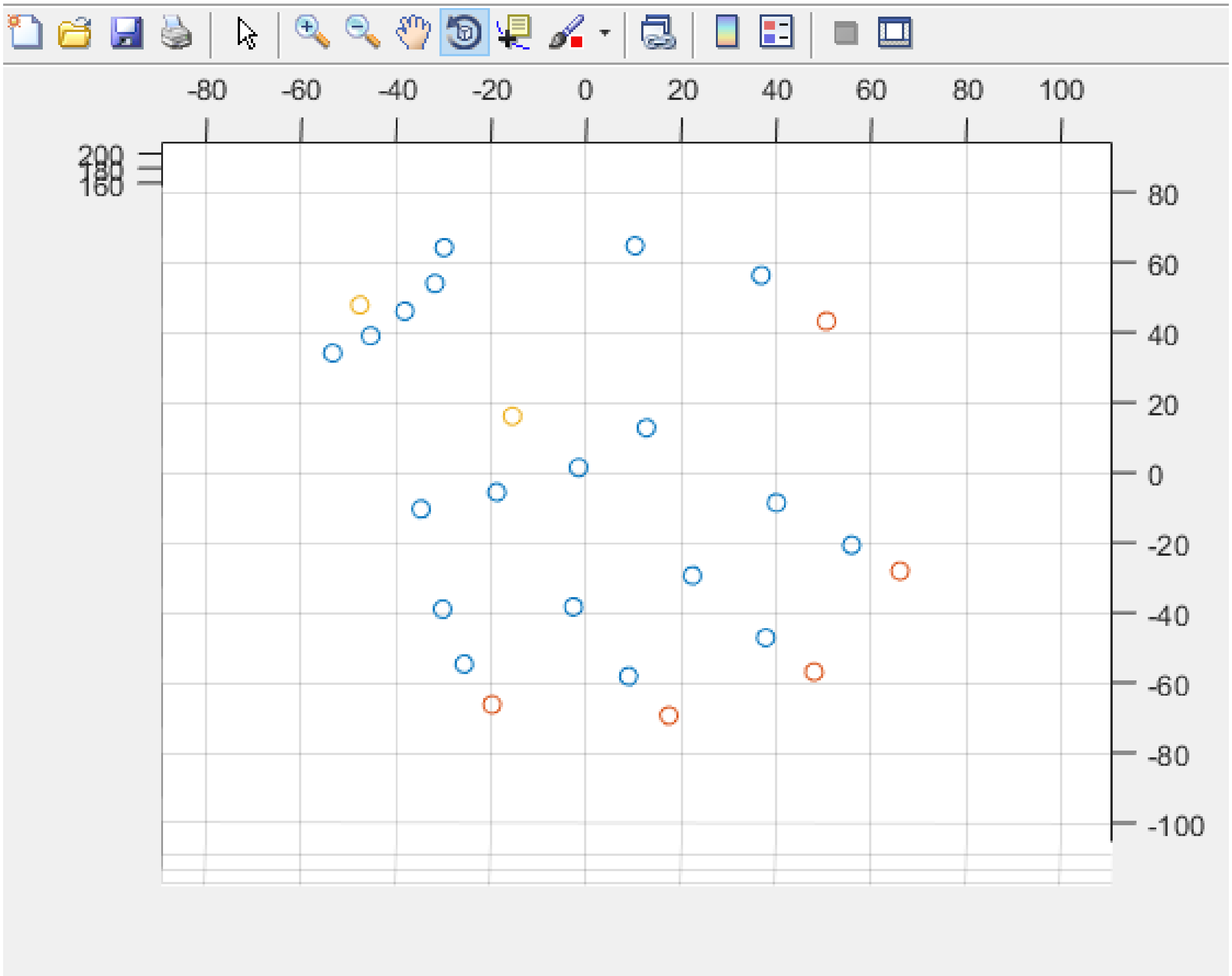



**Figure 3.17** We use matlab to show the 26 points of a hand from different perspective.

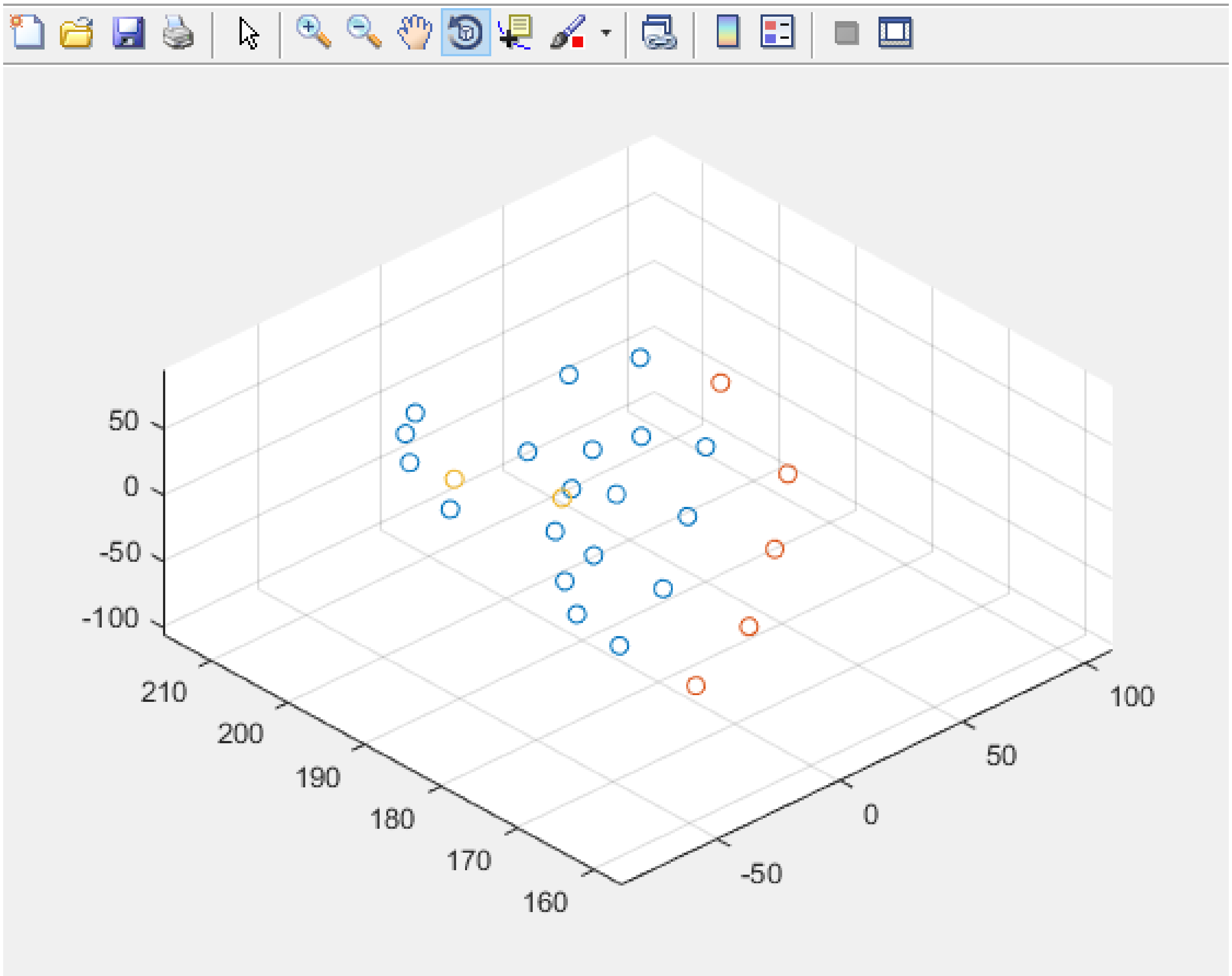



**Figure 3.18** We use matlab to show the 26 points of a hand from different perspective.

**Table 3.9** A sample of key points of hand.

| | RIP | MCP | PIP | DIP | TIP |
|---|---|---|---|---|---|
| Thumb | (-29.683, 187.676, 70.2258) | (10.4183, 195.15, 69.1618) | (37.0169, 194.366, 60.871) | (50.7828, 188.273, 49.065) | |
| Index | (-31.7254, 201.708, 57.0196) | (12.8738, 192.849, 17.6586) | (40.2285, 195.255, -4.23539) | (56.0532, 191.925, -15.588) | (66.2999, 184.474, -21.4666) |
| Middle | (-38.0803, 200.975, 49.2481) | (-1.41412, 191, 6.71806) | (22.5608, 192.846, -24.4958) | (38.0038, 188.491, -41.3158) | (48.2535, 180.377, -49.3829) |
| Ring | (-45.2578, 198.118, 42.831) | (-18.6058, 187.641, 0.381797) | (-2.49091, 187.777, -32.3944) | (9.15772, 182.434, -51.0645) | (17.6732, 173.942, -60.4801) |
| Small | (-53.192, 190.273, 39.5348) | (-34.5376, 181.285, -3.1037) | (-29.9593, 181.157, -31.6433) | (-25.4107, 177.046, -46.4094) | (-19.4905, 168.963, -56.3175) |

### 3.3.2 Normalization and Scale

We find that the finger shape information has no relationship with position. So we do the normalization of finger shape data. First we translate the finger shape points until the palm center is on the origin coordinate. Then rotate the points around the palm center until the palm normal vector is on the -y coordinate axis. And rotate again the points around the y coordinate axis until the palm direct vector is on the -z coordinate axis. The normalization procedure just like the figure shows.

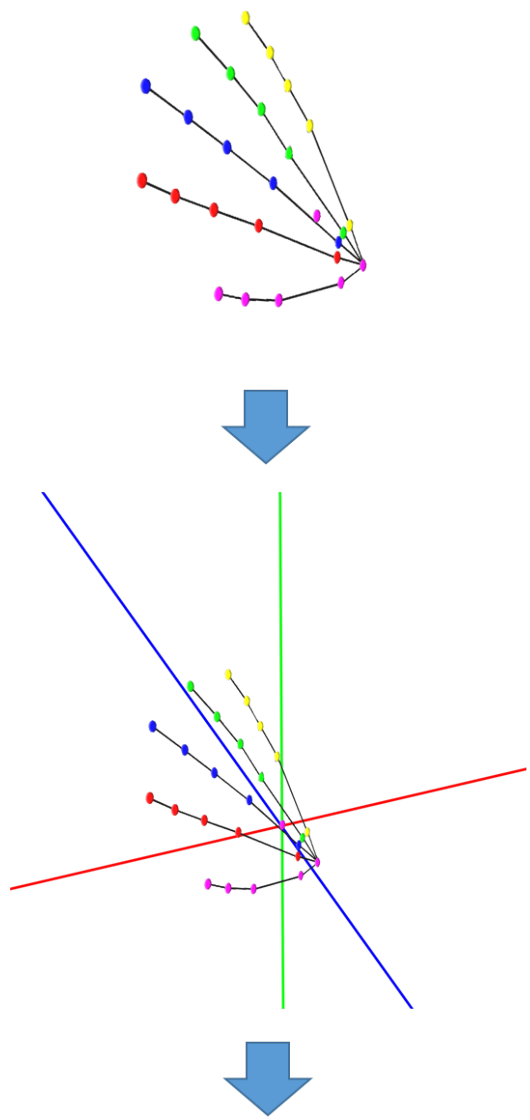


**Figure 3.19** Normalization procedure.

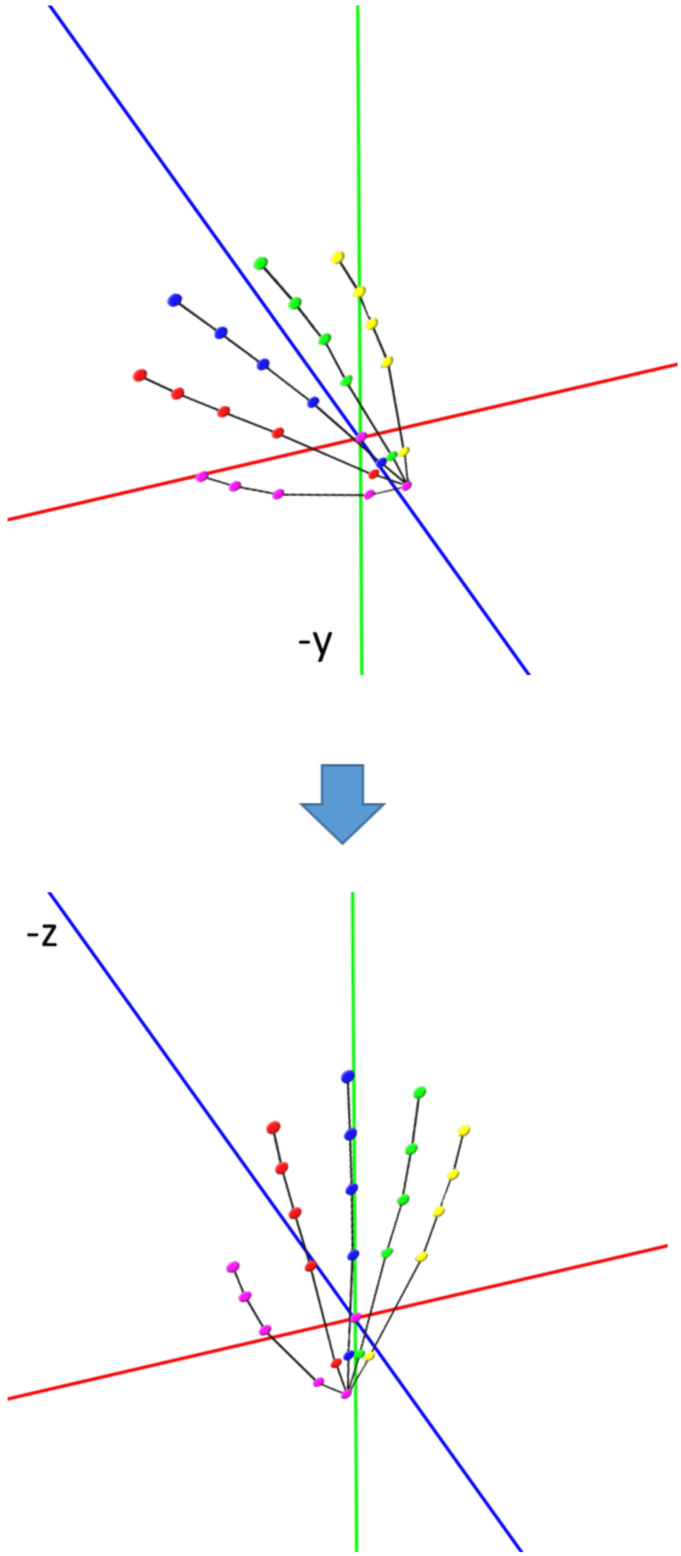



**Figure 3.20** Normalization procedure.

1 1:-4.86865 2:194.184 3:32.4861 4:-32.5424 5:193.665 6:75.1422 7:-8.95536 8:185.188 9:88.7377 10:34.2937 11:190.431 12:72.7743
13:61.3869 14:190.849 15:56.3381 16:78.3081 17:189.618 18:42.7229 19:-13.3084 20:203.566 21:77.39 22:25.659 23:199.6 24:21.3364
25:48.6802 26:184.255 27:-7.45978 28:58.4066 29:167.401 30:-18.6884 31:62.8745 32:152.854 33:-23.2312 34:-22.3151 35:204.522 36:70.3931
37:6.9986 38:200.243 39:12.7059 40:25.5264 41:210.265 42:-26.8354 43:37.6846 44:211.487 45:-50.2728 46:46.2622 47:209.15 48:-65.3082
49:-32.0744 50:202.868 51:64.6994 52:-13.8687 53:198.409 54:9.57528 55:-4.41606 56:214.735 57:-27.4219 58:2.61256 59:222.036 60:-51.0926
61:8.1359 62:224.788 63:-67.326 64:-42.3148 65:195.048 66:61.7001 67:-32.8871 68:192.61 69:8.69009 70:-37.7271 71:205.875 72:-20.9883
73:-39.5024 74:211.283 75:-38.2541 76:-40.182 77:214.13 78:-54.006
1 1:-3.63356 2:192.968 3:32.9769 4:-31.3616 5:193.121 6:75.6005 7:-7.83361 8:184.698 9:89.3314 10:35.5626 11:189.932 12:73.7696
13:62.8235 14:190.3 15:57.6119 16:79.8146 17:188.953 18:44.095 19:-12.0813 20:202.953 21:77.7489 22:26.9342 23:198.093 24:21.7994
25:49.9145 26:182.476 27:-6.88342 28:59.6233 29:165.541 30:-18.0049 31:64.066 32:150.952 33:-22.4347 34:-21.0746 35:203.866 36:70.729
37:8.28781 38:198.72 39:13.1376 40:26.961 41:208.021 42:-26.5113 43:39.1454 44:208.949 45:-49.9486 46:47.7198 47:206.484 48:-64.9653
49:-30.835 50:202.189 51:65.044 52:-12.5845 53:196.95 54:10.0034 55:-2.8141 56:212.692 57:-27.1642 58:4.34536 59:219.758 60:-50.8668
61:9.90319 62:222.492 63:-67.0916 64:-41.1105 65:194.384 66:62.1297 67:-31.6304 68:191.235 69:9.16649 70:-36.2973 71:204.249 72:-20.6503
73:-38.0065 74:209.56 75:-37.9531 76:-38.6665 77:212.384 78:-53.7099
1 1:-13.411 2:191.615 3:33.9719 4:-40.4189 5:194.261 6:76.9743 7:-16.9203 8:185.699 9:90.6689 10:26.4479 11:190.652 12:74.938 13:53.0157
14:190.023 15:57.6719 16:69.0639 17:187.689 18:43.1728 19:-20.8313 20:203.6 21:78.4341 22:17.1037 23:195.394 24:22.1331 25:38.7255
26:179.121 27:-7.23248 28:47.2169 29:161.103 30:-17.6247 31:50.3743 32:145.836 33:-20.6486 34:-29.9088 35:204.5 36:71.5216 37:-1.65687
38:196.227 39:13.7392 40:15.6902 41:204.054 42:-26.8201 43:27.3838 44:201.735 45:-50.4105 46:35.5259 47:195.414 48:-64.5123 49:-39.8054
50:202.887 51:66.0578 52:-22.6201 53:194.949 54:10.9974 55:-14.6781 56:210.701 57:-26.5994 58:-8.11955 59:215.965 60:-50.9362 61:-2.65888
62:215.99 63:-67.4221 64:-50.3476 65:195.278 66:63.6144 67:-41.8337 68:189.768 69:10.6802 70:-47.5537 71:202.676 72:-18.9989 73:-49.5185
74:206.838 75:-36.5866 76:-49.9315 77:207.259 78:-52.5973
1 1:-9.30211 2:194.096 3:32.2338 4:-36.5695 5:194.172 6:75.1539 7:-12.9112 8:185.664 9:88.6053 10:30.4476 11:190.883 12:72.9345
13:57.5327 14:191.125 15:56.4814 16:73.8371 17:189.288 18:42.1984 19:-17.2485 20:203.962 21:77.1239 22:21.1534 23:199.197 24:20.7432
25:43.4931 26:182.934 27:-8.08598 28:52.392 29:165.142 30:-18.5261 31:56.1085 32:150.143 33:-22.188 34:-26.3151 35:204.913 36:70.2041
37:2.41601 38:199.886 39:12.2847 40:20.6707 41:207.806 42:-27.8559 43:32.679 44:207.092 45:-51.3914 46:41.0946 47:202.976 48:-66.1337
49:-36.1392 50:203.272 51:64.6194 52:-18.4922 53:198.164 54:9.37011 55:-9.58703 56:213.613 57:-28.1364 58:-2.82984 59:220.025 60:-52.1413
61:2.56261 62:221.862 63:-68.5471 64:-46.4604 65:195.494 66:61.7926 67:-37.5569 68:192.488 69:8.7212 70:-42.9937 71:205.537 72:-20.9495
73:-45.0775 74:210.681 75:-38.2614 76:-45.9989 77:213.152 78:-54.0644
1 1:-1.45545 2:192.396 3:26.9166 4:-29.2747 5:192.54 6:69.4808 7:-5.8171 8:183.959 9:83.2341 10:37.4879 11:189.233 12:67.4335 13:64.7935
14:189.818 15:51.3579 16:81.9026 17:188.699 18:37.9697 19:-9.95235 20:202.273 21:71.7033 22:29.1607 23:197.411 24:15.822 25:52.1227
26:180.736 27:-12.2734 28:61.6999 29:163.54 30:-23.1057 31:65.9739 32:148.841 33:-27.3319 34:-18.9259 35:203.252 36:64.6672 37:10.5363
38:198.156 39:7.12227 40:29.6701 41:206.125 42:-32.5971 43:41.9834 44:206.52 45:-55.9819 46:50.5758 47:203.875 48:-70.9577 49:-28.6819
50:201.641 51:58.9556 52:-10.3375 53:196.497 54:3.93731 55:-0.228971 56:211.532 57:-33.432 58:7.01361 59:218.392 60:-57.17 61:12.584
62:221.058 63:-73.4016 64:-38.9883 65:193.895 66:55.9933 67:-29.4086 68:190.876 69:3.04043 70:-34.0313 71:203.649 72:-26.8873 73:-35.8321
74:209.044 75:-44.1546 76:-36.6556 77:212.094 78:-59.8616


**Figure 3.21** The original data of 26 points.


1 1:0 2:0 3:0 4:-11.1566 5:23.041 6:40.2185 7:12.7489 8:15.6185 9:49.6094 10:40.1386 11:-12.6884 12:31.141 13:55.682 14:-34.0922
15:17.6047 16:57.3758 17:-51.6732 18:7.40401 19:7.57373 20:27.2595 21:33.3459 22:23.9732 23:-7.45239 24:-18.003 25:35.0438 26:-16.5956
27:-52.5826 28:39.7971 29:-33.8665 30:-63.666 31:40.4746 32:-48.621 33:-61.7785 34:-2.38019 35:27.5282 36:29.3445 37:5.00767 38:-5.69841
39:-21.0355 40:5.76294 41:-8.92601 42:-62.9115 43:7.5842 44:-24.6942 45:-81.9436 46:9.26933 47:-40.7091 48:-84.9275 49:-12.8971
50:26.3606 51:27.7098 52:-14.4842 53:-3.27617 54:-18.1061 55:-21.9721 56:-2.88121 57:-56.3159 58:-25.2163 59:-15.3178 60:-76.7527 61:-
25.4536 62:-30.4971 63:-82.6414 64:-23.7249 65:21.3315 66:31.0742 67:-32.1237 68:-3.66853 69:-12.0301 70:-48.2294 71:-3.82159 72:-38.3017
73:-55.487 74:-12.8884 75:-50.7786 76:-57.6685 77:-27.0343 78:-55.3389
1 1:0 2:0 3:0 4:-11.1784 5:23.1061 6:40.214 7:12.8544 8:15.9295 9:49.5229 10:40.1002 11:-12.4384 12:30.8658 13:55.3117 14:-34.0188
15:17.1886 16:56.3047 17:-51.7061 18:7.05177 19:7.49266 20:27.4702 21:33.2305 22:23.9691 23:-7.30864 24:-18.1034 25:34.8443 26:-16.1542
27:-52.8499 28:39.6317 29:-33.4699 30:-63.8761 31:40.3941 32:-48.2081 33:-61.8202 34:-2.48923 35:27.6344 36:29.2735 37:4.96118 38:-
5.73584 39:-21.0526 40:5.43352 41:-8.90366 42:-62.9663 43:7.266 44:-24.9045 45:-81.8245 46:9.06084 47:-40.9865 48:-84.4194 49:-13.0101
50:26.3652 51:27.693 52:-14.5508 53:-3.47673 54:-18.0359 55:-22.1925 56:-3.22742 57:-56.2437 58:-25.3451 59:-16.0311 60:-76.4869 61:-
25.3639 62:-31.3949 63:-81.9147 64:-23.7832 65:21.2479 66:31.1307 67:-32.1688 68:-4.00552 69:-11.869 70:-48.4017 71:-4.43254 72:-38.0841
73:-55.5558 74:-13.8329 75:-50.3887 76:-57.3941 77:-28.1684 78:-54.5222
1 1:0 2:0 3:0 4:-11.0083 5:22.6918 6:40.5384 7:13.154 8:15.5636 9:49.6043 10:40.6649 11:-12.246 12:30.4263 13:55.1457 14:-33.9944
15:16.1847 16:54.5575 17:-51.6588 18:5.94566 19:7.59749 20:27.23 21:33.4474 22:23.8884 23:-7.011 24:-18.3659 25:35.0714 26:-14.9615 27:-
53.2612 28:40.203 29:-32.0179 30:-64.5638 31:41.1511 32:-46.7663 33:-62.5778 34:-2.42286 35:27.3729 36:29.5655 37:4.83534 38:-5.51959
39:-21.1575 40:5.26691 41:-7.97397 42:-63.1512 43:7.22504 44:-23.5832 45:-82.3467 46:9.18698 47:-39.657 48:-84.9499 49:-12.9562
50:26.0566 51:28.0527 52:-14.6801 53:-3.39862 54:-17.9689 55:-22.3824 56:-3.00685 57:-56.1934 58:-25.4303 59:-15.7401 60:-76.5187 61:-
25.2683 62:-31.126 63:-81.9185 64:-23.6814 65:20.8425 66:31.5282 67:-32.2602 68:-4.0853 69:-11.668 70:-48.5239 71:-4.09653 72:-37.8949
73:-55.5788 74:-13.4759 75:-50.2904 76:-57.1117 77:-27.8862 78:-54.3283
1 1:0 2:0 3:0 4:-11.4907 5:22.6371 6:40.4814 7:12.719 8:15.783 9:49.6959 10:37.6652 11:-13.6564 12:29.3946 13:51.0543 14:-35.8474
15:14.7231 16:51.3887 17:-53.4283 18:4.2973 19:7.11748 20:27.3766 21:33.4807 22:24.0847 23:-6.7739 24:-18.2408 25:34.6845 26:-15.446
27:-53.1793 28:39.4781 29:-32.8605 30:-64.1104 31:40.4477 32:-47.6375 33:-62.2603 34:-2.89056 35:27.4035 36:29.5406 37:5.01624 38:-
5.49635 39:-21.1408 40:5.28438 41:-7.66142 42:-63.1864 43:7.274 44:-23.3484 45:-82.3415 46:9.29945 47:-39.297 48:-85.6539 49:-13.4088
50:25.9674 51:27.9706 52:-14.5548 53:-3.58646 54:-18.0592 55:-22.185 56:-3.04707 57:-56.329 58:-25.1134 59:-15.5955 60:-76.8098 61:-
24.9718 62:-30.7114 63:-82.9608 64:-24.1032 65:20.6346 66:31.397 67:-32.1748 68:-4.461 69:-11.8486 70:-48.4322 71:-4.78537 72:-38.1071
73:-55.5082 74:-14.0156 75:-50.6213 76:-57.4784 77:-28.1279 78:-55.4838
1 1:0 2:0 3:0 4:-11.3026 5:22.9913 6:40.3825 7:12.9005 8:16.0491 9:49.6073 10:38.3983 11:-13.3445 12:29.8622 13:52.3421 14:-35.4735
15:15.5735 16:53.0284 17:-53.154 18:5.30427 19:7.33508 20:27.5797 21:33.3171 22:24.0334 23:-7.04088 24:-18.2469 25:34.6072 26:-15.9225
27:-53.1708 28:39.3376 29:-33.4079 30:-64.0467 31:40.2205 32:-48.1802 33:-62.0326 34:-2.68557 35:27.6391 36:29.3896 37:4.95376 38:-
5.66985 39:-21.1276 40:5.16836 41:-8.37434 42:-63.1738 43:7.04552 44:-24.3076 45:-82.1605 46:8.97734 47:-40.3389 48:-85.1775 49:-13.2215
50:26.2529 51:27.8441 52:-14.6154 53:-3.62341 54:-18.0257 55:-22.3779 56:-3.29829 57:-56.3012 58:-25.4556 59:-16.0303 60:-76.668 61:-
25.402 62:-31.2656 63:-82.5526 64:-23.9488 65:21.0013 66:31.3249 67:-32.2435 68:-4.3541 69:-11.7771 70:-48.5893 71:-4.8671 72:-38.005
73:-55.7297 74:-14.2483 75:-50.3871 76:-57.6719 77:-28.481 78:-54.9349


**Figure 3.22** The data after normalization.

When we do the normalization to data, we can see that the palm center point become the coordinate origin. So we remove this point.

```
1 1:-9.5942 2:23.6473 3:40.2513 4:13.2689 5:13.8114 6:49.9874 7:43.307 8:-13.8495 9:35.0267 10:61.4611 11:-34.5057 12:23.7976 13:64.8032
14:-52.1084 15:14.0664 16:9.58289 17:26.0553 18:33.7708 19:23.4533 20:-9.61583 21:-17.6401 22:36.0881 23:-17.2762 24:-52.0292 25:41.4671
26:-31.7867 27:-66.3093 28:42.3097 29:-46.5338 30:-68.147 31:-0.213518 32:27.3444 33:29.5936 34:4.82197 35:-5.96701 36:-20.9961
37:6.93388 38:-9.81926 39:-62.758 40:8.17861 41:-23.1878 42:-83.5781 43:8.71121 44:-38.5592 45:-89.1879 46:-10.7571 47:27.2384 48:27.7554
49:-14.3814 50:-1.6571 51:-18.3957 52:-20.9943 53:-1.40085 54:-56.754 55:-25.0484 56:-11.9271 57:-78.0898 58:-26.8576 59:-26.228 60:-
85.6509 61:-22.0872 62:23.2753 63:30.8646 64:-32.0807 65:-0.359497 66:-12.6484 67:-48.127 68:0.80542 69:-38.9184 70:-56.521 71:-6.41432
72:-51.8709 73:-60.9118 74:-19.5368 75:-57.7039
1 1:-9.67264 2:23.7542 3:40.1817 4:13.1315 5:13.8679 6:50.0204 7:43.305 8:-13.8991 9:35.5082 10:61.758 11:-34.5 12:24.6542 13:64.9492
14:-52.2033 15:15.0462 16:9.53874 17:26.0952 18:33.7647 19:23.4938 20:-9.71829 21:-17.5414 22:36.1872 23:-17.6485 24:-51.8565 25:41.4642
26:-32.4831 27:-65.8453 28:42.1477 29:-47.2796 30:-67.3482 31:-0.240863 32:27.4053 33:29.5489 34:4.88134 35:-6.0209 36:-20.9721
37:6.98978 38:-10.1171 39:-62.7201 40:8.16817 41:-23.8396 42:-83.319 43:8.60622 44:-39.3734 45:-88.4819 46:-10.7803 47:27.3264 48:27.6724
49:-14.3228 50:-1.64966 51:-18.4484 52:-20.9838 53:-1.60591 54:-56.8078 55:-25.1089 56:-12.4712 57:-77.9651 58:-26.959 59:-26.9584 60:-
85.1603 61:-22.1354 62:23.401 63:30.7486 64:-32.0427 65:-0.289902 66:-12.7663 67:-48.113 68:0.672058 69:-39.0376 70:-56.5083 71:-6.75261
72:-51.8777 73:-60.8525 74:-19.9973 75:-57.473
1 1:-9.61065 2:23.6036 3:40.2992 4:13.225 5:13.6936 6:50.0553 7:43.4363 8:-13.9671 9:35.3932 10:61.6984 11:-34.6446 12:24.3473 13:64.1305
14:-52.3979 15:14.6018 16:9.58983 17:25.9829 18:33.8516 19:23.4699 20:-9.63852 21:-17.6281 22:36.2348 23:-17.7616 24:-51.8796 25:41.5634
26:-32.6124 27:-65.8382 28:42.2833 29:-47.4097 30:-67.3449 31:-0.200661 32:27.302 33:29.6595 34:4.84432 35:-5.9406 36:-21.0083 37:6.96423
38:-10.2692 39:-62.7407 40:8.16719 41:-24.0671 42:-83.2938 43:8.63372 44:-39.6041 45:-88.4549 46:-10.7455 47:27.2225 48:27.8033 49:-
14.3616 50:-1.59128 51:-18.4294 52:-21.0743 53:-1.76347 54:-56.7872 55:-25.2017 56:-12.647 57:-77.9403 58:-27.0357 59:-27.1254 60:-
85.1647 61:-22.0936 62:23.2768 63:30.8881 64:-32.0745 65:-0.264755 66:-12.7055 67:-48.2153 68:0.547836 69:-38.9459 70:-56.6368 71:-
6.81369 72:-51.8097 73:-61 74:-20.0254 75:-57.4757
1 1:-9.5398 2:24.1401 3:40.0072 4:13.19 5:14.1554 6:49.9489 7:45.534 8:-11.6313 9:36.4368 10:64.0707 11:-32.1523 12:25.5396 13:65.3509
14:-50.0114 15:15.7586 16:9.70304 17:26.2904 18:33.5901 19:23.4483 20:-10.0298 21:-17.4498 22:36.616 23:-19.6244 24:-51.1734 25:41.9209
26:-35.2348 27:-64.2942 28:42.3333 29:-50.116 30:-64.7724 31:-0.0664873 32:27.6417 33:29.3524 34:4.8594 35:-6.21872 36:-20.9296
37:7.40248 38:-12.7105 39:-62.3664 40:8.75269 41:-27.3946 42:-82.2934 43:9.19516 44:-43.1363 45:-86.8075 46:-10.6088 47:27.6268
48:27.4647 49:-14.3223 50:-1.6839 51:-18.4583 52:-20.5219 53:-3.67087 54:-56.8604 55:-24.3077 56:-15.6726 57:-77.4717 58:-25.9537 59:-
30.5511 60:-83.8919 61:-22.0007 62:23.8078 63:30.5599 64:-32.0449 65:-0.148743 66:-12.8029 67:-47.6009 68:-0.00898743 69:-39.4141 70:-
55.5405 71:-8.26292 72:-52.0458 73:-59.118 74:-22.1356 75:-56.5784
1 1:-9.5368 2:23.8714 3:40.1818 4:13.2195 5:13.8486 6:50.0375 7:45.6634 8:-11.7543 9:36.3902 10:63.9994 11:-32.3433 12:25.2692 13:64.5143
14:-50.1693 15:15.3506 16:9.70072 17:26.0895 18:33.7609 19:23.4497 20:-9.88695 21:-17.5393 22:36.6792 23:-19.6179 24:-51.2072 25:42.0059
26:-35.2685 27:-64.2774 28:42.4379 29:-50.1512 30:-64.7741 31:-0.0757809 32:27.4559 33:29.5401 34:4.84928 35:-6.07671 36:-20.978
37:7.4574 38:-12.6311 39:-62.4087 40:8.86194 41:-27.4076 42:-82.2691 43:9.34518 44:-43.1525 45:-86.7789 46:-10.6213 47:27.4394 48:27.6613
49:-14.3396 50:-1.58224 51:-18.4594 52:-20.5059 53:-3.55222 54:-56.8749 55:-24.2406 56:-15.4809 57:-77.543 58:-25.8492 59:-30.3072 60:-
84.0997 61:-22.0072 62:23.5846 63:30.7422 64:-32.0622 65:-0.106827 66:-12.777 67:-47.5948 68:0.050354 69:-39.4084 70:-55.5138 71:-8.07776
72:-52.1384 73:-59.093 74:-21.8948 75:-56.8451
```

**Figure 3.23** Data without the palm center(coordinate origin).

There is another problem that some data very far from the leap motion looks very small and some data very near looks very big. But we know the finger shape information has no relationship with how big the model is. To solve the problem and also improve the classification accuracy, we try to scale all of the data to [-1, 1].

```
1 1:0.944557 2:0.935779 3:-0.813972 4:0.916528 5:0.93332 6:0.827565 7:-0.634289 8:-0.982908 9:0.822703 10:-0.956174 11:-0.993749
12:0.874563 13:-0.616956 14:-0.991638 15:0.8958 16:0.91947 17:0.941668 18:-0.8879 19:-0.990567 20:-0.93064 21:-0.376906 22:0.49938 23:-
0.387769 24:-1 25:0.817439 26:0.0173935 27:-1 28:0.551639 29:-0.0881546 30:-1 31:0.921085 32:0.941749 33:-0.879658 34:-0.958835 35:-
0.93056 36:-0.969143 37:-1 38:0.863787 39:-1 40:-0.998297 41:1 42:-1 43:-0.987578 44:1 45:-1 46:0.926583 47:0.940461 48:-0.864068 49:-
0.126185 50:-0.933163 51:-0.902356 52:-0.987859 53:0.262696 54:-0.922648 55:-0.986862 56:0.23562 57:-0.801563 58:-0.987865 59:0.00720202
60:-0.461705 61:0.945673 62:0.93598 63:-0.805835 64:-0.413176 65:-0.899317 66:1 67:-0.438367 68:-0.85937 69:-0.514241 70:-0.474723 71:-
0.843792 72:-0.00799636 73:-0.120248 74:-0.86596 75:0.836596
1 1:0.908163 2:0.94107 3:-0.835359 4:0.805395 5:0.936089 6:0.842703 7:-0.635676 8:-0.990117 9:0.883243 10:-0.804872 11:-0.993492
12:0.934125 13:-0.60103 14:-0.994379 15:0.948628 16:0.902646 17:0.944082 18:-0.889049 19:-0.933549 20:-0.940477 21:-0.293073 22:0.553958
23:-0.436108 24:-0.967218 25:0.816457 26:-0.0627442 27:-0.940861 28:0.50506 29:-0.170989 30:-0.876854 31:0.911018 32:0.945572 33:-
0.887412 34:-0.827228 35:-0.936259 36:-0.952686 37:-0.987334 38:0.7976 39:-0.993369 40:-0.999854 41:0.918034 42:-0.974824 43:-1
44:0.898598 45:-0.935554 46:0.917771 47:0.945917 48:-0.879117 49:0.0916187 50:-0.931961 51:-0.938774 52:-0.986266 53:0.2455 54:-0.93366
55:-0.992047 56:0.210054 57:-0.79166 58:-0.994774 59:-0.0192898 60:-0.437608 61:0.922581 62:0.943103 63:-0.838711 64:-0.263717 65:-
0.870893 66:0.800136 67:-0.432393 68:-0.873893 69:-0.557243 70:-0.471613 71:-0.863875 72:-0.0096268 73:-0.110532 74:-0.883579 75:0.870119
1 1:0.936924 2:0.933616 3:-0.799253 4:0.881021 5:0.927548 6:0.858712 7:-0.544613 8:-1 9:0.868784 10:-0.835244 11:-1 12:0.912785 13:-
0.690338 14:-1 15:0.924667 16:0.922114 17:0.937287 18:-0.872682 19:-0.967197 20:-0.932818 21:-0.366713 22:0.580173 23:-0.450793 24:-
0.971603 25:0.850069 26:-0.0776233 27:-0.939956 28:0.544048 29:-0.185439 30:-0.876345 31:0.925819 32:0.939086 33:-0.868228 34:-0.909291
35:-0.927766 36:-0.977509 37:-0.993123 38:0.7638 39:-0.996973 40:-1 41:0.889425 42:-0.972376 43:-0.996746 44:0.869867 45:-0.933089
46:0.930989 47:0.939475 48:-0.855383 49:-0.0525925 50:-0.92253 51:-0.925644 52:-1 53:0.232288 54:-0.929444 55:-1 56:0.201793 57:-0.789691
58:-1 59:-0.025347 60:-0.437824 61:0.942607 62:0.936065 63:-0.799175 64:-0.388791 65:-0.860623 66:0.903204 67:-0.476042 68:-0.887422
69:-0.524162 70:-0.503079 71:-0.867502 72:0.00667762 73:-0.134698 74:-0.884654 75:0.869727
1 1:0.969796 2:0.960169 3:-0.888979 4:0.852712 5:0.950176 6:0.809904 7:0.910254 8:-0.660508 9:1 10:0.373694 11:-0.887835 12:0.995689
13:-0.55721 14:-0.931069 15:0.987038 16:0.965255 17:0.955894 18:-0.921933 19:-0.997607 20:-0.970383 21:-0.215272 22:0.790114 23:-0.692657
24:-0.837552 25:0.9712 26:-0.379394 27:-0.743167 28:0.558424 29:-0.486022 30:-0.479758 31:0.975215 32:0.960414 33:-0.921495 34:-0.875863
35:-0.957183 36:-0.923544 37:-0.89382 38:0.221289 39:-0.931482 40:-0.912682 41:0.470979 42:-0.875171 43:-0.930321 44:0.429961 45:-
0.782709 46:0.982909 47:0.964539 48:-0.916776 49:0.093477 50:-0.937492 51:-0.945615 52:-0.916167 53:0.0723391 54:-0.944427 55:-0.923383
56:0.0596254 57:-0.75248 58:-0.92628 59:-0.149598 60:-0.375308 61:0.987113 62:0.966154 63:-0.892191 64:-0.27237 65:-0.813242 66:0.738091
67:-0.213893 68:-0.948061 69:-0.693068 70:-0.234625 71:-0.953538 72:-0.0499323 73:0.173642 74:-0.965393 75:1
1 1:0.971188 2:0.94687 3:-0.835328 4:0.876572 5:0.935143 6:0.850547 7:1 8:-0.678385 9:0.994141 10:0.337359 11:-0.896431 12:0.976887 13:-
0.648471 14:-0.93563 15:0.96504 16:0.964371 17:0.943737 18:-0.889765 19:-0.995636 20:-0.956669 21:-0.29129 22:0.824921 23:-0.691814 24:-
0.843968 25:1 26:-0.383272 27:-0.741026 28:0.588499 29:-0.489932 30:-0.48002 31:0.971794 32:0.948749 33:-0.888938 34:-0.898296 35:-
0.942162 36:-0.956732 37:-0.881376 38:0.238933 39:-0.938883 40:-0.89639 41:0.469344 42:-0.87281 43:-0.912572 44:0.427943 45:-0.780098
46:0.978161 47:0.952922 48:-0.88113 49:0.0291767 50:-0.92107 51:-0.946376 52:-0.913739 53:0.0822888 54:-0.947395 55:-0.917632
56:0.0686331 57:-0.758141 58:-0.91916 59:-0.140752 60:-0.385514 61:0.983999 62:0.953506 63:-0.840525 64:-0.340413 65:-0.796122
66:0.781997 67:-0.21129 68:-0.941599 69:-0.691012 70:-0.228087 71:-0.942545 72:-0.072135 73:0.177738 74:-0.95618 75:0.96128
```

**Figure 3.24** Data after scale to [-1, 1].

### 3.3.3 Model Train

We choose 3 finger shape of 5 typical kinds of gestures to classify.

**Table 3.10** 5 typical kinds of gestures.

| Action name | How to use | Function |
|---|---|---|
| a. Scroll, slide, swipe | Slide two fingers up or down on the touchpad | To scroll. |
| b. Zoom in/out | Pinch with two fingers on the touchpad | To zoom in or out. |
| c. Rotate | Move two fingers around each other on the touchpad. | To rotate a photo or other item. |
| d. Swipe between pages | Swipe left or right with two fingers on the touchpad | To show the previous or next page. |
| e. Three finger drag | Use three fingers drag on the touchpad | To drag items on your screen, then click or tap to drop. |

Because the gesture a and d, b and c have the same finger shape, so we only need to classify 3 kinds of finger shape.

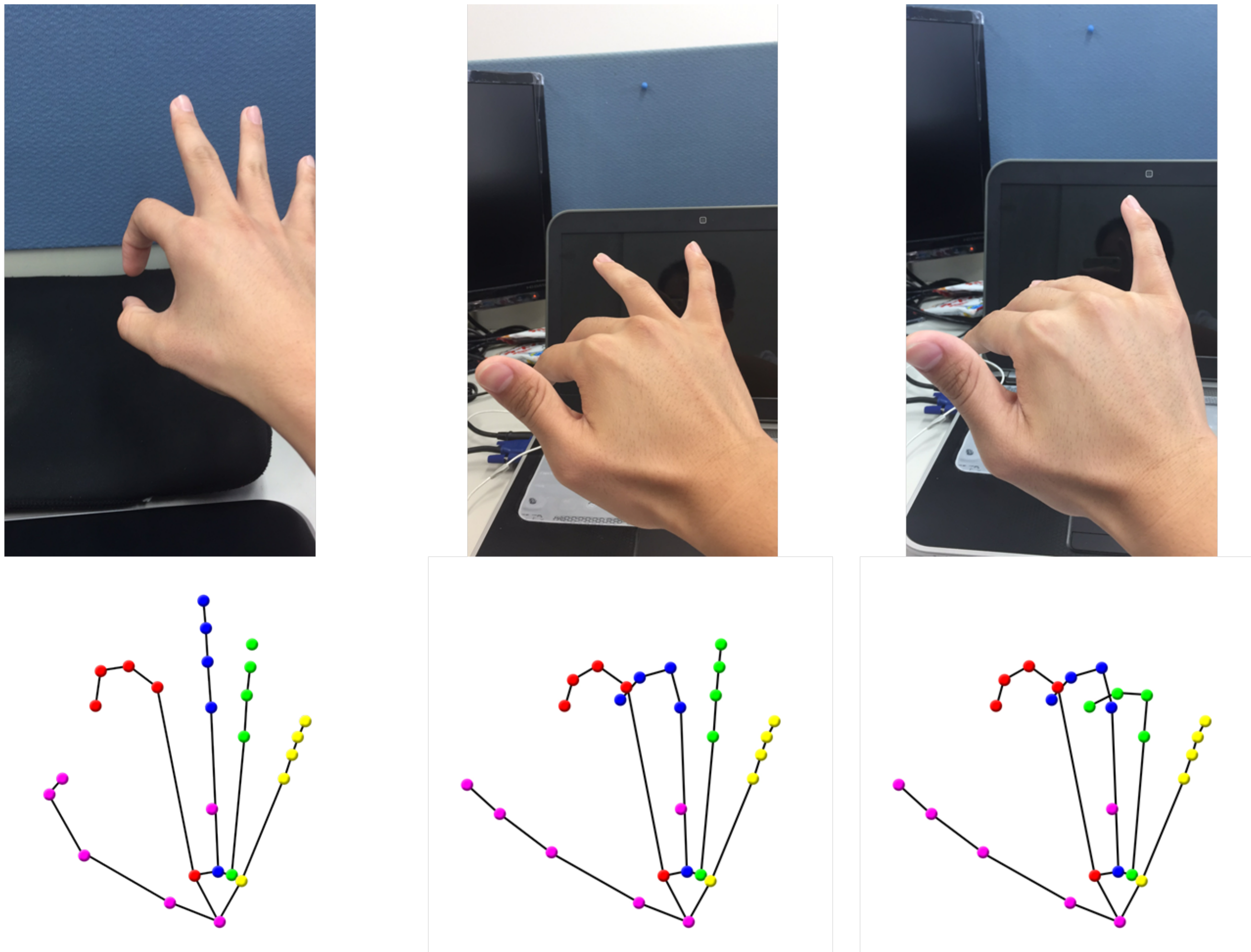


**Figure 3.25** 3 kinds of finger shape and the points model.

Then we get 540 sets of data as our training sample, using SVM train a model to predict the new data.

```
D:\workplace\libsvm-3.22\windows>svm-train data.txt data.model
.*.*
optimization finished, #iter = 189
nu = 0.395589
obj = -17.207068, rho = -0.068689
nSV = 74, nBSV = 0
.*.*
optimization finished, #iter = 203
nu = 0.462001
obj = -21.001957, rho = 0.147802
nSV = 80, nBSV = 4
.*.*
optimization finished, #iter = 198
nu = 0.399497
obj = -17.975762, rho = -0.027078
nSV = 77, nBSV = 0
.*.*
optimization finished, #iter = 227
nu = 0.382799
obj = -18.015198, rho = -0.026246
nSV = 79, nBSV = 2
.*.*
optimization finished, #iter = 157
nu = 0.502075
obj = -19.394104, rho = 0.207652
nSV = 70, nBSV = 2
.*.*
optimization finished, #iter = 159
nu = 0.435666
obj = -16.772395, rho = 0.041694
nSV = 67, nBSV = 0
.*
optimization finished, #iter = 157
nu = 0.415013
obj = -16.807195, rho = 0.043828
nSV = 69, nBSV = 0
.*.*
optimization finished, #iter = 167
nu = 0.507486
obj = -20.375132, rho = -0.167891
nSV = 73, nBSV = 3
.*
optimization finished, #iter = 165
nu = 0.479062
obj = -20.397107, rho = -0.175260
nSV = 75, nBSV = 5
.*.*
optimization finished, #iter = 176
nu = 0.417585
obj = -17.539819, rho = 0.002158
nSV = 71, nBSV = 0
Total nSV = 184
```

**Figure 3.26** The training procedure.

```
svm_type c_svc
kernel_type rbf
gamma 0.0133333
nr_class 3
total_sv 34
rho 0.337649 0.284837 -0.0574405
label 1 2 3
nr_sv 9 13 12
SV
1 1 1:-9.5942 2:23.6473 3:40.2513 4:13.2689 5:13.8114 6:49.9874 7:43.307 8:-13.8495 9:35.0267 10:61.4611 11:-34.5057 12:23.7976 13:64.8032 14:-52.1084
15:14.0664 16:9.58289 17:26.0553 18:33.7708 19:23.4533 20:-9.61583 21:-17.6401 22:36.0881 23:-17.2762 24:-52.0292 25:41.4671 26:-31.7867 27:-66.3093
28:42.3097 29:-46.5338 30:-68.147 31:-0.213518 32:27.3444 33:29.5936 34:4.82197 35:-5.96701 36:-20.9961 37:6.93388 38:-9.81926 39:-62.758 40:8.17861
41:-23.1878 42:-83.5781 43:8.71121 44:-38.5592 45:-89.1879 46:-10.7571 47:27.2384 48:27.7554 49:-14.3814 50:-1.6571 51:-18.3957 52:-20.9943 53:-1.40085
54:-56.754 55:-25.0484 56:-11.9271 57:-78.0898 58:-26.8576 59:-26.228 60:-85.6509 61:-22.0872 62:23.2753 63:30.8646 64:-32.0807 65:-0.359497 66:-12.6484
67:-48.127 68:0.80542 69:-38.9184 70:-56.521 71:-6.41432 72:-51.8709 73:-60.9118 74:-19.5368 75:-57.7039
0.1214482678063466 0.09202078807937432 1:-9.67264 2:23.7542 3:40.1817 4:13.1315 5:13.8679 6:50.0204 7:43.305 8:-13.8991 9:35.5082 10:61.758 11:-34.5
12:24.6542 13:64.9492 14:-52.2033 15:15.0462 16:9.53874 17:26.0952 18:33.7647 19:23.4938 20:-9.71829 21:-17.5414 22:36.1872 23:-17.6485 24:-51.8565
25:41.4642 26:-32.4831 27:-65.8453 28:42.1477 29:-47.2796 30:-67.3482 31:-0.240863 32:27.4053 33:29.5489 34:4.88134 35:-6.0209 36:-20.9721 37:6.98978
38:-10.1171 39:-62.7201 40:8.16817 41:-23.8396 42:-83.319 43:8.60622 44:-39.3734 45:-88.4819 46:-10.7803 47:27.3264 48:27.6724 49:-14.3228 50:-1.64966
51:-18.4484 52:-20.9838 53:-1.60591 54:-56.8078 55:-25.1089 56:-12.4712 57:-77.9651 58:-26.959 59:-26.9584 60:-85.1603 61:-22.1354 62:23.401 63:30.7486
64:-32.0427 65:-0.289902 66:-12.7663 67:-48.113 68:0.672058 69:-39.0376 70:-56.5083 71:-6.75261 72:-51.8777 73:-60.8525 74:-19.9973 75:-57.473
0.1492212612006958 0.1312332181127358 1:-9.61065 2:23.6036 3:40.2992 4:13.225 5:13.6936 6:50.0553 7:43.4363 8:-13.9671 9:35.3932 10:61.6984 11:-34.6446
12:24.3473 13:64.1305 14:-52.3979 15:14.6018 16:9.58983 17:25.9829 18:33.8516 19:23.4699 20:-9.63852 21:-17.6281 22:36.2348 23:-17.7616 24:-51.8796
25:41.5634 26:-32.6124 27:-65.8382 28:42.2833 29:-47.4097 30:-67.3449 31:-0.200661 32:27.302 33:29.6595 34:4.84432 35:-5.9406 36:-21.0083 37:6.96423
38:-10.2692 39:-62.7407 40:8.16719 41:-24.0671 42:-83.2938 43:8.63372 44:-39.6041 45:-88.4549 46:-10.7455 47:27.2225 48:27.8033 49:-14.3616 50:-1.59128
51:-18.4294 52:-21.0743 53:-1.76347 54:-56.7872 55:-25.2017 56:-12.647 57:-77.9403 58:-27.0357 59:-27.1254 60:-85.1647 61:-22.0936 62:23.2768 63:30.8881
64:-32.0745 65:-0.264755 66:-12.7055 67:-48.2153 68:0.547836 69:-38.9459 70:-56.6368 71:-6.81369 72:-51.8097 73:-61 74:-20.0254 75:-57.4757
0.4785791235854997 0.4615213167457944 1:-9.5398 2:24.1401 3:40.0072 4:13.19 5:14.1554 6:49.9489 7:45.534 8:-11.6313 9:36.4368 10:64.0707 11:-32.1523
12:25.5396 13:65.3509 14:-50.0114 15:15.7586 16:9.70304 17:26.2904 18:33.5901 19:23.4483 20:-10.0298 21:-17.4498 22:36.616 23:-19.6244 24:-51.1734
25:41.9209 26:-35.2348 27:-64.2942 28:42.3333 29:-50.116 30:-64.7724 31:-0.0664873 32:27.6417 33:29.3524 34:4.8594 35:-6.21872 36:-20.9296 37:7.40248
38:-12.7105 39:-62.3664 40:8.75269 41:-27.3946 42:-82.2934 43:9.19516 44:-43.1363 45:-86.8075 46:-10.6088 47:27.6268 48:27.4647 49:-14.3223 50:-1.6839
51:-18.4583 52:-20.5219 53:-3.67087 54:-56.8604 55:-24.3077 56:-15.6726 57:-77.4717 58:-25.9537 59:-30.5511 60:-83.8919 61:-22.0007 62:23.8078 63:30.5599
64:-32.0449 65:-0.148743 66:-12.8029 67:-47.6009 68:-0.00898743 69:-39.4141 70:-55.5405 71:-8.26292 72:-52.0458 73:-59.118 74:-22.1356 75:-56.5784
0.02424562056122262 0.02977958318443163 1:-9.4747 2:24.1694 3:40.1128 4:13.3076 5:14.1443 6:50.0272 7:44.537 8:-12.5247 9:35.3927 10:62.1936 11:-33.5126
12:23.7954 13:63.1738 14:-51.3135 15:13.797 16:9.79422 17:26.3151 18:33.6534 19:23.4557 20:-10.054 21:-17.5272 22:36.3082 23:-20.5207 24:-51.19 25:41.211
```

**Figure 3.27** This is the model file after training with 3 classes.

### 3.3.4 Predict Test and Evaluation

We get 60 sets of data as our predict data. The first result is the predict accuracy without normalization and scale.

```
D:\workplace\libsvm-3.22\windows>svm-predict predict.txt data.model data.predict
t
Accuracy = 78.3333% (47/60) (classification)
```

**Figure 3.28** Accuracy without normalization and scale.

This is the accuracy after normalization but without scale.

```
D:\workplace\libsvm-3.22\windows>svm-predict predict.txt data.model data.predict
t
Accuracy = 91.6667% (55/60) (classification)
```

**Figure 3.29** Accuracy after normalization.

This is the accuracy after normalization and scale.

```
D:\workplace\libsvm-3.22\windows>svm-predict predict_scale.txt data_scale.model
data_scale.predict
Accuracy = 96.6667% (58/60) (classification)
```

**Figure 3.30** Accuracy after normalization and scale.

## 3.4 Motion Recognition

Finally, we need to detect the motion of fingers to give the feed back. There are some sample effects of our system.

After we classify by finger shape, we get three possibilities. The first is three fingers drag, second is zoom in/out or rotate, third is two fingers scroll or two fingers swipe. Now we need to judge which is the specific gesture by the movement trajectory.

In the case of thumb and index finger touched, we can switch between zoom gesture and rotate gesture freely. That means when we do a zoom gesture, but at that time we also want to do a rotate gesture, we need not relax the two fingers and touch again, we can do it directly. We can get the two fingers' fingertip position information easily, and then we can calculate the distance of thumb fingertip and index fingertip, we marked as d. And the initial distance we marked as $d_0$, that means the distance of two fingertip just when classification finished.

And we monitor d frame by frame. For each frame, we calculate the ratio of d to $d_0$. And at the same time we zoom the graphical interface by the same ratio.

We give a sample application just like a photo viewer to show the effect of gesture recognition. We can also use our system in any other application which need a gesture interaction, but now we give a sample application as a photo viewer just like the following figures show.

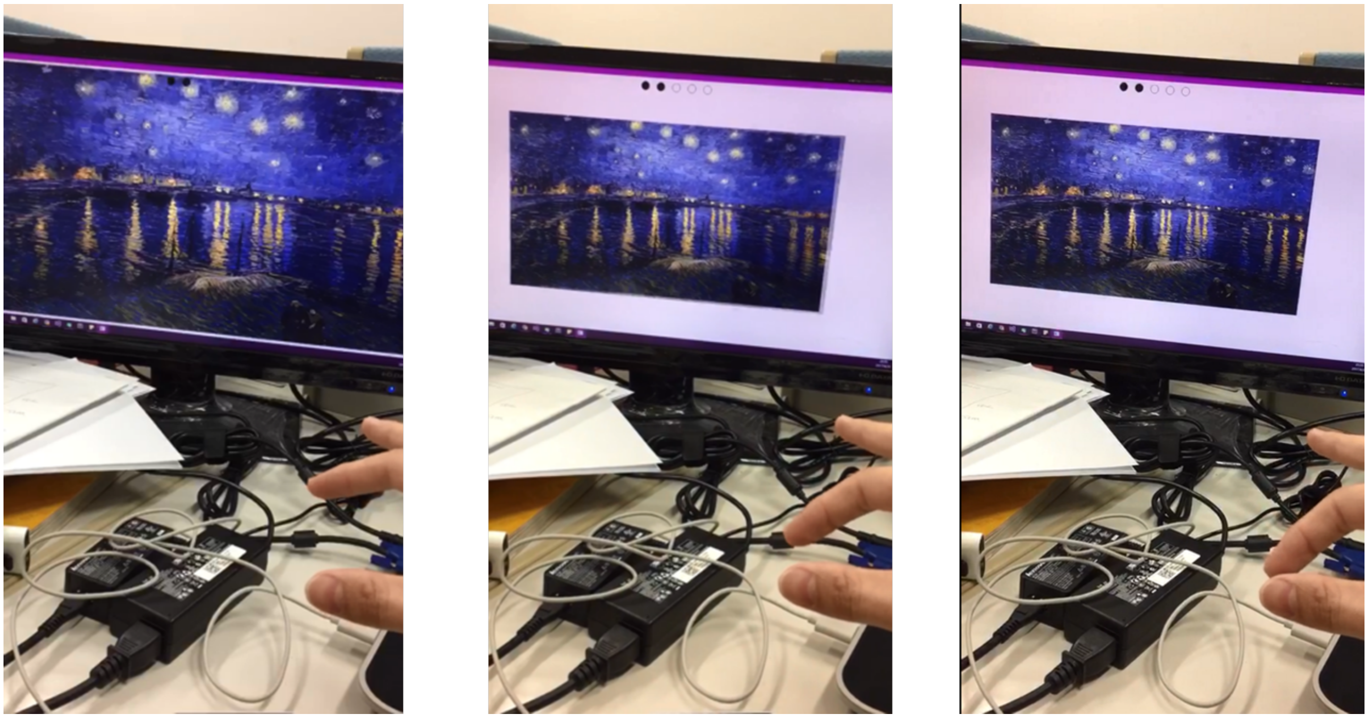


**Figure 3.31** Effect of zoom in/out.

At the same time, we also get the vector from thumb fingertip to index fingertip, we marked as $\mathbf{v}$. And the initial vector we marked as $\mathbf{v_0}$, that means the vector from thumb fingertip to index fingertip just when classification finished. For each frame, we calculate the angle between $\mathbf{v}$ and $\mathbf{v_0}$, And at the same time we rotate the graphical interface follow the angle.

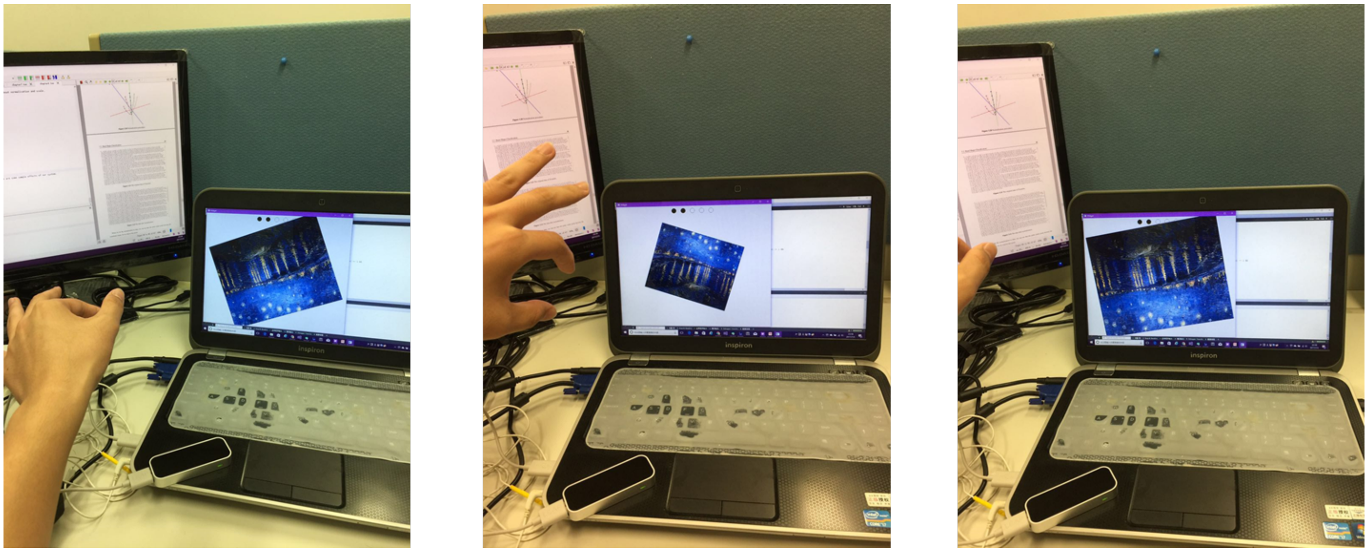


**Figure 3.32** Effect of rotate.

In the case of three fingers drag, we can also get the middle fingertip position information easily. we mark the initial middle fingertip coordinate as $(x_0, y_0, z_0)$, and we get the middle fingertip coordinate $(x_i, y_i, z_i)$ frame by frame. For each frame, we calculate the distance and direction between $(x_i, y_i, z_i)$ and the previous frame $(x_{i-1}, y_{i-1}, z_{i-1})$, and then make the appropriate movement.

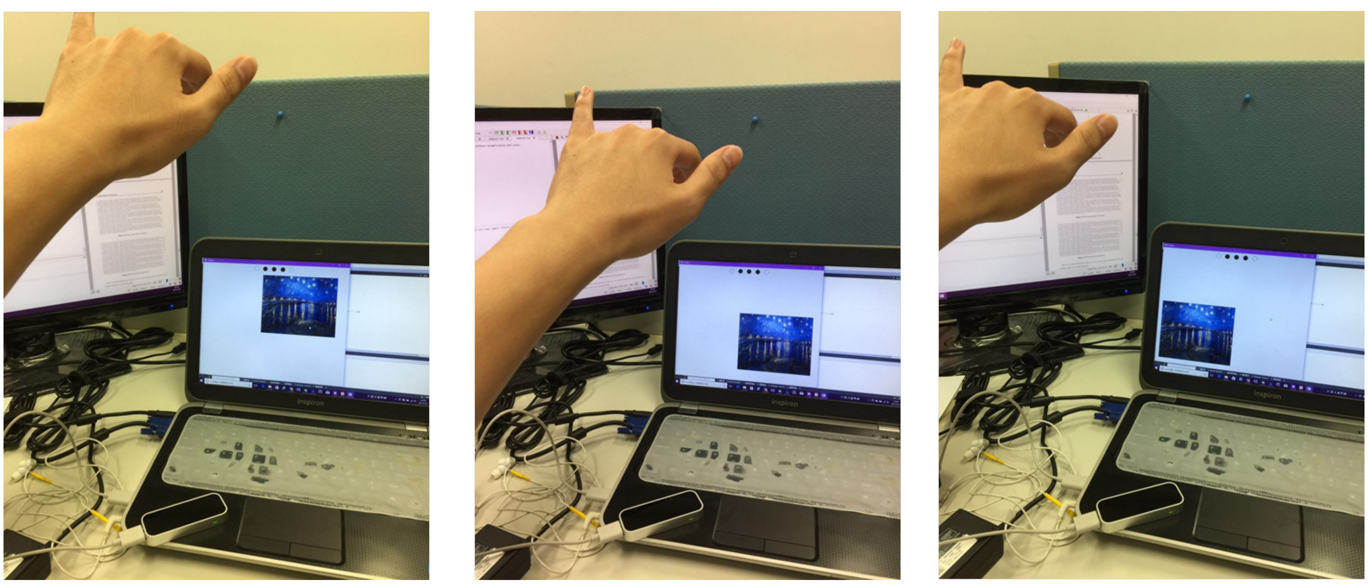


**Figure 3.33** Effect of three fingers drag.

In the case of two fingers scroll or swipe, it is very similar to three fingers drag. The difference is we just need to judge the movement of each frame is horizontal or vertical. If we get a horizontal movement, then we give a swipe feedback. If we get a vertical movement, then we give a scroll feedback.

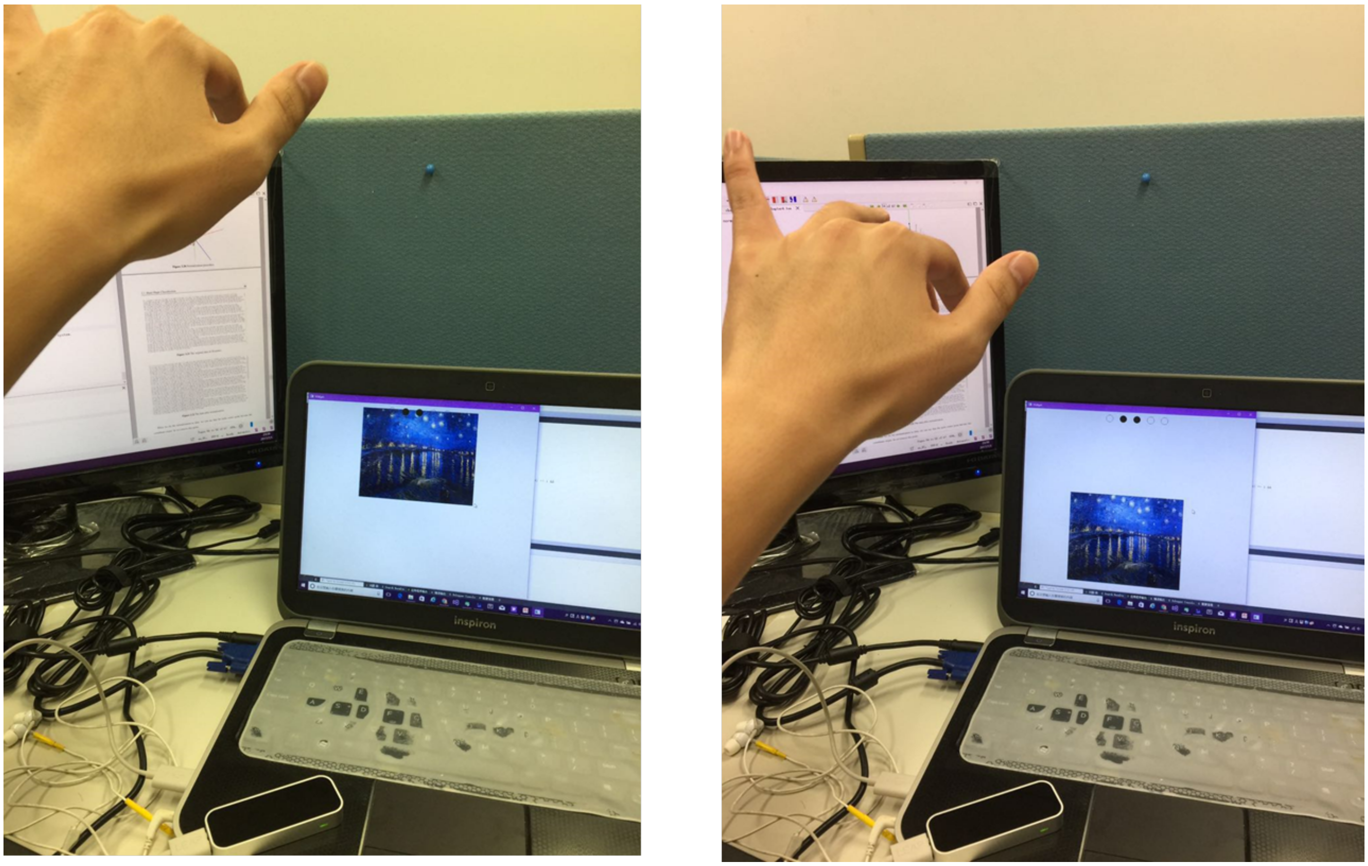



**Figure 3.34** Effect of scroll.

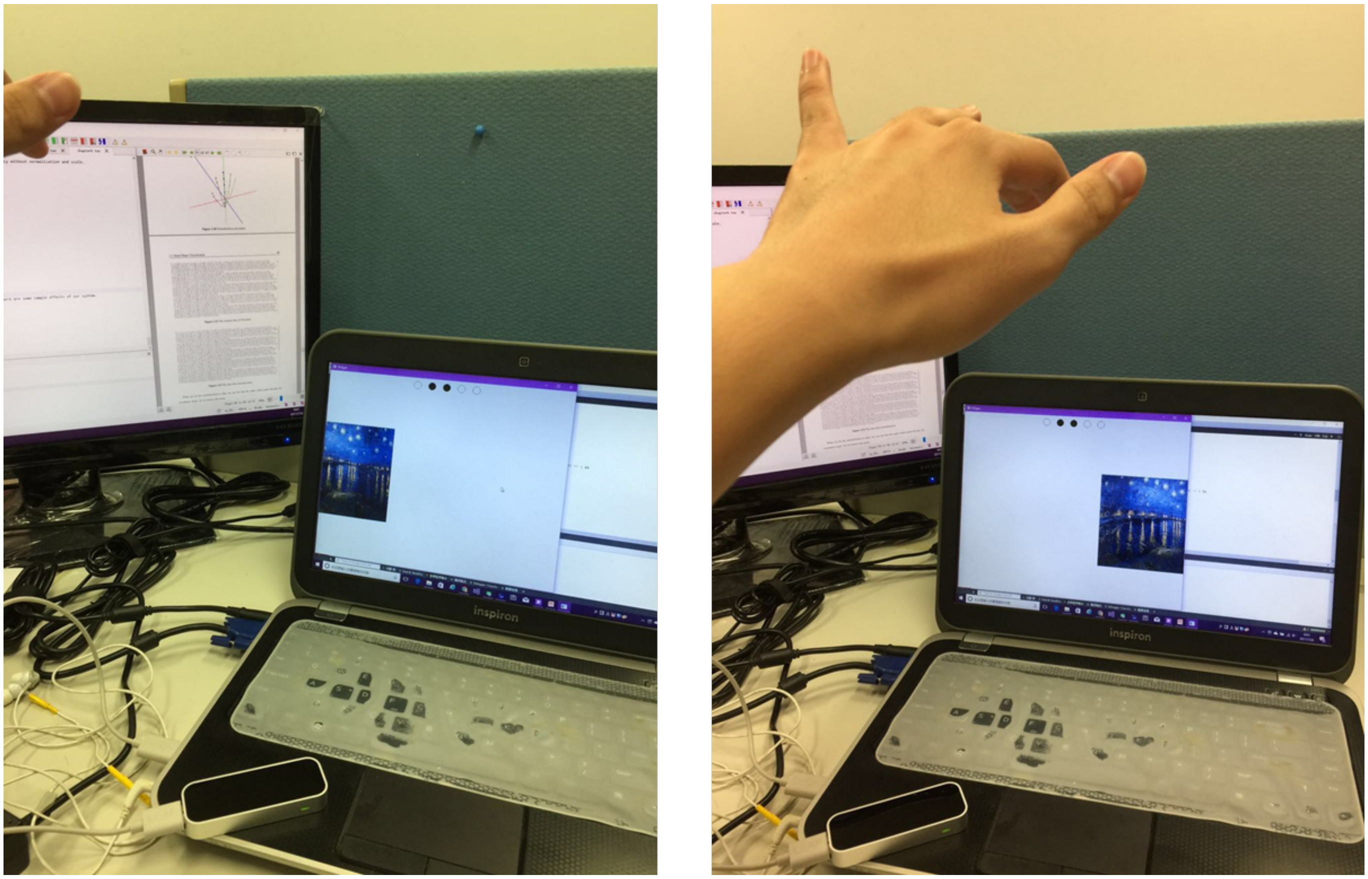


**Figure 3.35** Effect of swipe.

## 3.5 Experiment

We mainly did a experiment with the accuracy of our system.

In this experiment, we invited 3 users aged 22-25, each of the user have no experience with such kind of system but have related knowledge about how to use Leap Motion. The experiment step are as follows

1. Each user perform the 5 gestures one by one in two kinds of different speed for 20 times.
2. We calculate the accuracy of each gesture in different speed.

The results are as the table showed.

**Table 3.11** Result of experiment.

| | User No.1 | User No.2 | User No.3 |
|---|---|---|---|
| Two fingers scroll(slow) | 100%(20/20) | 95%(19/20) | 95%(19/20) |
| Two fingers scroll(fast) | 90%(18/20) | 90%(18/20) | 90%(18/20) |
| Two fingers swipe(slow) | 100%(20/20) | 100%(20/20) | 95%(19/20) |
| Two fingers swipe(fast) | 90%(18/20) | 95%(19/20) | 90%(18/20) |
| Zoom in/out(slow) | 100%(20/20) | 95%(19/20) | 95%(19/20) |
| Zoom in/out(fast) | 95%(19/20) | 95%(19/20) | 90%(18/20) |
| Rotate(slow) | 95%(19/20) | 95%(19/20) | 95%(19/20) |
| Rotate(fast) | 95%(19/20) | 95%(19/20) | 90%(18/20) |
| Three fingers drag(slow) | 100%(20/20) | 100%(20/20) | 100%(20/20) |
| Three fingers drag(fast) | 100%(20/20) | 95%(19/20) | 100%(20/20) |

We calculate the average accuracy of each gesture. The result is as the figure shows.

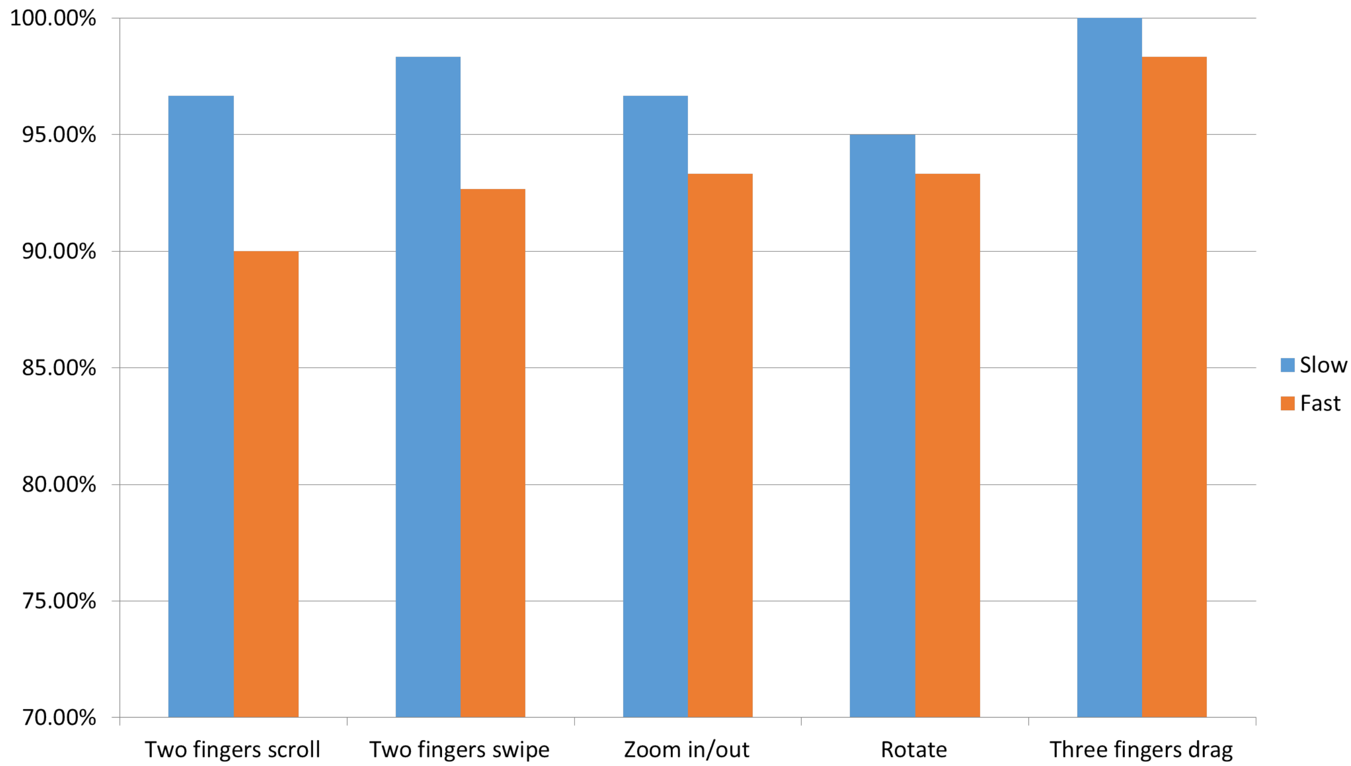



**Figure 3.36** Result of experiment.

From the result we can see speed has some effect on accuracy.

**Table 3.12** The effect on accuracy of speed.

| | Scroll | Swipe | Zoom | Rotate | Drag |
|---|---|---|---|---|---|
| Slow | 96.67% | 98.33% | 96.67% | 95% | 100% |
| Fast | 90% | 92.67% | 93.33% | 93.33% | 98.33% |

And then we calculate the average accuracy of each gesture. The result is as the figure shows.

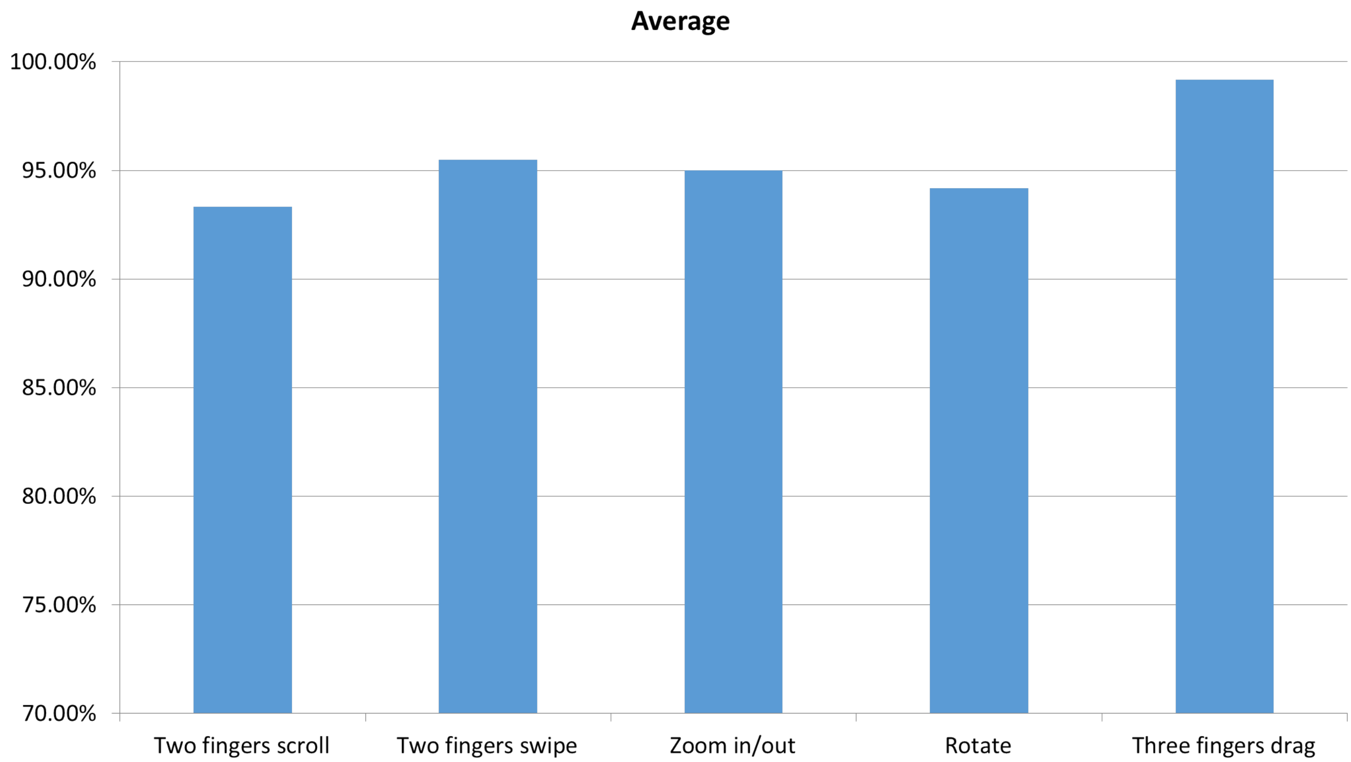



**Figure 3.37** Average accuracy of each gesture.

Finally we get the accuracy of each gesture and we can calculate the average accuracy of our system is 95.43%.

## 3.6 Summary of this chapter

In this chapter, we introduce our implementation details step by step. In our system, we combine the click part, classification part and movement recognition part as one system.

First we detect "click down" for every finger. If any fingers Triggered the threshold, that means get into "touch" state. Then we get the finger shape information at the same time, and classify the static data at that frame by the model that we have trained. Now we can classify 3 kinds of finger shape, but we can change the model file and classify more finger shape at any time when we need. And then we can further determined gesture by the movement trajectory of fingertip. And then give users a real time feedback. Finally, if we detect any fingers get out of "touch" state or any new fingers get into "touch" state, we move to the classification procedure and begin a new gesture recognition process.

In our work, we combine static finger shape and dynamic movement trajectory to recognize gestures. We use some math method to calculate some data and also use machine learning method to classify data by finger shape.

# Chapter 4

# Conclusion and Future Work

## 4.1 Conclusion

Because of 3D interactions becoming more and more popular. And also 2D gesture interaction seems a little weak in interacting with machine. So in our research, we presented a natural 3D multi-touch interaction in 3D space using depth camera.

In this work, we generated a 3D multi-touch interaction by using Leap Motion. User can interact with computer by 3D multi-touch gestures over Leap Motion. Our approach is combine static data classification and dynamic data detect. Our system can gestures in 3D space using a depth camera Leap Motion. And also we give a sample application just like a photo viewer. It can detect 5 kinds of different gestures and give feedback. And if we need to detect more gestures, we can change the model file at any time. And if we need to detect more gestures, we can change the model file at any time. Moreover, because of that there is no touching action in 3D space, so we define a concept of "touch" state just like tap or click. It is a very important part for multi-touch gestures. We design a method “click down” and “click up”. When the fingers “click down” means activate the fingers and get into “touch” state, “click up” means release the fingers and back to “nature” state. We can describe the “touch” state as activating some fingers by “click down” condition and release by "click up" condition.

Our system can interact with users very friendly. For every part of our system, its accuracy of recognition is more than 90%. Although in this work we only test our gesture interaction in a sample application as photo viewer, it also suitable for other possible application and cooperative work if gesture interaction is needed.

## 4.2 Future Work

Our research is aimed to realize a 3D multi-touch interaction. We mainly make some progress on system design and implementation, but there are still exist some aspects that can be improved.

1. Our recognition rates are not so perfect, we can also do something to improve the accuracy.

2. We can implement more kinds of gestures and even define some new gesture in 3D space.

3. When we get into "touch" state, sometimes skip out before we finish a gesture. So maybe we can improve this part of system in the future.

# References


[1] https://en.wikipedia.org/wiki/Multi-touch.

[2] I. Albrecht, J. Haber, and H.-P. Seidel, "Construction and animation of anatomically based human hand models," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 98–109.

[3] H. Benko, A. D. Wilson, and P. Baudisch, "Precise selection techniques for multi-touch screens," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 1263–1272.

[4] J. Segen and S. Kumar, "Look ma, no mouse!" *Communications of the ACM*, vol. 43, no. 7, pp. 102–109, 2000.

[5] C. Von Hardenberg and F. Bérard, "Bare-hand human-computer interaction," in *Proceedings of the 2001 workshop on Perceptive user interfaces*. ACM, 2001, pp. 1–8.

[6] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[7] P. Garg, N. Aggarwal, and S. Sofat, "Vision based hand gesture recognition," *World Academy of Science, Engineering and Technology*, vol. 49, no. 1, pp. 972–977, 2009.

[8] Y. Boussemart, F. Rioux, F. Rudzicz, M. Wozniewski, and J. R. Cooperstock, "A framework for 3d visualisation and manipulation in an immersive space using an untethered bimanual

gestural interface," in *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM, 2004, pp. 162–165.

[9] J. Triesch and C. Von Der Malsburg, "Robotic gesture recognition by cue combination." in *GI Jahrestagung*, 1998, pp. 223–232.

[10] S. Chu and J. Tanaka, "Hand gesture for taking self portrait," *Human-computer Interaction. Interaction Techniques and Environments*, pp. 238–247, 2011.

[11] J. Y. Han, "Low-cost multi-touch sensing through frustrated total internal reflection," in *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 2005, pp. 115–118.

[12] J. Accot and S. Zhai, "More than dotting the i's—foundations for crossing-based interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2002, pp. 73–80.

[13] G. Apitz and F. Guimbretière, "Crossy: a crossing-based drawing application," in *Proceedings of the 17th annual ACM symposium on User interface software and technology*. ACM, 2004, pp. 3–12.

[14] J.-C. Lévesque12, D. Laurendeau, and M. Mokhtari, "An asymmetric bimanual gestural interface for immersive virtual environments," 2013.

[15] M. Liu, M. Nancel, and D. Vogel, "Gunslinger: Subtle arms-down mid-air interaction," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015, pp. 63–71.

[16] A. Kulshreshth and J. J. LaViola Jr, "Exploring the usefulness of finger-based 3d gesture menu selection," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 2014, pp. 1093–1102.

[17] J. Shukla and A. Dwivedi, "A method for hand gesture recognition," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*. IEEE, 2014, pp. 919–923.

[18] https://support.apple.com/en-us/HT204895.

[19] https://en.wikipedia.org/wiki/Leap_Motion.

[20] https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html.

[21] https://en.wikipedia.org/wiki/LIBSVM.

[22] A. Sugiura, M. Toyoura, and X. Mao, "A natural click interface for ar systems with a single camera," in *Proceedings of Graphics Interface 2014*. Canadian Information Processing Society, 2014, pp. 67–75.

[23] https://en.wikipedia.org/wiki/Qt_(software).

[24] H. Karam and J. Tanaka, "Two-handed interactive menu: An application of asymmetric bimanual gestures and depth based selection techniques," in *International Conference on Human Interface and the Management of Information*. Springer, 2014, pp. 187–198.