# Development of a Haptic Video Chat System

by

# Longyu Zhang

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies In partial fulfillment of the requirements For the M.A.Sc degree in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering School of Electrical Engineering and Computer Science Faculty of Engineering University of Ottawa

© Longyu Zhang, Ottawa, Canada, 2012

## Abstract

Since conventional audio-video teleconferencing systems have reached their limits, the needs of integrating new sensations to improve users' telecommunication experience are growing. Haptics, the sense of touch, which includes handshake, comforting hug, encouraging pat, and other physical contacts, is of great importance for interpersonal communication, since it allows people to express and receive intimate affection, intention or emotion efficiently. Motivated by a few haptic telecommunication softwares, this thesis presents an innovative webcam-based touchscreen to replace the haptic device, which is used in HuqMe [30][26][27] system as human hand, to further increase people's degree of immersion with the audio-video-haptic teleconferencing system at a more reasonable price. With our webcam-based touchscreen, the user could directly use his/her bare hand to touch the image of the person who he/she is chatting with, instead of controlling a haptic device as the medium to realize indirect touch. This thesis also gives details of the touchscreen method and our proposed mathematical models for touch position calculation. Experimental results show that our system is accurate and robust, while maintaining high compatibility with conventional audio-video teleconferencing systems for combination. With our haptic jacket, the passive user can feel the touch of the active user at the right position. Also, additional applications with the touchscreen, such as writing and drawing, are developed and tested. Finally, we draw the conclusions and talk about future work.

#### Acknowledgements

I would like to give my sincerest gratitude and appreciation to my supervisor Prof. Abdulmotaleb El Saddik for his continuous guidance and support not only in academic level but also in my personal life.

Special and sincere thanks go to Dr. Jamal Saboune for the invaluable assistance, guidance, and feedback he provided throughout my research, as well as his revision of this thesis, Dr. Jongeun Cha for his profound description about *HugMe* system and the help on my courses, and PhD candidate Feng Shi for the support with programming and writing.

I would also like to thank all colleagues in DISCOVER and MCRLab and all my friends for their contributions throughout the research and their help on my life.

Finally, I am grateful to my family. Their inspiration, understanding and support made this thesis possible. This work is dedicated to them.

# Contents

1	Intr	oducti	on	1					
	1.1	Backgr	ound and Motivation	1					
	1.2	Existin	g Problems	6					
	1.3	Object	ive and Contribution	7					
	1.4	Thesis	Organization	9					
<b>2</b>	Literature Review								
	2.1	Audio-	Video Telecommunication Systems	10					
	2.2	Integra	ting Haptics to Telecommunication Systems	13					
		2.2.1	Vibrating Tattoo	14					
		2.2.2	The Hug Shirt	15					
		2.2.3	SMS Text Based Affective Haptic Application	16					
		2.2.4	Huggy Pajama	17					
		2.2.5	Kiss Transmission Device	18					
		2.2.6	Interpersonal Haptic Communication in Second Life	19					
		2.2.7	iFeel_IM!	19					
		2.2.8	HugMe	20					
	2.3	Touchs	screen Technologies	22					
		2.3.1	Resistive	22					
		2.3.2	Capacitive	23					
		2.3.3	Infrared	24					
		2.3.4	Other Technologies	25					
		2.3.5	The video-based touchscreen	26					
	2.4	Finger	Tip Detection Methods	29					
		2.4.1	Contour and color based approaches	30					
		2.4.2	Pattern-based	31					

		2.4.3 Marker-based approaches	33
3	The	e Proposed Haptic Video Chat System	36
	3.1	Webcam Position	36
	3.2	Initialization and Calibration Phase	38
		3.2.1 The screen's bottom border detection	39
		3.2.2 Two reference images with pre-set positions	43
	3.3	Realtime Touch Position Calculation	46
		3.3.1 The pinhole camera model	47
		3.3.2 Realtime finger's vertical position calculation	48
		3.3.3 The realtime finger's horizontal position calculation	52
	3.4	More Filtering for Finger Detection	55
		3.4.1 The continuity property	56
		3.4.2 The shape property	57
	3.5	Combination of the Touchscreen and the Haptic Jacket	59
4	Imp	plementation and Results	62
	4.1	Hardware and Software Configuration	62
	4.2	Experimental Result for the Touchscreen Algorithm	63
		4.2.1 Qualitative results:	63
		4.2.2 Quantitative results:	67
	4.3	Human Body Detection Result	68
	4.4	Other Applications	71
<b>5</b>	Con	clusion and Future Work	74
	5.1	Conclusion	74
	5.2	Future Work	75

# List of Tables

4.1	Accuracy of Calculated	Touch Positions .								•							6	7
-----	------------------------	-------------------	--	--	--	--	--	--	--	---	--	--	--	--	--	--	---	---

# List of Figures

1.1	Haptics enables bidirectional flow of information	2
1.2	HugMe synchronous haptic teleconferencing system	4
1.3	Haptic jacket and its inner part	5
2.1	Videophone from D-Link	1
2.2	A webcam from Microsoft	2
2.3	A telepresence room from Cisco	3
2.4	Vibrating tattoo alerts by Nokia	5
2.5	Hug Shirt from CuteCircuit	6
2.6	SMS service high level design	7
2.7	Overview of Huggy Pajama 18	8
2.8	Kiss transmission device	9
2.9	Second Life and haptic communication system block diagram	0
2.10	The architecture of iFeel_IM!	1
2.11	Haptic device from Sensable Company 22	2
2.12	An exploded view of a resistive touchscreen	3
2.13	An illustration for touch events detection of infrared touchscreen 24	4
2.14	User is browsing relics using Magic Crystal Ball	5
2.15	The illustration of the dTouch pointing system	6
2.16	The view and segmentation result of Visual Screen	7
2.17	Position of webcams of Hoang <i>et al.</i>	8
2.18	Improved hand contour and ratio curve of original contour	9
2.19	Glove-based sign language by Liang <i>et al.</i>	1
2.20	Finger tips detection process of Magic Crystal Ball 32	2
2.21	Foreground detection and hand template tracking results from Gorodnichy	
	<i>et al.</i>	3
2.22	3D drawing with LED markers	4

3.1	Flowchart of our proposed method	37
3.2	Webcam position and its field of view	38
3.3	Detection result of the screen bottom border	39
3.4	Screen background setting	40
3.5	All detected long lines in the image	41
3.6	Triangle edges and their point of intersection	41
3.7	Finger detection result at the first pre-set position	42
3.8	The two pre-set positions (white dots)	43
3.9	User's finger is pointing the first indicated position (the white dot)	44
3.10	Finger detection result at 22cm	45
3.11	Pinhole camera model	46
3.12	The geometry of a pinhole camera seen from another axis	47
3.13	Combination of 10cm and 22cm calibrations and the mathematical model	48
3.14	Mathematical model for vertical position calculation in another situation	51
3.15	Mathematical model for horizontal position calculation	53
3.16	Mathematical model for horizontal position calculation in another situation	56
3.17	Some pixels representing the mirror finger are mistaken as belonging to	
	the real finger given their color features	57
3.18	Our strategy works well to stop the large increase in finger width	58
3.19	Logo is mistaken as human face	59
3.20	Face detection and upper body dividing result	60
4.1	Full-screen black window with red test points and white calculated points	64
4.2	Original figure with a <i>Skype</i> chat image as background	65
4.3	Testing result with a <i>Skype</i> chat image as background	65
4.4	Testing results with other chat images as background	66
4.5	The boxplots of touch points' vertical and horizontal absolute errors	68
4.6	Upper body dividing results of other images	69
4.7	Touch activates number 13 box	70
4.8	Arduino Duemilanove microcontroller board	71
4.9	Writing word "MCR"	72
4.10	Figure drawing	73

# **Glossary of Terms**

- HD High-definition
- **3D** Three-dimensional
- 2.5D Two-and-a-half-dimensional
- **RGB** A color model which uses red, green and blue light to reproduce a broad array of colors
- **HSV** A color model which uses Hue, Saturation and Value to reproduce a broad array of colors
- YUV A color space typically used as part of a color image pipeline
- ${\bf IP}\,$  Internet protocol
- **IDE** Integrated Development Environment
- **VoIP** Voice over Internet Protocol
- **PSTN** Public Switched Telephone Network
- VGA Video Graphics Array
- $\mathbf{PC}$  Personal Computer
- $\mathbf{Hz}$  Hertz
- **DOF** Degree of Freedom
- **IM** Instant Message
- **VR** Virtual Reality

 ${\bf SMS}\,$  Short Message Service

## ${\bf API}$ Application Programming Interface

# Chapter 1

# Introduction

# 1.1 Background and Motivation

Telecommunications refers to the transmission of information over a significant distance to communicate. In earlier times, people mainly use visual signals, such as smoke, beacons, and signal flags, or audio messages, such as loud whistles, horns, and drumbeats, for telecommunications. In the modern age of electricity and electronics, telecommunications also involves the use of electrical devices, such as telephone, teleprinter, and telegraph, as well as the use of radio and microwave communications, and also the use of orbiting satellites and the Internet, along with the utilization of fiber optics and their associated electronics.

When we mention telecommunications, one sure thing to talk about is telecommunication systems. Basic telecommunication systems consist of three primary units which are usually present in some forms:

- A transmitter which takes information and converts it to a signal.
- A transmission medium, which is also known as the "physical channel" that carries the signal.
- A receiver which takes the signal from the medium and converts it back into usable information.

Nowadays, The teleconferencing systems are widely used in our everyday life; they refer to the live exchange and mass articulation of information among several people and machines remote from one another but connected by a telecommunications system.



Figure 1.1: Haptics enables bidirectional flow of information

Telecommunications system may support the teleconference by offering audio, video, and/or data services. Currently, available Internet teleconferencing systems includes telephone conferencing, video conferencing, web conferencing, and a few Augmented Reality conferencing [23]. Popular teleconferencing softwares for personal use, including *Skype* [14], *Google Talk* [3], *Windows Live Messenger* [18], and *Yahoo! Messenger* [19], became extremely popular in the world, and turned into part of our daily lives. More and more people now are willing to use those applications to video/audio chat with their family, while the number of companies or groups which utilize teleconferencing systems for their business is increasing.

The benefits of teleconferencing systems are obvious. First, their cost is much lower compared with that of ordinary cellphone calling and texting. Actually, most of these systems allow members to make free phone calls between one and another when logged to the service with a computer or via mobile applications. Also they help save the transportation fees and help, since it may not be necessary for people to travel to solve problems. Work which previously required much travelling time could be done with a simple laptop or cellphone through this kind of systems.

However, despite the rapid development of conventional personal-use teleconferencing systems, the improvement of audio and video contents for teleconferencing has reached its limits with the high-definition (HD) video and 3D displays. The idea of making telecommunications systems more interactive and enabling them to convey people's emotion

#### Introduction

more directly has attracted a lot of attention. Fuelled by several exciting discoveries, researchers nowadays have fostered their interest on incorporating the sense of touch in teleconferencing systems [31]. Touch is a unique human sensory modality since it enables a bidirectional flow of energy and information between the real, or virtual, environment and the end user, unlike other human modalities like vision and audition, which are all unidirectional. Figure 1.1 from El Saddik *et al.* [50] illustrates this distinguishing feature of haptics. By combining the touch sense with teleconferencing systems, researchers would be able to offer the users a much more realistic and intimate experience of communication.

The term related to the sense of touch is haptics, which was derived from the Greek verb "haptesthai" meaning "of or relating to the sense of touch". Haptics refers to the science of manual sensing and manipulation of surrounding objects and environments through the sense of touch, and the "touching" of objects and/or environment could be made by humans, machines, or a combination of both; while the objects and environments can be real, virtual, or a combination of both [51]. Haptics is important for interpersonal communication, since it can provide a much more direct way for people expressing and receiving intimate affection, intention or emotion. One example to prove the significance of haptics is that physical contacts, such as handshakes, encouraging pats, comforting hugs or other body contacts, are more commonly used compared with spoken language when people need to show their deep feelings. Thus, combining haptics with conventional video-audio teleconferencing systems would considerably improve the users' experience.

An existing example of this combination is the *HugMe* system [30] [26] [27]. *HugMe* is an interpersonal haptic teleconferencing system developed by the Multimedia Communications Research Laboratory at the University of Ottawa. The system block diagram, which includes both the passive side and the active side, is shown in Figure 1.2. From the diagram, we can see that the passive user (child) is wearing a haptic jacket with vibrators and heaters inside, and his image is captured by a 2.5D camera. Then his 2.5D model is transmitted to the active side, and the active user (parent) could use a haptic device which is represented by a sphere avatar in the screen to touch the model of their child. Once a collision is detected between the sphere avatar and the 2.5D human model, the haptic device will produce a force feedback to the active user's hand, and the system will transmit the collision point's position from the active side to the passive side through the Internet. Upon receiving the touch position, the passive side of the system will inform the control centre of the haptic jacket by Bluetooth. The control center will then activate the corresponding vibrators embedded inside the jacket to offer the passive



Figure 1.2: *HugMe* synchronous haptic teleconferencing system

user the feeling of physical contact. At the same time, the heaters are activated in order to simulate the warmth of a touch. Figure 1.3 shows the haptic jacket used and its inner part which contains several vibrators.

The *HugMe* system provides an effective way for parents who are always far away from their children to communicate with their son or daughter "physically" in the most realistic manner. This system is also of great importance for children since the absence of the affectionate contact could result in health and social problems according to several studies [7] [39]. With this system, when the child is crying, the remote parent can pat or hug him/her to stop the child crying easily. *HugMe* is not only used to coax the child, but also could be utilized between friends or business partners to interact with each other



Figure 1.3: Haptic jacket and its inner part

physically. Therefore, it is of a great practical use. In fact, *HugMe* offers the users a much better teleconferencing experience by adding the extra haptic features which are not available in the traditional teleconferencing systems.

However, due to the existence of the expensive haptic device and the 2.5D camera, the cost of the *HugMe* system is too high to be commercialized. And most recently popular teleconferencing softwares do not support 2.5D video information, which means combining this system with those softwares to benefit more people is difficult. Also, using a sphere avatar to represent the active user's hand and controlling it through a haptic device would generate a distance in the feeling between this virtual touch and the real touch. Thus, we came up with the idea of using a touchscreen to replace and eliminate the haptic device, so that the parent can touch the image of his/her child directly with his/her real hand to express the emotion. Touching the screen will also provide the parent with the feedback of touch though the feeling may differ from that of touching a real human body.

The interaction between the parent and the child would thus be described as follows: when the parent is not able to provide a face-to-face touch to the child, he could touch the child's image shown on the screen. On the other side, the receiver (child) will feel the touch at the appropriate location through the haptic jacket.

Then we investigated the market and found that though many cellphones and tablets do support touchscreens, few desktop and laptop screens offer this function, and the supported ones are much more expensive than the ordinary ones. Therefore we had the idea of using a simple webcam to transfer a regular monitor into a touchscreen, and then combine this webcam-based touchscreen with any recently popular teleconferencing software to produce an enhanced haptic-audio-video teleconferencing system. Through this system, the users can experience a more intimate emotion transmission at a reasonable cost.

In fact, our idea is to simulate a touchscreen by detecting the position of the contact point between the user's finger and the regular screen through the video feed provided by the webcam.

## 1.2 Existing Problems

The challenges in transforming a regular screen into a touchscreen using a webcam and utilizing this innovative touchscreen to replace the haptic device in the *HugMe* system can be summarized in the following points:

- The webcam position: The webcam position selection is important because it can significantly influence the finger's detection result in the images. Finger detection is the key for the touch position calculation, but it could be easily affected by surrounding environments, such as lighting condition, unrelated objects within the detection region, screen image colors and many other factors. Thus, choosing an appropriate webcam position where the influence of those factors can be minimized is crucial for the system's robustness and accuracy.
- Finger/hand shapes and colors: A finger can be detected in an image given some distinctive features, such as color or shape. However, a person's hand and fingers could distinctly differ from those of another person in shapes, colors, or even both. Thus, making the system capable of working efficiently with most users is a great challenge to researchers. Moreover, the finger may merge with the background which contains similar colors, and this increases the difficulty of extracting a finger from the background in the image. Also, since we need to use the finger to touch the "touchscreen", the color of the finger's tip will be influenced by the screen's light reflection as it is close to the screen. All of those factors will lead to inaccurate finger detection results, and cause the touch position calculation failure.
- Touch/contact position calculation: Even if we can detect a finger accurately, mapping its position in the image captured by the webcam to a touch position on

the monitor is another problem. Algorithms may vary from one to another because of the diverse webcam positions, and various algorithms could result in different system accuracies. Thus, researchers are required to advance accurate algorithms or mathematical models to process this mapping.

• User control and freedom: *HugMe* uses the Augmented Reality Tool Kit [1] to track markers on the haptic jacket to track the passive users and map the touch point to the sensors embedded inside the clothes. This tracking method is accurate. However, it also limits the user's choice of clothes, since markers must be on outer layer of the clothes within the detection region. If we change the haptic jacket to a inner layer shirt and give the users more freedom of choosing their outfit, tracking markers would be difficult. To make the system more usable we need to eliminate as many constraints as possible, and allow the interaction to be as natural and intuitive as possible. Thus, we adopt human proportions to make the received touch more realistic and accurate by allowing action at different areas of the jacket and associate them with the corresponding body parts in the image (shoulder, chest, *etc.*).

## **1.3** Objective and Contribution

In this research work, we first try to transform a regular screen into a touchscreen using a simple webcam. we then use it to replace the expensive haptic device in the *HugMe* teleconference system, and make it compatible with other teleconferencing softwares. With our system, the users can experience a more intimate emotion transmission, and enjoy the advantages of haptic telecommunications at a reasonable price. The receiver will feel a touch whenever the user on the other end touches the image. Also, we realized functions like writing and drawing with this webcam-based touchscreen.

The main contributions of the research work can be summarized in the following points:

• The proposed webcam positioning: As mentioned above, the webcam position is significant for finger detection because it will decide the influence of ambient conditions. To improve the webcam-based touchscreen's robustness, we opt to position the webcam on top of the monitor to have its image plan perpendicular to the screen plan. Thus, only the screen and a small area below will be captured by the webcam, so even if there are big background changes around the users, such as people or pets passing by, surrounding object changes, or if the user changes position or posture, the system will not be affected. Also, after testing different positions, we find that this position is less influenced by illuminating devices as most light will not shine directly into the camera at such position. The proposed webcam position is an essential contribution of our research.

- The finger detection and segmentation algorithm: Given the webcam position we select, we advanced our own generic and simple finger detection algorithm based on the screen lay out. By adding some strategies to eliminate the background effect, we realize good detection results with finger of different shapes and colors. Our system can in fact work effectively with most users and in different configuration (screen or light). Also, our application is fast enough for realtime calculation because of the simple detection algorithm we advanced.
- The touch position calculation mathematical models: Positioning our webeam on top of the monitor makes it hard to calculate the touch position, but we advanced some mathematical models, based on the geometry of the scene to solve this problem. With the proposed mathematical models and a simple calibration phase, we can calculate both vertical and horizontal touch position precisely, and the experimental results prove that the touch position calculated through the mathematical models is accurate.
- The body parts detection: As mentioned above, some systems use markers to track different body parts in the image. In order to give the users more choices of their outer layer clothes, we do not rely on any marker in our approach. Actually, we use face detection, which is highly reliable even under low image resolutions, to detect the passive user in the image, and use the Vitruvian Man [17] model to estimate the positions of the different body parts. Thus, people can wear haptic clothes as inner layer while wearing any outfit they want.
- The compatibility with the conventional teleconferencing systems: We captured several real video-chatting window images with different users, various backgrounds, and diverse resolutions to test the robustness of our webcam-based touchscreen. Results show that our webcam-based touchscreen method works well on those video-chatting images. Also, mapping the remote user's upper body parts with the sensors embedded inside the haptic clothes produced good results. Given

this, we can conclude that our system has high compatibility with popular teleconferencing systems to which it can be combined with to realize a synchronous haptic-video-audio teleconferencing system at a reasonable price.

• Providing a more intuitive and realistic interaction: As mentioned before, our system is fast enough to realize realtime applications. By adding some control functions, such as setting black background color, starting and stopping detecting finger, we also realized a writing and drawing application with which users can write and draw whatever they want. Of course, more applications like games could be developed based on our webcam-based touchscreen system.

## 1.4 Thesis Organization

This Thesis is organized as follows:

- Chapter 2: This chapter presents a review of the literature about all the related works. It describes the development of the telecommunications systems and haptics, the different touchscreen types, the various finger detection methods, and the definition and nature of similar triangles which are highly related to our proposed mathematical models used to calculate the touch position.
- Chapter 3: In Chapter 3, we discuss our proposed method of transforming a regular monitor into a touchscreen using a webcam. Details about the webcam position, the initialization and calibration phase, the mathematical models and the finger detection strategies are explained.
- Chapter 4: The experimental results are exposed in this chapter. We describe the experimental hardware and software settings, the complexity and the system accuracy. Through the *Skype* chatting window and realtime applications such as writing and drawing, we show some qualitative and quantitative results.
- Chapter 5: In the final chapter, we summarize the contributions and progress in this research with a conclusion and we put forward the future work that can improve this field.

# Chapter 2

# Literature Review

# 2.1 Audio-Video Telecommunication Systems

Along with the rapid development of the world' economy came the fast growth of the Internet. The Internet is an exchange of messages between computers [5]. Each computer on the Internet has a unique IP address for other computers routing information to it. Thus, any computer on the Internet could send messages to any other ones using its IP address. And these messages carry with them the original computer's IP address allowing two-way communication.

This global system of interconnected computer networks has reshaped or refined most traditional communications media, such as telephone, film, and television. It uses the standard Internet Protocol suite to serve billions of people worldwide. Moreover, Internet enabled many entirely new forms of social activities and interaction while producing a tremendous impact on areas like social networking and entertainment, politics and political revolutions, philanthropy, censorship, and many other fields. According to Hilbert *et al.* [35], "telecommunication has been dominated by digital technologies since 1990 (99.9% in digital format in 2007)". From this we can learn that a lot of multimedia communications is delivered over the Internet.

The audio telecommunications domain has been influenced by the Internet greatly. Internet telephony, also known as Voice over Internet Protocol (VoIP), Internet calling, or Internet phone, is a technology that is revolutionizing the communications market [33]. Compared with the traditional telephone company infrastructure which uses the public switched telephone network (PSTN), Internet telephony offers its communications services such as voice, fax, and/or SMS via the Internet, and it has reduced the price,

#### Literature Review



Figure 2.1: Videophone from D-Link

accelerated the rate of innovation, and made advanced services affordable and accessible to a greater population.

While audio telecommunication was not enough to satisfy people's needs, audio and video telecommunications grew up. Videophone, which is a telephone with a video display, shown in Figure 2.1 from D-Link company [16], provided the first version of video telephony, followed by webcams and high-definition telepresence. It is capable of providing simultaneous video and audio communication between people in realtime.

Compared with videophones, webcams are much more popular. They are relatively low-cost devices which can provide live video and audio streams via personal computers, and can be used with many software clients for both audio calls and videoconferencing [52]. The webcam is also known for its flexibility, users can put it almost anywhere on the desk or the monitor. Most consumer webcams are able to provide VGA resolution video at a frame rate of 30 frames per second, and some newer webcams can offer even higher resolution video, more frames per second, or both, such as the Microsoft webcam [9] shown in Figure 2.2 that is capable of offering High-Definition (HD) videos. As webcam capabilities have been combined with instant messaging, text chat service such as *AOL Instant Messenger*, and VoIP service such as *Skype*, peer-to-peer live video communication over the Internet has reached millions of mainstream PC users all over the world. New features like automatic face tracking, autofocus, and automatic lighting control, further increase the popularity of webcams. Moreover, with increased bandwidth



Figure 2.2: A webcam from Microsoft

and sophisticated software clients, current telecommunications systems allow both pointto-point and multi-party calls.

Telecommunication softwares are mainly developed for individual use. On the other hand, a telepresence system is a high-end videoconferencing system and service commonly employed by enterprise-level corporate offices. The conference rooms use state-of-the art room design, video cameras, displays, sound systems and processors, also coupled with high-to-very-high capacity bandwidth transmissions. Figure 2.3 shows a telepresence room from Cisco TX1300 Series[15]. Rather than travelling great distances to have a face-to-face meeting, it is now common to use a telepresence system, which is a multiple codec video system, to hold a teleconference. After "dialling" with a telepresence room, each party can see and talk to every other party on a screen or screens as if they were in the same room. This provides enormous time and cost benefits, and greatly enhances communications since it allows for perceptions of facial expressions and other body language. A telepresence system could also enable a remote user to "visit" some location that is monitored by a collection of webcams, and can provide users with any current view from viewing points while possibly offering the sense of being physically present at a remote location through interaction with the system interface [53].

### Literature Review



Figure 2.3: A telepresence room from Cisco

# 2.2 Integrating Haptics to Telecommunication Systems

Haptics, the sense of touch, is considered to be one of the most complex senses. It can be divided into two sub-senses: kinesthetic and cutaneous. Kinesthetic stimulation is roughly related to force exerted on the joints and muscles, while cutaneous stimulation could be felt through the skin. For instance, kinesthetic stimulation could be felt by carrying heavy objects which may be felt on wrists and elbows, and cutaneous stimulation could be felt by identifying different frequencies of vibration or different levels of heat, or even the sense of balance [45].

Haptic feedback includes kinesthetic and tactile feedback. Kinesthetic feedback refers to the sensation of joint movement and muscle tension, while tactile feedback relies on providing force at the skin surface with millions of tiny mechanoreceptors. Actuators are widely used to apply forces. The early designs of haptic feedback mainly used electromagnetic technologies, while the new haptic actuator technologies, such as electroactive polymers, piezoelectric, electrostatic and subsonic audio wave surface, are beginning to emerge. Providing haptic feedback in Virtual Reality simulations is difficult because of different requirements. The actuators should be lightweight, low cost, have a fast response time (around 1.5kHz) and deliver a large range of forces over small and large spatial areas alike [25]. Despite those problems, there are still several commercial systems available now, such as Cyber Grasp gloves [32] and Phantom Device [4]. They work well to deliver such feedbacks.

While audio and visual information can be easily transmitted remotely using audio and video signals, the tactile information needs the two parties involved (the active user and the receiver/passive user) to be physically present at the same location. However, with the advances of touch simulation technology, we can use actuators to simulate the tactile cue remotely in an action.

Conventional telecommunications systems usually just support sensory channels such as audio and/or vision, and ignore the importance of social communication mediums like haptics. With the recent advances in technologies like HD and 3D video and high-quality audio, improvements in user experience that can be obtained by improving audio and video quality have reached their limits [43]. To make the video conferencing experience more realistic, many researchers are investigating the possibilities of integrating other sensory cues. such as touch or odor, to the early systems. In fact, many researchers are now focusing on merging haptics, which means the sense of touch, with traditional audio and video telecommunication systems to offer users a much more immersive telecommunication environment. Following are some related approaches.

### 2.2.1 Vibrating Tattoo

Nokia recently filed the patent of a vibrating tattoo alerts technology to the US Patent and Trademark Office [10]. By tattooing, stamping or spraying "ferromagnetic" material onto a user's skin, and pairing it with a mobile phone device, users could create a range of alerts with different vibrations, as shown in Figure 2.4. Applications of the vibrating tattoo alerts may contain low battery indication, received message, received call, calendar alert, change of profile, or any other. Also, the magnetised marking could be used as an identity check through picking a certain shape by the user. This application indicates that haptics is of practical use in our real life.



Figure 2.4: Vibrating tattoo alerts by Nokia

### 2.2.2 The Hug Shirt

The Hug shirt invented by CuteCircuit [6] (Figure 2.5) allows people to send hugs over distance. Sensors are embedded in the shirt to feel the strength of the touch, the skin warmth and the heartbeat rate of the sender, while actuators recreate the sense of touch, warmth and emotion of the hug to the shirt of the distant loved one. This invention has been awarded as one of the Best Inventions of the Year by Time Magazine. Since the hug data transmitted to a mobile phone can be delivered to the Hug Shirt seamlessly through Bluetooth technology, sending hugs is as easy as sending ordinary SMS. If you do not have such a Hug Shirt but know your friend has one, you can still send them a

### Literature Review



Figure 2.5: Hug Shirt from CuteCircuit

hug.

## 2.2.3 SMS Text Based Affective Haptic Application

Alam *et al.* [20] proposed a novel SMS Text Based Affective Haptic Application which automatically detects secondary level emotions from SMS text by applying a lexical rules based analysis and maps emotions to special vibrotactile haptic actions. Figure 2.6 shows the high level design of this proposed system. Through this application, a SMS receiver would feel affective haptic effect through his/her mobile phone or wearable haptic jacket before reading any SMS. Such preview of the SMS content's emotion would guide the receiver to select a message for reading or discarding at various contexts. Emotion detection from text is managed by the J2SE based Emotion Serve, while Mobile Affection Manager would control the affection processing. In addition, this emotion analysis model could detect up to 25 emotions from the text and map them to the corresponding haptic



Figure 2.6: SMS service high level design

vibrations. Thus, users could know the emotion contained in a message in advance, and then make a better decision about whether they are going to read the message.

## 2.2.4 Huggy Pajama

Tactile information was also implemented in the Huggy Pajama [55] which uses a mobile doll embedded with sensors as the medium to transfer the hug, thus further enhancing the user experience. Figure 2.7 shows the overview of this system. Using the Huggy Pajama, when users hug or touch the light doll, the touch sensors will measure the varying levels of the force applied on the doll interface and send this message to a mobile phone or computer through Bluetooth. On receiving the hug or the touch message, the remote



Figure 2.7: Overview of Huggy Pajama

person who is wearing the pajama would feel the touch or the hug. To increase the visual effect, a color change of the haptic pajama is used to convey information like the relative distance separating the parent and the child; in fact, cold colors are used to represent a far distance of the parent, while color changes gradually to a warmer and brighter one when parents return closer to the child.

## 2.2.5 Kiss Transmission Device

Another application of the haptic transmission is the Kiss Transmission system [8], invented by the Kajimoto Laboratory at the University of Electro-Communications. This system is used to transmit the feeling of a kiss effectively. The device they developed can well simulate the kiss sensation. As shown in Figure 2.8, when the user puts one device (the white device) in his/her mouth and turn it with the tongue, the other device (the black one) turns in the same way, so the partner's device turns whichever way the user's own device turns. Currently, the synchronous rotations of the devices are controlled by a PC, and the rotation angle information is sent reciprocally by both devices through a network to maintain the same position. To realize the remote communication, a system which can handle the values over a network is needed.

### Literature Review



Figure 2.8: Kiss transmission device

### 2.2.6 Interpersonal Haptic Communication in Second Life

Social physical interactions like handshake, hug, or tickle would stimulate emotions and cause an emotional reaction for people. Researchers from the Multimedia Communications Research Lab, University of Ottawa developed an add-on [37] that works with Second Life [13], a 3D virtual World, to offer physical communication functions to the users through a haptic jacket. The haptic jacket is embedded with a set of carefully placed vibrotactile actuators to provide haptics to users based on signals from the Second Life add-on. Signals are produced by annotating various parts of the Second Life avatar, and then they are used to actuate the corresponding actuators in the jacket. Figure 2.9 shows the block diagram of this system.

### 2.2.7 iFeel\_IM!

As many companies are offering remote online communications, combining immersive technologies is a new feature that is getting more attention. 3D virtual worlds like Second Life have embedded chat and IM features to encourage people to establish or strengthen interpersonal relationships, share ideas, and feel genuine emotions during their Virtual Reality (VR) adventures. However, the lack of visual emotional signals, the simple support of textual cues and the manual control of the expressiveness of graphical representations of the users greatly decrease the degree of immersion. Thus, iFeel\_IM [57] [56] was developed to augment emotions during online communications like the



Figure 2.9: Second Life and haptic communication system block diagram

Second Life. This system's architecture is shown in Figure 2.10. With this system, users' emotions would be governed by four additional haptic (tactile) channels, which include physiological changes (heart rate and body temperature), physical stimulation (tickling), social touch (hugging) and emotional haptic design (a device's shape, material or texture). Also, the detected emotions will be visualized through avatars in the virtual world to increase the visual effects. By adding extra visual and haptic features, iFeel\_IM significantly enhance communication in online virtual environments and social life.

## 2.2.8 HugMe

As mentioned in the first chapter, the HugMe system [30] [26] [27], which is an interpersonal audio-video-haptic synchronous teleconferencing system, shown in Figure 1.2, was developed by researchers from the University of Ottawa to add the touch information to communication systems. This implementation aimed at offering the users precise and intimate communication. It is compatible with a tolerable bandwidth of 30-60 Hz for haptic data, and with a 3-DOF haptic device (Figure 2.11) from Sensable Company [4]. The active user can "touch" the whole upper body even the back of the passive users. In fact, the active user is using the haptic device as the intermediate to touch the image proxy of the passive user and specify the touched body part. Also, this haptic device is used to provide the active user with the force feedback when a touch is detected. With markers detection method, mapping from built 2.5D human model to the real passive user is done in an accurate manner. This system is also one of the main motivation of



Figure 2.10: The architecture of iFeel\_IM!

this thesis.

The interaction diagram in the Hugme system is as follows: the active user "touches" the image of the passive user through a haptic device. Then this touch is transferred through the jacket to the passive user, while providing the active user with a force feedback through the haptic device.

In order to make the interaction in the Hugme system more fluid and natural, we decided to eliminate the need of using the haptic device or any other device (mouse for example) to interact with the receiver's proxy (image). In fact, we want the user to touch the image directly and transmit it at the appropriate position. Thus, we are making the interaction more realistic. When the active user is not able to touch the passive one, he or she could touch the image, and the passive user could still feel it. To reach this goal, we thus need a device which would enable us to detect when and what region of the screen/image are touched. Hence, we thought of integrating a touchscreen in the interaction diagram.



Figure 2.11: Haptic device from Sensable Company

## 2.3 Touchscreen Technologies

A touchscreen is an electronic visual display that can detect the presence and location of a touch within the display area. It generally refers to touching the display of the device using a finger or a hand. Touchscreens have two main attributes: one is that they enable a person to interact directly with what is displayed rather than indirectly doing it with a pointer (through a mouse or a touch pad), which will significantly improve the users' experience. The other is that they allow one to do so without requiring any intermediate devices that would need to be held in the hand. With those advantages, touchscreens have been widely applied in smartphones, tablet computers, portable video game consoles, and the medical field, the heavy industry, as well as in kiosks, where the keyboard and the mouse systems do not offer a suitably intuitive or rapid interaction.

Many technologies have been used to realize a touchscreen. In this section, we will explain some of those technologies, such as resistive, capacitive, infrared, webcam-based, and some other technologies. Also, we will explain our choice to develop a new solution.

### 2.3.1 Resistive

A resistive touchscreen panel consists of several layers, and among those layers, the two most important ones are thin, transparent electrically-resistive layers separated by a thin space. Figure 2.12 from Phares *et al.* [49] shows an exploded view of a resistive touchscreen. When an object or a finger presses down on the outer surface, the two



Figure 2.12: An exploded view of a resistive touchscreen

layers touch and is connected at that point. Then the touch position will be calculated from the obtained data.

The resistive touch screen has a high resistance to liquids and contaminants, and it has low cost, so it is widely used in restaurants. However, since users have to press down the screen, there is a risk of damage by sharp objects. Also, it suffers from poorer contrast because of the existence of additional reflections from the extra layer of material placed over the screen.

### 2.3.2 Capacitive

A capacitive touchscreen panel consists of an insulator such as glass, coated with a transparent conductor. Since human body is an electrical conductor, touching the surface of the screen will lead to a distortion of the screen's electrostatic field, measurable as a change in capacitance. Kable *et al.* [48] patented their capacitive apparatus with accuracy performance in 1986.

There are many great advantages of capacitive touchscreens. They are more re-



Figure 2.13: An illustration for touch events detection of infrared touchscreen

sponsive than resistive touchscreen and can support multi-touch which is important for gesture control, so many tablets PCs and capacitive smartphones adopt this technology to offer users more interesting and convenient experience with gesture control commands. However, capacitive touchscreens are much more expensive to manufacture and offer a much less degree of accuracy than resistive touchscreens. Moreover, they cannot be used with most types of electrically insulating material, such as gloves and writing pens which can be used on resistive touchscreens. This disadvantage especially affects usability in consumer electronics, such as touch tables and capacitive smartphones in cold weather.

## 2.3.3 Infrared

An infrared touchscreen uses an array of X-Y infrared LED and photodetector pairs laid around the edges of the screen to detect a disruption in the pattern of LED beams. Figure 2.13 from Xu [60] illustrates one kind of systems for detecting touch events by an optical touch screen where transmitters and receivers are positioned according to an

### Literature Review



Figure 2.14: User is browsing relics using Magic Crystal Ball

alternating scheme on each side of the touchscreen.

A major advantage of infrared touchscreens is that they can detect any input including fingers, gloves, stylus or pens, so users can choose whatever object they want to interact with these touchscreens. This kind of touchscreen also supports multi-touch and gesture control, but costs more money compared with capacitive touchscreen and resistive touchscreen.

### 2.3.4 Other Technologies

There are also many other less used touchscreen technologies such as optical imaging [38], dispersive signal [47], acoustic pulse recognition [46]. This thesis will not introduce them in detail since those technologies are not widely used.

Also, some innovative touchscreens are developed for specific applications. Chan *et al.* [28] developed an interactive visual display system named Magic Crystal Ball based on finger tip template matching to detect gestures. With this application, users can manipulate the displayed content inside the ball with bare hands as if they are using



Figure 2.15: The illustration of the dTouch pointing system

this Magic Crystal Ball to show scenarios of the past, the present, or the future in the magic world. By changing touchscreen into a particular shape (sphere), they merged the touchscreen with they application well.

### 2.3.5 The video-based touchscreen

Despite the fast development of touchscreen technologies, the majority of the computer screens used does not support the touch option. This is mainly due to the high cost of the already exposed technologies. In order to overcome this problem, many video-based approaches have been proposed. The principle of these methods is to detect the hand/finger in video feed during the interaction and estimate its position when touching a regular screen, and as a result simulate a touchscreen.

Cheng *et al.* [29] developed a novel 3D pointing system that allows users to interact with large displays by using a single webcam. Figure 2.15 shows the illustration of this system. With their system, users can use bare hand to point naturally, and the virtual touchscreen coincides with the large display as shown in the figure, which makes the interaction more direct. Also, the dTouch enables users to roam around the room and still be able to interact with the arm's reach as long as users are within the camera's field of view. The localization is based on the face and fingertip detection: they first use face


Figure 2.16: The view and segmentation result of Visual Screen

detection to find the dominant eye and determine the depth from the width of the face; then they detect the fingertip by considering it as the lowest skin color pixel from the view of the camera. Experimental results show that the dTouch technique is accurate, though the trial time is long when cursor feedback is needed.

Since fingertip is hard to be detected from a frontal view with segmentation techniques, users were asked to lower their fingertip to make sure that it would appear as the lowest skin color pixel. And users' body postures were restricted as slight deviation would increase the system's estimation error. The second problem is mainly caused by the inaccuracy of the face detection.

Zhang *et al.* [62] realized their Visual Screen by putting a webcam along or not too far from a line normal to the center of the monitor and using the color difference between the hand color and the screen color to segment the human hand. Figure 2.16 shows the image captured by their webcam and the hand segmentation result. Based on the detected hand, they were able to use finger features, such as finger tip shape, to find the finger tip location as the touch point. Visual Screen can also correct the non-flatness of the computer screen to get accurate testing results.

The disadvantage of this approach comes from the fact that the hand segmentation is simplistic and necessitates having an image of the screen background without the hand. Also, the hand detection will be erroneous if the screen displays colors similar to the flesh color. In addition, this method is very sensitive to any noise/object present in the camera field of view.

Hoang et al. [36] positioned one webcam at the same location as that of the Visual



Figure 2.17: Position of webcams of Hoang et al.

Screen to detect the finger touch point, while using another additional webcam WC1, shown in Figure 2.17, to detect if the finger is really touching the screen. If a touch event is detected by WC1, it will activate WC2, and WC2 will begin detecting the finger tip position through three stages: preprocessing with Color Edge Detection using the Gradient, intermediate processing with motion-based detection, and pattern recognition using finger tip template matching.

However, this method still suffers from the problem of occlusion and noise in the images. Also, it requires complex time consuming operations such as the fingers pattern recognition.

Though Cam-based touchscreens are less accurate than other touchscreens, they can transform most ordinary monitors into touchscreens at a rather low cost with computer vision techniques. The loss in accuracy is highly compensated by the low cost of the technologies used, providing such a big advantage over the conventional touchscreens. However, the two webcam-based touchscreen methods with regular monitor mentioned above both have large restrictions on the user's sitting position, posture, and moves due to the cam position. For example, if a tired user puts an arm on the desk leaning



Figure 2.18: Improved hand contour and ratio curve of original contour

his/her head on his/her hand, and uses another hand to interact with the touchscreen, the system may fail because the user's body could block the view of the cam. Their methods also rely heavily on color segmentation, which can be greatly affected by factors such as illumination conditions, the user skin color, and background changes. Also, the Visual Screen system may take any hand detected as a touch event even if the hand is far from the monitor, while Hoang's method may not work well with certain finger tips or nails since they are using their own template database for matching. Given that, we decided to develop our own approach for transforming the screen into touchscreen using an inexpensive cam. We also need to make our system robust to noise and changing lighting conditions.

### 2.4 Finger Tip Detection Methods

To transform a regular monitor into a touchscreen using a webcam, we must accurately detect the user's finger tip in the image, which is always considered as the indicator of the touch position. The first step in estimating the position of a finger tip on the screen using video is to detect and isolate the finger in the image. Many approaches have been developed to solve this problem. Most of these techniques rely on the shape or color of the hands and fingers. We summarize some commonly used finger tip detection methods in the following section.

#### 2.4.1 Contour and color based approaches

Since human hands have distinct colors and shapes, many techniques used these features to segment the hands from the background, and then use these detected hands contours to find the finger tips based on their special curvatures. Figure 2.18 shows a hand contour with two fingers after Fourier Transform and the ratio curve of the hand contour from Wang *et al.* [59]. From this figure, we see that the finger tips always have the highest ratio curve in the hand contour which could be used for finger tips extraction.

Though many algorithms distinguish finger tips in the same way, they apply different methods to detect the hands.

Some researchers use the original hand color ranges to find it. Kang *et al.* [40] assume that the skin color vary in certain ranges, and they indicate that red component of the skin color is in the range of 37 to 60, whereas the green component of the skin color is between 28 to 34. Then they use **OpenCV** to perform a faster RGB to YUV conversion in order to make the color ranges more robust to changes in brightness and intensity. They also use morphological operation to eliminate palm before using the contours to find the fingers to minimize the effect of the palm. Lee *et al.* [42] use a webcam to detect a control motion first, and then zoom in the camera, when a control signal is given, to recognize the exact control gesture using the original skin color information, which is about relationships among finger's red, green and blue colors; Researchers of Visual Screen [62], described earlier, observe in their experiments that the images of screen pixels have some degrees of invariance in the color space even if contents change frequently, so they use the found dominating blue color to differentiate hand from the background.

Though a hand may have certain color ranges, its colors could change according to the surrounding light conditions or the color reflection of nearby objects. Thus, Liang-Guo Zhang *et al.* [61] advanced a sign language recognition system based on glove colors, as shown in Figure 2.19. Their left-hand glove is purple, while each finger of the right-hand glove has a specific color (red, yellow, orange, blue and green). First, they locate the reference point using the pupils detection. Then they use color difference to find hands and fingers. Since fingers of the right hand have different colors in this case, they can easily distinguish the position of each finger, and then calculate the exact finger postures including finger tips. As the glove colors are much more constant than natural finger colors, they are reliable features to detect. This method has a detection accuracy rate of 91%, but wearing gloves may affect the user's freedom.



Figure 2.19: Glove-based sign language by Liang et al.

Given the uncertainty of the skin color under different lighting conditions, other features were used to detect a hand in the image. The hand contour features can also be used to distuiguish hands from other objects. Anagnostopoulos *et al.* [21] find all objects contours in the image captured by a webcam, and use hand features like dimensions of the object area and its shape to wipe out non-hand objects. Then they trace the hands contour curve, identify a set of dominant curve points and use k-curvature, which is mainly used to detect the dominant points on curves, to decide which points are finger tips. Since their system just distinguishes two types of real objects: hands and all the rest, their application supports multi-hands detection.

#### 2.4.2 Pattern-based

Since finger or hand shapes remain the same in different configuration or various scenes, some researchers realized hand detection using pattern matching with pre-defined tem-



Figure 2.20: Finger tips detection process of Magic Crystal Ball

plates.

The Magic Crystal Ball [28] exposed earlier uses a finger tip template matching approach to verify the finger tip candidates which have passed the filtering step and remove the false matching candidates, as shown in Figure 2.20. In their process, candidates with low scores from the matching evaluation are discarded, and for false matching removal, if the pixels in the diagonal direction on the boundary of the finger tip patch coexist, then it is not considered as the finger tip and is removed; Hoang *et al.* [36] also use a  $60 \times 80$  templates with finger-like shapes as the specific patterns to match with sub-images in the images captured from a webcam. Then the sub-image with the highest correlation coefficient among all the sub-images is considered as the finger tip.

Sometimes the templates learning database base may not contain templates representing all the possible configurations. Thus, an initialization and an extensive learning phase of the system is necessary for each user. Gorodnichy *et al.* [34] get the hand tem-



Figure 2.21: Foreground detection and hand template tracking results from Gorodnichy  $et \ al.$ 

plates through ground subtraction in the initialization phase, and then use deformable templates to track the hand which could be highly deformable (Figure 2.21). The foreground detection extracts the blobs corresponding to the hand images (left column), while the hand template tracking allows to detect partially occluded hands (right column).

#### 2.4.3 Marker-based approaches

To realize precise detection results of finger gestures, markers can be attached to the fingers for tracking. Kavakli *et al.* [41] uses four optical sensors (cameras) to track up to eight infrared LED markers. The three-dimensional location of each marker is calculated from the sensor input and is transmitted to a server. Once the tracking data is received and interpreted, the server sends the gathered data to the rendering machine to realize functions such as 3D drawing or tracking a person. Figure 2.22 illustrates the 3D drawing result with this application. The use of cameras and markers makes this



Figure 2.22: 3D drawing with LED markers

approach expensive and constraining to the user.

In order to avoid using complex pattern recognition techniques and color matching, we opted to take advantage of the camera positioning to find a generic and simple approach for finger detection. This technique will be explained in Chapter 3. no learning nor database is needed as we do not use any shape or color matching.

Once the finger tip is detected, another challenge would be estimating its position. To achieve this goal, the most common technique would be to use a stereo-vision technique. Stereo-vision is mainly used to realize the 3D reconstruction using two or more cameras. From a general point of view, the 3D reconstruction problem is mathematically determined only if enough information about the sensing subsystem is available in terms of optical/intrinsic parameters and geometrical/extrinsic parameters [22]. With a 3D model of the finger, we can find its location in the image easily.

Since we are using a single webcam, we will use the geometry of the scene to calculate

the finger's touching position on the screen. Details of the proposed approach are given in the next Chapter.

# Chapter 3

# The Proposed Haptic Video Chat System

In this chapter, a detailed description of the proposed system is given. In fact, we describe our proposed webcam-based touchscreen method and how we associate it with the haptic jacket. We start by introducing the webcam position we adopted and the advantages of this setting. Then we explain the approach we used to detect the finger in the image as well as the mathematical models enabling the estimation of the vertical and horizontal touch position. Details of the initialization and calibration phase are also exposed. Figure 3.1 shows the flowchart of our proposed method, and we will explain each step in the following sections. At the end, we expose the haptic jacket used and give details about how the corresponding haptic feedback was realised through the proposed touchscreen.

### 3.1 Webcam Position

In comparison to other webcam-based touchscreen methods, our method requires just a single webcam. In order to calculate the touch point between the finger and the monitor, we need to detect the finger from the image captured by the webcam. To make this complex task easier, we opt to position our webcam in a way to have its image plan perpendicular to the screen plan. Given that configuration, the only object (in the image) intersecting the screen's bottom border during the finger-screen's interaction would just be the finger itself. The finger detection in the image is then reduced to detecting the object intersecting the screen border. Figure 3.2 indicates the position and field of view



Figure 3.1: Flowchart of our proposed method

of the webcam which is located on top of the monitor and looking down at the desk. Thus, we can capture the finger length in the image and use it to calculate the touch position. From the view of the camera, we can see that only a little part of the desk is within the camera's field of view, while other surrounding environment objects will not be captured. As a result, the large ambient changes in the scene (noise, user movement, *etc.*) would not affect the detection results or make the detection more difficult. This webcam positioning makes the system less sensitive to surrounding environments and background changes, which significantly improves our system's robustness and precision. Moreover, our method has less restrictions on the user's sitting position and posture, since most of the time the user is out of the field of view of the webcam.



Figure 3.2: Webcam position and its field of view

# 3.2 Initialization and Calibration Phase

Our goal is to accomplish an accurate estimation of the contact point position between the finger and the screen, so an initialization and calibration phase is necessary for further calculation. During this phase, we will detect the reference line (the screen's bottom border) used for the finger detection, and collect data from two reference positions that are necessary for further calculations.



Figure 3.3: Detection result of the screen bottom border

#### 3.2.1 The screen's bottom border detection

The first step in our finger detection algorithm is detecting the screen's bottom border. Given the webcam position, at first we characterize this border by the fact that it occupies the longest horizontal line in the image. However, long lines in the image are always separated into many short lines in most cases because of the image distortion, the lighting condition, the camera resolution, and some other factors. To solve this problem, we use the function CVLINE provided by OpenCV to find all the lines in the image, and then calculate their angles to the horizontal level to extract all horizontal lines (to make our system robust, a certain threshold is allowed). After detecting all the horizontal lines, we connect the lines with close horizontal levels together to calculate each level's whole length (each level's overlapped length will just be calculated once, which means if the image's resolution is  $1280 \times 720$ , the greatest possible length is 1280). In Figure 3.3, all the white fine lines are qualified detected lines, while the blue bold line shows the bottom border detection result.

However, sometimes this method is not accurate since some other lines, such as the whole monitor's bottom line and its mirror line on the screen, which are shown by white fine lines in Figure 3.3, also have the chance to become the longest horizontal line in the



Figure 3.4: Screen background setting

image. Thus, we use additional constraints to improve the detection result. As shown in Figure 3.4, we add a black triangle into the background and fix one of its vertices on the screen's bottom line. With the distinct color difference, we can easily detect the two edges of this triangle from the image, as shown in Figure 3.5. By adding conditions such as line length and inclination angle, we can easily find the black triangle's two edges from all the detected lines, then we can calculate the intersecting point of those two lines, as shown in Figure 3.6, the white solid circle is the point of intersection. We consider this point as one point of the screen's bottom line, and then use the same horizontal lines detection method mentioned before to find the long line passing this point, and we consider it as the screen's bottom edge. Actually, in order to make the detection results more robust, we move the bottom edge a little up in case it is right on the display screen and affected by the image background changes.

As mentioned above, when the finger touches the screen, it will be the only object intersecting this detected bottom border line in the camera's field of view. Thus, based on the detected bottom border line, we use its pixels' color difference and a pseudo gradient descent technique to detect the finger. In our application, we extract the red



Figure 3.5: All detected long lines in the image



Figure 3.6: Triangle edges and their point of intersection



Figure 3.7: Finger detection result at the first pre-set position

color value from the RGB color components of the screen's bottom border line detected above, as well as its upper and lower horizontal levels since we consider that this bottom edge line can be represented by those 3 lines. As a result, each point on the bottom edge will be represented by 3 pixels having 3 successive vertical positions. We then calculate the average red value of each point(3 pixels), divide those points in groups of 10, and calculate each group's average red value. For each vertical separation line between two of these groups, we estimate the red value averages' difference between its left and right group. The two lines with the biggest difference values are considered as the initial lines separating the screen and the finger.

Since each group contains 10 pixels, the error of the finger edge position should be within 10 pixels. In order to minimize this error, we apply the same technique on the 30 points interval surrounding the two detected initial lines separately, but with a smaller group division (3 points each group). Then we can find the more accurate finger edge positions. The reason we use larger groups to find the initial lines first is to reduce the detection time, so our system could respond faster in real-time application. The two vertical blue lines in Figure 3.7 shows the detected finger edges.

The reason we choose the red component as the criterion is that the red components of the finger and the monitor's bottom edge have large difference. To avoid some small



Figure 3.8: The two pre-set positions (white dots)

dots, which may be present on the finger or the monitor's bottom edge, to cause sudden color change, we process the image with erosion, dilation, and smoothing functions before any detection. Also, the distance between the two found lines should be larger than 20 pixels as the reference object in the image should not be very small.

#### 3.2.2 Two reference images with pre-set positions

As we are aiming to develop a generic system that can be adapted to any webcam and situation, and estimation of the used camera's position and focal length is necessary. This estimation is based on a simple camera model (no distortion or optical center displacement).

In this step, we try to acquire some reference images with the user's finger pointing to two well known positions in the screen separately. In our system, one pre-set position is the screen's horizontal midpoint at 10cm deep from the screen's top border when the screen's height is 30cm; the other position is 3cm (101 pixels when the screen's resolution is  $1280 \times 1024$ ) to the right from the screen's vertical central line at 22cm deep from the top border. Figure 3.8 shows these two pre-set point's positions (the white dots) in the screen. Actually, these two points will not be shown at the same time, we just put them together for easy explanation. For each step, a white dot shows the indicated pointing



Figure 3.9: User's finger is pointing the first indicated position (the white dot)

position in a full-screen black window. Then the user can point and touch the white dot while keeping his/her finger normal to the screen. Figure 3.9 shows the user is touching the first pre-set point (10cm deep from the screen's top border), then we repeat the same action with the second pre-set point (22 cm deep from the screen's top border).

Distinguishing the finger edges from the monitor's bottom line is realized using the detection method explained in the previous paragraph. After detecting the exact finger lines from the screen's bottom edge, we can easily locate the central part of the finger at this line (from 1/4 to 3/4 of the finger's width). Then we extract these central part pixels' color components and calculate their average value; these color values will be used as the standard to find all the pixels, whose color components differences with the standard value are within certain thresholds (RGB color components' possible values are from 0 to



Figure 3.10: Finger detection result at 22cm

255, while Hue color component varies from 0 to 179), and are located between the two vertical blue lines (Figure 3.7 and 3.10). Pixels satisfying these conditions are considered as the representation of the finger. At first, we were using only RGB color model to detect the qualified "finger" pixels, but later we found that the RGB model is too sensitive to light change, especially when finger is so close to the screen in our case. Thus, we also use the HSV color space for the detection. The HSV stands for the Hue, Saturation and Value. It offers the perception representation according to the human visual feature [54], so it is widely used for human hand or face detection. For each color component, we use different color thresholds to determine the colors' differences conditions when detecting the whole finger. Given our test, setting the following thresholds gave the best result: the red threshold is 50, the blue threshold is 70, the green threshold is 70, while the hue threshold is relatively low as 25. Pixels with all these four qualified color components and located between two detected finger lines will be considered as finger pixels and be flagged with a red color. The red parts in Figure 3.7 and Figure 3.10 show the finger detection result at 10cm and 22cm deep separately. Though some parts of the finger are not detected in Figure 3.7 due to the light reflection and the finger's intrinsic color difference, they are not going to affect our detection accuracy, since in our mathematical models, only the detected finger length (vertical direction) is used, while the finger width



Figure 3.11: Pinhole camera model

(horizontal direction) is just used to find a more accurate finger length. This also shows the robustness of our system.

With those steps, we have finished the initialization and calibration phase, and the data collected from those steps are used in the mathematical model we proposed and the equations exposed in the next section.

## **3.3** Realtime Touch Position Calculation

The realtime finger detection uses the same finger detection method mentioned in initialization and calibration phase, and the color difference thresholds we adopt are still the same. After we finish detecting the finger in a image, we can get the current finger length l in the image, and then calculate the finger's touch point's horizontal position  $F_h$ and vertical position  $F_v$  on the screen with our mathematical models. Once we get  $F_h$ and  $F_v$ , we have acquired the touch position. As a result, we realize a touchscreen with a webcam. Our calculation is based on the camera pinhole model.



Figure 3.12: The geometry of a pinhole camera seen from another axis

#### 3.3.1 The pinhole camera model

The mathematical models we adopt to calculate the finger tip's both vertical and horizontal positions are based on the pinhole camera model, as well as the Tsai camera calibration model [44]. In this part, we give a detailed description of the pinhole camera model.

The pinhole camera model illustrates the mathematical relationship between the coordinates of a 3D point and its projection on the image plane of an ideal pinhole camera, where the camera aperture is described as a point and no lenses are used to focus light. Figure 3.11 and 3.12 from WIKIPEDIA [12] are a diagram of a pinhole camera and the geometry of a pinhole camera seen from another axis separately. The pinhole aperture of the camera, referred to as the optical center, allows the projection lines to pass and form images on the image plane.

One concept closely related to the pinhole camera model is the similar triangles principle. If two geometrical objects have the same shapes, they are called similar. In Figure 3.12,  $\triangle OAQ$  and  $\triangle OBP$  are similar triangles. Thus, we can state that



Figure 3.13: Combination of 10cm and 22cm calibrations and the mathematical model

$$OA/OB = OQ/OP = AQ/BP$$

and the altitudes of two similar triangles have the same proportion as that of these corresponding sides. In this case, sides OA, OB, AQ, and BP are also altitudes of the triangles.

#### 3.3.2 Realtime finger's vertical position calculation

The finger's vertical position (depth to the screen's top border) calculation is mainly based on the 10cm and 22cm reference images and the pinhole model of the camera. Actually, given any two configurations of the finger touching the screen at any two points at different heights, we can establish the following mathematical model. Figure 3.13 shows the picture of these two calibrations' combination, and indicates the mathematical model we adopt. Though for the camera, the 10cm and 22cm calibrations are not on the same plan, we project them on the same plan to build the model, as neither the finger length nor any vertical distance will change. In fact, in the real application we are not going to touch those two points at the same time; we combine them together in one image just to explain the mathematical model better. In this figure, we assume that the red virtual hand is pointing at a point 10cm deep from the top edge of the screen, and the real hand is pointing at a point 22cm deep from the top edge of the screen.

In the mathematical model,

- The point O refers to the optical centre of the camera, and point E is point O's perpendicular projection on the screen. On the screen, we draw a line through E and we make it perpendicular to the webcam's image plan. Since we assume the webcam's image plan is perpendicular to the screen plan, this line will also be perpendicular to the screen's top and bottom border at the same time. F marks the intersection point of this line and the screen's top border, while K is the point with the screen's bottom edge, so FK refers to the monitor's vertical line whose length is equal to the screen height  $m_h$  (30cm in our case). EF is then the vertical distance d between the camera's optical centre and the screen's top edge.;
- The line AD represents a line on the camera's image plan, which is used to form the images captured through the optical centre, and OA is the vertical distance between the camera's image plan and the camera's optical centre;
- G and I are the finger's touch positions on the screen. If we assume they are the 10cm and 22cm calibration settings, we can get that FG=10cm  $(h_1)$  and FI=22cm  $(h_2)$ . Actually, if the webcam is not located right on the middle point of the screen, we can project the touch positions to this line, and the detection results will not change;
- OK will then be the line relying the optical centre O to the screen's bottom border point K, and thus will intersect the finger in different positions (H, J) depending on their distance (depth) to the camera. GH and IJ refer to the 3D portions (of lengths l'<sub>1</sub> l'<sub>2</sub>) of the detected touching finger length, and they are represented in the camera's image plan by BD and BC (l<sub>1</sub> and l<sub>2</sub>). (B is the image of K, J, H, while D is the image of G, and C is the image of I.)

Given the webcam position, we assume that the line OA is perpendicular to both AD and EO, and EK is perpendicular to EO. Thus, from the similar triangles  $\triangle OGH$  and  $\triangle ODB$ , we can state that

$$GH/DB = (FG - EF)/OA \tag{3.1}$$

Same for the similar triangles  $\triangle OIJ$  and  $\triangle OCB$ 

$$IJ/CB = (FI - EF)/OA \tag{3.2}$$

The relationship between IJ and GH can be derived from similar triangles  $\triangle KIJ$ and  $\triangle KGH$ 

$$IJ/GH = (KF - FI)/(KF - FG)$$
(3.3)

If we stop considering IF as the 22cm deep calibration setting, but assume that it is a random height, namely if the finger's vertical position  $F_v$  (IF) is unknown, we can use equations (3.1),(3.2),(3.3), and BC which then refers to the detected finger length l(pixels) in image, to calculate IF

$$IF = \frac{BD * FK * (FG - EF) + EF * BC * (FK - FG)}{BC * FK - BC * FG - BD * EF + BD * FG}$$
(3.4)

namely

$$F_v = \frac{l_1 * m_h * (h_1 - d) + d * l * (m_h - h_1)}{l * m_h - l * h_1 - l_1 * d + l_1 * h_1}$$
(3.5)

In equation (3.4) and (3.5), the only value that remains unknown is EF(d), the vertical distance between the screen's top border and webcam's optical centre. So we need to use both the 10cm and 22cm reference images to calculate EF, in other words, when FG=10cm, IF=22cm, and BC is  $l_2$ . From equations (3.1),(3.2) and (3.3), we can get

$$EF = \frac{FK * FI * BC - FG * FI * BC - FK * FG * BD + FG * FI * BD}{FI * BD + FK * BC - FK * BD - FG * BC}$$
(3.6)

namely

$$d = \frac{m_h * h_2 * l_2 - h_1 * h_2 * l_2 - m_h * h_1 * l_1 + h_1 * h_2 * l_1}{h_2 * l_1 + m_h * l_2 - m_h * l_1 - h_1 * l_2}$$
(3.7)

Thus, once we have the distance EF(d), then for the realtime detection we just need the data from one of the two reference images to calculate any realtime finger touch's vertical position with equation (3.4) or (3.5).



Figure 3.14: Mathematical model for vertical position calculation in another situation

If the camera's optical center is higher than the top of the screen, the mathematical model would be slightly modified (Figure 3.14). The equations (3.1) and (3.2) would change into the following equations separately:

$$GH/DB = (EF + FG)/OA \tag{3.8}$$

$$IJ/CB = (EF + FI)/OA \tag{3.9}$$

Also, the equation for calculating EF would be:

$$EF = \frac{FK * FI * BC - FG * FI * BC - FK * FG * BD + FG * FI * BD}{FK * BD - FI * BD - FK * BC + FG * BC}$$
(3.10)

And the equation to measure the touch point's vertical distance to the top of the screen IF changes into

$$IF = \frac{BD * FK * (EF + FG) - EF * BC * (FK - FG)}{BC * FK - BC * FG + BD * EF + BD * FG}$$
(3.11)

Thus, even if the webcam's optical centre is higher than the screen's top border, we can still get the vertical touch position with this second mathematical model.

Actually, equations for either mathematical model would suit any of these two situations. In fact, we can use the first set of equations (3.1),(3.2),(3.4),(3.6) on the second one, which is using equations (3.8),(3.9),(3.10),(3.11), to calculate the position in any camera configuration (below or above the screen top): if we are in the first camera configuration and we use the second set of equations, EF (equation (3.10)) will be negative and the calculated vertical touch position IF (3.11) will thus be equivalent to the one in the first set. By symmetry, EF (equation (3.6)) will also be negative if we are in the second camera configuration and we use the first set of equations, and IF (3.4) is equivalent to the one in the second set.

#### 3.3.3 The realtime finger's horizontal position calculation

The finger's horizontal position calculation requires information about the two calibration positions, the screen's resolution, the calculated distance d from the camera's optical centre to the screen's top border, and the calculated finger's vertical position  $F_v$ . With those values, we will be able to calculate the touch point's horizontal position. This calculation is done as follows.

Figure 3.15 shows both the real scene and our proposed mathematical model for the horizontal touch position calculation. In this model,

- Same as in the mathematical model we use in Figure 3.13, F marks a point on the screen's top border, K is a point of the screen's bottom edge, and FK refers to the monitor's vertical line whose length is equal to the screen height  $m_h$  (30cm in our case). The webcam's optical centre O's perpendicular projection on the screen is E, so EF is the same value as we get from equation (3.6);
- The line  $M_1N_1$  refers to the camera's image plan's horizontal projection line on the screen (this line  $M_1N_1$  is perpendicular to the image plan's line AD in Figure 3.13).  $A_1$  is point E's perpendicular projection on  $M_1N_1$ . The reason we do this projection is to put all related points and lines on the same plan (the screen plan) for an easier mathematical model building, and this projection will not affect the object's width or its horizontal position in the camera's horizontal image plan;



Figure 3.15: Mathematical model for horizontal position calculation

• The line GU is the vertical central line of the screen, and the Point G is the first reference touch point, which is the screen's horizontal midpoint at 10cm deep from the screen's top border. I is the second reference point, which is the 3cm to the right of GU at 22cm deep from the top border. Line EK is the line passing by the camera's optical centre's projection E, and is perpendicular to  $M_1N_1$ . L and U are I's projections on line EK and GU respectively, and P is G's projection on line EK.

In our model, we assume the camera's image plan is perpendicular to the screen plan, which means GU is parallel to EK. If the optical centre of the camera is right in the middle of the screen, GU and EK will be on the same line. However, we cannot guarantee this, so each time we calculate U's position in the image plan  $R_1$  from the reference point G's position in the image plan  $S_1$ .

For the similar triangles  $\triangle EGP$  and  $\triangle ES_1A_1$ , we know

$$GP/S_1A_1 = (FP - EF)/EA_1$$
 (3.12)

Same for the similar triangles  $\triangle EUL$  and  $\triangle ER_1A_1$ 

$$UL/R_1A_1 = (FL - EF)/EA_1 (3.13)$$

Since GU is parallel to EK, we can get

$$GP = UL \tag{3.14}$$

If we assume I is a random touch point at an unknown depth instead of the second reference touch point, then UI will be unknown, but we can still get its height  $F_v$  (FL in this figure) from the vertical position calculation model. Since U is the screen's midpoint at depth  $F_v$  (FL), if we know UI, we have got I's horizontal position. Thus, our goal is to calculate the distance between U and I.

From similar triangles  $\triangle EUI$  and  $\triangle ER_1D_1$ , we know

$$UI/R_1D_1 = (FL - EF)/EA_1 (3.15)$$

In this equation,  $R_1D_1$  is unknown. However, we know

$$R_1 D_1 = R_1 A_1 + A_1 D_1 \tag{3.16}$$

As  $D_1$  is the random touch point *I*'s position in the camera's image plan, and  $A_1$  is the midpoint of this image plan, we can get  $A_1D_1$  easily. Also from equations (3.12),(3.13), and (3.14), we can get  $R_1A_1$ 

$$R_1 A_1 = \frac{FP - EF}{FL - EF} * S_1 A_1 \tag{3.17}$$

Then to calculate UI from equation 3.15, the only value remains unknown is  $EA_1$ . Because  $EA_1$  is a constant value, once we know it, we will not need to recalculate it any more. Assume I is not a random touch point but the second reference point, then UIwill be 3cm, and we can get the value of  $EA_1$  from equation 3.15. Knowing  $EA_1$ , we can calculate any random touch point I's horizontal position UI with

$$UI = (R_1A_1 + A_1D_1)(FL - EF)/EA_1$$
(3.18)

Thus, we have got the touch point's horizontal position  $F_h$  on the screen.

Same as in the mathematical models for the vertical position calculation, if E is higher than F, as shown in Figure 3.16, then EF would be negative to suit this situation. In fact, if we use the first set of equations for vertical and horizontal position calculation, it will suit both situations; or we can use the second set of equations for vertical position calculation and change EF to negative for the horizontal position calculation, we will still be able to get the correct position from both situations.

With the vertical and horizontal positions  $(F_v \text{ and } F_h)$  of the touch position, and the screen's resolution, we can transfer the touch point's position onto the screen precisely, and could react to the corresponding actions. So far, we have successfully changed an ordinary monitor into a novel touchscreen with a webcam.

### **3.4** More Filtering for Finger Detection

To obtain more accurate finger detection results, we also adopt some strategies to improve the accuracy of our detection algorithm. In our method, when we search for the whole finger between the two detected finger edge lines, using only the color threshold we may falsely make "non-finger" pixels qualified as finger, such as the mirrored finger pixels produced by reflection on the screen (Figure 3.17) or the pixels with similar colors in the background (Figure 3.18). In order to solve this problem, we add two filtering steps to our finger detection algorithm.



Figure 3.16: Mathematical model for horizontal position calculation in another situation

#### 3.4.1 The continuity property

Since finger itself is a whole part, we set the requirements about the continuity when looking for the qualified pixels as follows: when we detect the finger from the image, we start from the screen's bottom border, then search downwards to find more finger parts. to explain our filtering better, we rank the screen's bottom border as the 1st line, than horizontal lines below it will be ranked as 2nd, 3rd... line in order. After we find the intersection part between the finger and the border, for the first 3 horizontal lines we just use the color conditions mentioned before to select the qualified pixels between the two detected finger edge lines. From the 4th line downwards, only when most pixels (more that 2/3 of all pixels), located between the most to the left and the most to the right pixels which satisfy our color requirements on a given line, are connected with some



Figure 3.17: Some pixels representing the mirror finger are mistaken as belonging to the real finger given their color features

qualified pixels on the previous line (such as qualified pixels on the 3rd line for the 4th line detection), we consider the qualified pixels on this line as finger parts; otherwise, we will take this line as the finger tip and stop searching. Thus, the problem is solved by adding this additional continuity requirement.

We set this requirement as we consider that the finger is a plain object with a uniform color. Thus, all pixels belonging to it should have similar color and should occupy most of the pixels between its border.

#### 3.4.2 The shape property

Despite the first filtering step, some non-finger pixels (such as the mirrored finger pixels) are so close to the real finger pixels that they all pass the first step. Thus, we need to add



Figure 3.18: Our strategy works well to stop the large increase in finger width

another step based on the finger's shape to further increase the accuracy of our detection result.

Given the shape of the finger, when going from the 1st horizontal line to the finger tip, there should not be large increase of the finger width in most cases, and actually the width would decrease gradually. Thus, we also restrict the finger width changes when detecting the whole finger to rule out non-finger elements. When scanning the "finger" from the 1st line to the bottom of the image with color and continuity condition, we disqualify all the pixels belonging to any horizontal level with sudden large finger width increase. Those disqualified pixels may represent the mirror finger on the screen or pixels with similar colors in the background. Figure 3.18 shows that the finger detection scanning process is stopped when a large increase of the finger width happens below finger tip. In this situation, the touch point would be considered as the midpoint of the level containing the last qualified group of pixels.

Using this strategy, we successfully eliminate some false detections based on the finger's shape feature.



Figure 3.19: Logo is mistaken as human face

# 3.5 Combination of the Touchscreen and the Haptic Jacket

In the *HugMe* system, the remote user is tracked through fiducial markers on the haptic jacket, shown in Figure 1.3 in Chapter 1. With these markers, the system can accurately recognize the remote user and map the touch position to the corresponding sensors embedded inside the jacket. Thus, the passive user would feel where the active user is touching. In our application, we need to accomplish the mapping process in a different manner. As we want to make the system usable by as many users as we can and put less restrictions on the users, we are not using clothes with specific colors or feature markers to locate the human in the image. In fact, we need to detect the body part touched from the image and send the command to activate the corresponding actuators.



Figure 3.20: Face detection and upper body dividing result

After several tests, we opted to use the face detection method which is highly reliable for human detection. Actually, we used the haarcascade model in OpenCV, which works well for frontal face detection. Haar classifier is an object detection application based on a boosted cascade of simple features, it works well for rigid objects and characteristic views [58]. The OpenCV distribution contains a trained frontal face detector which works well. Though occasionally there are some errors, such as graphs or logos on the clothes are mistaken as human faces, or the active user's face which is usually shown at the corner of the chatting window is also detected, as shown in Figure 3.19, we can easily solve those two problems by limiting face size and location.

Though we can detect the face easily and accurately using the haarcascade model, detecting the exact upper body parts locations including the shoulders and the arms is difficult in the content of our application. In fact, people may video-chat with their friends or relatives in whatever clothes or environments they like. Thus, the light conditions may vary, and the clothes they are wearing could be confused with the background. This problem especially happens in images with low resolution. However, for our application we do not require extremely accurate upper-body parts detection since we cannot embed too many vibrators inside the clothes; an approximate touch location would be sufficient for emotion conveyance of ordinary users.

The body parts detection method we use is based on the detected face and the Vitruvian Man model [17] (sometimes called the Canon of Proportions). The Vitruvian Man is a world-renowned drawing created by Leonardo da Vinci, based on the correlations of ideal human proportions along with the geometry described by ancient Roman architect Vitruvius. According to Leonardo's text accompanying this drawing: "from the hairline to the bottom of the chin is one-tenth of the height of a man" and "the maximum width of the shoulders is a quarter of the height of a man". If we consider the face length we detect as the distance from the hairline to the bottom of the chin, we can estimate that the maximum width of the shoulders is 2.5 times the face length. Thus we consider the area below the detected face, having a width of 2.5 the height of the face and having the same central line as the box containing the face, as the upper body; it covers the height going from the bottom of the image to the bottom of the face. We also divide this area into equal area contiguous squares, by fitting 5 squares per row. Each of the squares would represent a body part (shoulder, torso, *etc.*), and will be associated to a group of actuators inside the haptic jacket. The detection and division result is shown in Figure 3.20. In this figure, face detection result is shown with yellow box, while the upper body parts locating results are displayed with red boxes and numbers. As we can see, this type of upper body detection and division method could offer an acceptable approximation, though it is not extremely accurate. It actually satisfies our goal of transmitting emotion through touchscreen and haptic jacket.

After realizing the human body detection, once a touch is detected in a specific square (box), our system will show the touch region, and inform the control centre of the haptic jacket to activate the corresponding vibrators to produce the sense of touch.

As for the touch position transmission, our idea is to send the box number from the active side to the passive side through the text chat messaging system. On receiving the number, the passive side would send this information to the haptic jacket through a wireless technology, such as Bluetooth.

Therefore, we successfully combined our innovative touchscreen with the haptic jacket to offer user a much better haptic telecommunication experience.

# Chapter 4

# Implementation and Results

Experiments were conducted to test the performance of our proposed system; in this chapter we mainly give details about the implementation, as well as the testing results of our touchscreen approach. In order to measure the validity and accuracy of our finger detection and position estimation technique, we test the system by different users using different screens and cameras (with different resolution). We also test different screen contents during the interaction (black screens, webcam images, *etc.*). Furthermore, we show some qualitative results while using the system for some additional functions such as drawing or writing. On the other hand, we also detail the outcome of the complete system (delivering the touch sensation at the appropriate position)

## 4.1 Hardware and Software Configuration

The implementation of our touchscreen approach was done with the following hardware and software configuration:

- **PC configuration:** The PC we use has an Intel(R) Core(TM) i7-2760QM CPU 2.4GHz with Window 7 64-bit Operating System, and Ram of memory is 8GB.
- Screens: The proposed methods are tested on two screens: the first is an IBM Thinkvision 19.0-inch 9419 HB7 monitor, and the other is a Dell 19.3-inch 1905FP monitor. Both monitors' resolutions are 1280×1024, and their surrounding light conditions are different.
- Webcam: We choose the Microsoft LifeCam HD-5001 [9], shown in Figure 2.2, as the webcam to capture images. We select this webcam because its structure makes
it easy to be positioned on top of the monitor and this webcam offers up to 30 high-resolution  $(1080 \times 720 \text{ pixels})$  frames per second. Moreover, it allows setting its parameters, such as the focus, the brightness, the white balance, the saturation, the exposure and the contrast. The auto-focus function is disabled during our test, because when the webcam adjusts its focus parameter automatically, there will be a few blur images during the process. Following are all the parameters values we used: focus is 11, brightness is 133, contrast is 5, saturation is 83, exposure is -15, and white balance setting is automatic. Those values give the best quality images we are looking for.

- Haptic Jacket: The haptic jacket we use is developed by the Multimedia Communications Research Lab, University of Ottawa. As shown in Figure 1.3 in Chapter 1, the jacket is embedded with light weighted (1.2 grams) and small (10.0mm diameter × 3.4mm thick) vibrotactile actuators, which are controlled by an ATMega microcontroller with the Arduino Duemilanove microcontroller board [2].
- **Programming Environment:** Our algorithm is implemented using Microsoft Visual Studio 2010 C++ with OpenCV (Open Source Computer Vision) [11] which is a library of programming functions for realtime computer vision, and the OpenCV version we adopt is 2.3.1. And the microcontroller board of the haptic jacket is controlled by Arduino software.

## 4.2 Experimental Result for the Touchscreen Algorithm

### 4.2.1 Qualitative results:

To test the precision of our novel webcam-based touchscreen, we display several red test points on a full-screen window, and ask the users to touch those dots within the webcam detection region. We estimate the horizontal and the vertical touch points' positions on the screen, and draw the estimated locations on the same window with white dots. Figure 4.1 is an example of this process on a black background.

Since our motivation is to combine this touchscreen with the conventional audiovideo teleconferencing software, we capture several chatting images from *Skype* or any webcam, and test the touch point position on these images to verify the compatibility



Figure 4.1: Full-screen black window with red test points and white calculated points

between our application and the chat software. In order to test the robustness of our system, the resolutions of these chatting images differ from  $1280 \times 853$  to  $640 \times 480$ , the light conditions and backgrounds vary with the chatting environment, while the color pattern of the images are also different. Figure 4.2 shows a full-screen window with a *Skype* video chat image, and Figure 4.3 shows our experimental results with the white dots. Figure 4.4 shows other original images (left column) and our test results with those images (right column).

From these experimental results, we can see that despite some small errors, our touchscreen is accurate and robust. During the test, we realized that some colors from the background image may affect the color of the finger tip given the light reflection and thus influence the detection and calculation results. Actually, the white color would bring the greatest impact since its color reflection is the strongest compared with other colors, and it does have the similar color as real finger sometimes. We can also see that the test results with touch points near the screen's bottom are less precise than those at upper positions. This is due to the fact that at the bottom of the screen only a small



Figure 4.2: Original figure with a *Skype* chat image as background



Figure 4.3: Testing result with a *Skype* chat image as background



Figure 4.4: Testing results with other chat images as background

part of the finger is detected, and this part can be easily affected by the light reflection and cause some inaccuracy.

#### 4.2.2 Quantitative results:

Table 4.1. Accuracy of Calculated Touch Tostitons							
Image	Number	Number	Detection	Vertical	Vertical	Horizontal	Horizontal
	of	of	Failure	Average	Absolute	Average	Absolute
	Test	Detection		Absolute	Error	Absolute	Error
Number	Points	Failure	$\operatorname{Ratio}(\%)$	Error(mm)	$\operatorname{Ratio}(\%)$	Error(mm)	$\operatorname{Ratio}(\%)$
1	74	0	0	3.46	1.15	2.61	0.69
2	159	7	4.40	3.27	1.09	1.84	0.48
3	139	3	2.16	4.15	1.38	2.06	0.54
4	141	7	4.96	3.70	1.23	2.03	0.53
5	137	14	10.21	3.25	1.08	1.91	0.50
6	103	1	0.97	4.10	1.37	2.33	0.61
Average	125.50	5.33	4.25	3.66	1.22	2.13	0.56

Table 4.1: Accuracy of Calculated Touch Positions

Furthermore, in order to obtain precious data to evaluate the accuracy of our algorithm, we calculate the average absolute error of the estimated touch points (white dots) within each image, and also the global average of all images' average absolute errors. For these testing images, the vertical and the horizontal distances between any two close pre-set displayed red dots are both 3cm. For the convenience of managing images, we number the pictures with black background in Figure 4.1 as Image 1, the picture with a Skype chat window in Figure 4.3 as Image 2, and pictures in the right column of Figure 4.4 as Images 3 to 6. Results are shown in Table 4.1. Since occasional finger detection failure occurs in some points close to the bottom line of the screen, we remove those points when calculating the average absolute errors. The number of detection failure points and the ratio of this number to the total tested number are recorded and shown in this table, as well as the vertical and horizontal average absolute errors along with their ratio to the whole screen's height (300mm) and width (380mm). From this table, we can see that total finger detection failure points ratio is 4.25%, when vertical and horizontal total average absolute errors are 3.66(mm) and 2.13(mm) respectively and their ratios to the screen's height and width are just 1.22% and 0.56%. In order to show more details



Figure 4.5: The boxplots of touch points' vertical and horizontal absolute errors

of our experimental results, we also add the boxplots of the touch points' vertical and horizontal absolute errors, as shown in Figure 4.5. Results show that our system has a rather high successful detection rate, along with an accurate touch positions estimation.

Same experiments were conducted with other screens in different environment and different users, and the results were similar. This shows that our system is robust with different ambient conditions and configurations. Also, for each image, the detection and calculation time is from 0.1s to 1.3s with the frame size  $1280 \times 1024$ . The closer the finger is to the camera, the longer the time will be, because more pixels need to be processed. However, if we disable the continuity property strategy, which is mentioned in chapter 3, the time will be reduced to 0.1s to 0.3s, since this strategy costs much time especially when the finger is close to the camera.

## 4.3 Human Body Detection Result

As mentioned in Chapter 3, we use the haarcascade model in OpenCV to detect face in the image, and then locate the upper body parts based on the detected face and the Vitruvian Man human body proportions model. Figure 4.6 shows the detection results



Figure 4.6: Upper body dividing results of other images

using this method with some users, same as Figure 3.20 in Chapter 3, face detection result is shown with yellow box, while upper body dividing results are displayed with red boxes and numbers. As we can see, though these results are not so accurate, they can still offer good estimations of the human body parts. As we can see from these results, the upper body detection and division method could offer an acceptable approximation, though it is not extremely accurate. It actually satisfies our goal of transmitting emotion through touchscreen and haptic jacket.

Figure 4.7 shows that when user is touching a point within number 13 box, our system successfully detects this touch position and then activates the corresponding actuators inside the haptic jacket.

As mentioned before, the microcontroller board of the jacket is the Arduino Duemilanove microcontroller board, shown in Figure 4.8. The Arduino IDE is mainly designed



Figure 4.7: Touch activates number 13 box

to make the process of using electronics in multidisciplinary projects more accessible, and since it is an open source software, it could be downloaded from the Arduino website for free. We use Arduino software to write, compile and upload programs to the microcontroller boards in C language [24]. Since we map each number with a certain kind of vibration, once the jacket receives a box ID, which represents a upper body part, it will active the vibration at the corresponding position. In this way, we successfully combined our touchscreen with the haptic jacket.

From those experimental results, we can conclude that our system could work well with conventional audio-video telecommunication software to provide users more effective and intimate emotion interaction.



Figure 4.8: Arduino Duemilanove microcontroller board

## 4.4 Other Applications

Since our touchscreen algorithm is accurate and fast, we can use it for other educational or recreational applications, such as drawing, writing, and games. By adding some control keys like "stop", "start", and "clear background", we have used our novel touchscreen to realize functions of drawing and writing. Figure 4.9 shows the results of writing the word "MCR", which represents our Multimedia Communications Research Laboratory. Figure 4.10 shows the drawing of a human figure. Of course, we can also realize more applications with our proposed systems, and we will try to do this in the future.



Figure 4.9: Writing word "MCR"



Figure 4.10: Figure drawing

# Chapter 5

# **Conclusion and Future Work**

## 5.1 Conclusion

With the fast development of the Internet and its related applications, the traditional audio-video teleconferencing systems fail to satisfy people's needs for improved realistic communication. As we have reached the most realistic experience for the visual and auditive aspects of the interaction/chat through the 3D and HD cameras and the stere audio, it would be natural to try to integrate another sense to this interaction. We actually opt to integrate the touch sense to the chat as it is an intimate bidirectional sense. With haptics, the telecommunication systems would be more interactive by enabling people to express and receive intimate emotion, affection or intention more directly.

Motivated by the *HugMe*, a synchronous audio-video-haptic teleconferencing system developed by the Multimedia Communications Research Laboratory of the University of Ottawa, this thesis presented a webcam-based touchscreen method to enhance the users' experience when using this system by allowing the user to touch the image of another user with their bare hand directly, and to transmit the sense of touch to the remote user equipped with a haptic jacket. Without using any I/O devices, like a mouse, a keyboard, or any haptic device, users could convey their emotions more directly by simply touching the image on the screen. Also, with less device required, the price of such system would be more reasonable.

By choosing an appropriate webcam position and its field of view direction, acquiring essential data from the initialization and calibration phase, proposing efficient finger detection method and strategies, and advancing precious mathematical models to calculate the touch point's horizontal and vertical position, we have successfully realized an innovative webcam-based touchscreen, which is accurate while being robust to lighting and reflection changes.

Also experiments were conducted to test the precision of our touchscreen, as well as its compatibility with conventional audio-video teleconferencing systems. Using a highly reliable face detection method and the Vitruvian Man human proportions, we could recognize a human in the chat image and divide his/her upper body into several numbered boxes to map them to the corresponding sensors embedded inside the haptic jacket, worn by the remote user. Thus, when the active user touches the human image, we are able to detect and show the touched box and its ID. Once receiving the touch area ID, the passive user would feel the "touch" in this region. Compared to methods using markers on the clothes to track a user, our body parts detection method could offer users more freedom of choosing their clothes and the environment.

Since our touchscreen is accurate and fast enough for realtime applications, it can be used in other educational or recreational applications. In this thesis, we showed functions for drawing and writing. The experimental results show that the webcam-base touchscreen works well for those applications.

## 5.2 Future Work

As future improvements of our touchscreen system, an automatic intrinsic camera parameters adjustment should be added to capture images with better quality. With an appropriate focus, brightness, white balance, saturation, exposure and contrast setting to suit the various light conditions, the finger detection would be much more accurate and easier, since the color difference between the finger and the screen would be larger. Also, the finger color would be less influenced by the light reflection, and the mirror finger's influence would be minimized.

Also, a more precise human detection method should be developed to satisfy situations in which a more accurate mapping between a touch point and the corresponding sensors is needed. The tracking method we adopt now, which is mainly based on the estimation of the human body proportions, is satisfactory to detect large areas on the body.

Furthermore, we need to combine our touchscreen and tracking method with some conventional chat systems to put it into practical use. In this thesis, we just test the feasibility of adding a webcam-based touchscreen and a human body parts detection to chat systems. Though it is successful, we may still meet problems when implementing it, because many softwares are not open source or their Application Programming Interface (API) are not user-friendly. Since we need to use the video from the software to locate the faces, and using the chat text messaging to send the touch position to a remote part, certain control of the telecommunication software is necessary.

In the future, we will also try to develop more applications based on the webcambased novel touchscreen. As more and more applications are available for smartphones and tablets, transforming them onto our system would greatly enrich the use of our system. To realize more functions, a multi-finger detection and gesture control should be added to our application.

Also, in order to make the users' experience more realistic, it would be appropriate to provide an efficient haptic feedback on the fingers of the active user.

# References

- AR Toolkit website. http://www.hitl.washington.edu/artoolkit Accessed August 25, 2012.
- [2] Arduino Duemilanove. http://arduino.cc/en/Main/arduinoBoardDuemilanove Accessed August 25, 2012.
- [3] Google Talk. http://www.google.com/talk/ Accessed August 25, 2012.
- [4] Haptic device from sensable. http://www.sensable.com/ industries-application-development.htm Accessed August 25, 2012.
- [5] How internet infrastructure works. http://computer. howstuffworks.com/internet/b\discretionary{-}{}{asics/internet\ discretionary{-}{}{infrastructure.htm Accessed August 25, 2012.
- [6] Hug Shirt from Cutecircuit. http://www.cutecircuit.com/collections/ special\discretionary{-}{}{projects/hug\discretionary{-}{}{shirt/ Accessed August 25, 2012.
- [7] The importance of touch. http://library.adoption.com/Parenting-Skills/ The-Importance-of-Touch/article/3060/1.html Accessed August 25, 2012.
- [8] Kiss transmission device. http://www.diginfo.tv/v/11-0090-r-en.php Accessed August 25, 2012.
- [9] Lifecam HD5001. http://www.microsoft.com/hardware/en-us/p/ lifecam-hd-5001/GNF-00001 Accessed August 25, 2012.
- [10] Nokia vibrating tattoo alerts patent. http://appft1.uspto.gov/ netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetahtml%2FPTO%

2Fsearch-bool.html&r=10&f=G&l=50&co1=AND&d=PG01&s1=Nokia.AS.&OS=AN/ Nokia&RS=AN/Nokia?fvrewsd Accessed August 25, 2012.

- [11] OpenCV. http://opencv.willowgarage.com/wiki Accessed August 25, 2012.
- [12] Pinhole camera model from wikipedia. http://en.wikipedia.org/wiki/Pinhole\_ camera Accessed August 25, 2012.
- [13] Second Life. http://www.secondlife.com Accessed August 25, 2012.
- [14] Skype. http://www.skype.com Accessed August 25, 2012.
- [15] Telepresence room from cisco. http://www.cisco.com/en/US/products/ps7060/ index.html Accessed August 25, 2012.
- [16] Videophone from Dlink. http://www.dlink.com/products/?pid=302 Accessed August 25, 2012.
- [17] Vitruvian Man. http://en.wikipedia.org/wiki/Vitruvian\_Man Accessed August 25, 2012.
- [18] Windows Live Messenger. http://windows.microsoft.com/en-US/messenger/ home Accessed August 25, 2012.
- [19] Yahoo Messenger. http://messenger.yahoo.com/ Accessed August 25, 2012.
- [20] Kazi Masudul Alam, Sandro Hardy, Aysha Akther, and Abdulmotaleb El Saddik. Sms text based affective haptic application. In *Proceedings of Virtual Reality International Conference (VRIC 2011)*, Apr 2011.
- [21] Achilleas Anagnostopoulos and Aristodemos Pnevmatikakis. A realtime mixed reality system for seamless interaction between real and virtual objects. In *Proceedings* of the 3rd international conference on Digital Interactive Media in Entertainment and Arts, DIMEA '08, pages 199–204, New York, NY, USA, 2008. ACM.
- [22] R. Anchini, C. Liguori, V. Paciello, and A. Paolillo. A comparison between stereovision techniques for the reconstruction of 3-d coordinates of objects. *Instrumentation and Measurement, IEEE Transactions on*, 55(5):1459–1466, oct. 2006.

#### References

- [23] Christian J. Breiteneder, Simon J. Gibbs, and Costas Arapis. Teleportan augmented reality teleconferencing environment. In *Proceedings of the Eurographics workshop* on Virtual environments and scientific visualization '96, pages 41–49, London, UK, UK, 1996. Springer-Verlag.
- [24] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 423– 432, New York, NY, USA, 2008. ACM.
- [25] C. Cameron, L. DiValentin, R. Manaktala, A. McElhaney, C. Nostrand, O. Quinlan, L. Sharpe, A. Slagle, C. Wood, Yang Yang Zheng, and G.J. Gerling. Using electroactive polymers to simulate the sense of light touch and vibration in a virtual reality environment. In Systems and Information Engineering Design Symposium (SIEDS), 2011 IEEE, pages 121–126, april 2011.
- [26] Jongeun Cha, M. Eid, L. Rahal, and A. El Saddik. Hugme: An interpersonal haptic communication system. In *Haptic Audio visual Environments and Games*, 2008. *HAVE 2008. IEEE International Workshop on*, pages 99–102, oct. 2008.
- [27] Jongeun Cha, Mohamad Eid, Ahmad Barghout, ASM Mahfujur Rahman, and Abdulmotaleb El Saddik. Hugme: synchronous haptic teleconferencing. In *Proceedings* of the 17th ACM international conference on Multimedia, MM '09, pages 1135–1136, New York, NY, USA, 2009. ACM.
- [28] Li-Wei Chan, Yi-Fan Chuang, Meng-Chieh Yu, Yi-Liu Chao, Ming-Sui Lee, Yi-Ping Hung, and Jane Hsu. Gesture-based interaction for a magic crystal ball. In Proceedings of the 2007 ACM symposium on Virtual reality software and technology, VRST '07, pages 157–164, New York, NY, USA, 2007. ACM.
- [29] Kelvin Cheng and Masahiro Takatsuka. Initial evaluation of a bare-hand interaction technique for large displays using a webcam. In *Proceedings of the 1st ACM SIGCHI* symposium on Engineering interactive computing systems, EICS '09, pages 291–296, New York, NY, USA, 2009. ACM.
- [30] M. Eid, Jongeun Cha, and A. El Saddik. Hugme: A haptic videoconferencing system for interpersonal communication. In *Virtual Environments, Human-Computer*

Interfaces and Measurement Systems, 2008. VECIMS 2008. IEEE Conference on, pages 5–9, july 2008.

- [31] Mohamad Eid, Mauricio Orozco, and Abdulmotaleb El Saddik. A guided tour in haptic audio visual environments and applications. *International Journal of Ad*vanced Media and Communication, 1(3):265–297, June 2007.
- [32] S. Moustakidis M. Strintzis G. Nikolakis, D. Tzovaras. Cybergrasp and phantom integration: Enhanced haptic access for visually impaired users. In *Conference Speech and Computer*, pages 507–513, St. Petersburg, Russia, 2004.
- [33] Martha A. Garcia-Murillo and Lee W McKnight. Internet telephony: Effects on the universal service program in the united states. *Review of Network Economics*, 4:1–15, 2005.
- [34] D. O. Gorodnichy and A. Yogeswaran. Detection and tracking of pianist hands and fingers. In Proceedings of 3rd Canadian Conf. Computer and Robot Vision, 2006.
- [35] Martin Hilbert and Priscila Lpez. The worlds technological capacity to store, communicate, and compute information. *Science*, 332(6025):60–65, 2011.
- [36] Minh Phuong Hoang, Tien Dzung Nguyen, and Thang Manh Hoang. Implementation of webcam-based touchscreen. In *Communications and Electronics (ICCE)*, 2010 Third International Conference on, pages 201–206, aug. 2010.
- [37] S.K.A. Hossain, A.S.M.M. Rahman, and A. El Saddik. Interpersonal haptic communication in second life. In *Haptic Audio-Visual Environments and Games (HAVE)*, 2010 IEEE International Symposium on, pages 1–4, oct. 2010.
- [38] Shahram Izadi, Steve Hodges, Alex Butler, Alban Rrustemi, and Bill Buxton. Thinsight: integrated optical multi-touch sensing through thin form-factor displays. In Proceedings of the 2007 workshop on Emerging displays technologies: images and beyond: the future of displays and interacton, EDT '07, New York, NY, USA, 2007. ACM.
- [39] Paula M. Duncan Joseph F. Hagan, Judith S. Shaw. Bright Futures: Guidelines for Health Supervision of Infants, Children, and Adolescents, Third Edition. 2008.
- [40] Sung Kwan Kang, Mi Young Nam, and Phill Kyu Rhee. Color based hand and finger detection technology for user interaction. In *Convergence and Hybrid Information*

Technology, 2008. ICHIT '08. International Conference on, pages 229–236, aug. 2008.

- [41] Manolya Kavakli, Meredith Taylor, and Anatoly Trapeznikov. Designing in virtual reality (desire): a gesture-based interface. In *Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*, DIMEA '07, pages 131–136, New York, NY, USA, 2007. ACM.
- [42] Daeho Lee and Youngtae Park. Vision-based remote control system by motion detection and open finger counting. *Consumer Electronics, IEEE Transactions on*, 55(4):2308–2313, 2009.
- [43] P. Lemmens, F. Crompvoets, D. Brokken, J. van den Eerenbeemd, and G.-J. de Vries. A body-conforming tactile jacket to enrich movie viewing. In EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint, pages 7–12, march 2009.
- [44] R.K. Lenz and R.Y. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3-d machine vision metrology. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(5):713-720, sep 1988.
- [45] Robert Lindeman, Yasuyuki Yanagida, Haruo Noma, and Kenichi Hosaka. Wearable vibrotactile systems for virtual contact and information display. *Virtual Reality*, 9:203–213, 2006.
- [46] Rodney Feldman Andre D. Cropper Michael J. Siwinski, Kathleen Kilmer. Organic electroluminescent display with integrated touch-screen. Patent, 07 2002. US 6424094.
- [47] S.J.V. Nichols. New interfaces at the touch of a fingertip. Computer, 40(8):12 –15, aug. 2007.
- [48] Dublin OH Robert G. Kable and Columbus OH Philip A. Schlosser. Electrographic apparatus. Patent, 05 1987. US 4665283.
- [49] Knoxville TN Robert Phares. Resistive touchscreen having multiple selectable regions for pressure discrimination. Patent, 09 1998. US 5815141.
- [50] A. E. Saddik, M. Orozco, M. Eid, and J. Cha. *Haptics Technologies : Bringing Touch to Multimedia*. Springer Series on Touch and Haptic Systems. Springer, 2011.

- [51] El Saddik. The potential of haptics technologies. Instrumentation Measurement Magazine, IEEE, 10(1):10-17, feb. 2007.
- [52] Michael E. Whitman Amy B. Woszczynski Ken Hoganson Herbert Mattord. Solomon Negash, Michael E. Whitman. Editors: Solomon Negash. Handbook of Distance Learning for Real-Time and Asynchronous Information Technology Education. Idea Group Inc, 2008.
- [53] Stanislava Soro and Wendi Heinzelman. A survey of visual sensor networks. Advances in Multimedia, 2009, 2009.
- [54] Ching-Hung Su, Huang-Sen Chiu, and Tsai-Ming Hsieh. An efficient image retrieval based on hsv color space. In *Electrical and Control Engineering (ICECE)*, 2011 International Conference on, pages 5746 –5749, sept. 2011.
- [55] James Keng Soon Teh, Adrian David Cheok, Roshan L. Peiris, Yongsoon Choi, Vuong Thuong, and Sha Lai. Huggy pajama: a mobile parent and child hugging communication system. In *Proceedings of the 7th international conference on Interaction design and children*, IDC '08, pages 250–257, New York, NY, USA, 2008. ACM.
- [56] D. Tsetserukou and A. Neviarouskaya. Affective haptic garment enhancing communication in second life. In Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on, pages 1-2, sept. 2009.
- [57] D. Tsetserukou and A. Neviarouskaya. iFeel\_IM!: Augmenting emotions during online communication. *Computer Graphics and Applications, IEEE*, 30(5):72–80, sept.-oct. 2010.
- [58] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. Conference on Computer Vision and Pattern Recognition, 1:511–518, 2001.
- [59] Xiying Wang, Xiwen Zhang, and Guozhong Dai. Tracking of deformable human hand in real time as continuous input for gesture-based interaction. In *Proceedings* of the 12th international conference on Intelligent user interfaces, IUI '07, pages 235–242, New York, NY, USA, 2007. ACM.
- [60] Lin XU. Infrared touch screen. Patent Application, 05 2011. US 2011/0115748 A1.

#### References

- [61] Liang-Guo Zhang, Yiqiang Chen, Gaolin Fang, Xilin Chen, and Wen Gao. A visionbased sign language recognition system using tied-mixture density hmm. In *Proceed*ings of the 6th international conference on Multimodal interfaces, ICMI '04, pages 198–204, New York, NY, USA, 2004. ACM.
- [62] Zhengyou Zhang and Ying Shan. Visual screen: Transforming an ordinary screen into a touch screen. In *Proceedings of IAPR Workshop on Machine Vision Applications*, pages 215–218, 2000.