# VISION-BASED HAND INTERFACE SYSTEMS IN HUMAN COMPUTER INTERACTION

SERKAN GENÇ

MARCH 2010

# VISION-BASED HAND INTERFACE SYSTEMS IN HUMAN COMPUTER INTERACTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SERKAN GENÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

MARCH 2010

Approval of the thesis:

**VISION-BASED HAND INTERFACE SYSTEMS IN HUMAN COMPUTER INTERACTION**

submitted by **SERKAN GENÇ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Prof. Dr. Volkan Atalay _____
Supervisor, **Computer Engineering Dept., METU**

**Examining Committee Members:**

Assoc. Prof. Dr. Veysi İşler _____
Computer Engineering Dept., METU

Prof. Dr. Volkan Atalay _____
Computer Engineering Dept., METU

Assoc. Prof. Dr. Uğur Güdükbay _____
Computer Engineering Dept., Bilkent University

Prof. Dr. Sibel Tarı _____
Computer Engineering Dept., METU

Assoc. Prof. Dr. Erkan Tın _____
Comp. Tech. and Inf. Systems Dept., Bilkent University

Date: 5 March 2010

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name :

Signature :

## ABSTRACT

VISION-BASED HAND INTERFACE SYSTEMS IN HUMAN COMPUTER INTERACTION

Genç, Serkan

Ph.D., Department of Computer Engineering

Supervisor: Prof. Dr. Volkan Atalay

March 2010, 83 pages

People began to interact with their own environment since their birth. Their main organs to sense their surroundings are their hands, and this is the most natural way of interaction in human-human interactions. The goal of this dissertation is to enable users to employ their hands in interaction with computers similar to human-human interaction. Using hands in the computer interaction increases both the naturalness of computer usage and the speed of interaction. One way of accomplishing this goal is to utilize computer vision methods to develop hand interfaces. In this study, a regular, low-cost camera is used for image acquisition, and the images from camera are processed by our novel vision system to detect user intention. The contributions are (i) a method for interacting with a screen without touching in a distributed computer system is proposed, (ii) a benchmark of four hand shape representation methods is performed using a comprehensive hand shape video database, and (iii) a vison-based hand interface is designed for an application that queries a video database system, and its usability and performances are also assessed by a group of test users to determine its suitability for the application.



Keywords: Hand Interface Design, Hand Shape Representation, Interface Usability

# ÖZ

İNSAN BİLGİSAYAR İLETİŞİMİ İÇİN GÖRÜNTÜ TABANLI EL ARAYÜZÜ SİSTEMLERİ

Genç, Serkan
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Volkan Atalay

Mart 2010, 83 sayfa

Bu tezde, el kullanımının insan-bilgisayar etkileşimini nasıl daha doğal ve verimli hale getirdiği gösterilmektedir. Bu amacı gerçekleştirmenin yollarından biri de kameralı sistemler kullanmaktır. Bizim çalışmamızda bir tane kamera etrafın görüntüsünü almak için kullanılmaktadır. Alınan bu görüntü, görüntü işleme algoritmaları kullanılarak, kullanıcının elinin yeri bulunmakta ve yapmış olduğu el hareketlerine göre de amacı belirlenmektedir. Bu tezde, el ile arayüz yapımına katkı sağlayacak üç unsur önerilmiştir. (i) Dağıtık sistemlerde kullanılmak üzere ekrana dokunmadan ekranla etkileşime izin veren yeni bir el arayüzü tasarlanmıştır. (ii) Nesne şekillerinin tanınmasında kullanılan genel dört yöntemin kabiliyetleri ve hızları, el şekilleri için test ölçülmüştür. (iii) Görüntü işleme yöntemi ile çalışan bir el arayüz tasarlanmış ve video veritabanına sorgu yapmak için kullanılan bir uygulamada, tasarlanan el arayüzü kullanıcılar tarafından uygunluğu ve performansı test edilmiştir.

Anahtar Kelimeler: El Arayüz Tasarımı, El Şeklinin Tanımlanması, El Arayüzü Kullanılabilirliği

## ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude to my supervisor, Professor Volkan Atalay. His understanding, supporting, encouraging and personal guidance have provided a good basis for the present thesis.

I also thank to other members of my thesis committee: Professor Uğur Güdükbay for his collaboration in query database project, Professor Erkan Tın for their constructive comments and for his important supports throughout this work, Professor Sibel Tarı and Professor Veysi İşler for their helpful comments and discussions.

During this work I have collaborated with many colleagues working in the Department of Computer Technology and Information Systems, Bilkent University, and I wish to extend my warmest thanks to all those who have helped me in my thesis.

I also wish to thank Muhammet Baştan and Professor Özgür Ulusoy for their collaboration and their precious comments on the hand interface project.

I would also like to thank my family for their support, and understanding they provided me through my entire life. I would not have finished this thesis without their valuable supports.

## TABLE OF CONTENTS

# LIST OF TABLES

TABLES

## LIST OF FIGURES

FIGURES

## LIST OF ABBREVIATIONS AND ACRONYMS

| | | |
|---|---|---|
| ANOVA | : | Analysis of Variance |
| API | : | Application Programming Interface |
| ASL | : | American Sign Language |
| CSS | : | Curvature Scale Space |
| DOF | : | Degree Of Freedom |
| DTW | : | Dynamic Time Warping |
| FD | : | Fourier Descriptors |
| FSM | : | Finite State Machine |
| GB | : | Giga Byte |
| GMM | : | Gaussian Mixture Model |
| HCI | : | Human Computer Interaction |
| HMM | : | Hidden Markov Model |
| HSL | : | Hue Saturation Lightness Color Space |
| HU | : | Hu Moment Invariant |
| ITOUCH | : | Intelligent Touch Screen |
| IR | : | Infrared |
| MBB | : | Minimum Bounding Box |
| MPEG-7 | : | Moving Picture Experts Group version 7 |
| NN | : | Neural Network |
| OH | : | Orientation Histogram |

| | | |
|---|---|---|
| PDA | : | Personal Digital Assistant |
| RAM | : | Random Access Memory |
| RGB | : | Red Green Blue Color Space |
| SD | : | Shape Descriptor |
| ToF | : | Time of Flight |
| WD | : | Wavelet Descriptor |

# CHAPTER 1

# INTRODUCTION

## 1.1 Traditional and Alternative User Interfaces

Keyboard and mouse are the major interaction devices with computers to input text data, and to point and select objects on the screen. They are artificial and complex to ordinary people who rarely work with computers and they are far from being similar to the interactions in our daily life. For example, we employ our hands, specifically index fingers to show the direction and the position of an object. With the traditional interaction devices, the user translates the pointing action into several artificial and unnatural mouse activities such as moving and clicking.

With the increase in processing power and decrease in the prices of processing hardware, many applications take advantage of using hands. There are several ways to incorporate hands into interfaces. One way is to use special hand device. For example, an electromechanical data glove consists of many sensors and trackers to provide the interface with essential data about the position of the hand palm and the configuration of fingers. An electromechanical data glove is shown in Figure 1. However, special hand devices are expensive and their use is complex for ordinary people. They require intensive calibration procedure. Many cables connected to the glove prevent users from moving freely in the environment.

Figure 1: An electromechanical data glove from CyberGlove System [1].


Figure 2: A sample vision-based interface system from Microsoft Natal Project [2].

An alternative is a vision-based interface system where a camera is employed as the sensor. Figure 2 shows a vision-based interface system with a camera on the television, and the user employs her hands to give commands to an application.
Vision-based interface systems have many advantages. They make use of low-cost cameras. The camera does not require any physical connection equipment with user, which creates the illusion of immersive experience.

There are several commercial applications of vision-based interface systems on the market. For example, Microsoft's Natal Project develops hand and body interfaces for games that do not require any external devices. Hitachi is also developing a television system that can be controlled by hand gestures to turn on and off the television, to switch channels and to adjust the sound volume. Moreover, many applications are available on the market for managing slides, drawing figures by fingers, and interacting with objects in augmented reality. Sign language systems are also developed for hear impaired people for human-human interaction mediated by a computer system.

## 1.2 Motivation and Problem Definition

Vision-based approaches mimic the human visual system to analyze images. Although, we do not yet have robust computer algorithms and systems to process images as humans do, there exist at least ad hoc techniques to solve problems for restricted environments.

A significant bottleneck in computer vision is the segmentation problem. Segmentation is a process to form meaningful regions of pixels. Among many segmentation algorithms, selecting the appropriate one for a specific application is essential, since each algorithm has its own power and restrictions. For example, background segmentation requires a static background with no obvious changes in the illumination conditions.

The recognition of commands described by hand is another difficult problem. Since hand is a complex object with 27 degrees of freedom, it can produce large number of possible shapes. Selecting the best classification and recognition method

in terms of discrimination power and speed is very important for the stability of a hand interface.

Finding the convenient application for hand interfaces is an issue, because there is no unique interface device for all applications. For example, a joystick for a flight simulator will be more convenient than a mouse or a hand interface.

In spite of several advantages, vision-based hand interfaces involve many bottlenecks restricting their usage in real life applications. The most essential problem is the lack of available vision algorithm to detect hand regions, track them and recognize their shapes in images robustly and in real-time. Additionally, we need to find out suitable applications for hand-based interfaces. Although this is relatively easier than the former problem, further research is required.

In short, the goal of this thesis is to propose methods to achieve novel vision-based hand interfaces that are robust, real-time and applicable in real-life environments. To this end, the issues that should be addressed are as follows:

1. There are many vision algorithms in the literature. However, the appropriate algorithms for the segmentation and recognition processes should be selected and integrated into a vision system that has high recognition rate of hand commands, and a low response time in the interaction. Robustness, fast response (real-time) and high discriminative power are the desired properties of the proposed vision-based hand system.
2. It is also essential to find out possible applications where the hand interface increases the naturalness and efficiency of the interaction. The suitability of the application should be proved by conducting the usability and performance experiments.

## 1.3 Key Contributions

Our study has three major contributions as stated below:

1. A novel concept called “distributed interaction” is proposed where a user can start an interaction in one computer and goes on in other computers. Completing a task requires many interactions with many different computers. To realize this idea, a new interface called ITouch is proposed. It is a vision-based system where a camera is located in such a point that it sees the whole screen. It monitors user’s hand and converts hand gesture into interface commands. It creates an interaction volume rather than interaction surface as it happens in touch screen monitors. A user can perform any predefined hand gestures in front of a regular monitor unlike a touch screen monitor which is limited by point and click operations. This proposed interface is more natural than regular mouse interface and more capable than touch screen monitors. Chapter 2 explains ITouch system and potential applications.
2. Hand shape recognition is one of the main tasks in the design of hand-based interface. The recognition process uses hand shape representations in its classification stage. This is why the selection of hand shape representation is very important and influences the performance of classification. Meaningful representation of hand shape is called shape descriptors. Although there are many shape descriptors methods in literature, nobody knows the suitable descriptor method for hand shapes. This is the first study that compares four general shape descriptors especially for hand shapes in terms of discrimination power and speed. After experiments, appropriate shape descriptor method is proposed and it can be used in real-time hand-based interfaces.
3. Users can experience immersive interaction with computers when they use appropriate interface devices. Like mouse and keyboard, hand is not appropriate for all kind of applications. Therefore, suitable application areas for hand interaction should be figured out. This thesis proposes a vision-based interaction system for querying video databases. The proposed vision-based hand gesture interaction system uses appropriate algorithms for segmentation, tracking and recognition for real-time and robust interaction. Four different

types of queries are experimented by a group of people who performs queries with a hand-based interface implemented by proposed vision-based system and a standard mouse-based interface. Test users compare two interfaces in terms of usability and performance criteria to determine the best interface for this application.

## 1.4 Organization of the Thesis

Chapter 2 explains Itouch system, and presents its difference from available interfaces such as touch screen monitors. Also, its application areas in distributed interaction are introduced.

Chapter 3 presents a shape descriptor method which is suitable for real-time and robust hand-based applications. Among many descriptors, the selection of candidate shape descriptor is explained, and selected four descriptors are compared in terms of discriminative power and speed. At the end, the experimental results are discussed.

Chapter 4 presents a novel vision-based system which segments, tracks hands and recognizes their shapes. This system is used in the implementation of a new application, a query application of video database system. The suitability of this application for hand interface is experimented and usability results of both hand and mouse-based interface are discussed.

Chapter 5 gives conclusions and suggestions for further improvements.

# CHAPTER 2

# INTELLIGENT TOUCH SCREEN

## 2.1 Abstract

The aim of Human Computer Interaction (HCI) is to design user interfaces such that they are natural, immersive, goal-oriented and easy to learn. For example, mouse is the widely-used device for pointing purpose. However, moving mouse to point a desired position is not an immersive action since we usually employ our index finger to point an object. To overcome this disadvantage of mouse, touch-screen monitors are emerged as an immersive interface. In this way, a user can interact with the monitor by point and click operations. However, these interfaces are not only expensive but also limit the number of interaction operations. The system that we describe in this thesis solves the limited operations of touch-screen monitors by using a vision based system in which a camera is positioned so that the entire screen and the user's hand can be viewed, a hand gesture recognizer processes the captured image and understands the command given by the user's hand. She can interact with monitor by her hand but not limited to point and click operations. To show the potential of this interface, we have developed an application which uses 4 hand gestures; point, click, copy and paste in order to transfer a selected object in one computer system to another system. The user picks an object such as file by applying a copy hand gesture in a computer system, then goes to another computer system, and employs paste hand gesture to transfer the object. This is an analogy to picking physically an object from one table and dropping it onto another one. All shared clipboard and network operations are handled by the system. This makes object

transfer between computers very easy, natural and immersive. Since the interaction starts in one computer system and finishes in another, it can be called as distributed interaction system.

## 2.2 Introduction

Traditionally, a good software system is considered to be the one working properly and correctly. However, besides fulfilling the desired functionality, usability of a system is also essential. Therefore, natural, immersive human computer interaction is a desired property for current software systems [3] [4][5][6] [7] [8] [9]. Furthermore, user must not be aware of the complexity behind the system.

The aim of this study is to explain the design of a natural, goal-oriented user interface functioning on a single or networked window system. In a single window system, user usually points and clicks to give commands to applications via a mouse device. At first sight, it is not obvious what the functionality of a mouse is. Moving mouse leads to moving of digital representation of mouse on the screen, that is, mouse pointer. Although this works, it is not immersive, because the user looks at the screen, but her hands or pointing actions take place somewhere else. Therefore, several studies simulate mouse device with a camera [10] [11][12]. In those studies, the user cannot point to the desired place with her hand on the screen directly. Instead, she moves her hand over or the side of the keyboard. Therefore, they have the same drawbacks as mouse devices. On the other hand, other studies project the screen onto a wall and user employs her hand to give commands [13] [14]. This type of system is useful particularly for presentations. In addition, touch-screen monitors have emerged as an immersive interface where a user interacts with the monitor by pointing and clicking directly. However, they are not only expensive but also have limited number of interactions (only point and click).

Our system overcomes above mentioned problems by tracking the hand on the screen. It employs a vision based system in which a camera is positioned so that the entire screen and the user's hand can be viewed. A hand gesture recognizer processes the captured image and understands the command given by the user's hand.

The user moves her hand on the screen for pointing purposes similar to Zhang's Virtual Screen [15]. However, our system not only tracks the position of fingertip but also recognizes other predefined hand gestures. In this way, an intelligent, cheap but low-resolution touch screen (ITouch) is simulated.

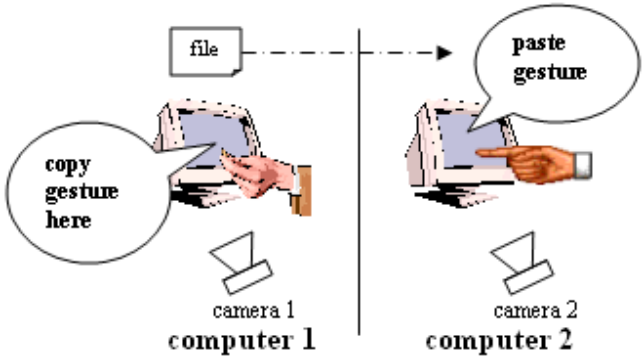



Figure 3: Copy-paste between computers in distributed system

In a multi-computer system, one of the desired operations is to perform "copy and paste" among computers. For example, in a two computer scenario as shown in Figure 3, to copy a file from computer 1 to computer 2, the user picks up the file from computer 1 by making copy-hand-gesture and then pastes the file into computer 2 by making paste-hand-gesture. In this way, we design a user interface that performs operations such as copy-and-paste and file transfers among computers in a very natural and immersive way. Although, there already exists a system that enables user to carry an object between PDA's using a special digital pen [16], our system does not need any special hardware. Our approach divides user interaction into several phases (selecting in one computer system, dropping in another) in order to fulfill a certain task and user interacts with more than one computer system; that is why we call it distributed.

## 2.3 System Overview

As in Figure 4, there are mainly two stages. The first stage is calibration stage or setup stage. A camera is placed so that the entire screen can be viewed, and the perspective transformation parameters are calculated by using calibration pattern.

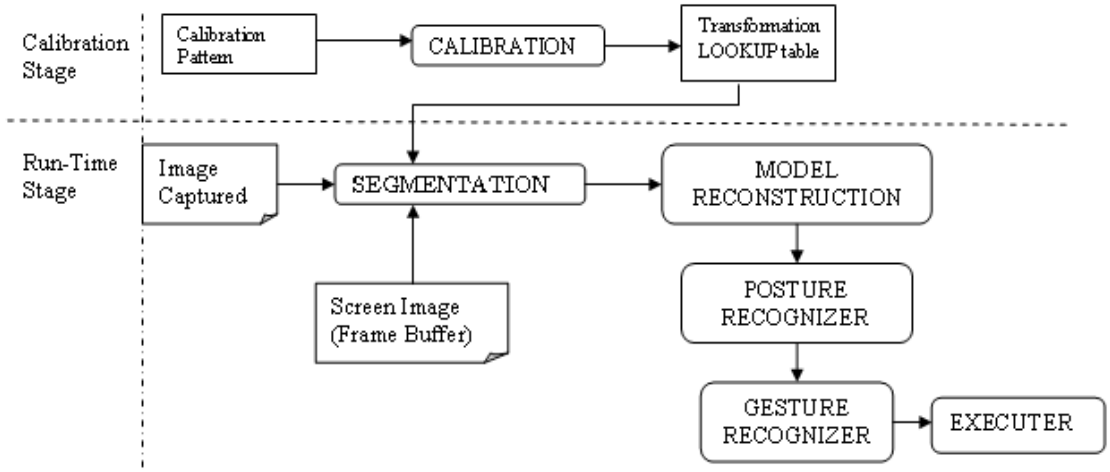



Figure 4: The System Overview of ITouch

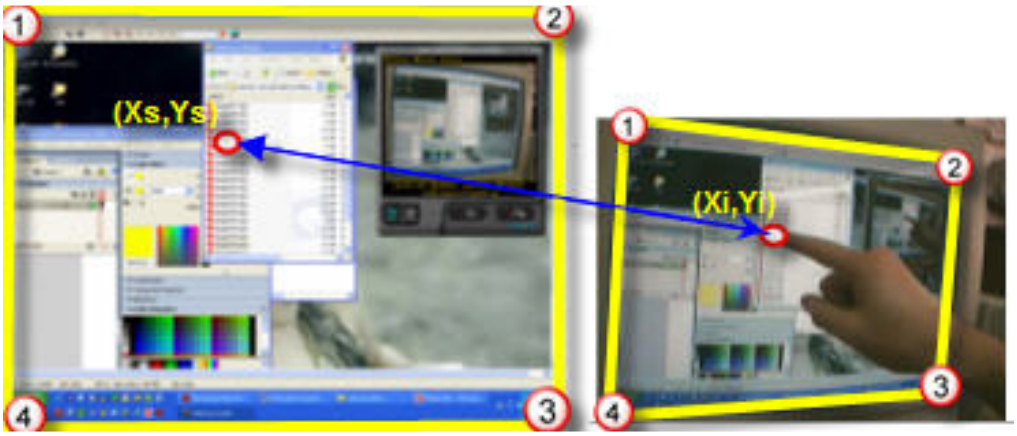
The second stage is the run time stage. The first process is segmentation that separates the hand pixels from the background, and then model reconstruction finds the model parameters from the segmented hand pixels. After that, posture recognizer uses the model parameters and decides the current state of the hand. The current posture is sent to the gesture recognizer, which is a state machine and concludes the command given by the user.

## 2.4 Camera Calibration

As mentioned before, a camera is used to capture the image of the screen, and the screen pixels can be accessed from the frame buffer. Camera calibration is used to find a mapping between a point in the captured image and a point on the screen.

Figure 5 shows that all pixels in yellow frame in the image maps to a point on the screen. There are many ways to map a point on the image to a point on the screen, such as bilinear transformation, polynomial transformation. However, the

correct mapping is perspective transformation due to the perspective nature of a pin-hole camera. This transformation is expressed in terms of a 3X3 matrix, and the coefficients can be determined by establishing four-point correspondences in the image and the screen, four corner points in Figure 5, [17].




Screen Coordinate Image Coordinate

Figure 5: ($X_i$ ,$Y_i$) in image coordinate maps to ($X_S$ , $Y_S$) in screen coordinates.

However, since the monitor is not flat, this perspective transformation does not lead to a correct solution. This is because there is a non-linear mapping between the screen and the image. An approximation method for this non-linear transformation is used in this study. The method is explained in details in [15]. It uses a calibration pattern with a grid of circles. The pattern is displayed on the screen, and using the four corner points as in Figure 5, an approximated transformation matrix is computed, the calibration circles in the image are mapped back to screen coordinates. Since we know the original position of the circles, the residual vector on each grid point is calculated. For the points other than those on the grid, the approximated position found by perspective transformation is shifted by a 2D vector found by a bilinear interpolation of surrounding 4 grid points' residual vector.

If the camera and the monitor are static, that is, no change in orientation and position, this mapping is determined once and each pixel's transformed position is

stored in a look-up table to speed up the transformation. Because screen-to-image transformation, that is, warping is used in Segmentation part. Storage requirement for screen-to-image transformation is $Screen_{width}$ * $Screen_{height}$ * sizeof(unsigned short)*2 (for x and y). LOOKUP($Screen_X$, $Screen_Y$) returns the corresponding ($Image_X$, $Image_Y$). However, for image-to-screen transformation, there is no need a look-up table since we mapped only 4 points in Model Reconstruction part from the image to screen.

### 2.5 Segmentation

In this study, the segmentation is the process to detect the pixels belonging to the hand region. There are two dominant methods for hand segmentation; color segmentation and background subtraction. Color segmentation models the color of hands and backgrounds in a preprocessing stage [11][18] [19][20]. In run-time, it classifies the pixels as foreground (hand) or background pixels according to the model calculated in preprocessing stage. The second method is background subtraction [21]. The idea is to subtract an image from the background. The result is the region of the hand in the image since the same pixels in the image and the background result in zero intensity. However, due to the noises in image acquisition, usually a threshold value is used to eliminate the noise effect.

$$\forall x, y; Dif(x, y) = \tau(|I(x, y) - B(x, y)|)$$

where Dif is the Difference Image where only different pixels are visible, I is captured image and B is the background, and $\tau$ is the threshold operator.

For dynamic scenes where the illumination changes or new objects appear in the scene etc, a background modeling technique is needed to update the background image. It is still an open problem. However, this is not a issue in our case if the following algorithm is applied:

**Segmentation Algorithm**

1. Lock the screen; none of the processes can update the screen.
2. Capture the screen using the camera as I

3. Copy the screen in frame buffer as B

4. Release the lock

5. Warp B (screen coordinate) to I (image coordinate)

6. Find Dif, now Dif has the hand pixels.

In this way, we ensure that the background of I is exactly the same with B. However, the image I includes both the background and the hand pixels. To subtract two images; I and B, they must be aligned. Therefore, B must be warped using the Look-up table determined at calibration stage. The image Dif consists of the hand pixels.

## 2.6 Skeleton Model

A model is a mathematical representation of an actual object. Usually, a hand is modeled by volumetric models consisting of cylinders, conics, etc., or skeleton models using lines. A full hand model can be represented by 21 parameters for joint angles and 6 parameters for hand pose, thus, 27 parameters in total [22]. Therefore, searching in such a high dimensional space cannot be handled in real time [23][24].

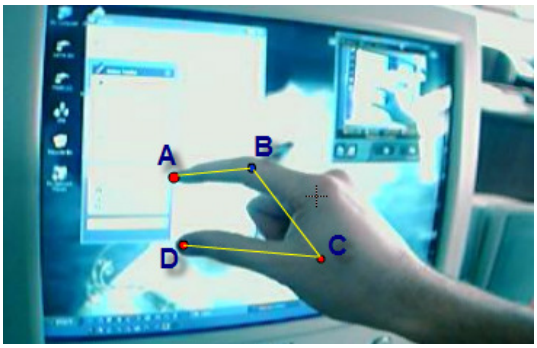



Figure 6: Partial skeleton model

One of the solutions to this problem is to simplify the hand model. In our case, there is no need to reconstruct the full hand model since only a part of the hand is visible in our system; thumb finger and index finger. Thus, a skeleton model for a

partial hand is employed. Our model is based on 4 vertices (A, B, C, D) and 3 edges as in shown Figure 6.

Due to the anatomy of two fingers, a number of constraints are attached to the model. For example, B cannot be under the line connecting A to C. These constraints are helpful in validating the skeleton model built from the image and if the model fits to constraints, it converts the vertices from image coordinate system to screen coordinate system by inverse perspective transformation matrix [17] [15].

To build the model, the user hand is marked at points A, B, C and D with different colors. At startup, these colors and initial positions are given to the color tracker manually. It constructs a color model for each point marked on the hand. Then, the color tracker searches the colors in a search window. The place of the interested points A, B, C and D are used in building the model.

### 2.7 Posture and Gesture Recognizer

In this study, a hand posture is defined as the stationary state of a hand. For example, OK sign by holding the thumb finger up is considered as a hand posture. On the other hand, a gesture is defined as a dynamic movement such as calling someone with the hand [25]. A gesture can be modeled by a series of hand postures. Therefore, the important postures should be defined for a gesture. This is analogous to key frames (postures) in an animation (gesture).

Our system consists of 3 postures; Close (CL), Open (OP) and Wide Open (WO) as shown in Figure 7.

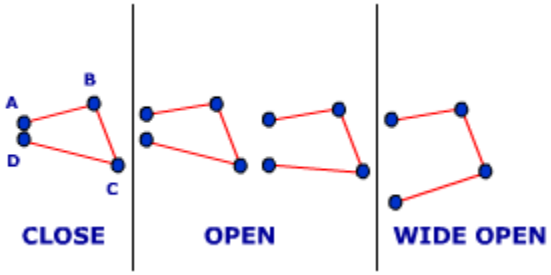



Figure 7: Hand postures in ITouch

$$Posture = \begin{cases} CL & , \quad Dist < \tau_1 \\ OP & , \quad \tau_1 \le Dist < \tau_2 \\ WO & , \quad otherwise \end{cases}$$

Dist is the distance between A and D, $\tau_1$ is the threshold to detect if the posture is CL or OP. $\tau_2$ is the maximum distance for OP posture.

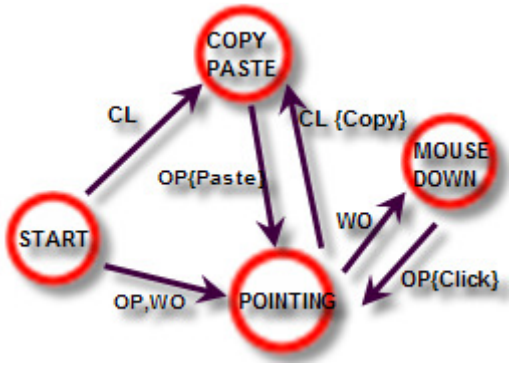



Figure 8: Finite State Machine Gesture recognizer

Gesture recognizer is modeled by a finite state machine (FSM) as presented in Figure 8. States of the machine are {START, COPY/PASTE, POINTING, MOUSEDOWN}, and the input alphabets are the outputs of Posture Recognizer; {CL, OP, WO}. The internal actions are {copy, paste, click}. START is the initial state, and all other states are the final states.

Initially, state machine is in START state, if the posture of the hand is CL, this means user has copied something before, and wants to paste it. The state is changed to COPY/PASTE. If the user opens her fingers (A and D) more than $\tau_1$, it detects paste gesture. Paste gesture action calculates the paste position as the mid-point of A and D as in Figure 9.

If the hand posture is OP or WO initially, it goes to POINTING state. Copy gesture is the result of OP and then CL. It copies the object pointed by the mid-point of A and D similar to Paste gesture. In the pointing state, if the user opens her fingers more than $\tau_2$, it changes the state as MOUSE DOWN state. It simulates a mouse

down operation. After a WO posture, if OP posture is applied, a click action sends a click message to the operating system.

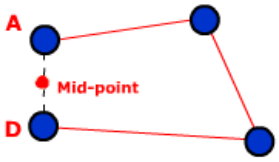



Figure 9**:** Copy/Paste position

## 2.8 Network

For the time being, a simple file transfer operation is handled by the system. There is no clipboard operation at all. For each computer, we created a shared folder accessible by all authenticated users in a local area network. After the user selects a file by copy gesture, the system copies the file to the shared folder. If the user makes the paste gesture in a computer, the system displays the list of all authenticated computers, she picks up one, and a network file copy operation from selected computer to local computer is performed.

However, intelligent agent architecture can be used for an intelligent user interface system. For example, agents can inform each other about which user is using which computer, or which user copied which object. This kind of information can be helpful in tracking of users in multi-computer system. Therefore, if a user copies an object from a computer, and wants to paste this object into another, agent automatically copies object without asking for the source computer.

## 2.9 Summary

In this study, an application, namely ITouch, is being developed to show how vision-based intelligent touch screen is suitable for distributed user interfaces. The purpose of ITouch system is to enable users to (1) employ their hands for the interaction with the monitor, and (2) an intuitive copy-paste operation between

computers. For the time being, only the file transfer is available. A user can grab a file from the monitor of a computer and paste it onto another's one. This is similar to carrying an object from one table to another. All network and file operations are transparent to user.

This study is different in several ways from [15] in which it focuses on camera calibration in virtual touch screen and it is limited to pointing operation for a single computer system. However, our study focuses mainly on hand gesture operations and its usage in a multi-computer environment. In our system, gesture operations are not limited due to continuous partial hand skeleton model. Furthermore, we applied background subtraction unlike [15], which employed color segmentation. Background subtraction is more effective and suitable since the screen does not change during the capture operation. Therefore, several important problems such as background update have been eliminated.

The use of hand gesture in HCI for multi-computer system is a novel idea of this study. This extends the interaction to multiple computers by unifying distributed interfaces into a global one. A gesture can start in a computer system and finish in another. Instead of reconstructing a full hand model, we used a partial hand model since the visible part of the hand is mainly the thumb and index fingers when the camera is positioned at the left side of the user. Since the model is simplified, this leads to real-time performance in model reconstruction.

For copy-paste operation between computers; immersive gestures are designed, for example, copy gesture is similar to grasping an object and paste gesture is dropping an object. This makes copy-paste operation very natural.

This research is our early attempts on hand gesture recognition for multi-computer systems. We believe that vision-based hand gesture systems are very suitable for a user to interact with multi-computer systems in a natural way.

# CHAPTER 3

# HAND POSTURE RECOGNITION

## 3.1 Abstract

Hand is a very convenient interface for immersive human-computer interaction. Users can give commands to a computer by hand signs (hand postures, hand shapes) or hand movements (hand gestures). Such a hand interface system can be realized by using cameras as input devices, and software for analyzing the images. In this hand interface system, commands are recognized by analyzing the hand shapes and its trajectories in the images. Therefore, success of the recognition of hand shape is vital and totally depends on the discriminative power of the hand shape representation. There are many shape representation techniques in the literature. However, none of them are working properly for all shapes. While a representation leads to a good result for a set of shapes, it may totally fail in another one. Therefore, our aim is to find the most appropriate shape representation technique for hand shapes to be used in hand interfaces. Our candidate representations are Fourier Descriptors, Hu Moment Invariant, Shape Descriptors and Orientation Histogram. Based on widely-used hand shapes for an interface, we compared the representations in terms of their discriminative power and speed.

## 3.2 Introduction

Hands play very important role in inter-human communication and we use our hands for pointing, giving commands and expressing our feelings. Therefore, it is reasonable to mimic this interaction in human-computer interaction [26]. In this way, we can make computer usage natural and easier. Although several electro-

mechanical and magnetic sensing devices such as gloves are now available to use with hands in human computer interaction, they are expensive and uncomfortable to wear for long times, and require considerable setup process. Due to these disadvantages, vision based systems are proposed to provide immersive human computer interaction. Vision systems are basically composed of one or more cameras as input devices, and processing capabilities for captured images. Such a system is so natural that a user may not be aware of interacting with a computer system. However, there is no unique vision based hand interface system that can be used in all types of applications. There are several reasons for this. First, there is no computer vision algorithm which reconstructs a hand from an image. This is because a hand has a very complex model with 27 degrees of freedom (DOF) [25]. Modeling the kinematical structure and dynamics are still open problems, and need further research [27] [28] [29]. Second, even if there was an algorithm which finds all 27 parameters of a hand, it would be very complex, and it may not be appropriate for real-time applications. Third, it is unnecessary to use complex algorithms for a simple hand interface, since it consumes considerable or even all available computing power of the system. Appearance-based techniques that analyze the image without using any 3D hand model work faster than 3D model based techniques and they are more appropriate for real time hand interface applications [30][27][22].

This study mainly focuses on appearance-based methods for static hand posture systems. However, the shape representations presented in this study can be incorporated as feature vectors to standard spatio-temporal pattern matching methods such as Hidden Markov Models (HMM) [31][32][33] [34][35] or Dynamic Time Warping (DTW) [36] to recognize dynamic hand movements or hand gestures.

Our initial motivation was to develop an application which was controlled by hand. In this application, the setting was composed of a camera located on top of the desk, and the user gave commands by hand. Although capturing from above limits possible hand shapes, this is very frequent setup for hand interface systems. For example, the system described by Quake *et al.* mimics the behavior of a mouse by a hand on keyboard [11]. ITouch uses hand gestures appearing on the monitor similar

to touch screen monitors [37]. Licsar and Sziranyi present another example that enables a user to manage presentation slides by hands [38]. Freeman *et al.* let the users play games by their hands [10]. Nevertheless, there is no study comparing techniques employed in such a setup. The aim of this thesis is to assess various representations for hand shape recognition system having a setup where a camera is located above the desk and it is looking downward to acquire the upper surface of the hand.

Usually, a hand interface system is composed of several stages: image acquisition, segmentation, representation, and recognition. Among those stages, segmentation is the main bottleneck in developing general usage HCI applications. Although there are many algorithms attempting to solve segmentation such as skin color modeling [18] [39], Gauss Mixture Model (GMM) [40], Background Subtraction [41][42] [43] and Neural Network (NN) methods [44], all of them impose constraints on working environments such as illumination condition, stationary camera, static background, uniform background, etc. When the restriction is slightly violated, a clean segmentation is not possible, and the subsequent stages fail. Remedy to this problem is to use more complex representations or algorithms to compensate the deficiency of segmentation. However, there is also a limit on compensation. As a result, for the time being, even using the state-of-the-art segmentation algorithms for color images, a robust HCI application is not possible. However, there is good news recently on segmentation with a new hardware technology, which is called Time-of-Flight (ToF) depth camera [45]. It captures depth information for each pixel in the scene and the basic principle is to measure the distances for each pixel using the round-trip delay of a light, which is similar to radar systems. This camera is not affected from illumination changes at all. With this technology, clean segmentation for HCI applications is possible using depth keying technique. Microsoft's Natal Project uses this technology to solve segmentation problem and enable users to interact with their body to control games [2]. Therefore, shape representations from clean segmentation can be used with this technology, and

we used clean segmented hand objects in our study. We believe that future HCI applications will be using ToF camera to solve segmentation problem.

The next stage after segmentation is representation, hand pixels are transformed into a meaningful representation which is useful at recognition stage. Representation is very important for recognition since unsuccessful representation gives unsatisfactory results even with state-of-the-art classifiers. On the other hand, good representation always results in an acceptable result with an average classifier [46].

This study compares four representation techniques which can be used in shape recognition systems. In the selection of these representations, the following criterions are used: discriminative power, speed and invariance to scale, translation and rotation. Selected representations are Fourier descriptors, Hu moment invariants, shape descriptors, and orientation histogram. In Section 2, these selected representations are explained in detail. To assess the representations, bootstrapping is used to measure the quality of representations while decision tree is used as the classifier. Section 3 gives all the details concerning tests. In Section 4, we comment on the results in terms of discriminative power and real-time issues. Finally, we conclude the study.

### 3.3 Shape Representation

Recognizing commands given by hand totally depends on the success of shape recognition, and thus, it is closely related to shape representation. Therefore, it is vital to select the appropriate shape representation for hand interfaces. Unfortunately, there is no unique representation that works for all sets of shapes. This is the motivation that leads us to compare and assess popular hand shape representations. Techniques for shape representation can be mainly categorized as contour-based and region based [47]. Contour based techniques use the boundary of the shape while region based techniques employ all the pixels within the shape. Each category is divided into two subcategories: structural and global. Structural methods describe the shape as a combination of segments called primitives in a structural way

such as a tree or graph. However, global methods consider the shape as a whole. Although there are many shape representation techniques in these categories, only some of them are eligible to be used in hand interfaces. We took into account certain criteria for the selection. The first criterion is the computational complexity of finding the similarity of two shapes, i.e. matching process [48]. Contour based and region based structural methods such as polygon approximation, curve decomposition, convex hull decomposition, medial axis require graph matching algorithm [49] for similarity, thus they are computationally complex. Zhang *et al.* show that Fourier Descriptor (FD) which is a contour based global method performs better than Curvature Scale Space (CSS) which is also a contour based global method, in terms of matching and derivation [50]. Another contour-based global method Wavelet Descriptor (WD) requires shift matching for the similarity, which is costly compared to FD. Therefore, we have selected FD as a candidate. Freeman *et al.* use Orientation Histogram which is a region-based global method, for several applications controlled by hand [10] [51]. Since there is no comparison of Orientation Histogram with others, and authors promote it in terms of both speed and recognition performance, we have also chosen it. Peura *et al.* claim that practical application does not need too sophisticated methods, and they use the combination of simple shape descriptors for shape recognition [52]. Since Shape Descriptors are semantically simple, fast and powerful [52][53], we have preferred Shape Descriptors (SD) as well. Each Shape Descriptor is either a contour or a region based global method. Flusser asserts that moment-invariants such as Hu Moment Invariants are important [54] since they are fast to compute, easy to implement and invariant to rotation, scale and translation. Therefore, Hu Moment Invariants are also selected for hand interface.

In conclusion, we have opted for four shape representation techniques; Shape Descriptors, Fourier Descriptors, Hu Moment Invariants and Orientation Histogram to assess their discrimination power and speed on a hand's shape data set. In the rest of this section, we describe each selected method.

### 3.3.1 Shape Descriptors

Shape Descriptor is a quantity which describes a property of a shape. Area, perimeter, compactness, rectangularity are examples of shape descriptors. Although a single descriptor may not be powerful enough for discrimination, a set of them can be used for shape representation [52][53]. In this study, five shape descriptors; compactness, ratio of principal axes, elliptical ratio, convexity and rectangularity are chosen because they are invariant to scale, translation and rotation, and easy to compute also they are reported as successful descriptors in [47][52][53][55].

#### 3.3.1.1.1 Compactness

A common compactness measure, called the circularity ratio**,** is the ratio of the area of the shape to the area of a circle (the most compact shape) having the same perimeter. Assuming *P* is the perimeter and *A* is the area of a hand shape, circularity ratio is defined as follows.

$$\Psi = \frac{4\,\pi\,A}{P^2}$$

For a circle, circularity ratio is 1, for a square, it is $\frac{\pi}{4}$, and for an infinite long and narrow shape, it is 0.

#### 3.3.1.1.2 Ratio of Principal Axes

Principal axes of a given object can be uniquely defined as the segments of lines that cross each other orthogonally in the centroid of the object and represent such directions that cross-correlation of point on object is zero [52]. Ratio of principal axes, ρ provides the information about the elongation of a shape. For a hand's shape boundary *B* which is an ordered list of boundary points, ρ can be determined by calculating covariance matrix $\sum$, of a boundary *B,* and then finding the ratio of $\sum$'s eigenvalues; $\lambda_1$ and $\lambda_2$. Eigenvectors; $e_1$, $e_2$ of $\sum$ are orthogonal and cross-correlation of points in *B* with $e_1$ and $e_2$ is zero since $\sum$ is a diagonal matrix. The values of λ equal to the length of the principal axes. However, to find the ratio of $\lambda_1$ and $\lambda_2$ or principal axes, there is no need to explicitly compute eigenvalues, and ρ can be calculated as follows [52]:

$$\Sigma = \begin{vmatrix} \sum_{xx} & \sum_{xy} \\ \sum_{yx} & \sum_{yy} \end{vmatrix}$$ , where $\sum_{ij}$ represents covariance of *i* and *j*.

$$\rho = \frac{\sum_{yy} + \sum_{xx} - \sqrt{\left(\sum_{yy} + \sum_{xx}\right)^2 - 4\left(\sum_{yy}\sum_{xx} - \sum_{xy}{}^2\right)}}{\sum_{yy} + \sum_{xx} + \sqrt{\left(\sum_{yy} + \sum_{xx}\right)^2 - 4\left(\sum_{yy}\sum_{xx} - \sum_{xy}{}^2\right)}}$$

#### 3.3.1.1.3 Elliptical Ratio

Elliptical Ratio, $\varepsilon$ is the ratio of minor axis, *b* to major axis, *a* of an ellipse which is fitted to boundary points.

$$\varepsilon = \frac{b}{a}$$

Ellipse fitting is performed using a least-square fitness function. In the implementation, ellipse fitting algorithm proposed by Fitzgibbon *et al.* [56] and provided by OpenCV is used [57].

#### 3.3.1.1.4 Convexity

Convexity is the ratio of perimeter of the convex-hull, $\pi_{convexHull}$ to the perimeter of the shape boundary, $\pi_{Shape}$, where convex hull is the minimum convex polygon covering the shape.

$$\varsigma = \frac{\pi_{convexHull}}{\pi_{Shape}}$$

#### 3.3.1.1.5 Rectangularity

Rectangularity measures the similarity of a shape to a rectangle. This can be calculated by the ratio of the area of the hand shape, $A_{handShape}$ to the minimum bounding box of hand shape, $A_{MBB}$. Minimum bounding box (MBB) is the smallest rectangle covering the shape.

$$\Gamma = \frac{A_{handShape}}{A_{MBB}}$$

For a rectangle, rectangularity is 1; for a circle, it is $\frac{\pi}{4}$.

### 3.3.2 Fourier Descriptors

Fourier Descriptors (FDs) represent the spectral properties of a shape boundary. Low frequency components of FDs correspond to overall shape properties; however, high frequency components describe the fine details of the shape.

FDs are calculated using Fourier Transform of shape boundary points, $(x_k,y_k)$, $k=0,\ldots, N-1$ where $N$ is the number of points in the boundary. Boundary can be represented by an ordered list of complex coordinates called complex coordinate signature, as $p_k=x_k+i\ y_k,\ k=0,\ldots,N-1$ or a boundary can be represented by an ordered list of distances, $r_k$, of each boundary point $(x_k,y_k)$ to centroid of the shape $(x_c,y_c)$ called centroid distance signature.

Zhang compared the effect of four 1 dimensional boundary signatures for FDs; these signatures are complex coordinates, centroid distances, curvature signature [58] and cumulative angular function [59]. The authors concluded that FDs derived from centroid distance signature is significantly better than the others. Therefore, we use centroid signature of the boundary. To calculate FDs of a boundary, the following steps are pursued.

- Calculate the centroid of the hand shape boundary
- $x_c = \frac{1}{N}\sum_{k=0}^{N-1} x_k \ \ and\ y_c = \frac{1}{N}\sum_{k=0}^{N-1} y_k$
- Convert each boundary points $(x_k,y_k)$ to centroid distance $r_k$,

$$r_k = [(x_k - x_c) + (y_k - y_c)]^2, k = 0, \ldots, N-1$$

- Use Fourier Transform to obtain FDs.

$$FD_f = \frac{1}{N}\sum_{k=0}^{N-1} r_k \cdot e^{-j2\pi fk/N}$$

*$FD_f$, f=0,...,N-1* are Fourier coefficients.

Calculated FDs are translation invariant since centroid distance is relative to the centroid. In Fourier Transform, rotation in spatial domain means phase-shift in frequency domain so using magnitude values of coefficients make FDs rotation invariant. Scale invariance is achieved by dividing FDs by $FD_0$. Since each $r_k$ is real valued, first half of FDs are the same with second half. Therefore, half of the FDs are enough to represent shape. As a result, a hand's shape boundary is represented as follows [59]:

$$FD = \left[\frac{|FD_1|}{|FD_0|}, \frac{|FD_2|}{|FD_0|}, \cdots, \frac{|FD_{N/2}|}{|FD_0|}\right]$$

### 3.3.3 Hu Moment Invariants, Φ

Hu derived 7 moments which are invariant to translation, rotation and scaling [54][60]. This is why Hu moments are so popular and many applications use them in shape recognition systems. Each moment shows a statistical property of the shape. Hu Moment Invariants can be calculated as follows. Remark that μ shows the 2nd and 3rd order central and normalized moments.

$$\phi_1 = \mu_{20} + \mu_{02}$$
$$\phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$
$$\phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$$
$$\phi_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$
$$\phi_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)$$
$$\phi_6 = (\mu_{20} - \mu_{02})((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$$
$$\phi_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})(3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2)$$

The main problem of Hu moments in classification is the large numerical variances in the values of moment invariants. Therefore, the use of Euclidian distance to compute similarity is not possible. In our implementation, decision tree is used as the classifier.

### 3.3.4 Orientation Histogram

Orientation Histogram (OH) is a histogram of local orientations of pixels in the image [51]. Freeman *et al.* applied the idea of OH to create fast and simple hand interfaces [10]. The basic idea of OH is that hand pixels are very sensitive to illumination, and pixel-by-pixel difference leads to huge error in total. Instead of using pixels themselves for comparison, their orientations are used to overcome the illumination problem. To make it translation invariant, orientations are collected in a histogram. Scale invariance is not pointed in [51]. Our implementation normalized OH to overcome scaling problem. Freeman proposed to train different orientation of the same gesture to overcome the problems due to rotation variance. Instead of using the whole image, its dimension is reduced to about 100 by 80 pixels for fast computation [51]. The problems of the method are also reported as two similar shapes can produce very different histograms, and hand shape must not be the small part of the image.

## 3.4 Experimental Results

To evaluate the performance of 4 shape representations, we collected 10160 samples of widely-used 15 hand shapes from 5 different people. There is

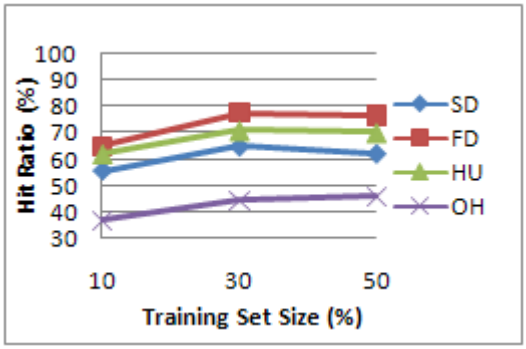
approximately the same number of samples for each person and hand commands. A sample set of collected 15 hand shapes are shown in Figure 10, and Appendix A shows hand shape database in more detail. The evaluation is based mainly on discrimination power and speed. Furthermore, we have also investigated the performance with respect to the number of people and samples in the training set.

Figure 10: Hand commands used in our experiments.

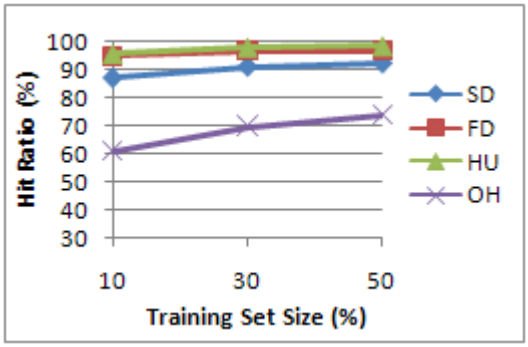
We have first divided the samples into two sets: training set and test set. Training set is used to train a decision tree for each representation, and samples in the test set are classified by the corresponding decision tree. Hit ratio, which is the percentage of correctly classified samples in the test set, is used as the measurement for discrimination power of each representation. The division of training and test set is based on two parameters: number of people in the training set, and the percentage of the samples of each person selected for training.

We first assessed the effect of training set size when the system is trained with samples only from one person. We selected randomly one person among 5 people, and used 10% of samples from each hand command of the selected person in training. All the remaining samples, that is, remaining 90% samples of the same person and all samples from other 4 people, were employed in test set. This test is repeated 5 times and the average of the hit ratios is used as the measurement of discrimination power for each representation. This procedure is repeated with 30% and 50% of samples in training for one person, and Figure 11.a shows the results

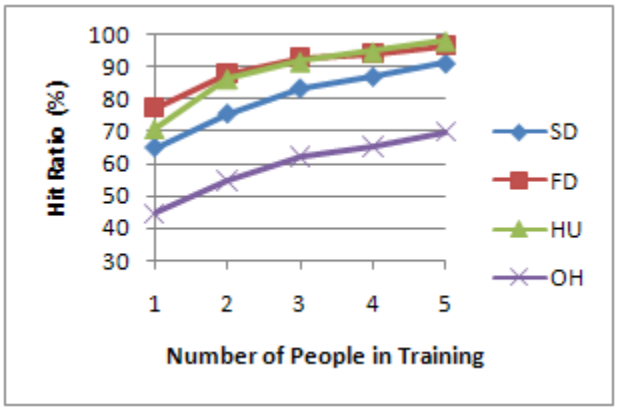
graphically. We repeated the above procedure for 2, 3, 4, and 5 people, and the results for 5 people is given in Figure 11.b. Table 1 presents all of the results.




(a)




(b)




(c)

Figure 11: Hit ratios for (a) randomly selected one person for training, (b) all 5 people used in training, (c) 30% of samples of selected number of people for training.

Figure 11.a and Figure 11.b show the hit ratios of representations when 10%, 30% and 50% of samples from the people used in training. It is observed that the performances of SD, FD and HU are not influenced considerably by increasing the number of samples from the same person. However, OH is getting better when it is trained with more samples. This is because OH is not rotation invariant, and it

memorizes the rotated instances of hand shapes. As a result, SD, FD and HU representations produce low variances for the representations of similar hand shapes from the same person. This is a desired property since a few training samples from a person are adequate to train the system for that person.

Figure 11.c indicates the influence of number of people in training on hit ratio. It is apparently observed that hit ratios of all representations improve drastically when the number of people participating in training increases. Thus, it is reasonable to use many people to train a hand interface system which uses one of four shape representations described in this chapter.

Our aim is to find out the best of four representations in terms of discrimination power and speed. According to Figure 11, FD and HU outperform SD and OH, and the results of FD and HU are very close to each other in terms of hit ratio. FD is slightly better than HU.

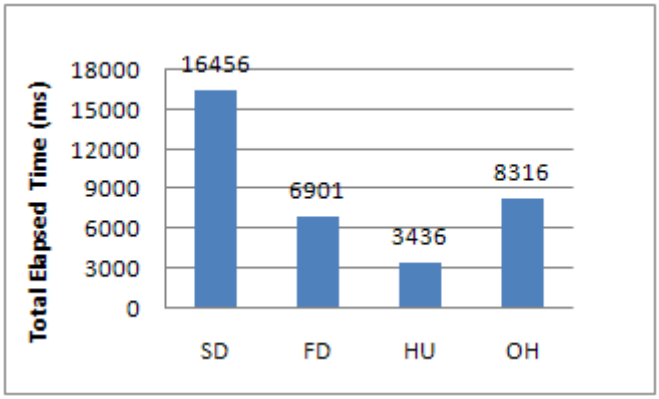



Figure 12: Running times of 4 shape representations for 10160 samples.

Real-time is an indispensable requirement for a hand interface system. Noticeable delay prevents user from immersive usage of the system. Therefore, we analyzed the running time performance of representations. To measure running time performances of each representation, we computed the total elapsed time of each representation for all samples. Figure 12 shows the results of total elapsed time of 10160 samples for each representation. According to the results, HU is the fastest

method among these four representations. Remark that FD is also a relatively fast method. On the average, calculation of a HU or FD representation of a segmented image is less than a millisecond with a Pentium IV – 3GHz computer with 1GB RAM of memory.

In conclusion, both HU and FD provide acceptable results in terms of discrimination power and speed, and they can be used for computer systems which employ static hand shapes as commands.

## 3.5 Summary

The main component of a hand interface is the part that recognizes the given hand command. Shape is an important property of a hand, and it can be used for representation of a hand. In this chapter, four representation techniques; Shape Descriptors (SD), Fourier Descriptors(FD), Hu Moment Invariants (HU) and Orientation Histogram (OH) are selected, and compared in terms of their discrimination power and speed. When forming the test environment, a widely-used hand interface setup is chosen. In this setup, a camera is located above the desk in such a way that its viewing direction is to the top of the desk. In the experiments, there are totally 10160 samples of 15 different hand commands from 5 people. Those samples are divided into training set and test set. Each sample in training set is converted to four representations, and each representation is used to train a decision tree classifier. These classifiers are used to recognize the samples in test set. In this way, hit ratios of decision trees are obtained and used as the measurements for discrimination power. Furthermore, running times of calculating the representations are accumulated and used as the measurement of speed for each representation. According to test results, HU and FD outperform SD and OH in terms of both discrimination power and speed. Therefore, HU and FD are reasonable to use in hand shape recognition systems as posture recognizer or as a feature vector to a spatio-temporal pattern recognizers such as HMM.

Table 1: Hit ratios for all parameters used in the experiment.

| Parameters | | Hit Ratios (%) | | | |
|---|---|---|---|---|---|
| Num. of people | Training Size (%) | SD | FD | HU | OH |
| 1 | 10 | 55.63 | 65.08 | 62.1 | 37 |
| | 30 | 64.99 | 77.5 | 70.97 | 44.76 |
| | 50 | 62.2 | 76.45 | 70.16 | 46 |
| 2 | 10 | 75.47 | 86.09 | 85.19 | 47.33 |
| | 30 | 75.54 | 88.11 | 86.48 | 54.8 |
| | 50 | 75.97 | 87.9 | 85.6 | 56.14 |
| 3 | 10 | 81.05 | 89.29 | 89.81 | 52.85 |
| | 30 | 83.44 | 92.98 | 91.8 | 62.29 |
| | 50 | 82.73 | 93 | 91.1 | 63.58 |
| 4 | 10 | 83.89 | 93.56 | 93.86 | 57.16 |
| | 30 | 87.05 | 94.12 | 94.82 | 65.37 |
| | 50 | 87.07 | 94.05 | 94.27 | 67.97 |
| 5 | 10 | 87.29 | 96.84 | 95.73 | 61.08 |
| | 30 | 91.18 | 98.64 | 98.13 | 69.73 |
| | 50 | 92.35 | 99.61 | 98.58 | 73.57 |

# CHAPTER 4

# MOTION AND SPATIO-TEMPORAL QUERY INTERFACE

## 4.1 Abstract

Using one's hands in human-computer interaction increases both the naturalness of computer usage and the speed of interaction. One way of accomplishing this goal is to utilize computer vision techniques to develop hand-gesture-based interfaces. A video database system is one application area where a hand-gesture-based interface is useful, because it provides a way to easily specify certain queries. This study presents a hand-gesture-based query interface for a video database system to specify motion and spatio-temporal object queries. We use a regular, low-cost camera for image acquisition. Images from the camera are processed to detect and track hands and their configurations. By utilizing the proposed hand detection and tracking method, we are able to specify different types of queries involving motion and spatio-temporal relations between objects. To measure the effectiveness of the proposed interface we prepared a set of queries, including spatio-temporal, trajectory and camera motion queries, and conducted a user study. The users were trained on the interface first and then specified the queries using the proposed interface and a mouse-based interface. They compared the two interfaces in terms of different usability parameters, including ease of learning, ease of use, ease of remembering, naturalness, comfortable use, satisfaction and enjoyment. The study shows that querying video databases is a promising application area for hand-based interfaces, especially for queries involving motion.

## 4.2 Introduction

Convenience of the interaction device is one of the main criteria for effective human-computer interaction. There is no one interface suitable for all applications; the one that makes users the most effective and comfortable is the most appropriate. For example, a steering wheel interface for a car simulation system perfectly fits to its objectives, whereas a mouse and/or keyboard interface would not result in the desired usability. Therefore, ensuring the most suitable interface makes a considerable difference in many ways.

Since the advent of computers, because there have been no options, people have been used to accomplish their tasks on a computer system using a mouse and/or keyboard whether it works well or not. However, this situation has been recently changing with advances in computer technology in terms of both processing power and price. For example, the human hand is very promising interface device for certain applications. In fact, hand interfaces are supersets of mouse-based interfaces because the capabilities of a mouse (i.e., point and click) can be easily simulated by a hand. Hand-based systems, however, have poor pixel accuracy due to the generally low resolutions of cameras. Fortunately, some applications, such as the query interface in this study, do not need precise pixel accuracies for object placement and path drawing tasks.

At first, electro-mechanical gloves that track palm and the fingers were used for hand interaction, but the method was uncomfortable, expensive and complex. Advances now enable interface designers to use vision-based solutions for interfaces. In this system, one or more cameras are placed in an indoor or outdoor environment, and the images acquired from those cameras are processed by vision algorithms. These algorithms determine the command performed by the user. This system does not require any wearable or attached electro-mechanical equipment, that is, it is a passive interface. Furthermore, it needs only an inexpensive camera for image acquisition and it provides the user with a completely immersive interaction experience. However, there are some problems with vision-based systems. Firstly, a stable, robust vision system has not yet been developed; many vision applications

work under some constraints. The lack of a robust, real-time segmentation algorithm with no constraints is the main source of bottlenecks. To overcome this problem, many designers impose constraints on the scene, such as static background, controlled environment or markers for users. Secondly, further research is needed to determine the appropriate applications for hand interactions.

This study contributes to the first problem by a novel vision-based interface system using two hands. It contributes to the second problem by investigating whether querying a video database system is an appropriate application for a hand-based interface. The idea is inspired by observing in BilVideo-7 [61], a recently developed, MPEG-7 [62] compatible video indexing and retrieval system, that motion and spatio-temporal object queries are difficult to formulate using a mouse-based visual query interface. Users can formulate text, color, texture, shape, location, motion and spatio-temporal queries on the visual query interface of BilVideo-7, whose composite query interface can be used to specify complex queries containing any type and number of video segments with their descriptors. The query interface would be easier and more intuitive to use if a hand-based interface complementary or alternative to the mouse-based interface were provided for motion and spatio-temporal object queries.

The aim of this research is to propose a hand-gesture-based interface and investigate the convenience and usability of this interface for querying video databases. The proposed vision-based system employs one webcam with a resolution of 320x240, located above the monitor and looking down to the keyboard to see the hand movements (Figure 13).

Due to the segmentation problems in vision-based systems, the user wears a different colored glove on each hand. This is the only constraint of our system. Time-of-Flight (ToF) cameras can determine the depth of objects in a scene for short distances (from 5 cm to 7 meters) [63], and after applying depth thresholding, segmented objects can be obtained robustly. However, the cost of ToF cameras is currently prohibitive. With colored gloves we can use a color segmentation algorithm to quickly detect hand regions from a camera image. The feature vector of the hand

as four shape descriptors, namely, compactness, rectangularity, axis ratio and convexity parameters, are calculated from the segmented hand regions. As the main data for the decision tree classifier, they are used for the training and recognition process. In this way, the vision system can determine the position, orientation, size and shape of the hands.

The proposed system consists of interfaces for four different query types: (1) spatial queries, (2) motion trajectory queries, (3) motion trajectory with temporal relation queries and (4) camera motion queries. We conducted a user study to evaluate the system. The users perform sample queries for each query type and fill out a questionnaire. The results are evaluated to judge the suitability of a hand-gesture-based interface for querying a video database.

The chapter is organized as follows: in Section 2, literatures on hand-based systems are reviewed; the various vision-based hand interfaces, application types and their successes are explained. Section 3 describes low level image processing for detecting and tracking hands, from camera capture to classification. Section 4 explains the proposed hand-based query formulation interface. Section 5 describes the testing environment, test users, test procedure, test queries and the questionnaire in detail and discusses the test results. Finally, Section 6 concludes the paper.

## 4.3 Related Work

Although the idea of using hands in human-computer interaction is promising, it is not a panacea for all applications. Appropriate applications for hand interfaces should be determined by considering users' assessments on usability criteria. There are several ways to integrate hands into human-computer interaction. Some authors preferred using electro-mechanical gloves for the stability reason noted above [32][63][64]. We, however, investigate only the vision-based hand interfaces.

Pointing is an essential activity in our daily lives, and many authors focused on simulating this activity in human-computer interaction with hands [11][37][65]. One of the earliest such applications was Queck's FingerMouse, which mimics the pointing behavior of a conventional mouse with hands. With the FingerMouse, the

user moves his index finger over the keyboard to move the mouse cursor on the screen [11]. The ITouch system improves Queck's system in two ways: (1) the user points to the monitor, which is more natural than pointing over the keyboard, and (2) the user can perform many hand postures, that is, he is not limited to pointing [37].

Some authors decided to use hand interaction in a projected environment. For example, Licsar and Sziranyi present an application that enables a user to manage presentation slides using hands [66]. Wellner designed DigitalDesk, where the computer display is projected onto a desk; a user employs his bare fingers to interact with projected objects such as a calculator. In this way, he merged a real desk, virtual objects and finger interaction [67]. Crowley *et al.'s* FingerPaint, a finger tracking application, draws objects on a workspace projected with an overhead projector [13].

Through their vision system, Hardenberg *et al.* demonstrated three applications that use hand interaction: a 2D drawing application, a presentation control system and an application for arranging positions of virtual items on a projected surface with one's hand during brainstorming sessions [14]. Oka *et al.* designed an augmented desk interface that tracks multiple fingertips using an infrared (IR) camera. Similar to DigitalDesk, their system detects trajectories and predefined finger gestures performed by users to interact with projected objects on a desk [68].

Controlling external devices with hands is also a promising application area [10][44] [69]. Freeman *et al.* developed a system that lets viewers control their television sets with hands instead of with a remote control [10]. Yin and Xie implemented a hand based interface to enable users to control the movement of a humanoid service robot [44].

Computer game applications are very appropriate for hand interactions to control objects in a game [10][12][70][2] [71]. For instance, Microsoft's Natal project is promising technology that uses a ToF camera to detect hand, arm and body motions and gestures. A user can control games using his hand and/or body gestures. This interaction, free from all external devices, results in immersive gaming experiences [2]. Segen and Kumar's GestureVR recognizes static hand postures.

Through this work, the authors demonstrated that one can easily play first-person shooting games with bare hands. For short periods, this interface provides a natural and enjoyable interaction experience, but after a while, muscle fatigue disturbs the user [12].

Several studies presented applications which recognize sign languages [72][33][73]. Aran *et al.* introduced an interactive sign language teaching system for deaf people [72]. Starner and Pentland presented a system to recognize sentence-level continuous American Sign Language (ASL) [33]. Watanabe *et al.* designed a system that recognizes 26 letters of the English alphabet using a specially designed colored glove [73]. Licsar and Sziranyi have developed a search-and-retrieval system for Hungarian folk songs stored in Tillarom, a comprehensive collection of original Hungarian folk songs, using Kodaly's hand signs [66].

Kolsch and Turk implemented a vision-based hand gesture interface, HandVu, to detect and track users' hands mainly for wearable, mobile applications. Speed and robustness against moving cameras, dynamic backgrounds and changing lighting conditions are distinguishing features of this system [74].

Hand interaction is suitable in medical applications as well [64][75]. Wachs *et al.* designed the Gestix application, a vision-based hand interface system allows doctors' hands to remain sterile while they are navigating and manipulating images in electronic medical records databases [75].

From stereo images, Sepehri *et al.* proposed a vision-based algorithm to estimate the position and orientation of a hand with respect to a camera. To demonstrate their system's abilities, they implemented a 3D model construction application based on the user's hand movement [76]. Rehg and Kanade calculated 3D hand parameters from stereo images to track a hand for a 3D mouse application [24]. Wang and Popovic developed a system that can reconstruct a hand pose from a single image of the hand wearing a specially designed multi-colored glove [77].

Determining the appropriate application for a hand interface is vital, and there is no hand interface proposed for querying a video database system. The aim of this study is to fill this gap.

## 4.4 The Proposed Vision-based System

### 4.4.1 System Architecture

In our hand-based interface, the user formulates queries using her right and left hands without wearing any electro-mechanical devices. To achieve this, we propose a vision-based system to detect, track and recognize the user's hands in low-resolution image sequences captured by a web camera. The basic setup of the system is shown in Figure 13, where a camera above the monitor looks down to the keyboard. The height of the camera determines the available working space of the user. The user wears two different solid-colored gloves for the left and right hands to enable a real-time and robust vision-based application that can run in complex and dynamic environments with changing lighting conditions.

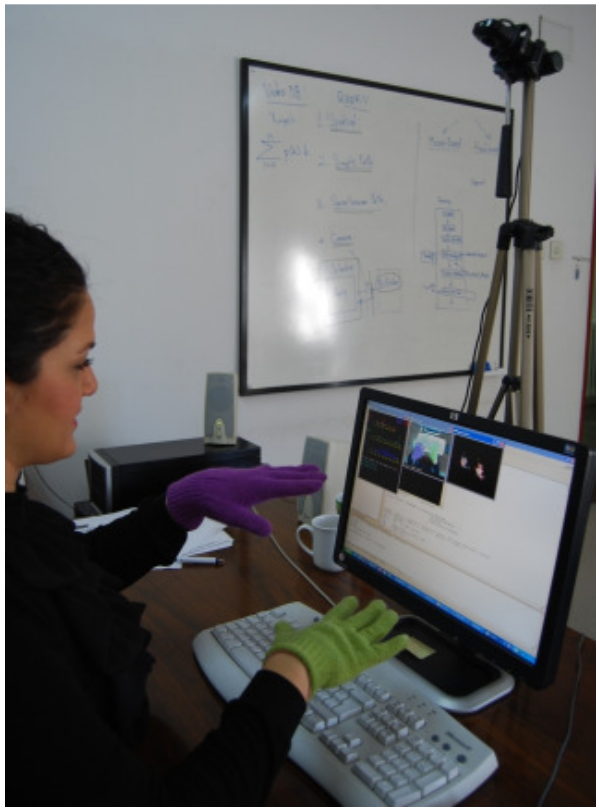


Figure 13: System Setup.

The proposed system is designed as a two-layer architecture: vision and application. The vision layer is a low-level image processing layer that provides the application layer with the left and right hand's properties. The vision layer has four modules: *Setup, Training, Detection* and *Recognition* (Figure 14).

### 4.4.2 Hand Segmentation

The setup module is used to determine the parameters of the color segmentation. The user selects a rectangular hand region on the image to see the histograms of the hue, saturation and lightness (HSL) channels (Figure 15). In this way, the boundaries of each channel in the HSL color space can be determined automatically or manually. A Gaussian low pass filter is used for smoothing the histograms to reduce the noise.

The detection module determines which hand is available and converts available hand regions to meaningful information. This module has two components: segmentation and feature extraction. Segmentation is the lowest level processing to determine which pixels belong to a hand. It merges these individual pixels into regions called `hand blobs' by using connected component analysis.

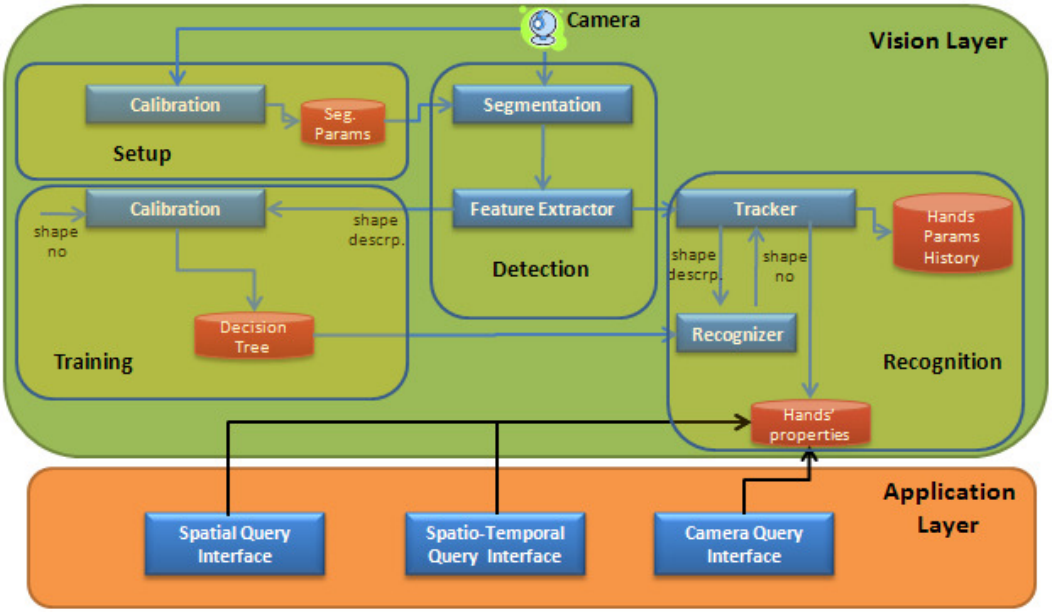
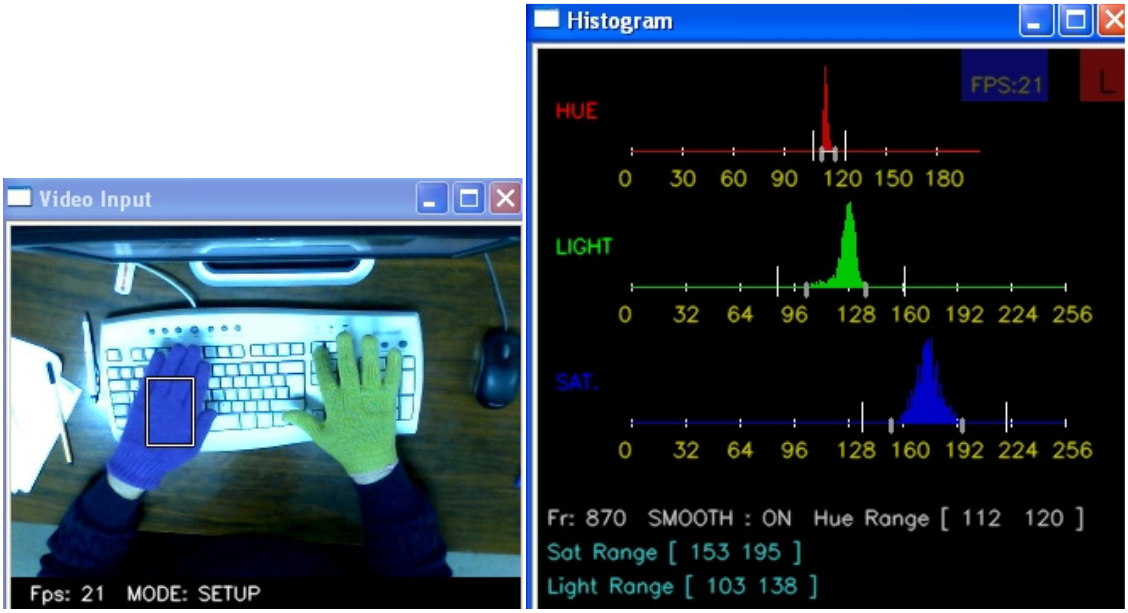



Figure 14: System Architecture.

There are only few segmentation algorithms executing in real time, namely; background subtraction [40] [21] and color segmentation [18] [39] [78]. Background subtraction aims to generate a background model that represents the background of the scene using many background images, and it subtracts the current frame from the background model. Although this algorithm works well for static backgrounds, it does not result in good segmentation when the environment conditions change. Therefore, we decided to use color segmentation, which works robustly under varying conditions. The only restriction is that the user must wear colored gloves.




(a) (b)

Figure 15: Setup Module (a) marking glove pixels with a rectangle,(b) corresponding HLS histogram.

The color segmentation used in our system is based on modeling gloves in HSL color space. Color samples marked by the user (Figure 15a) are used to model the glove as a rectangular prism in HSL space by adjusting the boundaries of the HSL channels using the histogram interface (Figure 15b). Boundaries are

automatically found by scanning the color histogram, but the interface enables the user to modify the boundaries manually as well. Six parameters per glove, for the upper and lower boundaries for each channel, therefore twelve parameters in total, are stored as the main model parameters for color segmentation.

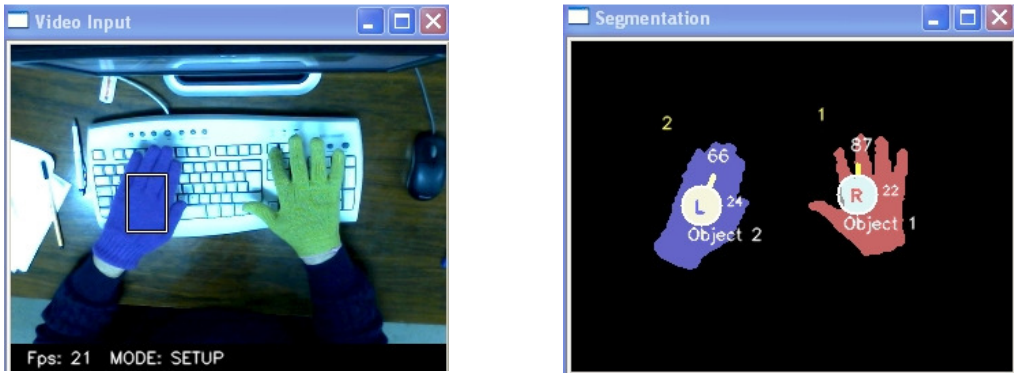
Initially, each pixel in the current image acquired from the camera is converted from Red, Green, and Blue (RGB) to HSL color space, and tested against the modeled rectangular volume. If the pixel is inside the volume, it is counted as a possible hand pixel. This test operation for two gloves (two color classes) requires twelve comparisons. However, linear thresholding to test if a pixel is in one of two color classes (as in our system) can be implemented by two AND operations, similar to the method used in [78], which encodes a linear color model as three arrays (one array for each channel in YUV color space). This reduces the computation from twelve comparisons with ten AND operations to three array lookups and two AND operations.

After detecting the pixels possibly belonging to the gloves, white noise pixels are eliminated by using a 3x3 median filter. The remaining pixels are grouped into regions by 8-connected component analysis to form hand blobs. The hand blob with the highest area for each glove is selected as the hand region. The other regions are determined to be noise, and eliminated.

### 4.4.3 Hand Shape Representation

Hand blobs are represented by their position, orientation, size and shape descriptors. Hand position is the center of gravity of the hand blob. The orientation of the hand blobs is calculated by fitting 2D ellipses to the hand blobs using least-squares fitting [57]. The major axis shows the direction of the hands, and size is the ratio of the hand blob area. Figure 16 shows an input image and the result of its segmentation. L and R indicate the left and right hand, respectively. The line from L and R shows the orientation of the hand and the length of this line is proportional to the hand size. The numbers to the upper left of the hand indicates the hand's shape

index. For example, the right hand is in an open posture, which in training has been labeled as shape index 1.




(a) (b)

Figure 16: (a) input image (b) corresponding segmentation result.

Hand shape is one of the most important features for hand interfaces. The aim is to recognize a trained hand posture from its blob. There are many shape representation and description techniques in the literature [47][79], however, none of them works perfectly for all sets of shapes. The shape descriptors method is widely used to represent hand shapes. It is easy to implement, fast and very robust. Therefore, the feature extraction component of this system computes four descriptors to represent hand blobs for recognition purposes: compactness, axis ratio, convexity and rectangularity. These descriptors are invariant to translation, rotation and scaling [80]. Many states of the same hand shape are used in shape training, as shown in Figure 17.

Figure 17: Different size, orientation and position of the same hand shape in training.

*Compactness* is the ratio of the area of the shape to the area of a circle (the most compact shape) having the same perimeter. *Axis ratio* is the ratio of the minor axis to the major axis of an ellipse that is fitted to boundary points. *Convexity* is the ratio of the perimeter of the convex-hull to the perimeter of the hand shape boundary, and *rectangularity* measures the similarity of a shape to a rectangle.

### 4.4.4 Hand Shape Recognition

The detection module provides the training and recognition modules with the hands' position, orientation, size and shape descriptors. The training module requires two inputs: an identification number for each shape, and its corresponding shape descriptors from the detection module. Based on hand shape samples, the training module calculates the parameters of the classifier, which in our case is a decision tree.

From methods among neural network, k-nearest neighbor, Adaboost classifiers and support vector machines, we decided to use a decision tree to classify hand shapes for its simplicity and efficiency [80]. This is a supervised learning system where the user specifies a shape number for a given hand posture in training and using the decision tree, the recognition module returns the shape number of the given hand shape. We tested the eight hand shapes shown in Figure 18 to measure the system's discriminative ability, and it recognizes these shapes successfully.

Figure 18: Some of the possible hand postures in a hand-based interface

After the training module has constructed the decision trees for the left and right hands, the recognition module uses them to classify the shape descriptors of the

current hand blobs. In other words, it finds the corresponding shape index (index of the hand posture) for each hand blob.

The recognition module is the last module that tracks hand properties and finds their shape indices. It has two components: *tracker* and *recognizer*. The tracker is responsible for keeping histories of both hands' properties. Furthermore, it gets all properties of the hands from the feature extractor, and estimates the current properties of hands based on the previous and current hand properties. Especially when one hand occludes the other, there is a need to estimate the properties of the occluded one. Tracking objects based on their features is a huge area in computer vision literature. Comaniciu *et al.* proposed a system which tracks non-rigid objects in clutter background using their color and texture distribution by a mean-shift analysis [81]. Shi and Tomasi showed that corner points of an object are good features to track instead of its color distribution [82]. Avidan turns tracking problem into a classification problem using AdaBoost classifiers [83] [84]. Bradski uses the color distribution of the object as the feature to track and searches the distribution in a window using continuously adaptive mean shift algorithm, namely, CAMShift algorithm [85].

Our tracking algorithm uses OpenCV's CAMShift algorithm to detect the hands in the current frame with an adaptive window size. This enables us to find hand pixels faster by ignoring the pixels outside the search window. In case of the lost of a hand object, tracker module rescans the whole image.

There are many options for the estimation of the hand properties in case of occlusions. The Kalman filter is a unimodal estimator that compromises a measured or observed value and an estimated value calculated by a linear model, proportional to their uncertainties. Another estimator is particle filtering. It tracks multiple hypotheses by sampling an unknown state distribution, unlike the Kalman filter, which assumes a Gaussian distribution of state variables [86]. The particle filter requires a considerable amount of computation due to its large number of particles [87]. In our case, real time performance is a vital constraint, and the Kalman filter is selected to track the position, size and orientation of hand blobs. In the case of

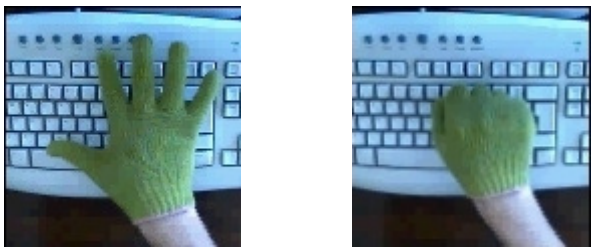
occlusion, the previous hand shape index is taken as the current one, assuming hand shape has not changed.

The tracker sends shape descriptors of the present hands to the recognizer, which finds the corresponding hand posture, i.e., shape index, from the decision tree constructed by the training module.

To sum up the process, an image is segmented into hand regions, hand properties, such as position, orientation and size, are extracted, and the shape index is found by the help of a decision tree. Using the appropriate API calls of the vision layer, hand-based applications can access any calculated property of a hand for the current frame. The vision layer's performance is measured in several applications, using a Pentium PC with 1.7 GHz CPU and 1 GB Memory, with a Microsoft VX 2000 webcam. It can run at a rate of 20-22 frames per second.

## 4.5 Hand-Based Query Formulation

Our hand-based interface covers four types of queries: *spatial, motion trajectory, motion trajectory with spatio-temporal relations* and *camera motion*. For each type of query, a separate hand-based query component is implemented by means of the functionalities in the vision layer (Figure 19).



(a) (b)

Figure 19: The two postures used in our tests (a) Posture #1 (b) Posture #2

For all query interfaces, only two hand postures are used to form queries (see Figure 19). The meanings of hand postures differ in each query interface. However,

the user can define any two hand shapes for postures #1 and #2 in the system setup stage.

The spatial query defines a scene where an object's position, size and orientation are determined by the user. The user's two hands represent two objects, and the user can specify two objects in the scene by the appropriate hand postures. Posture #1 describes the properties of the objects in the scene (Figure 20a); after posture #1, the user performs posture #2 to add the object itself into the scene (Figure 20b).

For example, Figure 20 shows the user's hands representing *man* and *airplane* objects, with their properties (taken from the size, shape and orientation of the hands), shown in Figure 20c. The yellow line shows the object's orientation, and the line's length represents the object's size. The blue hand in posture #2 adds the *airplane* with the given properties in Figure 20a into the scene. In this example, the following query is sketched by our hand interface: ``A man at the right side of the scene is looking at an airplane moving diagonally upward to the right."

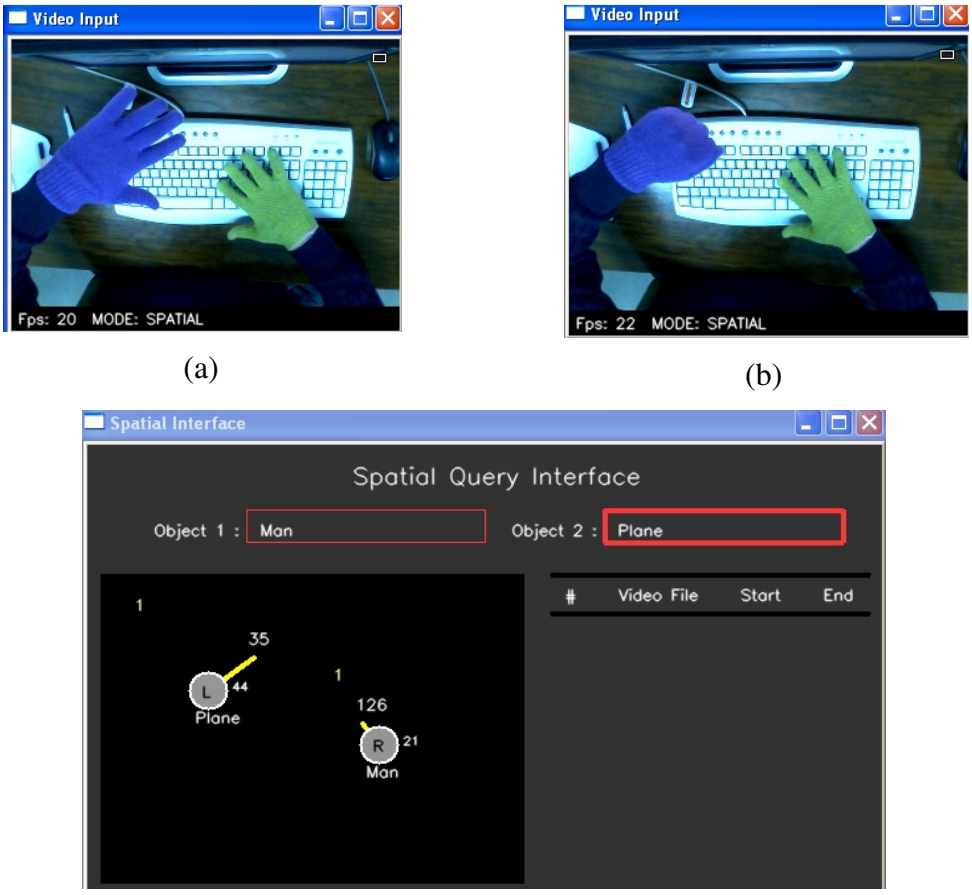
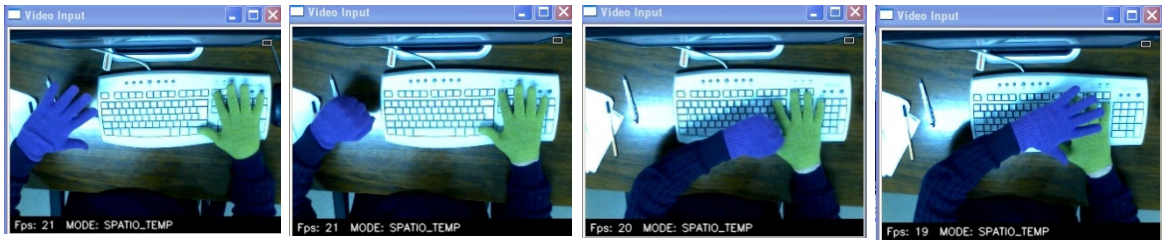


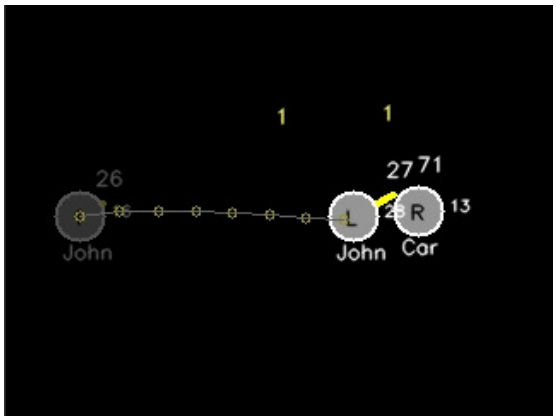

Figure 20: (a) Initial hand configuration, (b) "Add object" hand posture for *airplane,* (c) Hand properties in the scene.

The motion trajectory query is used to search for objects by their trajectories (paths). To add an object and its path, posture #1 is performed for the initial configuration of the object. After that, the user makes posture #2 and draws the path. Figure 21a through Figure 21d show how to add a moving object in a sequence of hand configurations. The blue glove represents *John* in this query, and the user places *John* in his initial position in Figure 21a, She marks the beginning of the path by setting posture #2 for *John* in Figure 21b, then draws his path in Figure 21c and

finishes the path by showing posture #1 again in Figure 21d. The result of those hand movements can be seen in Figure 21e, where *John* is added to the scene with his path.




(a) (b) (c) (d)




(e)

Figure 21: (a)-(d) Sequence of hand configurations, (e) Specify an object with its path.

In the implementation, this interface uses the two hands' properties, provided by the vision layer. The logic behind detecting the start and end of the path is achieved by using a finite state machine, shown in Figure 22. The machine has two states; $S_1$ is the state for the start and end of the path, and $S_2$ shows the path/motion drawing state. Initially, the user moves her hand in posture #1 to the starting position of the path. To start drawing, she performs posture #2 for consecutive 10 frames (approximately half a second) to eliminate temporal noises in shape recognition. This causes a transition from $S_1$ to $S_2$, marking the starting point of the path. In $S_2$, while

the hand is in posture #2, all hand positions are added to the path. To end the path, the user holds posture #1 for 10 consecutive frames.

The timing of the two paths is very important, especially when defining the collision of two objects. This kind of query requires specifying the objects' motion trajectories and their spatio-temporal relations. Using two hands at the same time can solve timing issues, formulating such queries naturally and efficiently. Figure 23 shows a query formulation for the collision of two cars sing the hand interface.

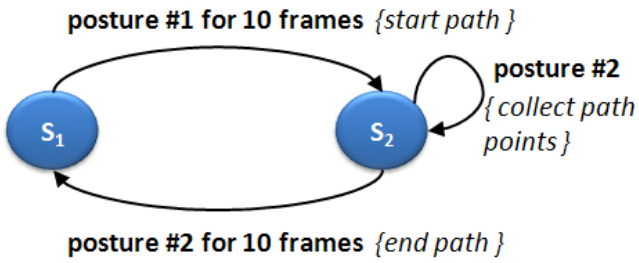
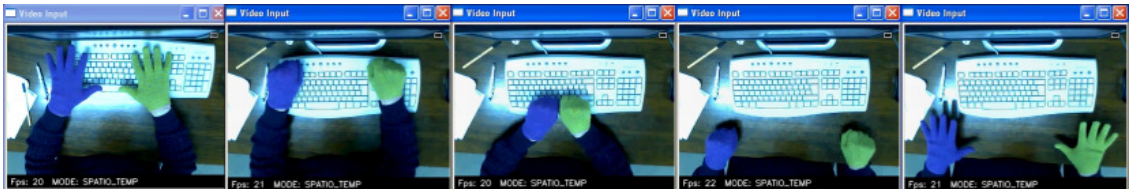
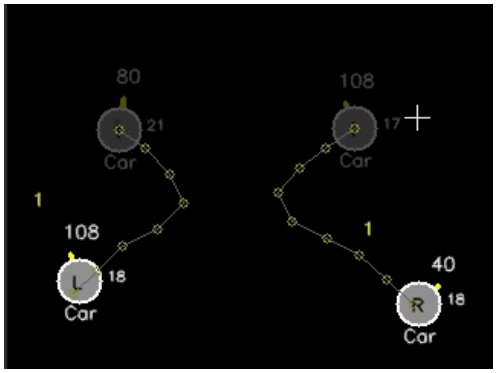



Figure 22: Finite State Machine for adding path.



(a)

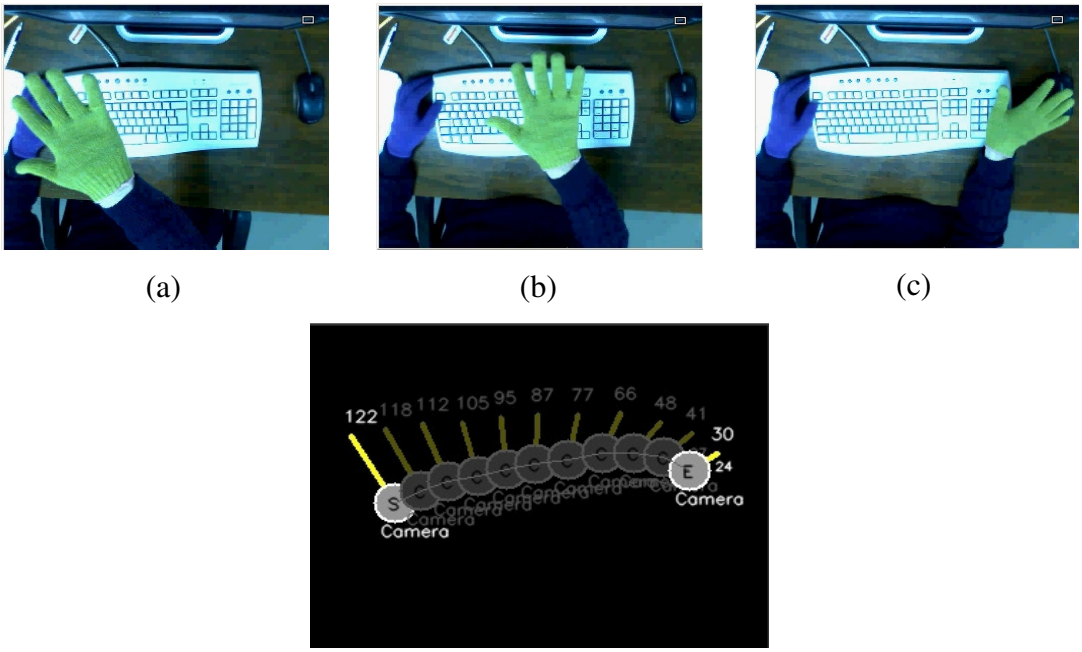



(b)

Figure 23: (a) Sequence of hand configurations for collision, (b) Defining a collision query.

The last type of query that can be formulated by our hand-based interface is called the camera motion query. A hand represents a camera, and its position is specified by moving the hand in spatial coordinates. Zoom in/out is realized by moving the hand downward and upward relative to the camera (Figure 24) and hand orientation determines the camera view direction.

A sample camera query could be ``Find all movies where a camera is moving from left to right while it is panning from left to right and zooming in." Figure 24 shows how to formulate this query using a hand. The hand (or the camera) is placed at the start point, as in Figure 24a, then, while moving her hand from left to right, the user moves her hand down and changes its orientation from left-facing to right-facing as in Figure 24(b,c). As can be seen from this example, a user can easily convert a textual query into hand movements. The distinguishing characteristic of this interface is that the user can specify many properties, e.g., location, size and orientation, at the same time. Such a query is very difficult to formulate using a mouse-based interface.




(a) (b) (c)

(d)

Figure 24: (a)-(c) Hand poses for camera query, (d) query definition.

## 4.6 Performance Evaluation

We conducted a user study to assess the usability of this system in terms of *usefulness, learning, remembering, naturalness, comfort, satisfaction* and *enjoyment*. Test users evaluated each parameter using a scale from 1 (strongly disagree) to 5 (strongly agree). Furthermore, the interface's performance in completing a query was measured in terms of the number of trials and elapsed time [88]. To compare our hand-based interface with a mouse-based interface (which is described in Section 4.6.1), both interfaces were tested by 12 users.

The test users were chosen from regular computer users who had never experienced this kind of application. The test users were first given a tutorial on how to use the hand- and mouse-based interface. Then they were asked to complete 10 given queries of four query types. For each query type, they filled out a questionnaire on attitude parameters as in Appendix B. A performance criteria questionnaire was filled in by the supervisor who observed the users' number of trials and the time needed to complete the query.

### 4.6.1 Mouse-based Interface

To enable users to compare the two interfaces in terms of usability and performance criteria, a simple mouse-based interface was implemented. The user specifies a spatial query by drawing rectangles representing objects in the scene and adjusting their properties by rotating and/or resizing (Figure 25a). To draw an object's trajectory, the user pressed the left mouse button and moved the mouse accordingly. Releasing the left mouse button marks the end of the trajectory. A trajectory is represented by a series of spatio-temporal points, each showing the timestamp in milliseconds as the path is being drawn (Figure 25b). For camera motion queries, the user draws rectangles and specifies their properties to indicate the camera in motion (Figure 25c). Figure 25 defines a query that the camera is moving from left to right while rotating from left to right and zooming out. Cameras 1 and 2 indicate the first and last states of the camera, respectively, and relative rectangle

sizes indicate ``zoom in" and ``zoom out". For complicated camera motion queries, more rectangles may be required.


(a) (b) (c)

Figure 25: Mouse-based Interface for (a) spatial, (b) motion trajectory, (c) camera motion queries.

### 4.6.2 Test Queries

Each user performed the following 10 queries for four query types using the mouse- and hand-based interfaces.

*Type 1: Spatial Queries*

1. Two objects are side by side, with the left one smaller than the right one. They are facing each other.
2. Two objects are under one another, with the upper one looking right, and the lower one looking left.

*Type 2: Motion Trajectory Queries*

3. An object is moving from left to right on a linear path.
4. An object is moving from right to left on a sinusoidal path.
5. Two objects are moving, with the left one moving from top to bottom, and the right one moving from bottom to top.

*Type 3: Motion Trajectory with Temporal Relation Queries*

6. Two objects are colliding at the center of the page. After collision, they move in opposite directions.

*Type 4: Camera Motion Queries*

7. The camera moves from left to right while it is zooming out.
8. The camera zooms in then zooms out.
9. The camera pans right to left while moving left.
10. The camera moves right while rotating from left to right and zooming in then zooming out.

### 4.6.3 Test Results

After performing each type of query, the test user fills out a questionnaire. The results from the 12 test users are consolidated for each query type and showed in the following figures. In Appendix C, all results are available in tabular form.

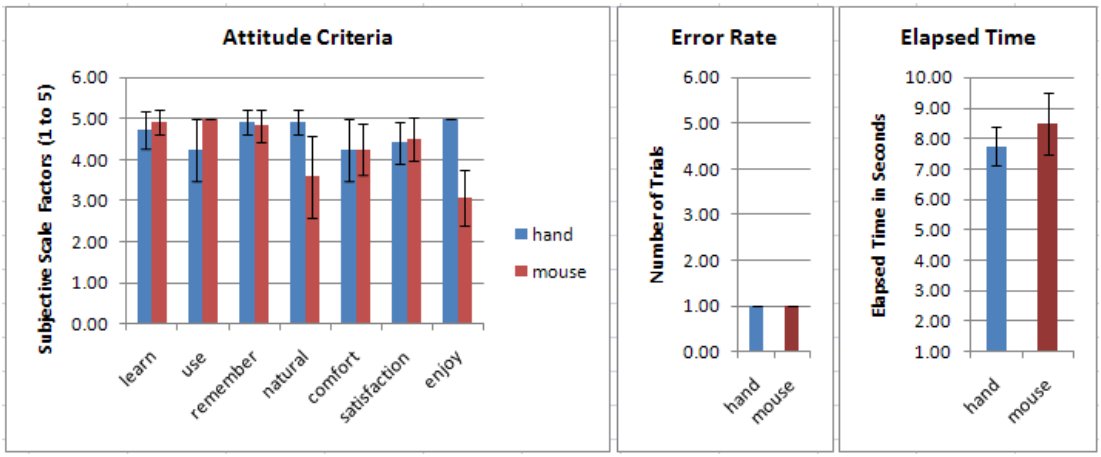



Figure 26: Spatial query evaluation.

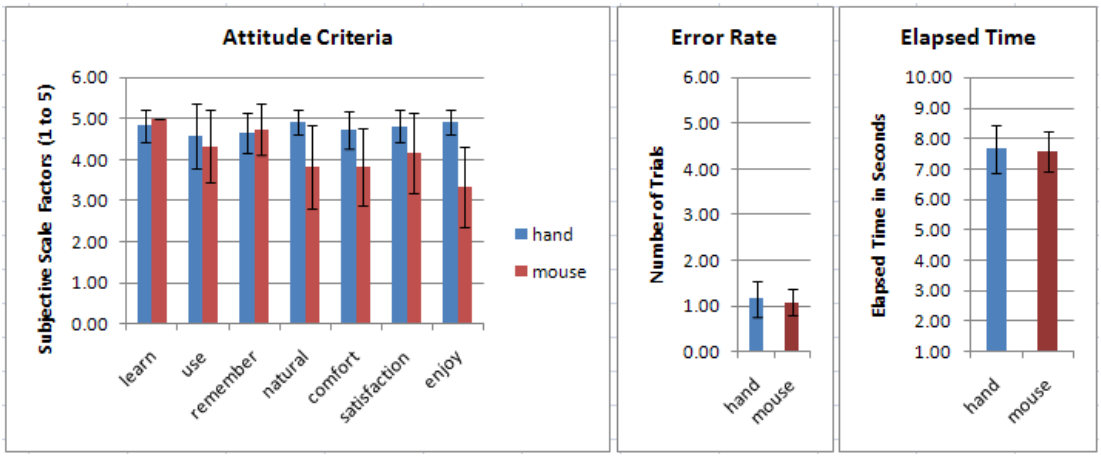



Figure 27: Motion Trajectory query evaluation.

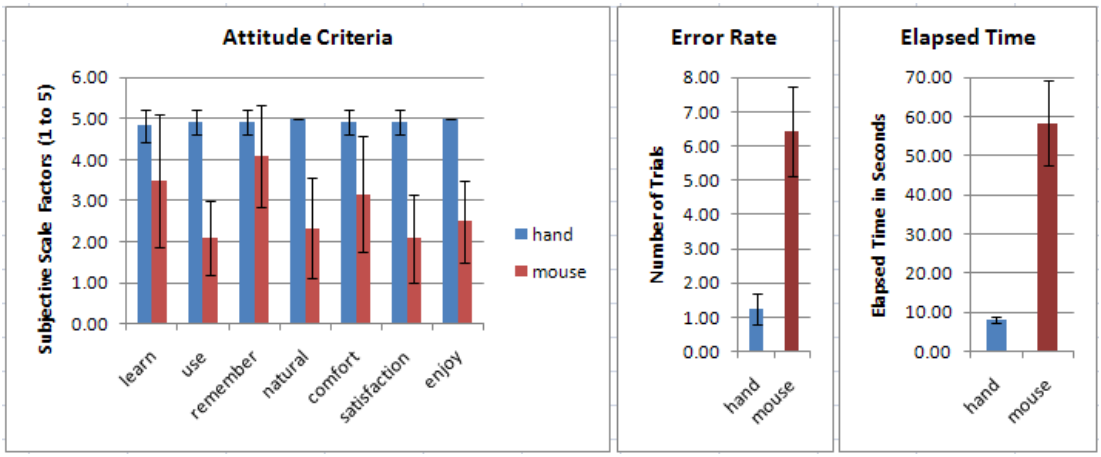



Figure 28: Motion Trajectory query with temporal relation evaluation.

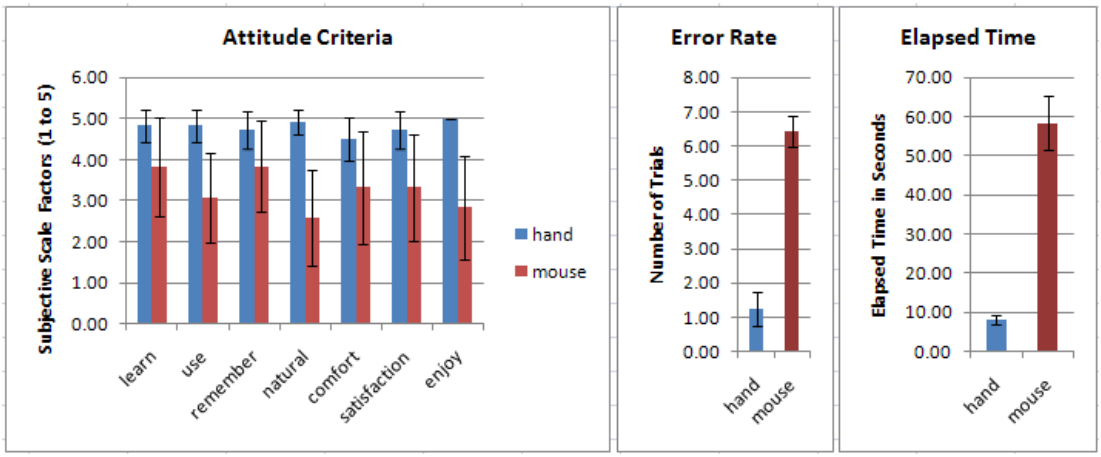



Figure 29: Camera Motion query evaluation.

## 4.7 Discussion of Results

In this section, we discuss the results of the experiments for each query type individually. All the results for the attitude and performance criteria are discussed based on their mean and standard deviation. The mean of a parameter indicates the average of users' assessments, and the standard deviation measures the degree of the users' agreement according to the attitude and performance criteria. We perform single factor ANOVA test with 0.05 alpha value for each criterion to find if the difference between hand- and mouse-based interface is statistically significant or not.

Referencing Figure 26 for the spatial query type, test users agree that the hand- and mouse-based interfaces are both easy to use, easy to learn, easy to remember and comfortable. However, the hand-based interface is more natural and enjoyable to use, while the mouse-based interface is a little easier to use. According to users' comments in the questionnaires, they are used to interacting with a computer through a mouse device, which is why they find the mouse-based interface easier to use. In terms of performance, there is no noticeable difference between two interfaces. Furthermore, the standard deviations are very small for all attributes, which shows that people are in fairly good agreement according to the criteria. Using ANOVA test for naturalness criterion, the hand-based interface ($\mu$=3.91, $\sigma$=0.083)

and the mouse-based interface ($\mu$=3.58, $\sigma$=0.99) result in F=19.8, p=0.0002 and $F_{crit}$=4.3. Since F is far larger than $F_{crit}$, and p is less than 0.05, this shows that their difference is statistically significant. Similar to naturalness criterion, ANOVA also finds that the difference of enjoy criterion in two interfaces is statistically significant, F=98.6, p=$1.3x10^{-5}$, $F_{crit}$=4.3. On the other hand, the differences in all other criteria are not statistically significant. Briefly, there is no outperforming interface between the two spatial query interfaces overall.

Figure 27 shows the results of the motion trajectory query types, which are similar to the spatial query results. Drawing an object's path with the hand-based interface is evaluated as “strongly agree" in terms of comfort, satisfaction and natural usage. Other criteria are similar in both interfaces based on ANOVA tests. The mouse-based interface is stated as being a little easier to learn and remember due to users' familiarity with the mouse device. Measured performances between the two interfaces in terms of time and number of trials are similar, and their differences are not statistically significant. Nevertheless, test users are somewhat biased towards the hand-based interface due to its feeling more natural.

Query Type 3 (motion trajectory with spatio-temporal relation) shows big differences between the two interfaces in attitude and performance evaluations. Specifying two paths while including temporal relations between objects is very difficult in the mouse-based interface because the user must draw each path separately while adjusting for the correct timing. In the hand-based interface, the user takes advantage of having two hands to represent the two objects in the scene. Moving two hands at the same time is easier than drawing two paths with a mouse at separate times. People had trouble with the timing issue in the mouse-based interface. Referencing Figure 28, it is obvious that the hand-based interface is much easier to use, more comfortable, more satisfying, more natural and more enjoyable than the mouse-based interface. ANOVA tests also confirm that differences are statistically significant for all criteria. The performance results are also very different in both the number of trials, and in the elapsed time. To adjust for the temporal relations with the mouse-based interface, users tried an average of six times in 58

seconds to perform correct queries, compared to one trial in eight seconds with the hand-based interface. The hand-based interface thus outperforms the mouse-based interface in simultaneous path queries in terms of both usability and performance criteria.

In the mouse-based interface, the user adds camera objects with different properties (position, size and orientation) one by one in sequence to define the camera motion. The user must decide on the number of camera objects and their properties before using the mouse-based interface. However, in the hand-based interface, one hand representing the camera object is enough to provide all camera properties. Based on the test results of Query Type 4, users find the hand-based interface considerably more natural (Figure 29). In all attitude criteria, the hand-based interface is preferred by users. ANOVA tests show that differences in all criteria are statistically significant. In camera motion queries with the mouse-based interface, users spend too much time (an average of 30 seconds) thinking about the number of camera objects and their properties. The standard deviation for this query is noticeably large, which indicates that performance varies significantly among the test users. Using the hand-based interface, users performed the camera motion queries in nine seconds. In short, compared to the hand-based interface, the mouse-based interface is not suitable for camera motion queries.

In summary, the two interfaces provide similar usability and performance results for Query Types 1 and 2, but the hand-based interface is generally more natural, easier to use, and enjoyable than the mouse-based interface. For Query Types 3 and 4, the hand-based interface is significantly more preferable in all aspects.

## 4.8 Summary

We proposed a novel vision-based system to formulate motion and spatio-temporal object queries for a video retrieval system, which, to the best of our knowledge, is the first in the literature. We defined four query types to be formulated by our hand-based interface. The usability and performance of each query type are

experimented with and compared by test users on a mouse-based and our hand-based interface, on the *BilVideo-7* video retrieval system. The results showed that the spatial and trajectory query interfaces give approximately the same results, but the hand-based interface is slightly more natural and enjoyable than the mouse-based interface. For motion trajectory queries that include spatio-temporal relations of two objects (such as collision) and camera motion queries, the hand-based interface outperforms the mouse-based interface in performance and usability. This research concludes that querying video databases is a promising application area for hand-based interfaces, especially for queries involving motion and spatio-temporal relations.

# CHAPTER 5

# CONCLUSION

Computers have become indispensible in our daily life and business activities. The wide range of applications cannot be handled efficiently by a small number of interfaces such as mouse and keyboard. Appropriate interfaces should be developed for natural and efficient interaction.

Humans take advantage of their hands to interact with their environments. Therefore, it is logical to use hands in computer interaction for similar activities as well. For example, pointing an object in real life is realized by our hands, and fingers. This can be simulated in computer interaction to provide familiar interaction for the people.

The main goal of this thesis is to propose appropriate methods to achieve novel vision-based systems with three important constraints. The first one is the real-time constraint, where the system should respond to the user activities without any delay. Robustness is the second constraint. This means the hand interface should execute in any complex environment. It should not fail in case of the changes in illumination or camera movement. The last one is the high recognition rate of the system, where hand interface should recognize given hand commands correctly.

Touch screen monitors are useful for limited number of natural interactions such as pointing and clicking. However, user may need many hand commands, other than pointing and clicking, such as turning page; zoom in and out, or rotating an object without touching the screen. To do this, this thesis proposed a novel way of interaction with the screen without any physical connection. This is called “intelligent touch screen” or ITouch. In Itouch, a camera, which is located a place

where it sees the screen completely, tracks the user's hand and its actions. Based on the images captured from the camera, Itouch recognizes the user's hand commands. In this way, it creates an interaction volume rather than a touching surface in touch screen monitors. This interface is useful in multi-computer systems as well. One part of an interaction may start in one computer and continue in another one. In other words, user may need to interact with several computers to complete her task, called *distributed computer interaction*. Itouch system enables this kind of distributed interactions. For instance, a copy-and-paste operation between two computers are simulated by grasping an object in one computer's screen by performing grasping hand gesture and pasting on another computer's screen. We proposed the concept of *distributed interaction*, and a feasible hand-based interface solution, Itouch, to realize this concept.

Among several crucial problems of vision systems, one of them is the recognition of hand shapes in the image. Depending on the sequence of hand shapes, hand interface generates appropriate commands to the system. Shape recognition is mainly composed of shape analysis and classification. The success and performance of recognition stage largely depends on the representation of the hand shape, and there is still no *de facto* standard for shape representation. This thesis compared four different general shape representation methods on a hand's shape database generated using an overhead camera. The recognition rate of all methods and the speed of all execution times are measured by using about 10000 samples for 15 hand postures with 5 different people. Among the following methods, shape descriptors, Fourier descriptors, Hu moments and orientation histogram; Fourier descriptors and Hu moments outperform others in term of discrimination power and speed. Nevertheless, Fourier descriptor is preferred since it is easier to implement for real-time vision-based hand interface systems.

We also describe a hand-based interface for an application that queries a video database. A vision-based architecture is proposed to realize this interface which runs in real-time and in any complex environment. Based on BilVideo video database system, four kinds of query types are designed to form spatial queries,

motion trajectory queries, motion trajectory with spatio-temporal relation queries, and camera motion queries. In the hand interface, users wear different colored-gloves on each hand in order to enable the hand interface to differentiate left and right hands, and it also makes the system robust against the environmental changes such as changes in illumination, and movements of the camera. Hand-based and mouse-based interfaces designed for this application are experimented by a group of test users with respect to usability and performance criteria. The results show that users found no difference between two interfaces in spatial and motion trajectory queries with the exception that they felt more natural in hand-based interaction. However, the users evaluated that motion trajectory with spatio-temporal relation and camera motion queries were more natural, easier, enjoyable and efficient in hand-based interface than in mouse-based interface. Results showed that forming complicated queries are far more suitable with hand-based interfaces.

The novelties of this thesis are (i) nobody proposed turning a computer screen into an input device that does not need any physical contact of the user before. This results in a very natural and rich way of interactions. (ii) For a general hand interface setup, the best hand shape representation method in terms of speed and discrimination power is determined among many methods. There is no shape comparison study on hand shape databases in the literature before. This study gives interface designers an initial starting point in the recognition stage of their hand interfaces. (iii) A novel vision-based interface system that recognizes two hands of a user is proposed. This is robust against changes in the environment, fast to response in real-time, and able to recognize hand shapes correctly. Many researchers use bare hands which is very natural for the interaction in their applications, but bare-hand interfaces work only in controlled environments, and not suitable for general usage. This is why we preferred using colored gloves which cause slightly less natural but very robust vision-based system. Therefore, if the robustness is the issue for the hand device, our proposed hand system is a good choice. (iv) A new application for hand-based interface, *querying video database application*, is proposed first time in the literature. Four query types and corresponding hand gestures are determined, and the

suitability of the hand interface is assessed after conducting usability and performance tests.

Vision-based systems still suffer from unstable image processing algorithms. For hand interfaces, especially segmentation process should be more stable and faster to detect hand pixels within the image. There is no solution to robust segmentation algorithm yet. However, a new technology called ToF (time-of-flight) camera can produce the depth of each pixel in the image very fast and robustly. This camera can be used as the input device in hand interfaces. Now, their prices are not affordable, but they have been going down dramatically. In the near future, interface devices will take advantage of using ToF cameras as their input devices. In this way, there will be no reason to use colored gloves to create robust and stable hand interfaces. As a future study, our proposed system can utilize ToF cameras for its segmentation stage.

Use of hand-based interfaces in applications is relatively a new research area. Revealing the potential applications of hand-based systems and the hand gestures which are suitable for the application, need further research. Moreover, the techniques to assess the quality of hand gestures is also a challenging problem in hand interface designs. Proposing new application areas that need distributed interactions are also a future study of this thesis.

Finally, we have described hand interfaces where the hand is the main device in the interaction. The proposed vision-based hand systems in this thesis prove that real-time and robust hand-based interfaces are not imaginary or fictional in the near future, and it will not be surprising that many devices, machines and applications will be controlled by users' hands in a natural way.

## REFERENCES


[1] CyberGlove. Company Web Site. [Online]. http://www.cyberglovesystems.com/ last visited on March 2010.

[2] Microsoft. (2010) Project NATAL. [Online]. http://www.xbox.com/en-US/live/projectnatal, last visited on March 2010.

[3] M. E. Erdem, I. A. Erdem, V. Atalay, and A. E. Çetin, "Computer Vision-based Unistroke Keyboard System and Mouse for the Handicapped," in *Proc. of IEEE International Conference on Multimedia and Expo*, 2003, pp. 765-768.

[4] R. Kjeldsen and J. Hartman, "Design Issues for Vision-based Computer Interaction Systems," in *Proc. of ACM Workshop on Perceptive User Interfaces*, 2001.

[5] I. V. Pavlovic, R. Sharma, and T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction:Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677-695, 1997.

[6] M. Turk and G. Robertson, "Perceptual User Interfaces," *Communications of the ACM*, vol. 43, no. 3, pp. 33-34, 2000.

[7] A. van Dam, "Post-WIMP User Interfaces," *Communications of the ACM*, vol. 40, no. 2, pp. 63-67, 1997.

[8] S. Oviatt and P. Cohen, "Multimodal Interfaces That Process What Comes Naturally," *Communications of the ACM*, vol. 43, pp. 45-53, 2000.

[9] J. W. Davis and S. Vaks, "A Perceptual User Interface for Recognizing Head Gesture Acknowledgements," in *Proc. of the Workshop on Perceptive User Interfaces*, 2001, pp. 1-7.

[10] W. T. Freeman, et al., "Computer Vision for Interactive Computer Graphics," *IEEE Computer Graphics and Applications*, vol. 18, no. 3, pp. 42-53, 1998.

[11] F. Quek, T. Mysliwiec, and M. Zhao, "FingerMouse: A Freehand Pointing Interface," in *Proc. of International Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 372-377.

[12] J. Segen and S. Kumar, "Shadow Gestures: 3D Hand Pose Estimation Using a Single Camera," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999, pp. 479-485.

[13] J. Crowley, F. Berard, and J. Coutaz, "Finger Tracking as an Input Device for Augmented Reality," in *Proc. of International Workshop on Gesture and Face Recognition*, 1995, pp. 195-200.

[14] C. von Hardenberg and F. Berard, "Bare-Hand Human-Computer Interaction," in *Proc. of the Workshop on Perceptive User Interfaces*, 2001, pp. 1-8.

[15] Z. Zhang, "Vision-based Interaction with Finger and Papers," in *Proc. of International Symposium on the CREST Digital Archiving Project*, 2003, pp. 83-106.

[16] J. Rekimoto, "Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments," in *Proc. of User Interface Software and Technology*, 1997, pp. 31-39.

[17] G. Wolberg, *Digital Image Warping*. IEEE Computer Society Press, 1992.

[18] V. Vezhnevetsi, V. Sazonov, and A. Andreeva, "A Survey on Pixel-Based Skin Color Detection Techniques," *Graphicon*, pp. 85-92, 2003.

[19] C. Zhang and P. Wang, "A New Method of Color Image Segmentation Based on Intensity and Hue Clustering," in *Proc. of International Conference on Pattern Recognition*, 2000, p. 3617.

[20] M. J. Jones and J. M. Rehg, "Statistical Color Models with Application to Skin Detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81-96, 2002.

[21] T. Horprasert, D. Harwood, and L. Davis, "A Statistical Approach for Real-Time Robust Background Subtraction and Shadow Detection," in *Proc. of IEEE*

*International Conference on Computer Vision*, 1999, pp. 256-261.

[22] Y. Wu and T. S. Huang, "Hand Modeling, Analysis, and Recognition," *IEEE Signal Processing Magazine*, vol. 18, no. 3, pp. 51-60, 2001.

[23] J. Lin, Y. Wu, and S. T. Huang, "Capturing Human Hand Motion in Image Sequences," in *Proc. of IEEE Workshop on Motion and Video Computing*, 2002, pp. 99-104.

[24] J. Rehg and T. Kanade, "Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking," *Lecture Notes in Computer Science*, vol. 801, pp. 35-46, 1994.

[25] J. J. La Viola, "A Survey of Hand Posture and Gesture Recognition Techniques and Technology," Department of Computer Science, Brown University, Technical Report CS-99-11, 1999.

[26] M. Turk, "Perceptive Media: Machine Perception and Human Computer Interaction," *Chinese Journal of Computers*, vol. 23, no. 12, pp. 1235-1244, 2000.

[27] A. Erol, G. ,. Bebis, R. D. Boyle, and X. Twombly, "A Review on Vision-based Full DOF Hand Motion Estimation," *Computer Vision and Image Understanding*, vol. 108, pp. 52-73, 2007.

[28] S. U. Lee and I. Cohen, "3D Hand Reconstruction from a Monocular View," in *Proc. of International Conference on Pattern Recognition*, 2004, pp. 310-313.

[29] R. Hassanpour, S. Wong, and A. Shahbahrami, "Vision-Based Hand Gesture Recognition for Human-Computer Interaction: A Review," in *Proc. of International Conference on Virtual Reality Continuum and its Applications in Industry*, 2009, pp. 157-162.

[30] J. Martin and J. L. Crowley, "An Apperance-based Approach to Gesture-Recognition," in *Proc. of International Conference on Image Analysis and Processing*, 1997, pp. 340-347.

[31] F. S. Chen, C. M. Fu, and C. L. Huang, "Hand Gesture Recognition Using a

Real-time Tracking Method and Hidden Markov Models," *Image and Vision Computing*, vol. 21, pp. 745-758, 2003.

[32] Y. Nam and K. Wohn, "Recognition of Space-Time Hand-Gestures Using Hidden Markov Model," in *ACM Symposium on Virtual Reality Software and Technology*, 1996, pp. 51-58.

[33] T. Starner, J. Weaver, and A. Pentland, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371-1375, 1998.

[34] G. Rigoll, A. Kosmala, and S. Eickeler, "High Performance Real-Time Gesture Recognition Using Hidden Markov Model," in *Proc. of International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, 1997, pp. 69-80.

[35] O. Aran and L. Akarun, "Üç Boyutlu El Hareketlerinin Tanınması için İki Boyutlu Özniteliklerin Birleştirilmesi," in *14th Signal Processing and Communications Applications (SIU)*, 2006.

[36] S. Nakagawa and H. Nakanishi, "Speaker-Independent English Consonant and Japanese Word Recognition by a Stochastic Dynamic Time Warping Method," *Journal of Institiution of Electronics and Telecommunication Engineers*, vol. 34, no. 1, pp. 87-95, 1988.

[37] S. Genç and V. Atalay, "ITouch: Vision-based Intelligent Touch Screen in a Distributed Environment," in *International Conference on MultiModal Interfaces, Doctoral Spotlight*, 2005.

[38] A. Licsar and T. Sziranyi, "Dynamic Training of Hand Gesture Recognition System," in *International Conference on Pattern Recognition*, 2004, pp. 971-974.

[39] Wu.Y, Q. Liu, and S. H. Huang, "Robust Real-Time Human Hand Localization by Self-Organizing Color Segmentation," in *Proc. of International Conference*

*on Computer Vision*, 1999, pp. 161-166.

[40] C. Stauffer and W. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999, pp. 246-252.

[41] Y. Ivanov, A. Bobick, and J. Liu, "Fast Lighting Independent Background Subtraction," *International Journal of Computer Vision*, vol. 37, pp. 49-55, 1998.

[42] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting Moving Objects, Ghosts, and Shadows in Video Streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1337-1342, 2003.

[43] A. Cavallaro, E. Salvador, and T. Ebrahimi, "Detecting Shadows in Image Sequences," in *Proc. of First European Conference on Visual Media Production*, 2004, pp. 165-174.

[44] X. Yin and M. Xie, "Finger Identification and Hand Posture Recognition for Human-Robot Interaction," *Image and Vision Computing*, vol. 25, no. 8, pp. 1291-1300, 2007.

[45] A. Kolb, E. Barth, and R. Koch, "ToF-sensors: New Dimensions for Realism and Interactivity," in *IEEE Computer Vision and Pattern Recognition, Workshop on ToF-Camera based Computer Vision*, 2008, pp. 1-6.

[46] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Willey-Interscience, 2000.

[47] D. Zhang and G. Lu, "Review of Shape Representation and Description Techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1-19, 2004.

[48] C. R. Veltkamp, "Shape Matching: Similarity Measures and Algorithms," in *Proc. of Shape Modeling International*, 2001, pp. 188-197.

[49] J. Triesch and C. von der Malsburg, "Classification of Hand Posture Against Complex Backgrounds Using Elastic Graph Matching," *Image and Vision Computing*, vol. 20, pp. 937-943, 2002.

[50] D. Zhang and G. Lu, "Content-based Shape Retrieval Using Different Shape Descriptors: A Comparative Study," in *Proc. of IEEE Conference on Multimedia and Expo*, 2001, pp. 317-320.

[51] W. T. Freeman and M. Roth, "Orientation Histograms for Hand Gesture Recognition," in *Proc. of International Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 296-301.

[52] M. Peura and J. Livarinen, "Efficiency of Simple Shape Descriptors," in *Aspects of Visual Form*, 1997, pp. 443-451.

[53] P. L. Rosin, "Measuring Shape: Ellipticity, Rectangularity, and Triangularity," *Machine Vision and Applications* , vol. 14, no. 3, pp. 172-184, 2003.

[54] J. Flusser, "Moment Invariants in Image Analysis," *World Academy of Science, Engineering and Technology*, vol. 11, pp. 196-201, 2006.

[55] R. Poppe and M. Poel, "Comparison of Silhouette Shape Descriptors for Example-based Human Pose Recovery," in *IEEE Conference on Automatic Face and Gesture Recognition*, 2006, p. 541–546.

[56] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct Least Square Fitting Ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476-480, 1999.

[57] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media Inc., 2008.

[58] J. R. Parker and Z. Zhou, "Object Recognition Using Signatures," in *Proc. of The IASTED International Conference on Signal and Image Processing*, 1998, pp. 238-243.

[59] D. Zhang and G. Lu, "A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures," in *Proc. of International Conference on Intelligent Multimedia and Distance Education*, 2001, pp. 1-9.

[60] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Transactions on Information Theory*, vol. 8, pp. 179-187, 1962.

[61] M. Bastan, H. Cam, U. Gudukbay, and O. Ulusoy, "An MPEG-7 Compatible Video Retrieval System with Intergrated Support for Complex Multimodal Queries," *IEEE MultiMedia*, in press..

[62] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*. Wiley, 2002.

[63] A. Sherstyuk, D. Vincent, and C. Jay, "Sliding Viewport for Interactive Virtual Environments," in *International Conference on Artificial Reality and Telexistence*, 2008, pp. 175-182.

[64] E. Zudilova-Seinstra, et al., "Evaluation of 2D and 3D Glove Input Applied to Medical Image Analysis," *International Journal of Human-Computer Studies*, in press.

[65] P. Nesi and A. Del Bimbo, "A Vision-based 3D Mouse," *International Journal of Human Computer Studies*, vol. 44, no. 1, pp. 73-92, 1996.

[66] A. Licsar, T. Sziranyi, L. Kovacs, and B. Pataki, "A Folk Song Retrieval System with a Gesture-Based Interface," *IEEE MultiMedia* , vol. 16, no. 3, pp. 48-59, 2009.

[67] P. Wellner, "Interacting with Paper on the DigitalDesk," *Communications of the ACM*, vol. 36, no. 7, pp. 87-96, 1993.

[68] K. Oka, Y. Sato, and H. Koike, "Real-time Fingertip Tracking and Gesture Recognition," *IEEE Computer Graphics and Applications*, vol. 22, no. 6, pp. 64-71, 2002.

[69] A. Malima, E. Özgür, and M. Çetin, "A Fast Algorithm for Vision-Based Hand Gesture Recognition for Robot Control," in *IEEE Conference on Signal Processing and Communications Applications*, 2006, pp. 1-4.

[70] C. Manresa, J. Varona, R. Mas, and F. Perales, "Real-time Hand Tracking and Gesture Recognition for Human-Computer Interaction," *Electronic Letters on Computer Vision and Image Analysis*, vol. 5, no. 3, pp. 96-104, 2005.

[71] S. C. Crampton and M. Betke, "Counting Fingers in Real-Time: A Webcam-

based Human-Computer Interface with Game Applications," in *Proc. of the Conference on Universal Access in Human-Computer Interaction*, 2003, pp. 1357--1361.

[72] O. Aran, et al., "SignTutor: An Interactive System for Sign Language Tutoring," *IEEE Multimedia*, vol. 16, pp. 81-93, 2008.

[73] K. Watanabe, Y. Iwai, Y. Yagi, and M. Yachida, "Recognition of Sign Language Alphabet Using Colored Gloves," *Systems and Computers in Japan*, vol. 30, no. 4, pp. 48-59, 1999.

[74] M. Kolsch and M. Turk, "Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition, Workshop on Real-Time Vision for Human-Computer Interaction*, 2004, p. 158.

[75] J. Wachs, et al., "A Gesture-based Tool for Sterile Browsing of Radiology Images," *Journal of the American Medical Informatics Association*, vol. 15, no. 3, pp. 321-323, 2008.

[76] A. Sepehri, Y. Yacoob, and L. Davis, "Employing the Hand as an Interface Device," *Journal of Multimedia*, vol. 1, no. 7, pp. 18-29, 2006.

[77] R. Y. Wang and J. Popovic, "Real-Time Hand-Tracking with a Color Glove," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, pp. 1-8, 2009.

[78] J. Bruce, T. Balch, and M. Veloso, "Fast and Inexpensive Color Image Segmentation for Interactive Robots," in *Proc. of International Conference on Intelligent Robots and Systems*, 2000, pp. 2061-2066.

[79] T. Heap and F. Samaria, "Real-Time Hand Tracking and Gesture Recognition Using Smart Snakes," in *Interface to Real and Virtual Worlds*, 1995, pp. 261-271.

[80] S. Genç and V. Atalay, "Which Shape Representation is the Best for Real-Time Hand Interface System?," in *International Symposium on Advances in Visual*

*Computing : Part I*. Springer, 2009, pp. 1-11.

[81] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 142-149.

[82] J. Shi and C. Tomasi, "Good Features to Track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593-600.

[83] S. Avidan, "Ensemble Tracking," Mitsubishi Electric Research Laboratories, Technical Report TR2005-065, 2005.

[84] Y. Freund and Y. V. Schapire, "A Decision Theoretic Generalization of Online Learning and Application to Boosting," in *Proc. of European Conference on Computational Learning Theory*, 1995, pp. 23-37.

[85] G. R. Bradski. (1998) Intel Technology Journal. [Online]. http://developer.intel.com/technology/itj/q21998/articles/art_2.htm, last visited on March 2010.

[86] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1-45, 2006.

[87] M. Isard and A. Blake, "Condensation Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision* , vol. 29, no. 1, pp. 5-28, 1998.

[88] B. Shackel, "Usability-Context, Framework, Definition, Design and Evaluation," *Interacting with Computers*, vol. 21, no. 5, pp. 339-346, 2009.

# APPENDIX A

## SHAPE SAMPLES IN HAND SHAPE DATABASE

Hand shape database is composed of 15 different hand shapes with various position, orientation and size. Following images in Figure A are samples among about 10000 images in the database.

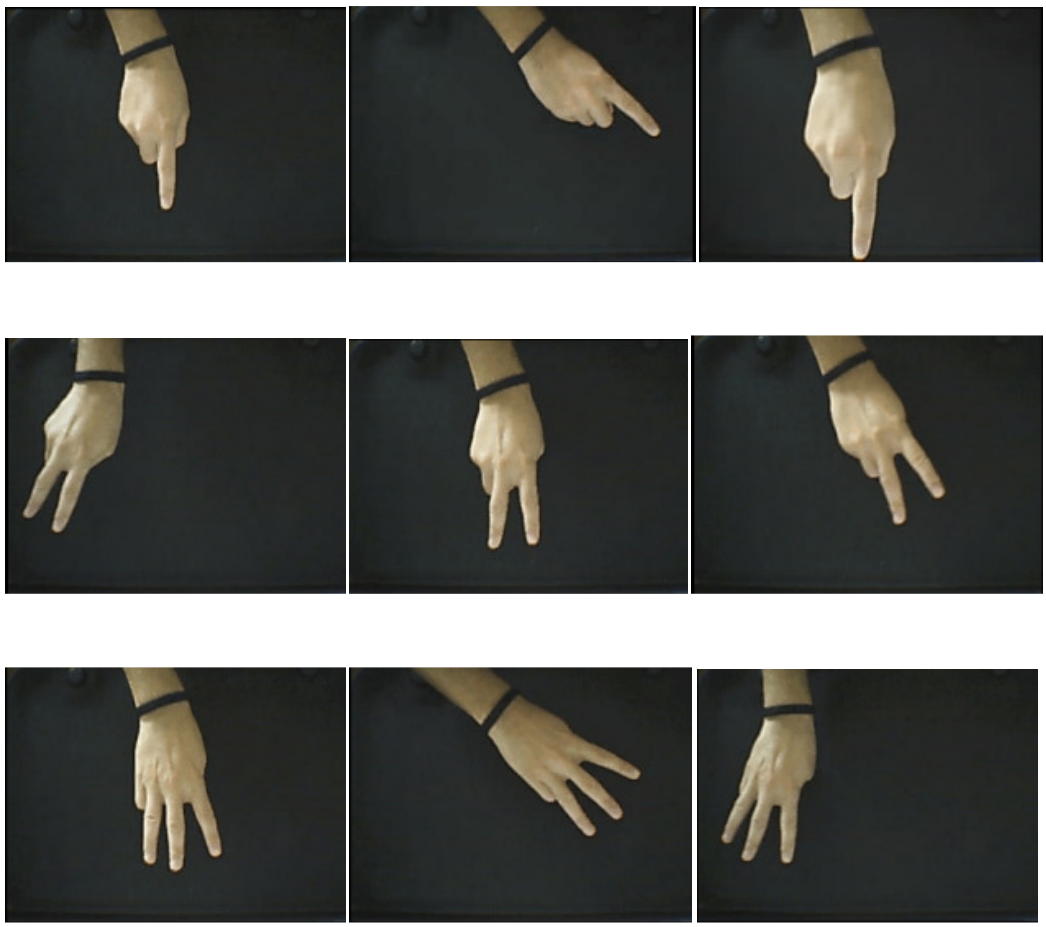


Figure A: Hand shape images.

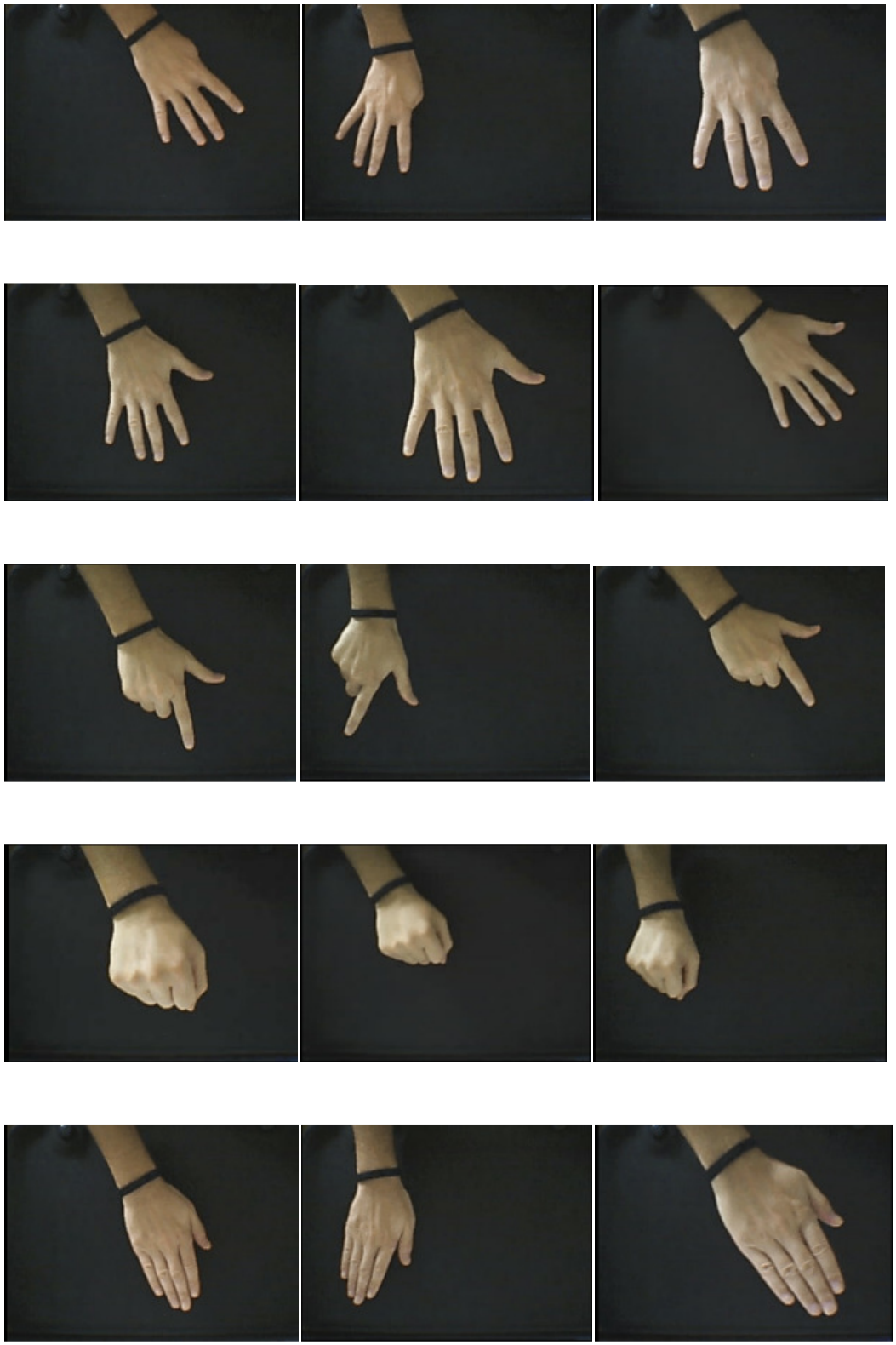


Figure A (continued)

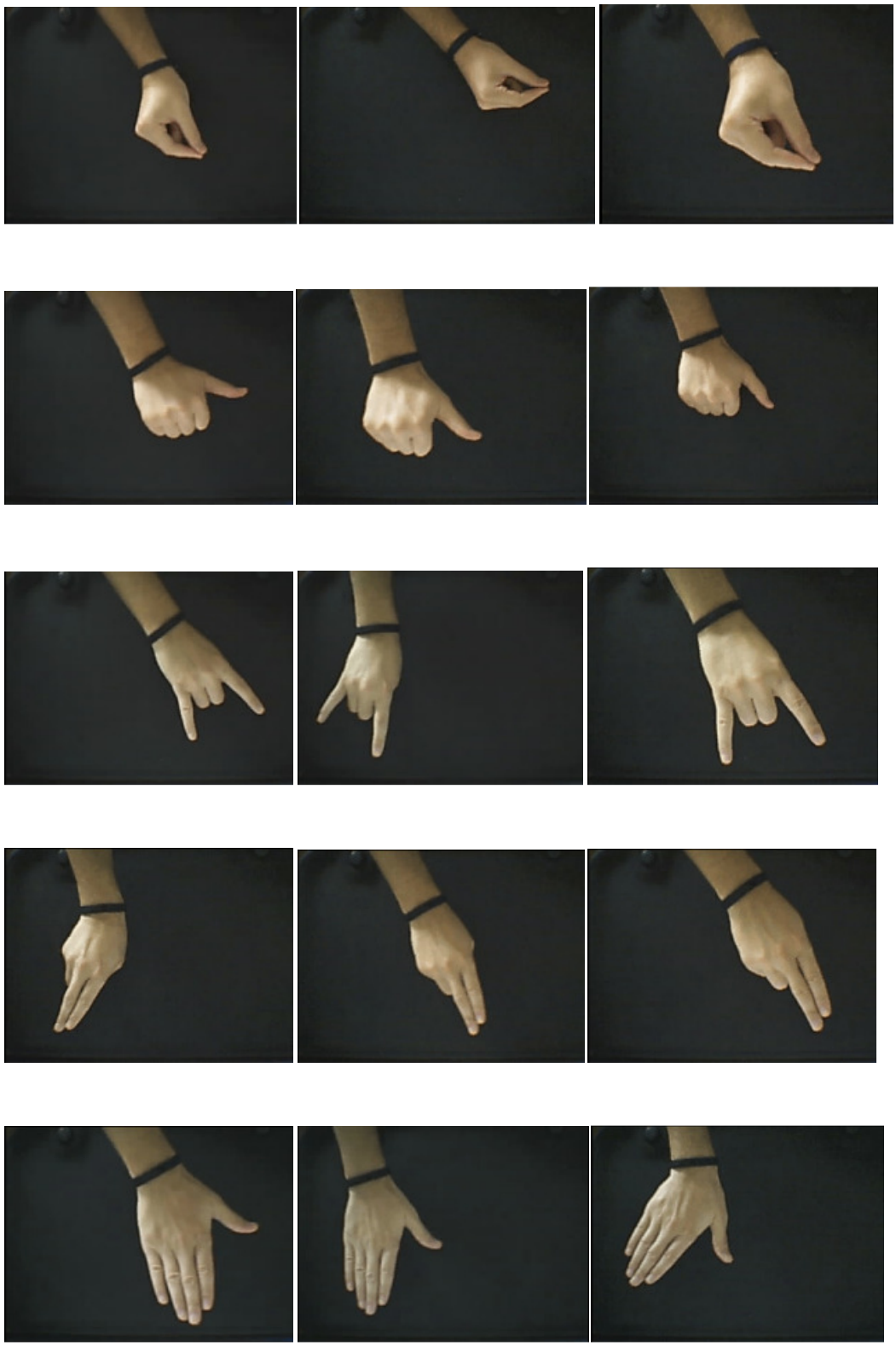


Figure A (continued)

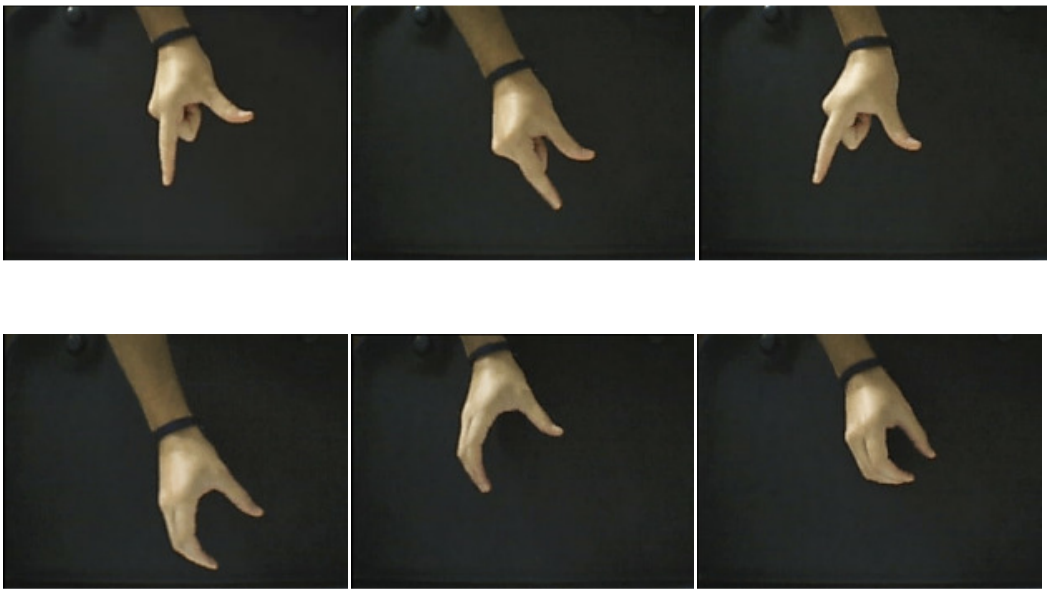


Figure A (continued)

## APPENDIX B

## THE QUESTIONNAIRE OF THE EXPERIMENT

*Query No:*

*SURVEY*

*Comparison of Hand and Mouse-based Interfaces*

*Name:* *Surname:* *Date:*
*Duration:*

*Subjective Criterias :*

| *no* | *Attributes* | *Hand-based Intf.* | *Mouse-based Intf.* |
|---|---|---|---|
| *1* | *Easy to* ***learn*** | | |
| *2* | *Easy to* ***use*** | | |
| *3* | *Easy to* ***remember*** | | |
| *4* | *Natural to* ***use (immersive)*** | | |
| *5* | ***Comfortable*** | | |
| *6* | ***Satisfactory*** | | |
| *7* | ***Enjoyable*** *to use* | | |

*Performance Criterias:*

| *no* | *Attributes* | *Hand-based Intf.* | *Mouse-based Intf.* |
|---|---|---|---|
| *1* | *# of trials for success* | | |
| *2* | *Time to complete task successfully* | | |

*Scale: 1: strongly disagree, 2:disagree, 3: neutral, 4: agree, 5:strongly agree*

*Comments:*

# APPENDIX C

## RESULTS OF USABILITY EXPERIMENT

The experiment was conducted using twelve different users. Each user assessed the subjective and performance criteria of mouse- and hand-based interface. There are totally ten queries in four different query types. For each query type, each user filled the questionnaire in Appendix B, and the results are tabulated in the following tables. The average and variance of each criterion are also shown for each query type at the bottom of the tables.

Table 2: Spatial Query Results.

*SPATIAL QUERY RESULTS*

| | *learn* | | *use* | | *remember* | | *natural* | | *comfort* | | *satisf.* | | *enjoyable* | | *trial* | | *time* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **User** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** |
| 1 | 4 | 5 | 4 | 5 | 5 | 5 | 5 | 3 | 4 | 5 | 4 | 4 | 5 | 3 | 1 | 1 | 8 | 9 |
| 2 | 4 | 5 | 3 | 5 | 5 | 5 | 5 | 3 | 4 | 4 | 5 | 5 | 5 | 2 | 1 | 1 | 7 | 9 |
| 3 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 4 | 5 | 5 | 3 | 1 | 1 | 8 | 8 |
| 4 | 5 | 5 | 5 | 5 | 4 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 1 | 1 | 8 | 7 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 4 | 5 | 5 | 2 | 1 | 1 | 7 | 7 |
| 6 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 5 | 5 | 3 | 1 | 1 | 7 | 8 |
| 7 | 5 | 5 | 4 | 5 | 5 | 5 | 4 | 4 | 3 | 4 | 4 | 5 | 5 | 4 | 1 | 1 | 8 | 8 |
| 8 | 5 | 4 | 4 | 5 | 5 | 5 | 5 | 3 | 5 | 3 | 5 | 4 | 5 | 3 | 1 | 1 | 8 | 8 |
| 9 | 4 | 5 | 4 | 5 | 5 | 4 | 5 | 2 | 3 | 5 | 4 | 4 | 5 | 3 | 1 | 1 | 8 | 9 |
| 10 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 1 | 1 | 7 | 9 |
| 11 | 5 | 5 | 3 | 5 | 5 | 5 | 5 | 2 | 4 | 4 | 4 | 4 | 5 | 3 | 1 | 1 | 9 | 10 |
| 12 | 5 | 5 | 4 | 5 | 5 | 5 | 5 | 4 | 4 | 4 | 4 | 4 | 5 | 3 | 1 | 1 | 8 | 10 |
| **μ** | 4.75 | 4.92 | 4.25 | 5.00 | 4.92 | 4.83 | 4.92 | 3.58 | 4.25 | 4.25 | 4.42 | 4.50 | 5.00 | 3.08 | 1.00 | 1.00 | 7.75 | 8.50 |
| **σ** | 0.5 | 0.3 | 0.8 | 0 | 0.29 | 0.39 | 0.3 | 1 | 0.8 | 0.6 | 0.5 | 0.5 | 0 | 0.67 | 0 | 0 | 0.6 | 1 |

Table 3: Motion Trajectory Query Results.

*MOTION TRAJECTORY QUERY RESULTS*

| | *learn* | | *Use* | | *remember* | | *natural* | | *comfort* | | *satisf.* | | *enjoyable* | | *trial* | | *time* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **User** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** |
| 1 | 4 | 5 | 3 | 5 | 4 | 5 | 5 | 3 | 4 | 4 | 4 | 4 | 5 | 5 | 1 | 1 | 7 | 8 |
| 2 | 4 | 5 | 5 | 3 | 4 | 5 | 5 | 3 | 5 | 4 | 5 | 5 | 5 | 2 | 2 | 1 | 7 | 7 |
| 3 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 3 | 5 | 3 | 5 | 3 | 1 | 1 | 8 | 8 |
| 4 | 5 | 5 | 3 | 4 | 4 | 5 | 4 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 1 | 8 | 7 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | - | - | 5 | 5 | 1 | 1 | 9 | 7 |
| 6 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 3 | 1 | 1 | 7 | 8 |
| 7 | 5 | 5 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 5 | 5 | 3 | 2 | 1 | 8 | 7 |
| 8 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 3 | 5 | 3 | 5 | 4 | 5 | 3 | 1 | 1 | 7 | 8 |
| 9 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 4 | 4 | 5 | 5 | 4 | 5 | 3 | 1 | 1 | 9 | 9 |
| 10 | 5 | 5 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 5 | 3 | 1 | 2 | 7 | 7 |
| 11 | 5 | 5 | 5 | 3 | 5 | 5 | 5 | 2 | 5 | 3 | 5 | 5 | 5 | 4 | 1 | 1 | 7 | 8 |
| 12 | 5 | 5 | 5 | 3 | 5 | 3 | 5 | 3 | 5 | 2 | 5 | 2 | 5 | 2 | 1 | 1 | 8 | 7 |
| **μ** | 4.83 | 5.00 | 4.58 | 4.33 | 4.67 | 4.75 | 4.92 | 3.83 | 4.75 | 3.83 | 4.82 | 4.18 | 4.92 | 3.33 | 1.17 | 1.08 | 7.67 | 7.58 |
| **σ** | 0.389 | 0 | 0.793 | 0.888 | 0.492 | 0.622 | 0.289 | 1.03 | 0.452 | 0.937 | 0.405 | 0.982 | 0.289 | 0.985 | 0.389 | 0.289 | 0.7785 | 0.669 |

Table 4: Motion Trajectory with Temporal Relation Query Results.

*MOTION TRAJECTORY with TEMPORAL RELATION QUERY RESULTS*

| | *learn* | | *use* | | *remember* | | *natural* | | *comfort* | | *satisf.* | | *enjoyable* | | *trial* | | *time* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| User | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** |
| 1 | 5 | 5 | 5 | 2 | 5 | 4 | 5 | 3 | 5 | 3 | 5 | 3 | 5 | 3 | 1 | 6 | 9 | 50 |
| 2 | 5 | 2 | 5 | 2 | 4 | 3 | 5 | 2 | 5 | 2 | 5 | 2 | 5 | 2 | 1 | 5 | 7 | 54 |
| 3 | 4 | 3 | 5 | 3 | 5 | 5 | 5 | 3 | 5 | 2 | 5 | 2 | 5 | 2 | 2 | 8 | 7 | 70 |
| 4 | 4 | 4 | 4 | 3 | 5 | 4 | 5 | 4 | 5 | 4 | 4 | 4 | 5 | 4 | 1 | 6 | 8 | 45 |
| 5 | 5 | 5 | 5 | 2 | 5 | 5 | 5 | 4 | 5 | 4 | 5 | 1 | 5 | 3 | 2 | 7 | 7 | 68 |
| 6 | 5 | 5 | 5 | 2 | 5 | 5 | 5 | 2 | 5 | 5 | 5 | 2 | 5 | 3 | 1 | 8 | 8 | 75 |
| 7 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 1 | 7 | 8 | 70 |
| 8 | 5 | 2 | 5 | 2 | 5 | 3 | 5 | 1 | 4 | 3 | 5 | 2 | 5 | 2 | 1 | 4 | 9 | 42 |
| 9 | 5 | 5 | 5 | 2 | 5 | 5 | 5 | 2 | 5 | 4 | 5 | 2 | 5 | 2 | 2 | 6 | 8 | 56 |
| 10 | 5 | 5 | 5 | 1 | 5 | 5 | 5 | 1 | 5 | 5 | 5 | 1 | 5 | 3 | 1 | 5 | 10 | 48 |
| 11 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 1 | 7 | 9 | 63 |
| 12 | 5 | 1 | 5 | 1 | 5 | 5 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 1 | 8 | 8 | 60 |
| **μ** | 4.83 | 3.50 | 4.92 | 2.08 | 4.92 | 4.08 | 5.00 | 2.33 | 4.92 | 3.17 | 4.92 | 2.08 | 5.00 | 2.50 | 1.25 | 6.42 | 8.17 | 58.42 |
| **σ** | 0.4 | 1.6 | 0.3 | 0.9 | 0.29 | 1.24 | 0 | 1.2 | 0.3 | 1.4 | 0.3 | 1.1 | 0 | 1 | 0.5 | 1.3 | 0.9 | 10.93 |

Table 5: Camera Motion Query Results

*CAMERA MOTION QUERY RESULTS*

| | *learn* | | *use* | | *remember* | | *natural* | | *comfort* | | *satisf.* | | *enjoyable* | | *trial* | | *time* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **User** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** | **H** | **M** |
| 1 | 5 | 1 | 5 | 1 | 5 | 2 | 5 | 1 | 4 | 1 | 5 | 1 | 5 | 1 | 1 | 1 | 9 | 25 |
| 2 | 5 | 3 | 5 | 2 | 4 | 2 | 5 | 2 | 5 | 3 | 5 | 2 | 5 | 2 | 1 | 1 | 12 | 30 |
| 3 | 5 | 4 | 5 | 4 | 5 | 5 | 5 | 3 | 5 | 2 | 5 | 2 | 5 | 2 | 2 | 2 | 9 | 34 |
| 4 | 5 | 3 | 4 | 3 | 5 | 4 | 5 | 3 | 4 | 3 | 4 | 3 | 5 | 3 | 1 | 1 | 9 | 37 |
| 5 | 4 | 5 | 4 | 2 | 4 | 5 | 4 | 2 | 4 | 3 | 5 | 5 | 5 | 2 | 1 | 2 | 11 | 34 |
| 6 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 5 | 5 | 3 | 1 | 1 | 7 | 24 |
| 7 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 5 | 5 | 5 | 5 | 4 | 2 | 1 | 10 | 22 |
| 8 | 4 | 3 | 5 | 3 | 4 | 4 | 5 | 2 | 4 | 3 | 4 | 4 | 5 | 3 | 2 | 1 | 9 | 33 |
| 9 | 5 | 4 | 5 | 3 | 5 | 3 | 5 | 2 | 4 | 5 | 4 | 3 | 5 | 4 | 1 | 1 | 8 | 25 |
| 10 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 3 | 5 | 5 | 5 | 4 | 5 | 4 | 1 | 2 | 9 | 43 |
| 11 | 5 | 5 | 5 | 3 | 5 | 3 | 5 | 3 | 5 | 3 | 5 | 3 | 5 | 5 | 2 | 1 | 9 | 21 |
| 12 | 5 | 5 | 5 | 3 | 5 | 5 | 5 | 1 | 5 | 2 | 5 | 3 | 5 | 1 | 1 | 1 | 10 | 37 |
| **μ** | 4.83 | 3.83 | 4.83 | 3.08 | 4.75 | 3.83 | 4.92 | 2.58 | 4.50 | 3.33 | 4.75 | 3.33 | 5.00 | 2.83 | 1.33 | 1.25 | 9.33 | 30.42 |
| **σ** | 0.389 | 1.1934 | 0.389 | 1.084 | 0.452 | 1.115 | 0.289 | 1.165 | 0.522 | 1.371 | 0.452 | 1.303 | 0 | 1.267 | 0.492 | 0.452 | 1.3027 | 6.986 |

## CURRICULUM VITAE

### PERSONAL INFORMATION

Surname, Name: Genç, Serkan
Nationality: Turkish (TC)
Date and Place of Birth: 1 September 1975
Email: sgenc@bilkent.edu.tr

### EDUCATION

| | | |
|---|---|---|
| MS METU | Computer Engineering | 1999 |
| BS METU | Computer Engineering | 1997 |
| High School | Eskişehir Science High School | 1992 |

### WORK EXPERIENCE

| | |
|---|---|
| 2001-Present | Instructor, Computer Technology and Inf. Sys., Bilkent University |
| 1999-2001 | Senior Software Engineer, iXec Gmbh, Munich, Germany |
| 1997-1999 | Teaching Assistant in Computer Engineering at METU |

### FOREIGN LANGUAGES

Fluent English, Intermediate German

### PUBLICATIONS

1. S. Genç, M. Baştan, U. Güdükbay, V. Atalay, Ö. Ulusoy, "A Hand-Gesture-Based Query Interface for a Video Retrieval System", Intl. Journal of Human-Computer Studies, Submitted, 2010.
2. S. Genç, V. Atalay, "Which Shape Representation is better for Real-Time Hand Interface System?", 5th International Symposium on Visual Computing, LNCS5875, pp. 1-11, Las Vegas, Nevada, USA, (2009)
3. Genç S., Atalay V.: ITouch: Vision-based Intelligent Touch Screen in a Distributed Environment. Int. Conf. on Multimodal Interfaces (ICMI), Doctoral Spotlight, Trento, Italy, October, (2005)
4. Genç S., Atalay V.: Texture Extraction from Photographs and Rendering with Dynamic Texture Mapping. ICIAP, Venice, Italy, (1999)
5. Genç S., Fatos T. Yarman-Vural: Morphing as a Tool for Motion Modeling. ICIAP, Venice, Italy, (1999)
6. Genç S., Atalay V., İşler V.: Görüntü tabanlı dinamik doku eşleme, SİU'99, Ankara, (1999)