2004

A vision-based approach for human hand tracking and gesture recognition.

Jiangnan Lu
University of Windsor

A Vision-Based Approach for Human Hand Tracking and Gesture Recognition

By

Jiangnan Lu

A Thesis
Submitted to the Faculty of Graduate Studies and Research
through Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2004

Your file Votre référence
ISBN: 0-612-96121-4
Our file Notre référence
ISBN: 0-612-96121-4

Abstract

Hand gesture interface has been becoming an active topic of human-computer interaction (HCI). The utilization of hand gestures in human-computer interface enables human operators to interact with computer environments in a natural and intuitive manner. In particular, bare hand interpretation technique frees users from cumbersome, but typically required devices in communication with computers, thus offering the ease and naturalness in HCI.

Meanwhile, virtual assembly (VA) applies virtual reality (VR) techniques in mechanical assembly. It constructs computer tools to help product engineers planning, evaluating, optimizing, and verifying the assembly of mechanical systems without the need of physical objects. However, traditional devices such as keyboards and mice are no longer adequate due to their inefficiency in handling three-dimensional (3D) tasks. Special VR devices, such as data gloves, have been mandatory in VA.

This thesis proposes a novel gesture-based interface for the application of VA. It develops a hybrid approach to incorporate an appearance-based hand localization technique with a skin tone filter in support of gesture recognition and hand tracking in the 3D space. With this interface, bare hands become a convenient substitution of special VR devices. Experiment results demonstrate the flexibility and robustness introduced by the proposed method to HCI.

DEDICATED

To my parents, my friends
and all who love me and beloved

Acknowledgements

I must begin by acknowledging my supervisor, Dr. Xiaobu Yuan, for sharing his wisdom and his experience while giving me constant advice throughout my study. I am also grateful to him for giving me the artistic freedom necessary to realize this task.

I would like to extend my sincere appreciation to my committee members, Dr. Fritz Rieger and Dr. Christie Ezeife, for all their kind help, encouragement and guidance.

My special thanks go to Yingxin Xia for her understanding and unbelievable supports.

Finally, I would like to express thanks to all others who have helped me to overcome the many technical difficulties in preparing this thesis.

Table of Contents

List of Figures

1. Introduction

The concept of Human-Computer Interaction (HCI) was born as early as the invention of computer itself. It mainly refers to devices and technologies that support essential communication between human and computers. The rapid growth of massive computerization has made effective human-computer interaction essential. Nowadays, it is becoming an increasingly important and indivisible part of our daily lives. A simple example is the ubiquitous graphical interface used by Microsoft Windows 95, whose original idea can be traced back to the interface of Macintosh, the work at Xerox PARC, all the way to the early research at Stanford Research Laboratory (new SRI) and at the Massachusetts Institute of Technology [21].

For a long time, however, research on HCI has been restricted to techniques based on the use of a graphic display, a keyboard, and a mouse. These devices have grown to be familiar but inherently limit the speed and naturalness with which we can interact with the computers. It is regrettable that these limitations have become the major bottleneck in the developments of related areas such as VR, robotics, game, remote sensing, robot-human collaboration, and virtual assembling ([1], [47], [68], [143]).

The long-term goal of HCI is to pursue the ease and naturalness with which the user can interact with the computers. Inspired by the natural means employed in the human-to-human communication environment, researchers have started to explore the potentials of introducing other communication means constantly used by humans into the HCI context. One of such means is the movement of human arms and hands, also called **hand gestures**. Human hand gestures are common in non-verbal interaction among people. They range from simple actions of using our hand to point at and move around objects to the more sophisticated ones that express our feelings and allow us to communicate with

others ([70], [108], [109], [133]). Thus, in recent years the integration of hand gesture into HCI context has obtained awareness of active researchers.

1.1 Previous Work

To introduce gestures as a new channel of HCI, it is necessary to develop techniques to allow computers recognize and interpret gestures. The recognition and interpretation of human gestures in HCI need to measure the dynamic and static configurations of human hands and other parts of the human body. Early solutions to this problem use mechanical devices that directly measure spatial position of hands and the joint angles of arms. A popular representative of such devices is glove-based device (**Figure 1**). To interact with computers with glove-based devices, users usually are required to wear a cumbersome glove with one or more cables that connect the glove to the computer ([7], [34], [92], [120]). The major disadvantages of such mechanical devices can be identified easily: unease and inconvenience. Furthermore, in remote-control environments, such glove-based devices are powerless for measuring the human hands.

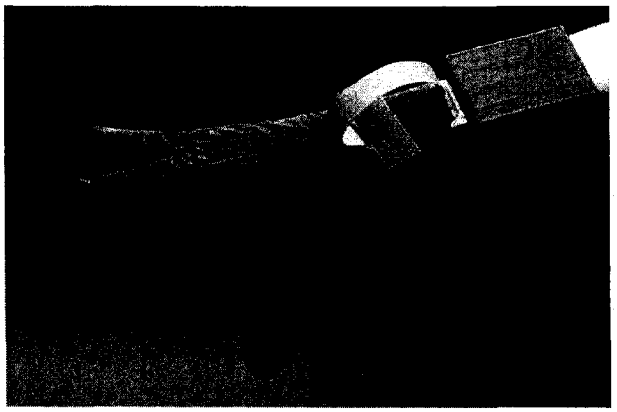


Figure 1: Digital glove

Natural interaction between humans does not involve devices since we are able to perceive our environment with eyes and ears. In principle, computers should be able to imitate those abilities with cameras and microphones. To overcome the weakness of using mechanical devices, in recent year, vision-based interaction techniques have been proposed that use a set of video cameras and computer vision techniques to recognize and interpret gestures [43]. The ease, convenience, and naturalness of this approach have attracted more and more attentions of researchers in the HCI community.

1.1.1 Vision-Based Hand Gesture Interface Architecture

In any typical vision-based hand gesture interface, the first stage is to choose a descriptive model for hand gestures. Hand features and hand gestures have to be selected properly in light of the requirement of particular applications. For instance, both the spatial and temporal characteristics of the hand and hand gestures may be considered in a mathematical model. As the foundation of hand recognition, the selection of models plays a pivotal part that will largely influence the performance of gesture recognition.

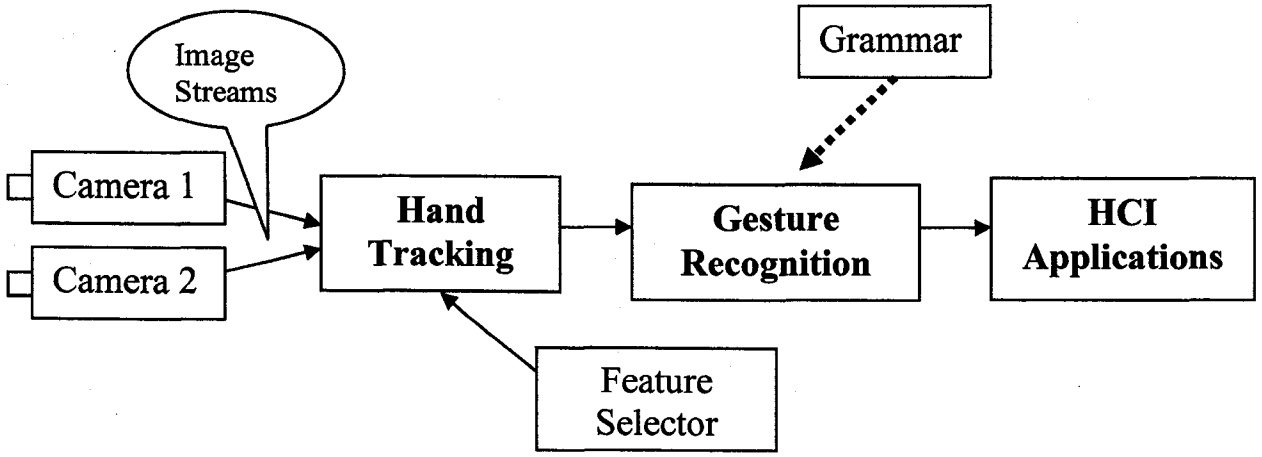


Figure 2: Architecture of vision-based hand gesture interface.

The chosen model is used in the analysis stage to obtain model parameters by extracting image features from image streams. The parameters are detailed measurements of hand

pose and trajectory varying depending upon the chosen model. The hand localization and hand tracking are major issues in this stage.

Model parameters are then classified according to certain pattern recognition rules. The rules may contain some grammars that reflect the internal syntax of gesture commands and perhaps interact with other communication modes like speech, gaze or facial expressions. The accuracy, robustness, speed and variability in different group of hand movements are involved in the evaluation of a particular recognition approach. **Figure 2** shows the typical architecture of vision-based hand gesture applications.

1.1.2 State-of-Art

Gesture interface has become increasingly popular since the concept of *virtual reality* was proposed. As the device-based hand gesture applications have to use expensive special hardware, the visual hand gesture interface has evoked the studies for possible replacements of device-based interface in HCI. Hand gesture can simply enhance the interaction in desktop computer applications by replacing the mouse or joysticks. Furthermore, the improvement of computing power impulses the studies of bare hand interface. Also the growth of various applications in the virtual environments is propelling the development of hand gesture interface.

A variety of vision-based applications have been successfully developed to replace the expensive glove devices. Some applications used separate images to visually estimate hand postures, while other applications attempt to track human hands in image streams. In Bell Laboratory, *Jakub Segen* developed a system in which 2D hand gesture is used as a navigator for computer games. *Soren Lenman*, a Swedish researcher, proposed an algorithm to employ hand gestures in a TV remote controller. In University of Berlin, *Christian Hardenberg* developed a finger-mouse system that mimics normal mouse actions ([73], [97]).

In virtual assembling domain, researchers attempted to use hand gestures as manipulators of *virtual objects* (VOs). VOs can be computer-generated graphics, like simulated 2D and

3D objects ([52], [126]) or windows [127], or abstractions of computer-controlled physical objects, such as device control panels ([39], [113]) or robotic arms ([21], [59], [125]). Often a combination of tracking and recognizing is involved in such applications to perform the manipulations of virtual objects. For example, to issue a command to rotate an object, a user may take two-step: "*select object*" and "*rotate object*". The first action uses coarse hand tracking to move a pointer in the Virtual Environment (VE) to the vicinity of the object [53]. The second action allows the user to rotate his/her hand back and forth producing a *metaphor* for rotational manipulation.

However, those existing vision based hand gesture applications usually only work well under a set of assumptions [8]. Some applications only provide two degrees of freedom (DOF). Some applications require a training process for a new user. In human hand tracking systems, the results are often unsatisfied in real environment. Color-based approaches need unchangeable lighting while motion-based approaches usually require static and monochroic background. A brief summary of constraints involved in existing applications is shown in **Table 1.**

Applications	Background	Lighting	Training Process	Degree of Freedom
CD Player Control Panel	Static+Monochroic	Variable	Required	2 DOF
2D game navigator [36]	Static+Monochroic	Unchanged	Required	2 DOF
Finger-Mouse [107]	Dynamic	Unchanged	Required	2 DOF
TV Display Control	Static	Unchanged	Required	2 DOF
Fingertip Tracker	Monochroic	Variable	Required	3 DOF

Table 1. Summary of constraints

1.2 Contributions

Although vision-based gesture applications have demonstrated success in some applications, it is not difficult to find out constraints and restrictions. Ideally, vision-based gesture interfaces are able to work well under any real outdoor environments. In such case, dynamic changing background, varying lighting and any movement of user hands will be allowed. If so, humans will be unaware that counterpart is not a human, but a "smart" computer. However, it is regrettable that current vision-based gesture interface fails to render satisfactory solutions to achieve this goal. Most of the gesture-based systems at the present time only provide limited functionalities.

In this thesis, we first address a new gesture definition that facilitates HCI studies. Based on our new gesture definition, we then contribute a novel framework for visually tracking human hand in outdoor workspace. The core algorithm presented in our framework is a hybrid appearance-based model for hand localization and segmentation. In the algorithm, skin color cue and hand motion cue are combined to deal with complex background and varying lighting. A skin tone filter is incorporated into the algorithm to enhance the ability of removing noisy hand. Another feature of the algorithm is a self-adaptive online training schema which improves the accuracy of skin color recognition. Another contribution of this thesis is an accurate and robust stereovision-based algorithm for 3D human hand position recovery.

1.3 Outline of Thesis

The bare-hand interface presented in this thesis consists of several key components involving various technologies in hand modeling, hand localization and hand gesture recognition. Therefore, to systematically describe the key components integrated in our bare-hand interface, we organized the thesis as follows. Chapter 2 first reviews the existing gesture definition and gesture categorization. Different solutions to the question

"How to mathematically describe human hand" are also presented in this section. After address the argument of hand gesture definition, we propos a new definition of hand gesture that is particularly suitable for HCI researches. Chapter 3 analyzes different hand localization techniques. Hand localization and segmentation are the foundation for hand tracking and gesture recognition issues. We emphasize color and motion characteristics of hand gesture and summarize the drawbacks respectively. Then we introduce our appearance-based hand localization algorithm. Chapter 4 surveys various technologies for tracking dynamic hand gestures through image stream. To overcome the weakness of existing tracking technologies, a stereovision-based 3D hand trajectory reconstruction schema is contributed. In Chapter 5, we analyze various issues in gesture recognition domain, including gesture vocabulary, feature space reduction and hand dynamics recognition. Our novel gesture-based interface is demonstrated in Chapter 6, showing improvements of functionality and flexibility. Finally, Chapter 7 concludes by reiterating the contributions of this method and proposes future research directions.

2. Hand Gesture Modeling

2.1 Hand Gesture Definition

Although gestures are used daily, there is no widely accepted authoritative definition because the definition involves various factors in psychology, sociology and linguistics. Outside HCI domain, definitions are particularly related to the communicational aspect of human hand and body movements. For example, Dictionary.com defines gesture as

> *"A motion of the limbs or body made to express or help express thought or to emphasize speech; the act of moving the limbs or body as an expression of thought or emphasis."*

However, inside HCI literature, researchers tend to narrow down this broad definition to facilitate studies in the specific domain. Hand tracking systems focus only on the hand palm, whereas computer controlled environments use human hands, even arms, to perform tasks that mimic both the natural use of the hand as a manipulator, and its use in communication. However, it is a fact that most of the studies in HCI framework are only concerned about the use of gestures as a communication method, usually called practical gestures.

Hand gestures and spoken language have similarities in communication. Gestures are conceptually formed in a gesturer's mind, possibly in conjunction with speech. They are realized in a way of motions of arms and hands, while speech is produced in a similar way by air stream modulation through the human vocal tract. The observers perceive gestures as streams of visual images and interpret them based upon the knowledge and experience they possess about those gestures. Due to the similarities of gestures and

spoken language, the models used in spoken language recognition can be employed to describe the production and perception of hand gestures [90]. A gesture model is shown in **Figure 3**.

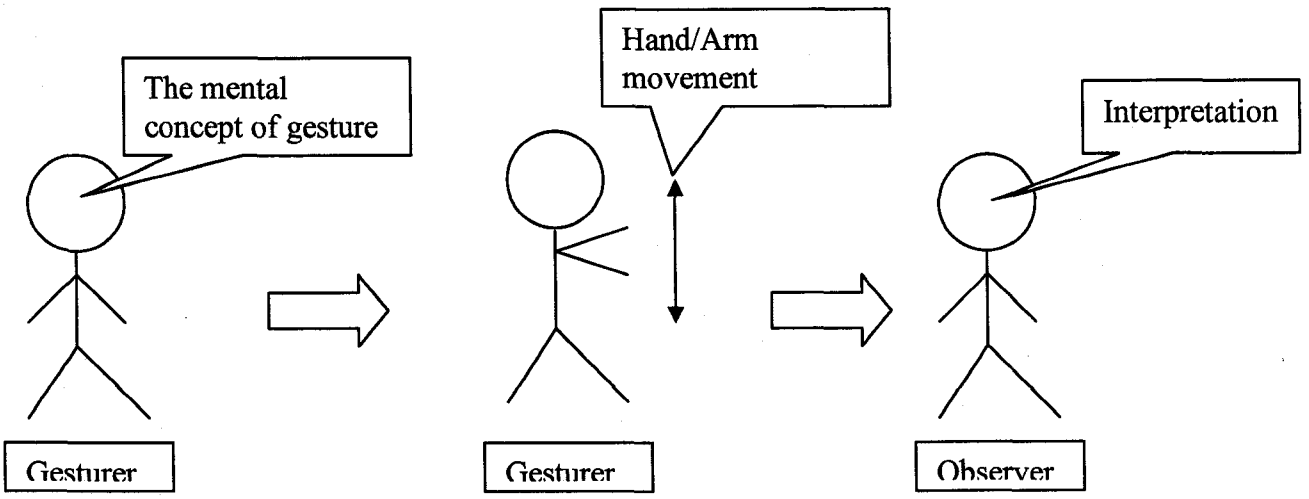


Figure 3: Production and perception of gestures.

From HCI viewpoint, the majority of gestures usually contain movement of hands and arms, and dynamic is therefore the key characteristic of gestures. Static postures are usually used only to describe the shape of objects or to convey a small set of meanings. When performing a gesture the gesturer moves his hands in 3D world, so the position of hands forms a trajectory in space.

Consequently, in this thesis, we use the definition of hand gestures as follows:

"***A hand gesture is a trajectory in 3D geometric space within a time interval.***"

The definition reveals two important characteristics of gestures: temporal nature and spatial nature. Therefore, further gesture analysis is closely based on this definition. **Figure 4** is the visualization of two hand gestures based on our new gesture definition.

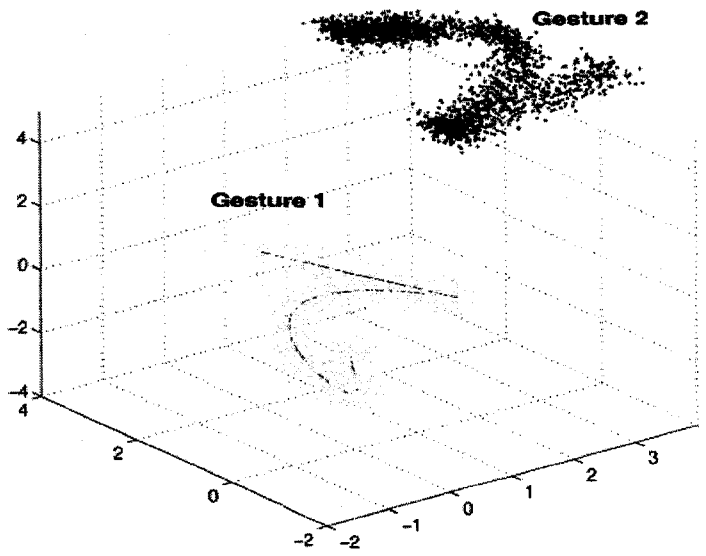


Figure 4: Two gesture trajectories in 3D space.

2.2 Gesture Categorization

Different taxonomies have resulted in various categorizations in the literature. Gestures can occur with or without speech. Thus, with the consideration of psychological aspects of gestures, Kendon suggested two groups of gestures: "autonomous gestures" and "gesticulation". The autonomous gestures occur independent of speech while the gesticulation occurs in association with speech. In 1994, Cadoz classified gestures into three types according to the functions of gestures:

- *semiotic:* those used to communicate meaningful information;
- *ergotic:* those used to manipulate the physical world and create artifacts;
- *epistemic:* those used to learn from the environment through tactile or haptic exploration.

McNeill and Levy provided a taxonomy that separates gestures into iconic, metaphoric, and beat. However, those categorizations seemed unsuitable for the usage of gesture

within HCI framework. In this thesis, we employ a taxonomy addressed by Quek in ([94], [95]). The hierarchy of Quek's taxonomy is shown in **Figure 5**.

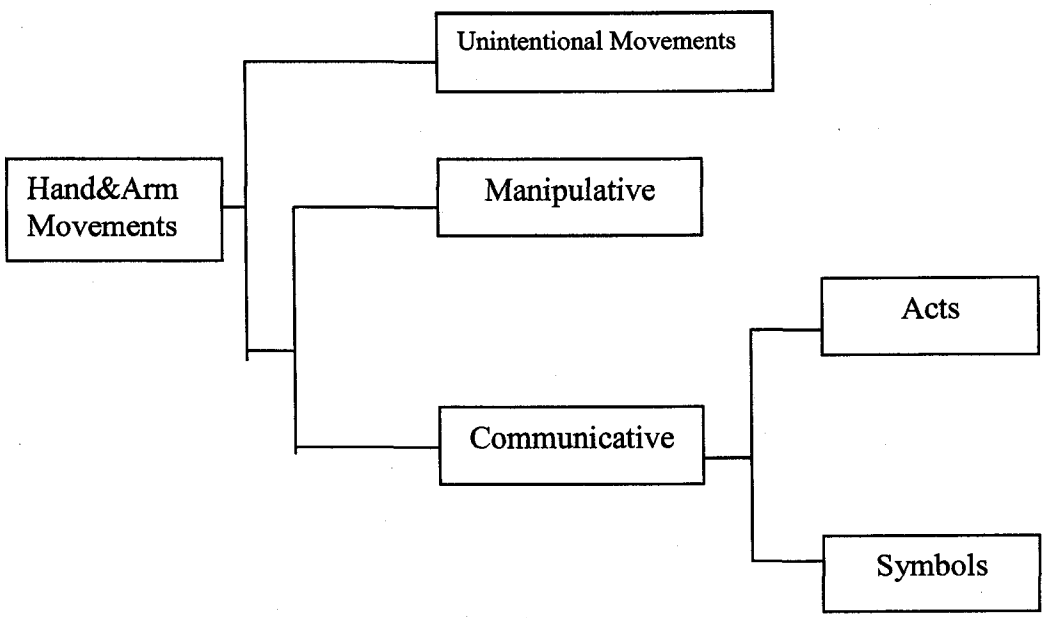


Figure 5: Taxonomy of hand gestures

In the first layer, all hand and arm movements are classified into two major classes:

- *gestures*;
- *unintentional movements*.

Unintentional movements are movements produced by gesturer without intention. They do not contain any meaningful information. Such gestures can occur when a person is talking with another through the phone.

In the second layer, gestures can be further classified into two groups as communicative and manipulative. Manipulative gestures are used to act on physical object in the real world, such as the rotation of a wheel and the stroke of a key in the keyboard. Like the unintentional movements, they do not convey much information. In comparison, communicative gestures are the ones used to express some meanings. In a human-human context, communicative gestures are often accompanied by speech to emphasize something.

In the third layer, communicative gestures may be either acts or symbols. Acts are gestures that are directly related to the interpretation of the movement. One of the components of acts is mimetic when someone says: "The plane flew like this", while moving his hand through the air like the flight path of an aircraft. Another component of acts is deictic which is used to point in a direction. Symbols are those gestures that play a linguistic role. Some abstracted simple actions (a motion of the index finger making a circle) could be considered as symbols [96]. It is symbols that are the most commonly used gestures in HCI domain since they have a rich set of meanings and can be easily represented by different static hand postures.

2.3 3D-based Gesture Modeling

Gestures are performed through hand and arm movements and actions in 3D space. In addition, humans interpret gestures according to the observation of the hand/arm movements. Gesture modeling deals with how to describe the hand. Numerous approaches have been proposed in HCI literature for hand modeling. These approaches can be categorized into two groups: appearance-based modeling and 3D-based modeling. **Figure 6** shows the different ways of hand modeling.

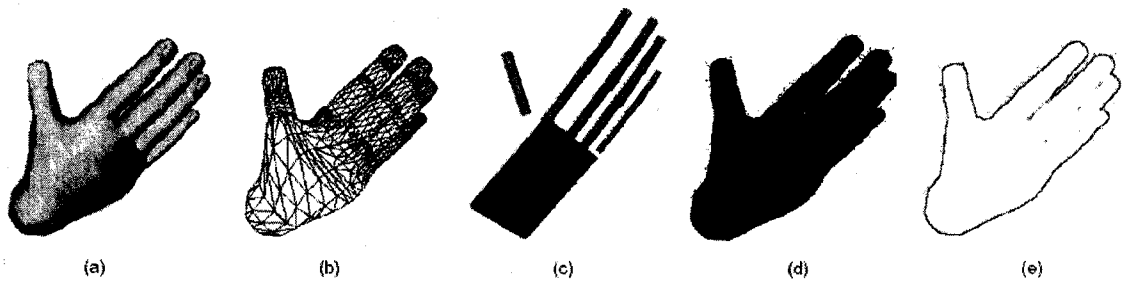


Figure 6. Different hand models representing the same hand posture. (From left: 3D Textured volumetric model; 3D Wireframe volumetric model; 3D Skeletal model; 2D Binary silhouette; 2D Contour.)

In 3D-based models, hand gestures are first considered as physical objects. The surface and volume are two characteristics to describe the hand. It mainly includes two large groups as:

- *volumetric models*;
- *skeletal models*.

Volumetric models deal with the 3D visual appearance of the human hand and arms. These kinds of models are commonly applied in the computer animation applications, and their possible usage in the field of computer vision has also been explored [32]. In the field of computer vision the volume of the human hand, arms or body are modeled for *analysis-by-synthesis* tracking and recognizing of the body's posture [66]. The analysis-by-synthesis approach is to analyze the body's posture by synthesizing the 3D model of the candidate human body and then varying its parameters until the model and the real human body appear visually same. To reduce the complexity of parameters in volumetric models, skeletal models were introduced. These models use joint angles with segment lengths as the parameters for modeling the hand and arm ([3], [6], [41], [58], [78], [99], [128]). However, even with the assistance of skeletal models, the complexity of 3D-based models is still too high for real-time computation.

2.4 Appearance-based Gesture Modeling

Appearance-based modeling, as the name suggests, is to model the hand gesture using the features directly obtained from visual images. This group of models is relatively simple since it deals with the appearance of hands and arms in the visual images. The model parameters are derived from the 2D images from camera or other input devices.

In this model, the gestures are modeled by comparing the appearance of target gesture to the appearance of the set of predefined template gestures. So far, most of models addressed in the HCI literature belong to appearance-based model because these models are easy to collect and compute [72]. For example, contour model and silhouette model rely upon deformable 2D template matching of human hands, arms, or even body ([12], [26], [56], [71], [75]). Deformable 2D templates are a set of points on the surface of an

object. The templates comprise the average point sets and point variability parameters. Average point sets are used to describe the general shape within a certain group of shapes. Point variability parameters represent the allowed shape deformation (variation) within that same group of shapes. For instance, the human hand in open position has one shape on the average, and all other instances of any open posture of the human hand can be formed by slightly modifying the average shape ([23], [48]). Parameters are obtained through statistical methods applied on a great number of training sets of data. Template-based models are typically used for hand-tracking task [57]. They can also be applied for simple gesture classification according to the multitude of classes of templates [20].

The selection of gesture model plays important role in gesture-based systems. The performance of such systems depends largely upon the computation complexity of the model. Thus, the selection of hand modeling is somehow application-oriented. For instance, in TV controller system, contour and silhouette models are usually sufficient. In virtual assembling area, where robot guidance is usually achieved by human point gesture, fingertip is the best choice ([2], [31]). On the other hand, when the requirements of certain applications need to recognize sophisticated hand gestures, saying American Sign Language (ASL), 3D hand models are desirable ([11], [30], [69]).

3. Gesture Analysis for Hand Localization

Without assistance of mechanic devices, all information about gestures reside in visual images or image sequences. Once the model of hand gesture is determined, all model parameters have to be extracted and estimated from images or image streams of hand gesture. However, raw images that are obtained from cameras are not directly usable for gesture analysis because they usually contain background and noise. Therefore, hand localization is the premier task in gesture analysis. In the process of gesture localization, the image region of the person and his gestures are separated from the rest of the visual image [114]. Nevertheless, unlike human beings, computers are not "smart" enough to easily segment hand gesture from raw images. Consequently, various efforts have been made to solve this non-trivial problem. Two major groups of localization techniques are discussed in the following subsection on color-based techniques and motion-based techniques.

3.1 Color-based Gesture Analysis

Color has been considered as a low-level yet an efficient visual feature for hand localization process because it is easy to detect and extract from raw color images. In earlier vision-based gesture interfaces, distinguishable color patches are attached to the user hands as markers ([22], [76]). These active markers yield a high contrast in the images, and therefore can be detected quickly. However, these solutions bring more inconvenience to the users, and changes of marker color may require new detection algorithm. In the natural environment, human can focus on hands without any artificial markers since human can identify the skin color of the hand effortlessly. Theoretically, computer is able to distinguish skin color as human do. Based on the studies of human abilities, color histogram technique is applied in skin detection algorithms ([49], [65], [80], [88]).

3.1.1 Skin Color Histogram

A color histogram is a distribution of colors in the color space and has long been used by the computer vision community in image understanding. For example, analysis of color histograms has been a key tool in applying physics-based models to computer vision ([15], [42]). In the mid-1980s, it was recognized that the color histogram for a single inhomogeneous surface with highlights would have a planar distribution in color space. It has since been shown that the colors do not fall randomly in a plane, but form clusters at specific points. It is wildly believed that the histograms of human skin color coincide with these observations. **Figure 7** shows the color distribution of these skin samples in the chromatic color space. Three views of the same skin histogram are shown in **Figure 8.**

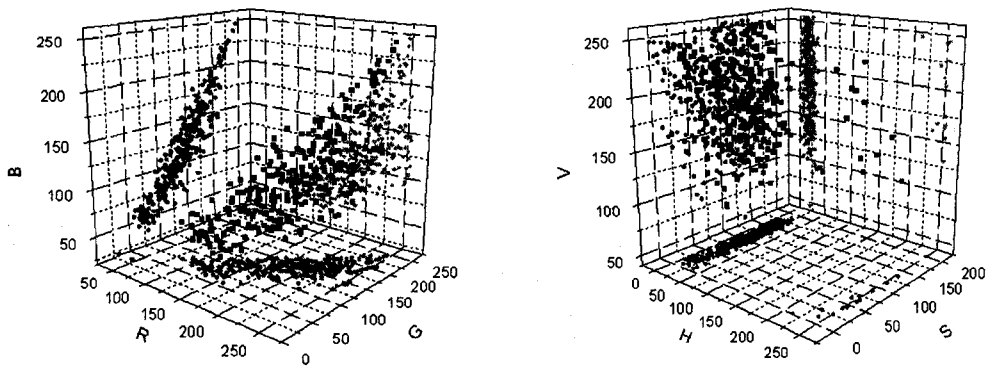


Figure 7: Skin color distribution in RGB color space

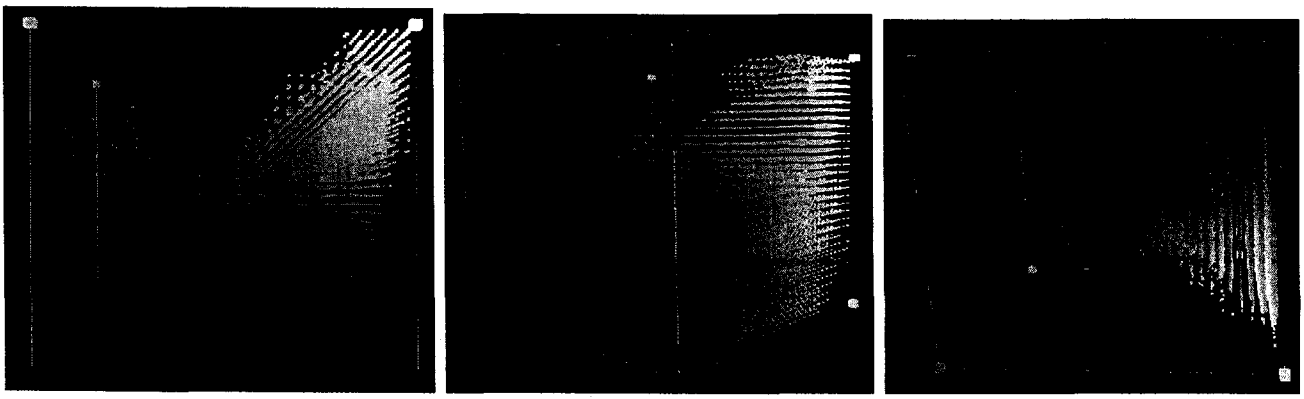


Figure 8: Different views of skin color histogram

3.1.2 Color Space

The studies of skin color histogram suggest that statistical models can be applied in skin detection process [129]. A majority of statistical models is based on classification theory. Due to the fact that the performance of skin detection algorithms is depending upon the separability of color space, it is necessary to systemically analyze different color spaces. Color spaces can be mathematically represented by three-dimensional coordinate systems. In different color spaces, the origins and three axes present different meanings. The color spaces frequently used in skin detection process include: RGB, HSV and YIQ. RGB (Red, Green, Blue) color space is widely used in computer display.

In RGB color space, we use three axes that are perpendicular to one another to represent red, green and blue respectively. HSV color space contains three components: Hue, Saturation and Value. Conceptually, the HSV color space is a trigonal cone [17]. Viewed from the circular side of the cone, the hues are represented by the angle of each color in the cone relative to the 0 line. The saturation is represented as the distance from the center of the circle. The brightness is determined by the colors vertical position in the cone. YIQ color space is based on luminance and chrominance. Here, Y is the luminance or brightness component. It gives all the information required by a monochrome television. I and Q are the chrominance or color components. YIQ color space has an important property. Y, the luminance information and I and Q, the color information are decoupled. **Figure 9** shows the different coordinate systems.

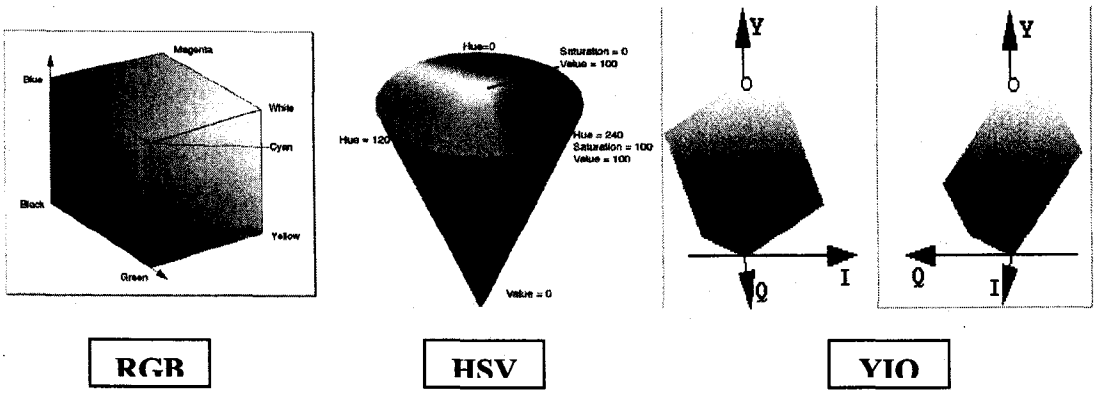


Figure 9: RGB color cube, HSV color cone and YIQ color space.

3.1.3 Color space reduction and transformation

One of the primary problems in skin detection is color constancy. The apparent color of human skin will be dramatically changed by the varying light brightness. All color spaces discussed above not only contain chromatic information but also include brightness value which definitely decrease the accuracy of skin detection algorithms. Due to the fact that varying illumination presents additional challenges to the task of skin detection, it is necessary to remove the illumination component of the color space to achieve illumination invariance [146]. Furthermore, "*Curse of Dimensionality*" in classification theory indicates adding more features may worsen the results if number of training data is not infinite. Consequently, color space reduction is an important pre-process in skin detection. According to the color space definitions, it is obvious that HSV and YIQ color spaces are easier for dropping brightness component than RGB color space. Thus, most of skin detection algorithms are based on HSV or YIQ color spaces. However, in vision-based gesture interface, the raw images of hand provided by cameras are usually in RGB format. To solve this problem, a color space transformation is required. Besides, color space transformation can improve classification process by increasing the separability between skin and non-skin classes and grouping the colors of different skin tones together.

3.1.4 Limitations

It is true that color is a powerful fundamental clue that can be used in the hand localization process because the color image segmentation is computationally fast and relatively robust for the changes in illumination, in scale and in rotation to shaded or complex background. However, problems of color-based localization may rise if the background or lighting conditions contain similar colors to human skin. In such cases, the color-based techniques are either unable to detected skin regions or falsely detect non-skin textures. The problem can be somewhat alleviated by sizing the regions of images to a certain size (scale filtering) [63] or restricting certain spatial position (positional filtering) ([4], [93], [117]). An alternative solution to the problem is the use of restrictive

backgrounds and clothing, such as uniform black background and long dark sleeves [104]. Finally, many of the gesture recognition applications rely on the use of uniquely colored gloves or markers on hands/fingers. The use of background restriction or colored gloves makes it easy to localize the hand efficiently and applicable in real-time, but imposes the extra restriction on the user and the interface setup. On the other side, the color-based localization methods without these restrictions are computationally intensive thus unable to implement in real-time.

3.2 Motion-based Gesture Analysis

Studies on human perception show that the visual system uses changes in luminosity in many cases to set the focus of attention. A change of brightness in one part of the visual field, such as flashing light, attracts our attention. Inspired by this principle, researchers have developed various motion-based techniques to assist in localizing human hands in images ([9], [10], [28], [33], [61]). Motion-based techniques work well when some assumptions about gesturer hold. For example, in the HCI context, it is usually believed that in most cases only one person gesticulates at any given time. Moreover, the body of the gesturer and the background are usually stationary. Consequently, the most active motion in the visual image is usually the motion of the arm/hand of the gesturer and can thus be used to localize her/him.

3.2.1 Image Differencing

Image differencing is a simple yet efficient method to monitor changes in image streams. The idea behind image differencing is to measure changes in and between consecutive images or between the current image and the background ([46], [102]). It tries to segment a moving foreground from a static background by comparing the gray-vales of successive frames in a pixel-by-pixel fashion. For example, two consecutive images I_{t-1} and I_t are taken from hand image sequence, where the camera position is fixed. Then image I_{t-1} is subtracted from I_t and the resultant image contains only information about the differences between those two frames. Usually the image I_{t-1} refers to the background

image (or referencing image) where no hand is presented. Since a fixed background is assumed, the only differences between frames should be hand that has moved. **Figure 10** shows the process of image differencing.

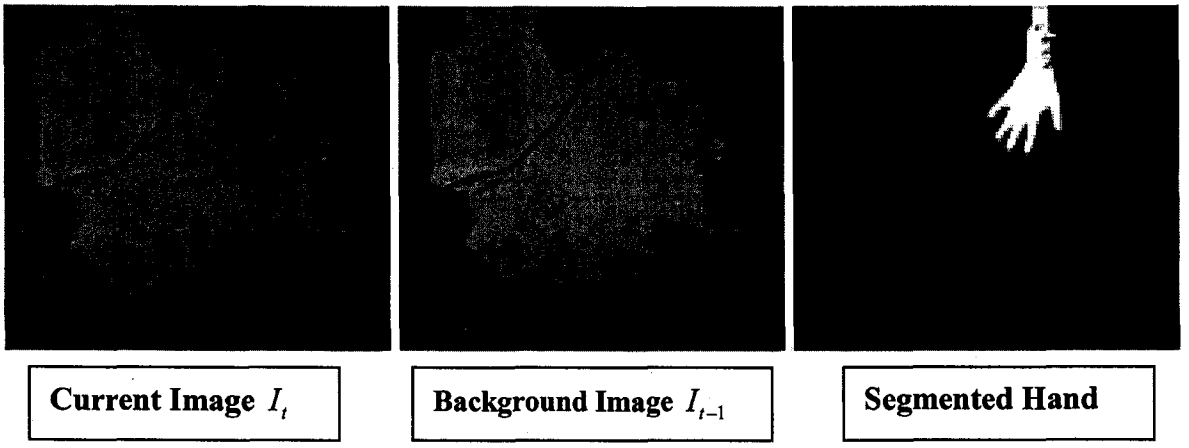


Figure 10: Image differencing process

3.2.2 Background Image Generation

Image differencing methods are widely used due to their simplicity. However, background subtraction methods are highly susceptible to noise and require that no change, including lighting changes, occur in background image ([62], [138]). Lighting conditions are paramount in this type of segmentation. Especially, outdoor scenes with lighting from the sun suffer dramatic lighting changes if the sun rays are blocked momentarily by a cloud. Consequentially luminosity will be changed in a scene, causing any simple background subtraction process to fail.

A method of overcoming this drawback is the lighting invariant background described by Wren and is often referred to in the literature as Pfinder ([84], [135]). It is possible to model the scene as a static background and dynamic foreground by building a background model of the variations of intensity such that each pixel has a Gaussian distribution. The foreground is modeled as a number of blobs each sharing statistically similar color and spatial properties. Statistical texture properties of the background observed over an extended period of time are used to construct a model. This model is used to decide which pixels in an input image fall into the background class [16].

Another solution for dealing with changes in illumination is to update reference image with newly arriving image using the following formula (1):

$$\forall x, \forall y \quad R_t(x,y) = \frac{N-1}{N}.R_{t-1}(x,y) + \frac{1}{N}.I(x,y) \qquad (1)$$

Where, R stands for the reference image and I for the newly arrived frame. The formula calculates a running average over all frames, with a weighting that decreases exponentially over time. With this adaptive background generator, the elimination of lighting changes will be maximized.

3.3 A Hybrid Model for Hand Localization

The localization process finds the locations of the interesting objects. Based on the discussions above, it is clear that neither color-based techniques nor motion-based algorithms offer robust performance respectively. For example, in case of a complex background, color cue may fail, but localization could be performed using the cue of motion [139]. On the contrary, in real environments where motion cue is nearly impossible to be used due to the dynamically changed lighting, color cue is usually the better choice [141]. Therefore, we contribute a hybrid model which integrates all these cues in order to put no restrictions on the user and the environments. In such scenario, most of the defects of both methods can be alleviated, if not completely be overcome. The hybrid appearance-based model consists of two algorithms for skin color detection and hand motion detection which will be described in Chapter 6.

4. 3D Hand Position Tracking

3D hand tracking has great potential as a tool for better human-computer interaction ([54], [142]). According to the definition of hand gesture, every hand gesture is performed by hand and arm movements in 3D space. Therefore, a stream of images is needed to represent the whole hand gesture. Each frame in the stream is a static image in which a hand appears. Hand tracking is to find corresponding hand in consecutive frames. Although rigid object tracking has been well studied, tracking hands, in particular finger motion, is a challenging problem because the motion exhibits many degrees of freedom and few features which are used in rigid object tracking can be applied ([44], [83], [119]). Estimation of hand position in only an image (or video sequence) of a hand is rather difficult. Other obstacles that have limited the use of hand trackers in real applications are the handling of self-occlusion (very common in hand motion) ([77], [100]), tracking in cluttered backgrounds, and automatic tracker initialization. Note that 3D tracking is different from gesture recognition, where there is a limited set of hand poses to recognize.

4.1 Feature Selection and Detection

Hand gesture features are the source used to estimate hand position. It will largely influence the performance of hand tracking. A widely used feature in gesture analysis is the fingertip. Both 3D-based models and appearance-based models can employ fingertip localization technique to obtain parameters. Other features used to estimate the parameters include hand/arm silhouettes and contours ([27], [140]). The computational complexity of the detection for those features is relatively low because most of the work is completed in the gesturer localization stage.

4.1.1 Fingertips

Fingertip is one of common features used in gesture-based interface. Various techniques exist to detect fingertip locations. A simple and effective solution to the fingertip detection problem is to use marked gloves or color markers to designate the targeted fingertips (**Figure 11**) [79]. Color histogram techniques can greatly help to extract the information of fingertip location. A different method to detect fingertips is using pattern-matching techniques in which templates can be images of fingertips or fingers or generic 3D cylindrical models. The pattern matching techniques can be improved by combining with other image features including contours ([35], [86]). Some fingertip extraction algorithms rely on the characteristic properties of fingertips in the image. For instance, curvature of a fingertip outline follows a characteristic pattern (low-high-low), which can be cues for feature detection ([40], [81], [89]). However, very often one or more fingers are occluded by the palm or other fingers from a certain viewpoint and direction. Using multiple cameras will solve this occlusion problem. Other solutions use the knowledge of the 3D model of the gesture to estimate the occluded fingertip positions.

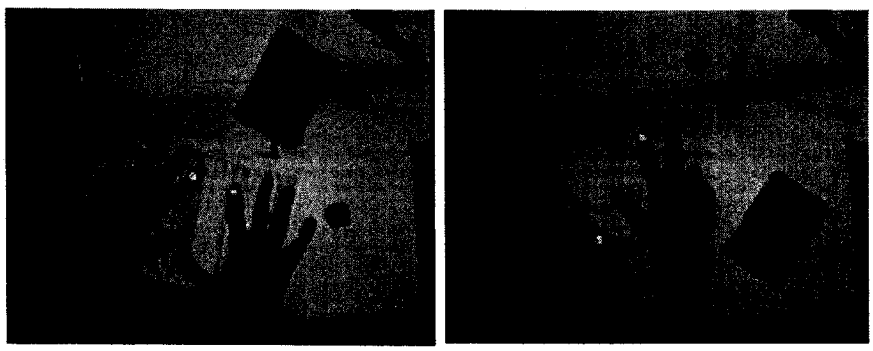


Figure 11: Marked fingertips in images

4.1.2 Silhouettes and Contours

Hand/arm silhouettes are the simplest, yet most frequently used features since silhouettes are easy to extract from local hand and arm images in a simple uniform background. While in the case of complex backgrounds, color histogram may be needed. Examples of

the use of silhouettes as features are found in both 3D-based models techniques as well as in the appearance-based algorithms [91]. Contours are also commonly used features. Some contours are produced from simple hand-arm silhouettes, while the others are extracted from color or gray-level images. For example, in 3D-based models analyses, contours can be used to select finger and arm link candidates through the clustering of the sets of parallel edges. **Figure 12** shows the silhouette and contour of human hand.

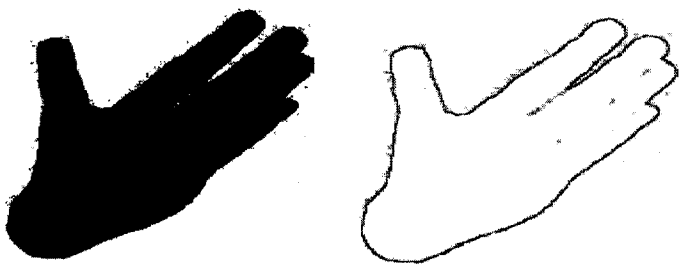


Figure 12: Hand features: silhouette and contour

4.1.3 3D-based Feature Estimation

The hand gestures in 3D-based model can be represented by two types of parameters: joint angles and dimensions of palm and finger [101]. To perform the estimation task, a recursive process is involved which requires initial values and an update schema. First the initial values of parameters have to be given. These values of finger lengths and palm dimensions often come from assumption based on training data. The initial values of joint angles are obtained via a complex and cumbersome process, which involves *inverse kinematics* algorithms. The 3D hand can be considered as a linked structure, in which the palm is the base and the fingertip is the end. Given a 3D position of finger, the inverse kinematics algorithms determine the joint angles for each joint. This process is computationally expensive and may result in multi solutions.

To speed up computation, some constraints are imposed on the parameter values. One approach is to use interpolation of the discrete *forward kinematics* mappings to approximate the inverse kinematics [3]. Given a table of the discrete values of the joint angles and the fingertip positions the approach can estimate the values of the joint angles for other fingertip positions which are not in table. Once the parameters of hand are initially estimated, the parameter estimates can be updated using some kind of prediction/smoothing scheme. A simple scheme reported in [56] can be employed. In this scheme, a simple silhouette matching between the 3D hand model and the real hand image was used to obtain satisfactory parameter estimation and update.

There are three major drawbacks associated with the mentioned 3D-based hand model parameter estimation approach. One has to deal with the computational complexity of any task involving the inverse kinematics. Another more serious problem is due to occlusions of the fingertips used as the model features. An obvious but expensive solution is to use multiple cameras. The third drawback roots in the employed assumption that the linear dimensions of the hand are known, which is essential in the inverse kinematics problems. Thus, any change in scale of the hand images always results in inaccurate estimates of the hand joint angles.

4.2 Projective Geometry

Projective geometry is the essential mathematical basis for computer vision. It explains how the points in 3D scene are projected onto image plane (see **Figure 13**). The center of projection is at the origin O of the 3D reference frame of the space. The image plane Π is parallel to the $(\vec{i},\vec{j})$ plane and displaced a distance f (*focal length*) along the $\vec{k}$ axis from the origin. The 3D point P projects to the image point p. The orthogonal projection of O onto Π is the principal point o, and the $\vec{k}$ axis which corresponds to this projection line is the optical axis.

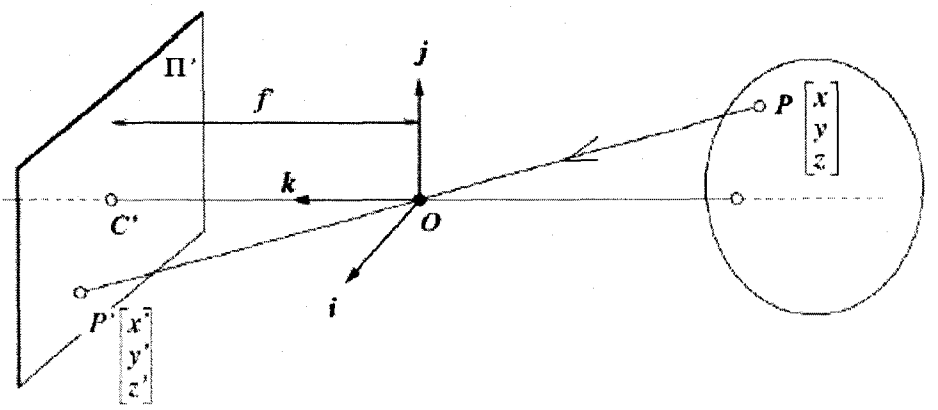


Figure 13: Projective geometry

4.2.1 3D Reconstruction of Single Point

With the knowledge of projective geometry, we can find that it is impossible to recover the depth of a point in 3D scene from a single image. For example, in **Figure 14**, both point P and Q are projected to p_1 on first image. Therefore, second image is definitely needed to identify the original point. In general, the projection of a 3D point $P = (X, Y, Z, 1)$ onto the pixel $p = (x, y, 1)$ is given by:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \qquad (2)$$

Where λ is a nonzero scaling factor.

Let p_1 and p_2 be the projection of a 3D space point P on the left and right images respectively. The projection yields 4 equations:

$$\begin{pmatrix} x_1 = \dfrac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \\ y_1 = \dfrac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \\ x_2 = \dfrac{n_{11}X + n_{12}Y + n_{13}Z + n_{14}}{n_{31}X + n_{32}Y + n_{33}Z + n_{34}} \\ y_2 = \dfrac{n_{21}X + n_{22}Y + n_{23}Z + n_{24}}{n_{31}X + n_{32}Y + n_{33}Z + n_{34}} \end{pmatrix} \qquad (3)$$

Where m_{ij} and n_{ij} are known as camera projection matrix. When the coordinates of points p_1 and p_2 are measured and camera matrix are estimated by calibration process, the 3D coordinates (X,Y,Z) can be obtained by solving 4 linear equations.

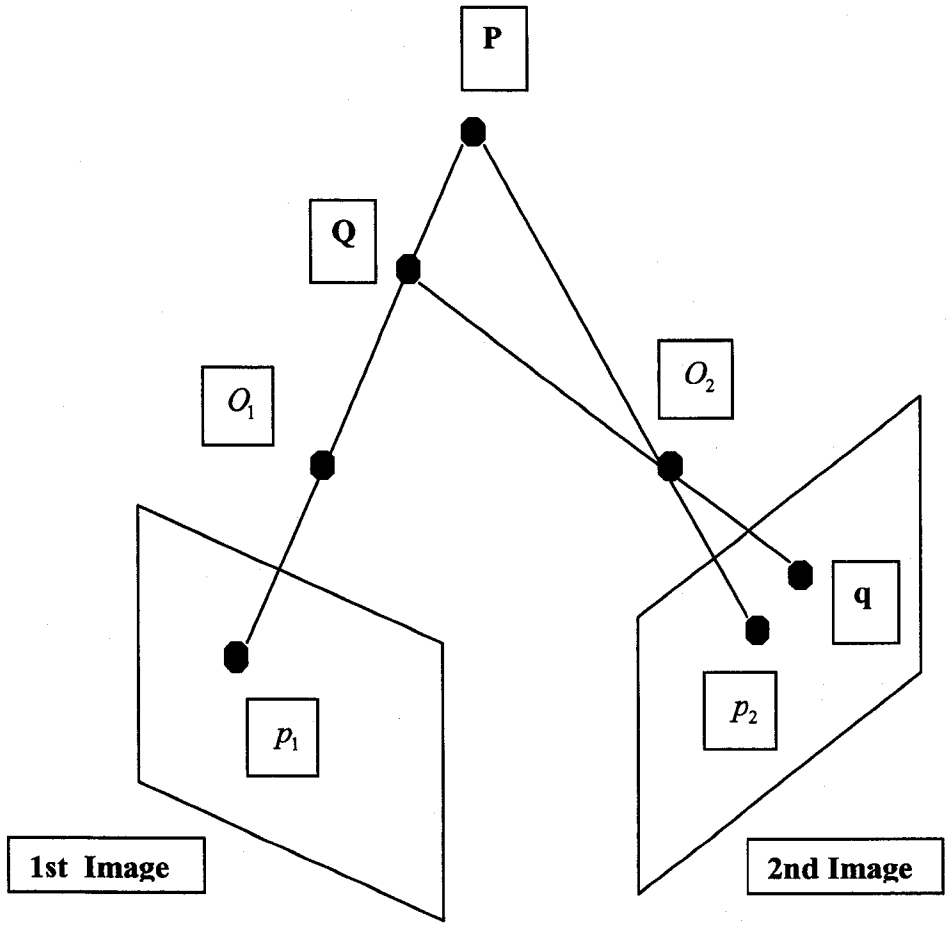


Figure 14: Stereovision geometry

4.2.2 Camera Calibration

The calibration process is necessary to estimate the value of camera projection matrix. In particular, the projection matrix is composed of two types of calibration parameters: internal and external. The external parameters are those that specify the pose and orientation of the camera in the world coordinate frame. They consist of 6 parameters: three angles for the scene-camera rotation and a translation vector that expresses the

origin of the scene coordinate system in the camera coordinate system. The internal calibration parameters are related to optical properties of the cameras. They consist of several parameters: two scale factors along the X-axis and the Y-axis and, the image coordinates of the center of the image. Recently, some researchers have taken geometry constraints into consideration and exploited the potentials of using un-calibrated cameras to recover the 3D scene. However, those approaches heavily rely on the result of edge and corner detection which is difficult in non-rigid object tracking. Additionally, a long stream of images (at least 200 frames) is often required ([132], [137]).

4.3 Hand Trajectory Reconstruction

In 3D hand tracking process, the central task is to recover hand trajectory in 3D space. Current research into tracking has somewhat diverged into two camps: informally these can be distinguished as low-level [82] and high-level [51]. Low-level approaches include "blob trackers" and systems which track sets of point features. Blob trackers perform low-level processing. For example color segmentation, usually applied on low-resolution images, are fast and robust but convey little information other than object centroid. Rigid object deformations can be tracked by matching point correspondence frame-to-frame, but this relies on a rich set of point features on the object of interest, and segmenting the sets of points into coherent objects is challenging.

An alternative is to use higher-level information, either by modeling objects with specific gray-level templates which may be allowed to deform, or with more abstract templates such as curved outlines [134]. By including high-level motion models, these trackers can follow complex deformations in high-dimensional spaces, but there tends to be a trade-off between speed and robustness. Kalman-filter based contour trackers which run in real time are very susceptible to distraction by clutter, and correlation-based systems are vulnerable to changes in object appearance and lighting [45], and rapidly slow down as the space of deformations increases in complexity.

Contour trackers have been constructed. They achieve highly robust to clutter by sacrificing real-time performance. The high-level approaches also tend to be economical on processing time by searching only those regions of the image where the object is predicted to be [135]. This diminishes robustness, and also precludes natural extensions of the trackers to perform initialization when the object could be anywhere in the space. The difficulty of initialization is compounded when the dimension of the tracking space increases, since it rapidly becomes impractical to perform an exhaustive search for the object.

In this thesis, we present a framework to bridge the gap between low-level and high-level tracking approaches. We focus on fingertip as the interesting points of hand and combine low-level and high-level tracking approaches to render the 3D trajectory of the hand. Let vector $V_t = (X, Y, Z, t)^T$ to be the 3D coordinate of fingertip at moment t. By tracking the fingertip frame by frame, we reconstruct the hand trajectory matrix M as follows:

$$M = (V_0, V_1, V_2, \bullet\bullet\bullet\bullet\bullet\bullet V_{t-1}, V_t) \quad (3)$$

Once the trajectory matrix M is acquired, the gesture classification then can be applied in gesture recognition phase. We discuss gesture recognition in the next chapter.

5. Gesture Recognition

Hand and arm gestures have received substantial attention among researchers who are working on performance improvement of HCI applications. Hand gesture recognition is a process in which the gestures performed by users are recognized and interpreted by computer. The vast majority of automatic recognition systems are for deictic gestures (pointing), emblematic gestures (isolated signs) and sign languages (with a limited vocabulary and syntax). Some are components of bimodal systems, integrated with speech recognition. Some produce precise hand and arm configuration while others only coarse motion. The existing applications of gesture recognition are mainly designed for communicative purpose. In such domain, Sign Language recognition is the long-term goal. Another group of recognition applications are for manipulative purpose. For example, in virtual assembling environment, gestures are used to issue commands to control robots without any physical contact between human and the computer.

5.1 Gesture Vocabulary

It is common that people do not understand or misunderstand other's hand gestures especially when the gesturers are from different regions. The ambiguity of gesture stemming from different cultures and custom is the major challenge in gesture recognition applications. Therefore, it is necessary to predefine gesture vocabulary for minimizing the ambiguity. In fact, gesture vocabulary is playing a pivotal role in those applications which aim at recognition of formal signed language.

Understanding a formal signed language, such as the American Sign Language (ASL), is naturally one of the driving tasks for gesture recognition systems. Such systems could play an important role in communication with people with a communication-impairment like deafness. A mechanism which could automatically translate ASL hand gestures into

speech signals would undoubtedly have a positive impact on such individuals. ASL is the native language of the American deaf community and each gesture consists of a specific hand-shape and a specific hand-motion.

However, it is true that the recognition of full, dynamic gestures representing words and concepts as they do in the ASL is undoubtedly the most difficult problem in such area. There has not been any system with these capabilities reported in the literature because ASL contains thousands of formal gestures. **Figure 15** shows first half alphabets in ASL. A practical solution to this problem is to reduce the vocabulary. Some applications implemented the recognition task based on a subset of full vocabulary (**Table 2**).

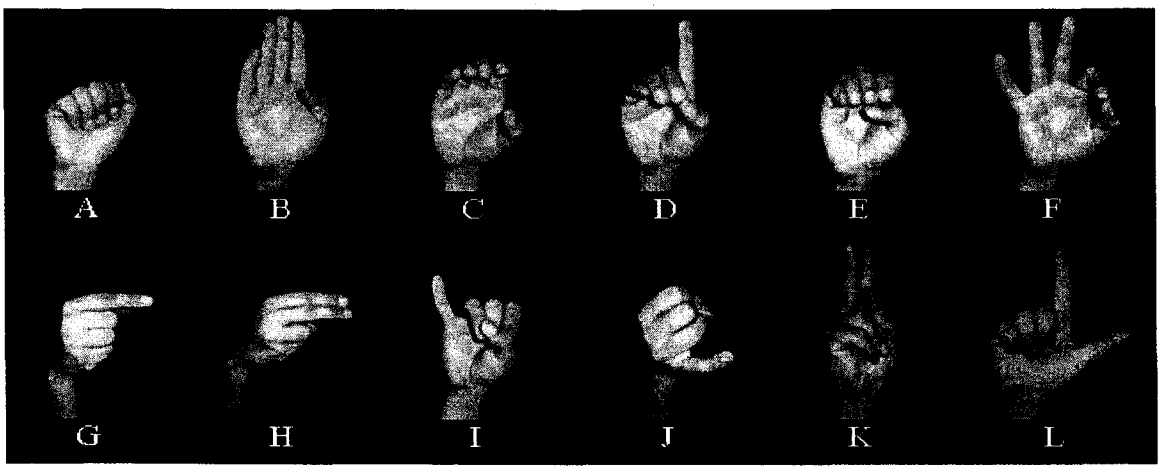


Figure 15: Gestures standing for letter 'A'-'L' in ASL.

Applications	Gesture Vocabulary	Recognition Rate
Tele-Assistance	6	87%
Robot Control	10	80%
Augmented Reality	15	78%
Writing Sign Language Recognition [25]	40	51%

Table 2: Gesture vocabulary in recognition applications

5.2 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a useful statistical technique for finding patterns in data of high dimensions. The objectives of PCA are to discover and reduce the dimensionality of the data set as well as to identify new meaningful underlying variables [116]. During the definition process of the gesture vocabulary, PCA on a large number of training data is usually involved to extract most distinguishable components of mathematical representation of each gesture for later classification step. Moreover, PCA allows us to represent images as points in a low-dimensional space.

For instance, if each image is composed of 32x32 pixels whose values vary from 0 to 255, then each image defines a point in a 1024-dimensional space. If a sequence of images representing a gesture is obtained, then this sequence will generate a sequence of points in space. However, this set of points will usually lie on a low-dimensional sub-space within the global 1024-dimensional space. The PCA algorithm allows us to find a sub-space, which usually consists of up to three dimensions. This enables the visualization of the sequence of points representing the gesture (**Figure 16**). PCA also facilitates graphic classification and comparison which are used in some applications.

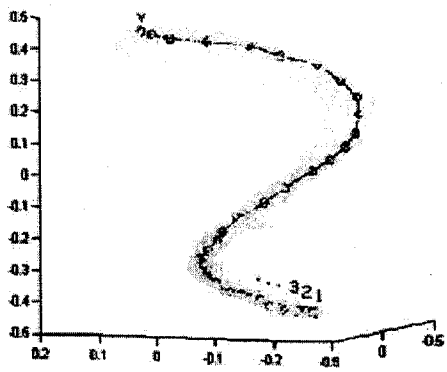


Figure 16: Visualization of a gesture.

5.3 Hand Posture Recognition

Hand postures express certain concepts through hand configurations which are represented by model parameters. The task of hand posture recognition is closely related to the choice of parameters. Optimization of the model parameters usually takes place

first before the classification. It is necessary due to the fact that some parameters selected in the gesture modeling stage may be improper for recognition. To recognize static gesture, parameter clustering techniques are usually performed. One of such techniques is vector quantization, which separate high dimensional space into lower dimensional space by hyper-planes, based on training examples. If the parameters are chosen particularly to ease the recognition, as in ([24], [122]), the separation of gestures could be done easily.

However, if the parameters are designed without the consideration of recognition, the classification of such gestures may be difficult or even impossible. For example, with contour descriptors, a few hand gestures will be confusing in the recognition process. Hence, contours are often used for hand or arm tracking instead of recognizing ([29], [85]). Another type of parameter, geometric moment parameters, encounters the same problem. This type of parameters is sensitive to rotation. So a slight change in rotation of the same hand posture can lead to false recognition.

Another approach is to introduce different parameters, say, Zernike moments [105] or orientation histograms [37], [38] or other invariants in projection of 3D to 2D [131], which are not only helpful for modeling the gestures but also insensitive to rotation. Alternatively, the use of nonlinear classifier, such neural network classifier in ([64], [130]) will improve the accuracy of recognition. In 2000, an approach [115] is proposed in which hand posture estimation is completed by combining 2D appearance-based and 3D model-based approaches.

5.4 Gesture Motion Recognition

Different from static gestures, dynamic gesture involve both the temporal and the spatial properties ([18], [50], [74]). Because of the temporal nature of gesture motion, the major requirement of any recognition algorithm is time instance invariant and time scale invariant. For example, a clapping gesture should be recognized correctly no matter how long the interval is between two claps [136]. As the similarities of spoken language and gesture, an approach that has solved the problem in speech recognition has been

successfully migrated to gesture recognition context. It is called automatic speech recognition (ASR).

In speech recognition field, a difficulty is to recognize the spoken words independent of their duration and variation in pronunciation. A technique, named *Hidden Markov Models* (HMM) ([98], [106], [123]), has made such recognition possible. Naturally, the HMM has been employed in gesture motion recognition. HMM is a doubly stochastic process dealing with hidden and observable states. The hidden states "drive" the model dynamics—at each time instance the model is in one of its hidden states. Transitions between the hidden states are controlled by probabilistic rules. The observable states produce outcomes during hidden state transitions or while the model is in one of its hidden states. Such outcomes are measurable by an outside observer. The outcomes are governed by probabilistic rules.

Another approach for gesture recognition is *motion energy* [118]. It is based on temporal templates which accumulate the motion history of a sequence of visual images into a single 2D image. The recognition task then can be performed by using any of the 2D image clustering algorithms.

5.5 Computational Complexity

The computational complexity of a recognition approach is critical in HCI applications. The cost came from model complexity and recognition expense. There is a conflict between model complexity and gesture recognition. The more complex the model is, the wider class of gestures can be recognized; consequently, the computational complexity of recognition increases. Most of the 3D model-based gesture models have more than 10 parameters. The successive approximation procedures of their parameter calculation are computationally expensive. Real applications based on such models rarely show close to real-time performance. For instance, the performance ranges from 45 minutes per single frame in [125] to 10 frames per second in [97]. With respect to the consideration of time performance, the appearance-based models are usually used for their lower cost in computation.

6. Our Vision-based Gesture Interface

We design a vision-based gesture interface with a new hybrid hand localization model that combines hand localization mechanism with robust 3D hand trajectory. In this chapter, our project is systematically described and our algorithms will be discussed in detail. Our contributions in this paper are also underlined.

6.1 Problem Domain

With the massive computerization in society, human-computer interaction has become an increasingly important part of our daily life. Furthermore, rapid developments of novel technologies, including virtual reality, augmented reality, robotics and virtual assembly, are fuelling in research toward novel devices and techniques that will support faster and easier HCI.

Virtual Assembly (VA) is a Virtual Reality (VR)-based engineering application that allows engineers to plan, evaluate, and verify the assembly of mechanical systems. The goal of virtual assembly systems is to produce optimized assembly sequence that is ready to direct robotic manipulators implementing assembly tasks and putting together machinery parts into products ([121], []). Recently, VA is gradually being accepted as a tool for digital prototyping in manufacturing industries, because it offers many advantages: rapid design/test cycles, low prototyping costs, efficient learning, convenient platform for simultaneous and even distributed engineering teams.

Virtual assembly, however, is one of the most challenging applications of virtual reality. This is mostly due to the very high interactivity: it is not only the high amount of functionality needed, but also because some of the interaction must be as natural as possible. After all, it is the interaction itself that is to be simulated, especially the type of

interaction that mostly involves human hands. This is in contrast to other VR applications like styling review, design review, or lighting simulation where the amount of immersive functionality is much less than virtual assembly.

For example, in virtual assembly, robots need to perceive dynamically change of real world in order to find paths automatically ([5], [145]). In addition, computer-controlled mechanic arms need to learn the process of assembling products. In both cases, traditional input interface such as the keyboard-mouse combination is no longer adequate. Since gestural language has always been an important aspect of human interaction, the development of a hand gesture system is highly advantageous for improvements of virtual assembling.

In many earlier hand gesture systems, hand tracking and gesture recognition are achieved with the assistance of specialized devices (e.g., data glove, markers, etc.) [144]. Although they provide accurate tracking and shape information, they are too cumbersome for use over extended periods. Meanwhile, the wires connected with devices often limit the distance of movement and inhibit freedom of orientation. Vision-based gesture interface seems to be a more suitable alternative that recognizes hand gestures with computer vision techniques. A typical virtual assembly workspace is shown in **Figure 17**.

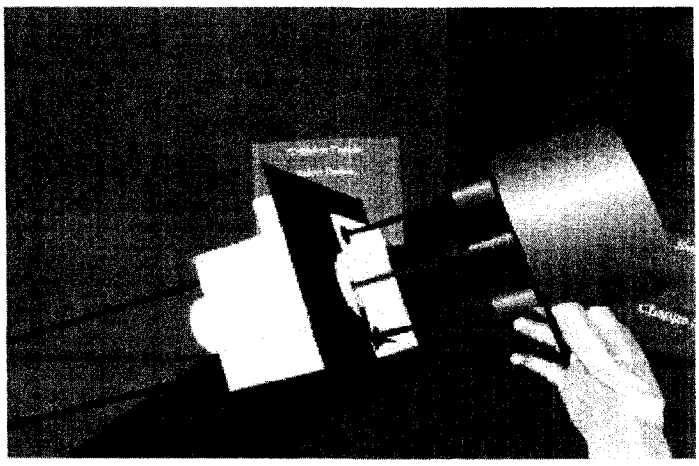


Figure 17: Virtual parts are being controlled by a virtual hand.

However, existing hand gesture recognition applications usually only work well under a set of assumptions as addressed in previous chapters. Some applications only provide 2 degrees of freedom (DOF). Some applications require a training process for new users [124]. In human hand tracking systems, the results are often unsatisfied in real environment. Color-based approaches need unchangeable lighting while motion-based approaches usually require static and monochroic background. It is becoming obvious that new localization algorithms and robust 3D hand trajectory methods are desirable. This is the motivation of developing novel vision-based hand gesture interface.

6.2 System Overview

6.2.1 System Requirements

It is nearly impossible to develop an "all-in-one" gesture-based interface which is applicable in any environment. We design a vision-based gesture interface whose requirements are closely associated with virtual assembly demands. In virtual manufacturing workspace, robots or mechanic arms will learn the assembling sequence through gesture-based interface. Human operators also will indicate the physical constraints by the interaction with machine. Having surveyed various existing VA applications in the literature, we found excess constraints imposed on the users and environments extremely hinder the advance of VA. Consequently, we stress system requirements as follows:

- Users can interact with system with bare hands. It is a device-independent system.
- System can tolerate complex & dynamic background, varying lighting.
- No training process is needed for new users.
- System can recognize a set of predetermined hand gestures.
- System can dynamically track user fingertip positions in 3D space.

6.2.2 System Architecture

Our interface design is tightly based on the system requirements, which aim at underscoring our contributions. As the 3D hand trajectories have to be rendered, we use two cameras (camera A and B in **Figure 18**) to look at user bare hands. Each camera provides a stream of images where the user hand is focused. As the projective geometry suggests, a camera calibration pre-process is involved. In the initialization stage, a user register function is called to obtain the skin tone pattern of user for later color-based analysis. Meanwhile, background information will be stored for motion-based analysis.

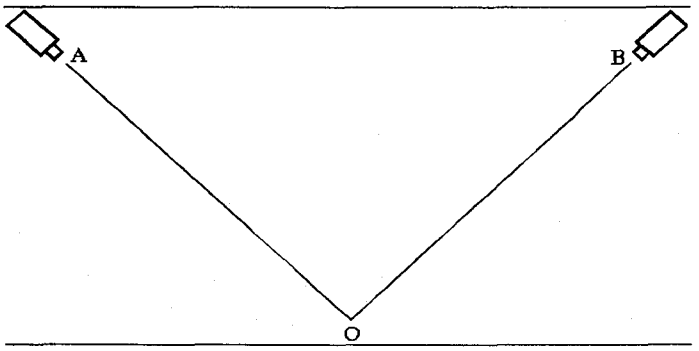


Figure 18: Two fixed cameras are placed in front of users.

In hand modeling stage, we choose contour as the descriptor of hand mainly for two reasons. First, hand contour is easy to detect and segment. Second, it supports fast computation in our fingertip detection algorithm. Once the raw image sequence is acquired, our localization/segmentation process is invoked. This process will search the image and find the region where the user hand resides. A skin detector and a motion detector are combined to quickly localize the hand. In the fingertip detection component, K-curvature algorithm is used to identify the portion of fingertip. After fingertip is focused correctly, either tracking process is called to monitor the hand position and yield

3D hand trajectory, or a gesture recognition function is launched to issue commands. **Figure 19** shows the architecture of our system.

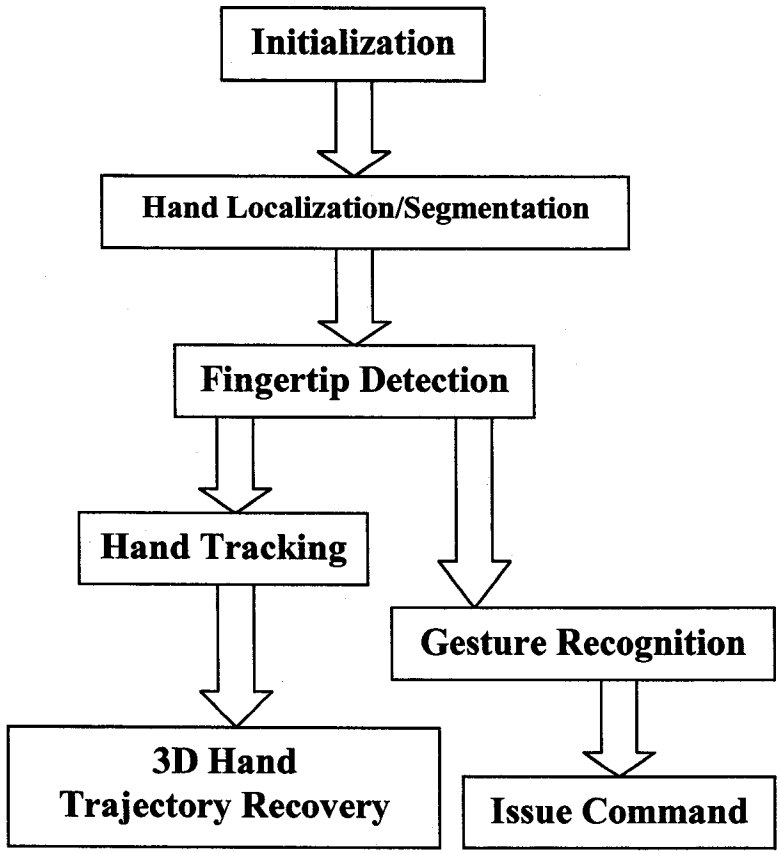


Figure 19: Gesture-based interface architecture.

6.3 Skin Detection Algorithm

Detection of skin in images is an important component of our system for detecting, recognizing, and tracking user hands. The skin detection method used in this thesis is color-histogram based approach that is intended to work with a wide variety of individuals, lighting conditions, and skin tones. As pixel classification performance is

mainly determined by different color spaces, a comparative evaluation of different color spaces is necessary.

6.3.1 Color Space Selection

Individual color spaces used in skin detection methods include HSV, a variant of Hue and Saturation, Normalized RGB, simple RGB, YIQ, and transformations from CIE XYZ, including Farnsworth and CIE L*a*b* [147]. We developed an experiment to decide which color space is yielding satisfactory result of pixel classification among simple RGB, HSV and YIQ color spaces. We used 48 images to train the classifier and 64 other images for testing. The images were downloaded from a variety of sources, including frames from movies and television, professional publicity photos and amateur photographs. The images were selected so as to include a wide range of skin tones, environments, cameras, and lighting conditions. Some of the images depicted multiple individuals and the quality of the images varied. The experimental result is shown in **Figure 20** where HSV color space, yielding 72% correct classification rate, is the best choice.

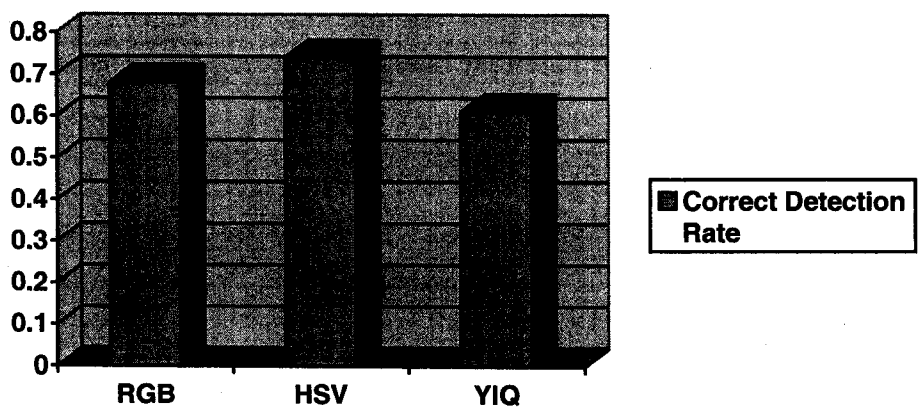


Figure 20: Different color spaces comparison.

The transformation formula used in our system from RGB to HSV is given as follows:

$$H = \begin{cases} H_1 & \text{if } B \le G \\ 360^\circ - H_1 & \text{if } B > G \end{cases} \quad (1)$$

where

$$H_1 = \cos^{-1}\left\{ \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}} \right\}$$

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)} \quad (2)$$

$$V = \frac{\max(R,G,B)}{255} \quad (3)$$

6.3.2 Bayesian Classifier

Our skin detection algorithm is based on pixel classification method which classifies each pixel in the raw image either into skin class or non-skin class. The theoretical basis of this method is Bayesian decision rule. In general, the classifier refers to discriminant function. In a c-class case, discriminant functions, denoted by $g_i(\vec{x})$, where i=1,2,...,c, are used to partition the feature space R^d as follows:

Assign $\vec{x}$ to class w_m if $g_m(\vec{x}) > g_i(\vec{x})$ $\forall i = 1,2,...c$ and $i \neq m$.

In Bayesian theory, the discriminant function is the form of

$$P(C_i \mid x) = \frac{p(x \mid C_i) P(C_i)}{p(x)} \quad (3)$$

where $P(C_i \mid x)$ is the *posteriori* probability, $P(C_i)$ is the priori class probability, and $p(x \mid C_i)$ is the probability distribution.

In our skin detection case, we have two classes: skin class C_1 and non-skin class C_2. $P(C_i \mid x)$ is the probability of observing a pixel belonging to class C_i given that its color is x. Thus, $P(C_1 \mid x)$ and $P(C_2 \mid x)$ are the respective *a posteriori* probabilities for skin and non-skin color classes. Since $\frac{P(C_i)}{p(x)}$ is just a scale factor which is unimportant for comparison, we just adopt the following decision rule:

classify x to a skin pixel if $P(x \mid C_1) > P(x \mid C_2)$,
classify x to a non-skin class, otherwise.

There are three approaches to the estimation of $p(x \mid C_i)$, namely parametric, non-parametric and semi-parametric. The parametric approach assumes a functional form of $p(x \mid C_i)$, which can be customized by a set of parameters [14]. The non-parametric approach, on the other hand, does not express $p(x \mid C_i)$ in a parametric form but allows it to be determined entirely by the training data [19]. Semi-parametric approach, most notably neural networks, uses very general functional form that can have a variable number of adjustable parameters. In our algorithm, we use Gaussian densities to approximate the $p(x \mid C_i)$ with the following formula:

$$p(x \mid C_i) = (2\pi)^{-d/2} \mid \Sigma_i \mid^{-1/2} \exp[-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)] \quad (4)$$

where

$$\mu_i = \frac{1}{N_i} \sum_k^{N_i} x_k$$

$$\Sigma_i = \frac{1}{N_i - 1} \sum_k^{N_i} (x_k - \mu_i)(x_k - \mu_i)^T$$

Since the pixel classifier may not yield 100% accuracy, it is true that occasionally some skin pixels are falsely classified into non-skin class. To deal with this problem, our

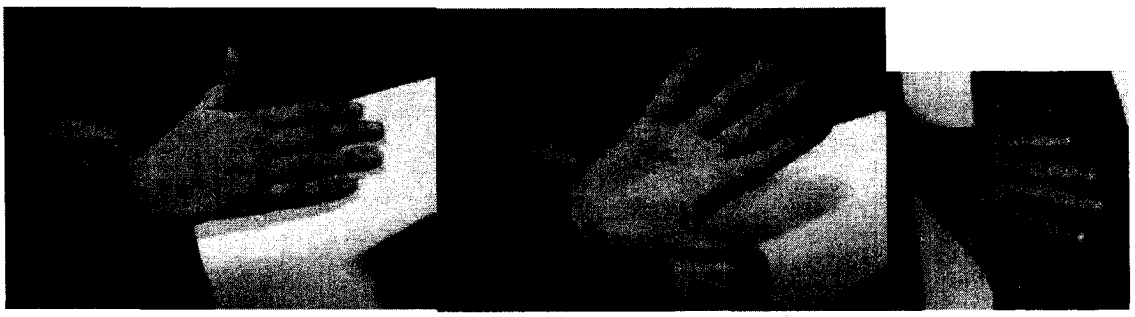
algorithm does not operate single pixel but deals with a window (3X3 pixels) of pixels. It is similar to use texture mapping mechanism where not only the candidate pixel is considered but also the neighborhoods of pixels are involved for helping classification. It will greatly eliminate the negative influence of outliners. Another improvement of the classifier in our system is to use online training schema. Once the user hand is localized correctly, the color data of the hand is immediately used for further training of the classifier. This heuristic training data updating makes classifier more sensitive to a specific user.

6.3.3 Skin Tone Pattern Filter

Skin color detection works well when there is only one user in the image. However, in real environments, background of the images may contain other people, even other hands which are named noise patches. Since the color detector is trained to distinguish skin pixels and non-skin pixels, we cannot simply reuse it to differentiate pixels of a "wrong hand" from pixels of "the right hand". In such case, the noise patches in background will also be recognized as skin incorrectly. To deal with this problem, we add a skin tone pattern filter which can remove the noisy hands in background. It is widely believed that the skin tone pattern of different people varies significantly. Our filter is based on this distinguishable characteristic. In HSV color space, tone is represented by components H and S. Suppose F_H is the skin hue pattern matching indicator, H_{user} is the hue pattern of user skin which has been previously calculated in the initialization stage, H_{noise} is the hue pattern of noisy hand, matching indicator is quantified by the following formula:

$$F_H = \frac{\sum_{(u,v)\in W} H_{user}(u,v) \bullet H_{noise}(x+u, y+v)}{\sqrt{\sum_{(u,v)\in W} (H_{user}(u,v) - \overline{H}_{user})^2 \bullet \sum_{(u,v)\in W} (H_{noise}(x+u, y+v) - \overline{H}_{noise})^2}}$$

where w is the search window and (x,y) is the coordinate in image. In our experiments, when we applied this formula on the one user's hand in the same image, F_H is 1. When other image of same user's hand is used, F_H varies from 0.8-1.0. However, when other user's hand comes, F_H is no greater than 0.65. **Figure 21** shows the sample images.



(a) (b) (c)

Figure 21: Skin tone filter. (a) is reference hand, (b) is the testing hand of same user, and (c) is a noisy hand from another user.

6.4 Gesture Recognition

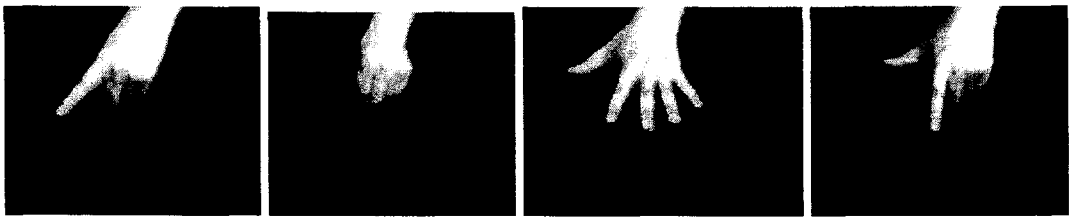
6.4.1 Gesture Vocabulary Definition

In a virtual assembly environment, all machinery parts are modeled as graphic objects. Robots then fit those parts together by performing assembly tasks in specified sequences [144]. With hand gesture interface, the human users are able to program the assembly sequences in an intuitive way where hand motions are continuously tracked and hand postures are recognized as gesture commands by robotic manipulators. For instance, a closing hand means the *hold* command to grasp a virtual object; an open hand issues a *free* command to release the object; a *point* sign invokes items for section; and an "OK" posture launch *quit* action.

To imitate such scenario, we design the gesture vocabulary as showed in **Table 3** and **Figure 22**:

Gesture	Meaning
A close hand with index finger stretch out (a)	Selection action
A closing hand (b)	Grasp and hold
An open hand (c)	Release
A close hand with thumb and index finger stretch out (d)	Quite Session

Table 3: Gesture vocabulary in the system.



(a) Select (b) Hold (c) Release (d) Quit

Figure 22: Gesture commands in our system

6.4.2 Fingertip Finder

Based on our gesture vocabulary, we center on fingertip as the hand local feature. From **Figure 22**, it is manifest that those gestures are easily distinguished by fingertip. Gesture (a) contains one fingertip, gesture (b) has no fingertip, gesture (c) has more than two (actually five) fingertips and gesture (d) has two fingertips. Thus, the number of fingertips can be used to classify gestures. The classification rule is described in **Table 4**.

Gesture	Number of Fingertips
Selection	=1
Grasp and hold	=0
Release	>2
Quit	=2

Table 4: Gesture classification rule based on number of fingertips.

We develop fingertip finder algorithm which is based on K-curvature measurement. Once the hand is localized and segmented from original images, the boundary pixels of the hand contour image are easily extracted. Those pixels that reside in the outline of the hand form a list in which each pixel is denoted by $P(i) = (x(i), y(i))$. The K-curvature is the angle C(i) between two vectors [P(i-k), P(i)] and [P(i), P(i+k)], where k is a constant. The pixels along the boundary where the curvature reach a local extremum, that is the "local features", are then identified. Some of these local features are labeled "peaks" or "valleys". Peaks are those features whose curvatures are positive (denoting a locally convex boundary) with magnitudes greater than a fixed threshold P_{thr}. Valleys are features whose curvatures are negative (denoting a locally concave boundary) with magnitudes less than a fixed threshold V_{thr}. In **Figure 23,** all three gestures in our vocabulary are segmented and contour pixels are extracted. Circles represent peaks and squares denote valleys after applying our fingertip finder. **Figure 24** is the local view of K-curvature measurement at fingertip and other boundary.

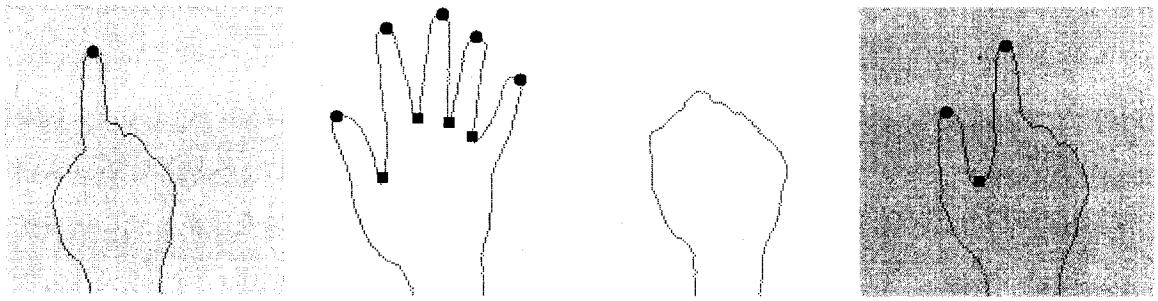
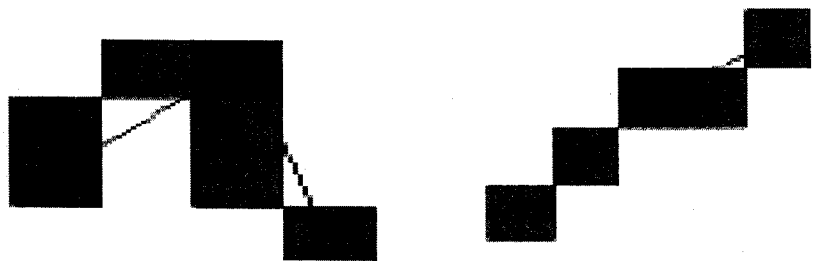


Figure 23: Local features are labeled based on fingertip finder algorithm.



(a) (b)

Figure 24: K-curvature finger finder. (a) is the K-curvature value of pixels at fingertip and (b) is the K-curvature value of pixels at palm.

6.5 Experimental Result

6.5.1 Initialization Data

We implemented our system using Microsoft MFC. The device is two calibrated Logitech cameras and a desktop computer. The camera projection matrix is given as follows:

$$Matrix_1 = \begin{bmatrix} -998.671 & -3.12962e-05 & -255.66 & -25566 \\ 2.88603e-05 & -998.671 & -255.66 & -25566 \\ 1.23056e-07 & -1.13641e-07 & -0.998672 & -99.8671 \end{bmatrix}$$

$$Matrix_2 = \begin{bmatrix} -970.423 & -145.242 & -321.632 & -23440.2 \\ 109.173 & -1024.38 & -70.7075 & -18354.6 \\ 0.0871855 & -0.173046 & -0.981395 & -103.141 \end{bmatrix}$$

The analysis result on offline training data for skin color classifier is shown as follows:

$$Mean = \begin{pmatrix} Red = 107.1 \\ Green = 81.6 \\ Blue = 66.3 \end{pmatrix}$$

6.5.2 Hand Localization and Segmentation

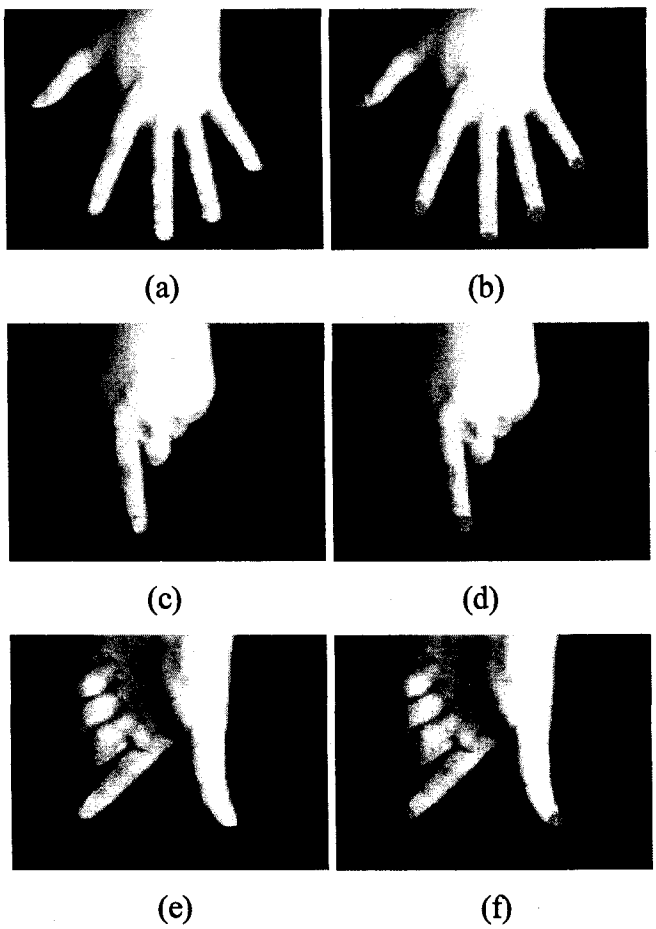
The hand localization and segmentation process is then applied on those frames. **Figure 25** shows the result.

(a) (b) (c)

Figure 25: Hand segmentation result. (a) is the background frame used as referencing image, (b) is the next frame where hand appears, (c) is the result image after segmentation process applied.

6.5.3 Fingertip Detection

To test our fingertip detection algorithm, we used images of various gestures as showed in Figure **26**.



(a) (b)

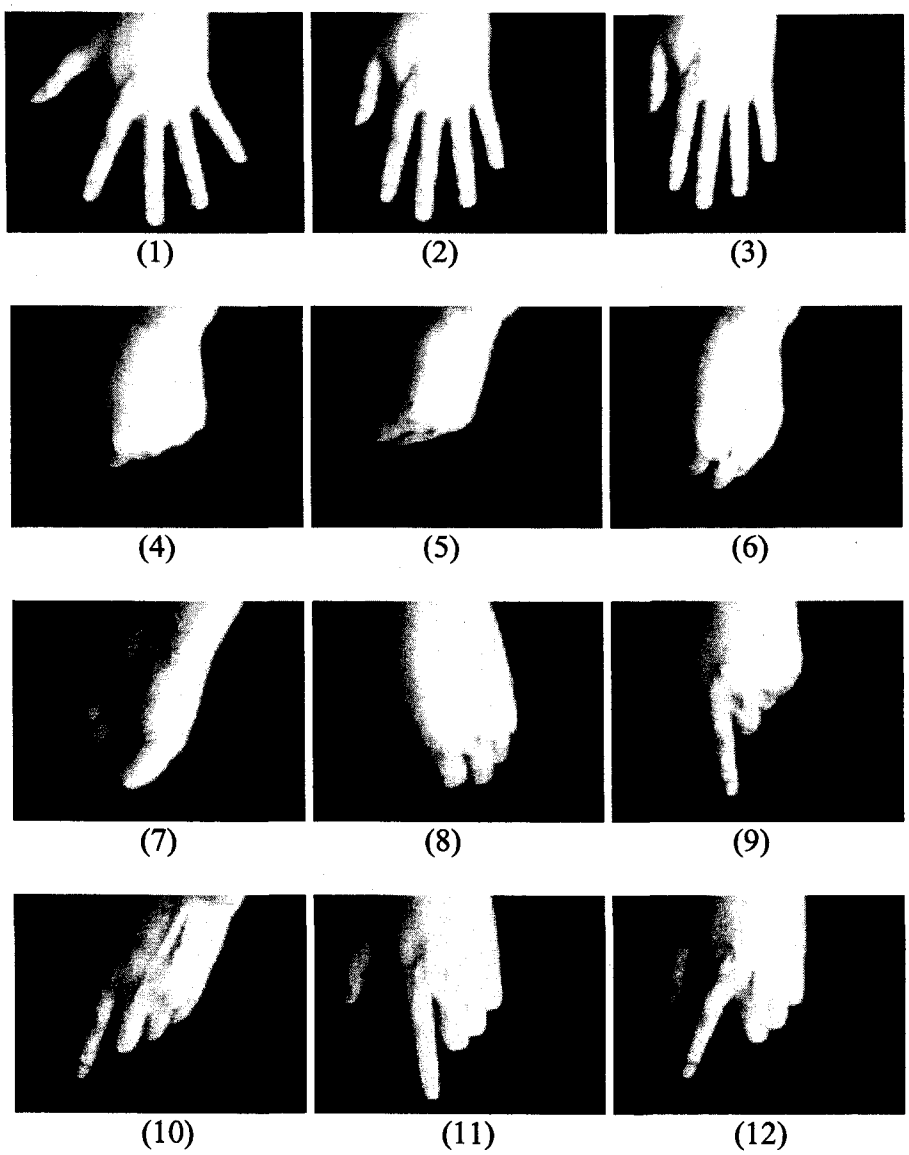
(c) (d)

(e) (f)

Figure 26: Fingertips are detected and labeled with green dots.

6.5.4 Gesture Recognition

In order to evaluate the robustness of gesture recognizer, we intentionally rotated the hand and joint angles to verify the recognition function. **Figure 27** shows the gesture images used as the test bed. Images (1) is the standard "*Release*" gesture while (2) and (3) are more nature gestures. Images (4) is the standard "*Hold*" gesture and (5)-(8) were casual closing hand. Image (9) and (10) are two pointing gesture. Image (11) is standard "*Quit*" gesture command and (12)-(15) are same gestures which are slightly rotated. Our experiment results showed that all of those gestures are recognized correctly by the recognizer.

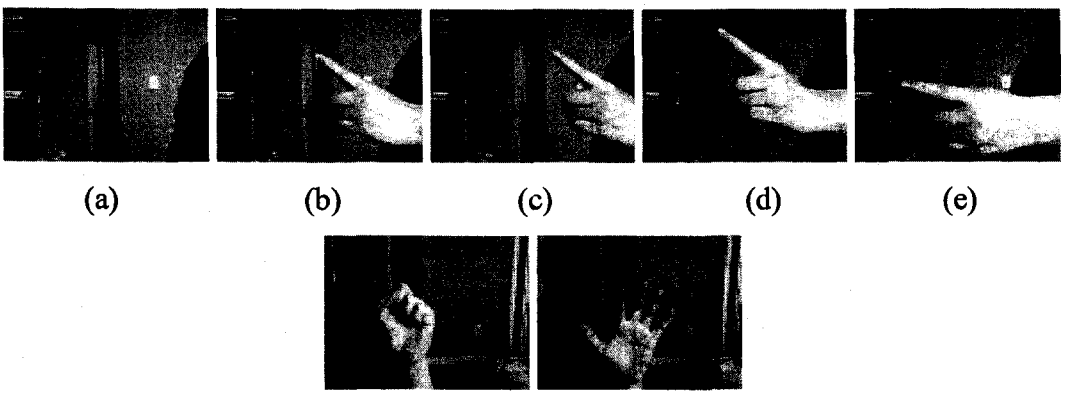


(13) (14) (15)

Figure 27: Images for gesture testing.

6.5.5 Hand Movement Tracking

The user's hand movement was recorded into two video files with "AVI" format. Some of key frames are shown in **Figure 28**.



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

(f) (g)

Figure 28: Hand movement images. (a) is the background image, (b) is the image where the user hand first appears, (c), (d) and (e) are images of hand movement. (f) is the "Grasp" gesture, (g) is the "Release" gesture.

In the tracking phrase, hand was used to virtually draw letter *'M'*. The trajectories of those hand movements are shown in **Figure 29**.

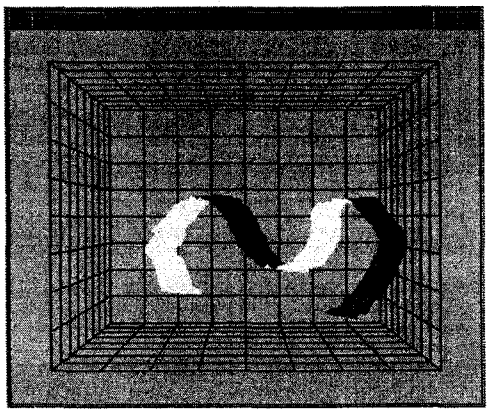
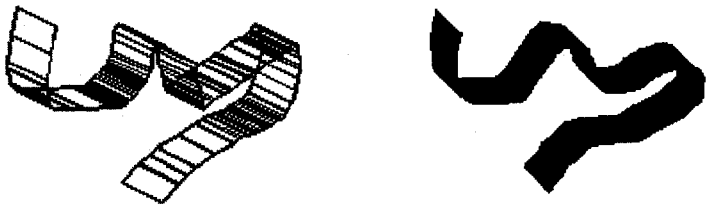


Figure 29: Hand trajectory of *'M"*-like movements.

6.5.6 Accuracy Evaluation of Tracking Algorithm

Our vision-based gesture interface is primarily designed for replacement of mechanic devices such as data gloves. In existing hand tracking applications, the most accurate results were obtained by using data gloves. However, it is nearly impossible to repeat exactly the same gestures twice, even by the same user. Thus it is not feasible to evaluate accuracy by directly comparing with results from data gloves.

Marker-based tracking system marks the user's hand with color and provides satisfying results. Similar to our bare-hand tracking, marker-based tracking is also based on continuous hand movements. The accuracy evaluation is therefore conducted by comparing results obtained from these two algorithms on the same hand movement. We reconstructed the trajectories for three different hand movements. **Figure 30** shows hand trajectories in test case 2. The red one (a) was obtained from marker-based algorithm and the black one (b) was the result of our visual tracking algorithm.



(a) (b)

Figure 30: Two trajectories from different tracking algorithms.

In each test case, we randomly chose ten time moments to extract the 3D coordinates of the points in both trajectories. Let P $(x,y,z)^T$ and Q $(a,b,c)^T$ represent the 3D coordinates of points in two trajectories respectively. We define the *Error Rate* formula as follows:

$$Error = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - a_i}{a_i} + \frac{y_i - b_i}{b_i} + \frac{z_i - c_i}{c_i}\right)$$

Table 5 shows the accuracy rates in three test cases.

Test Case No.	Number of Points	Accuracy Rate
Test Case 1	10	87.3%
Test Case 2	10	91.2%
Test Case 3	10	89.1%

Table 5: Accuracy Evaluation Results.

As having been reported in specification of commercial data gloves, the errors of glove-based tracker vary from 10% to 15%. In comparison, the results from our algorithm are in the same level of accuracy as data gloves (**Table 5**). Errors were accumulated from inaccurate calibration process and lower resolution of cameras. The evaluation process is based on the assumption that the true hand trajectories can be represented by the result of marker-based tracking algorithm. This assumption is reasonable because unique color can minimize these negative effects of various noises.

7. Conclusion and Future Work

Hand gesture recognition plays a pivotal role in the immersive virtual reality, the utmost natural interface. Vision-based hand gesture interface is one of most promising techniques in virtual reality. Using video cameras to capture hand images can provide adequate number of parameters for a dextrous interface and untie the user from mechanical devices. A great many researchers have contributed various approaches to vision-based hand gesture recognition and new algorithms in this area have been proposed more quickly than before. On the other side, however, it is not difficult to find a variety of constraints or restrictions are being imposed in current hand gesture applications. This shows that there is still a long way for natural HCI.

7.1Contributions of this thesis

The first issue encountered in gesture-based interface is the argument of gesture definition. The existing gesture definitions are either too broad and fuzzy or unable to unveil temporal and spatial characteristics of gesture. In this thesis, we first proposed a new gesture definition to facilitate mathematical modeling of hand and quantitative analysis of gesture. Then we developed a novel framework for visually tracking human hand without imposing constraints on the users. Those flexibilities are achieved by a hybrid appearance-based hand localization model. Neither constant lighting nor static background is required in our system. To preserve depth information, we also contributed an accurate and robust method for tracking hand in 3D space. To deal with the noisy hand problem, we add a skin tone filter which keeps the user hand focused through the images.

Additionally, we adopt a self-adaptive strategy to refine the skin classifier by an online training schema.

7.2 Future Research Directions

As many other technologies in HCI, vision based hand gesture interpretation undoubtedly is still in baby stage. The potential of using hand gesture recognition in HCI has not been fully exploited. Future theoretical researches aiming at natural interaction are needed before the hand gesture recognition can be used seamlessly. We propose future research directions as follows:

- **Two-hand gesture recognition**

  Two-hand gesture recognition will be a direction which not only improves the accuracy of recognition for gestures but also broadens the gesture vocabulary because two hands are naturally involved in gesticulation between human-human communications ([55], [87], [110]). However, some ambiguous situations may occur in two-hand recognition, such as frequent occlusion and distinction between left and right hands.

- **Multimodal interface**

  Recently, the relation of gestures with speech, body movement and gaze has been underlined in HCI studies ([13], [60]). The reason is that almost any natural communication among humans concurrently involves several modes of communication that accompany each other. For example, the sentence "look that" and a deictic gesture using index finger and an obvious movement of eyes may occur concurrently to express the same meaning. Then the recognition for speech, gesture and gaze can be mutually confirmed. In such multi-model interface, the complexity of analysis can be reduced and the performance can be improved ([67], [103], [111], [112]).

Bibliography

1. J.A. Adam. **Virtual Reality Is for Real.** *IEEE Spectrum, vol. 30, no. 10, pp. 22-29, 1993.*

2. S. Ahmad and V. Tresp. **Classification With Missing and Uncertain Inputs.** *Proc. Int'l Conf. Neural Networks, vol. 3, pp. 1,949-1,954, 1993.*

3. S. Ahmad. **A Usable Real-Time 3D Hand Tracker.** *IEEE Asilomar Conf., 1994.*

4. Y. Azoz, L. Devi and R. Sharma. **Reliable Tracking of Human Arm Dynamics by Multiple Cue Integration and Constraint Fusion.** *Proc. 1998 Computer Vision and Pattern Recognition. pp. 905-910.*

5. R. Bajcsy. **Active Perception.** *Proc. IEEE, vol. 78, pp. 996-1,005, 1988.*

6. N.I. Badler. **Real-Time Virtual Humans.** *Proc. 1997 Computer Graphics and Applications. pp: 4 –13.*

7. T. Baudel and M. Baudouin-Lafon. **Charade: Remote Control of Objects Using Free-Hand Gestures.** *Comm. ACM, vol. 36, no. 7, pp. 28-35, 1993.*

8. D.A. Becker and A. Pentland. **Using a Virtual Environment to Teach Cancer Patients T'ai Chi, Relaxation, and Self-Imagery.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, 1996.*

9. A.F. Bobick and J.W. Davis. **Real-Time Recognition of Activity Using Temporal Templates.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, 1996.*

10. Joseph Bray. **Markerless Based Human Motion Capture: A Survey.** *Vision and VR Group, Dept. of Systems Engineering, Brunel University. www.visicast.co.uk/members/move/Partners/ Papers/MarkerlessSurvey.pdf*

11. C. Bregler, S. Manke, H. Hild and A. Waibel. **Bimodal Sensor Integration on the Example of "Speech-Reading.** *Proc. of IEEE Int. Conf. on Neural Networks, San Francisco, 1993.*

12. U. Bröckl-Fox. **Real-Time 3D Interaction With Up to 16 Degrees of Freedom From Monocular Image Flows.** *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 172-178, June 1995.*

13. U. Bub, M. Hunke and A. Waibel. **KNOWING WHO TO LISTEN TO IN SPEECH RECOGNITION: VISUALLY GUIDED BEAMFORMING.** *Proc. IEEE Int. Conf. on Acoust., Speech, and Signal, 1995.*

14. T. S. Caetano, S. D. Olabarriaga, D. A. C. Barone. **Performance evaluation of single and multiple-Gaussian models for skin color modeling.** *Computer Graphics and Image Processing, 2002. Proceedings. XV Brazilian Symposium on, 7-10 Oct. 2002, Page(s): 275 –282.*

15. L.W. Campbell, D.A. Becker, A. Azarbayejani, A.F. Bobick, and A. Pentland. **Invariant Features for 3D Gesture Recognition.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, pp.157-162, 1996.*

16. Teofilo Emidio Campos. **3D Hand and Object Tracking for Intention Recognition.** *Dphil Transfer Report. http://www.robots.ox.ac.uk/~teo/TransferReport/deCampos_report.pdf.*

17. Darrin Cardani. **Adventures in HSV Space.** *Article in BUENA Software INC., July 2001.*

18. C. Cedras and M. Shah. **Motion-Based Recognition: A Survey** *Image and Vision Computing, vol. 11, pp. 129-155, 1995.*

19. D. Chai, S.L Phung, A. Bouzerdoum. **A bayesian skin/non-skin color classifier using non-parametric density estimation.** *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on, Volume: 2, May 25-28, 2003, Page(s): 464 -467*

20. K. Cho and S.M. Dunn. **Shape-based Object Recognition by Inductive Learning.** *Proc. of 11th IAPR International Conference, pp: 681 –684, 1992.*

21. R. Cipolla and N.J. Hollinghurst. **Human-Robot Interface by Pointing With Uncalibrated Stereo Vision.** *Image and Vision Computing, vol. 14, pp. 171-178, Mar. 1996.*

22. R. Cipolla, Y. Okamoto, and Y. Kuno. **Robust Structure From Motion Using Motion Parallax.** *Proc. IEEE Int'l Conf. Computer Vision, pp. 374-382, 1993.*

23. T. F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. **Active Shape Models—Their Training and Application.** *Computer Vision and Image Understanding, vol. 61, pp. 38-59, Jan. 1995.*

24. Y. Cui and J. Weng. **Hand Sign Recognition from Intensity Image Sequences with Complex Backgrounds.** *Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 88-93, 1996.*

25. Y. Cui and J. J. Weng. **Hand Segmentation Using Learning-Based Prediction and Verification for Hand Sign Recognition.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, pp. 88-93, 1996.*

26. T. Darrell, I. Essa and A. Pentland. **Task-Specific Gesture Analysis in Real-Time Using Interpolated Views.** *IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 12, pp. 1,236-1,242, 1996.*

27. T. Darrel, G. Gordon, M. Harville, J. Woodfill. **Integrated Person Tracking using Stereo, Color, and Pattern Detection.** *In Proceeding of Conference on Computer Vision and Pattern Recognition (CVPR'98), pp. 601-609, Santa Barbera, June 1998*

28. T. Darrell and A.P. Pentland. **Active Gesture Recognition Using Partially Observable Markov Decision Processes.** *Proceedings of 13th IEEE Intl. Conference on Pattern Recognition (ICPR), 1996.*

29. T. Darrell and A. Pentland. **Space-time gestures.** *Proc. of CVPR'93, pp. 335-340, 1993.*

30. J. Davis and M. Shah. **Determining 3D Hand Motion.** *Proc. 28th Asilomar Conf. Signals, Systems, and Computer, 1994.*

31. J. Davis and M. Shah. **Visual Gesture Recognition.** *IEE Proc. - Vis. Image and Signal Process., vol. 141, pp. 101--10, 1994.*

32. C. Downton and H. Drouet. **Model-based Image Analysis for Unconstrained Human Upper-body Motion.** *Proc. of IEEE International Conference on Image Processing and its Applications, pp. 274--277, 1992.*

33. Essa and S. Pentland. **Facial Expression Recognition Using a Dynamic Model and Motion Energy.** *Proc. IEEE Int'l Conf. Computer Vision*, 1995.

34. S.S. Fels and G.E. Hinton. **Glove-Talk: A Neural Network Interface Between a Data-Glove and a Speech Synthesizer.** *IEEE Trans. Neural Networks, vol. 4, pp. 2-8, Jan. 1993.*

35. J. Flachsbart, D. Franklin and K. Hammond. **Improving Human Computer Interaction in a Classroom Environment using Computer Vision.** *Proceedings of Intelligent User Interfaces. 2000. New Orleans, LA: ACM.*

36. W.T. Freeman, K. Tanaka, J. Ohta and K. Kyuma. **Computer Vision for Computer Games.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, pp. 100-105, Oct. 1996.*

37. W.T. Freeman and M. Roth. **Orientation Histograms for Hand Gesture Recognition.** *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, June 1995.*

38. W.T. Freeman, D. Anderson, P. Beardsley, C. Dodge, H. Kage, K. Kyuma, Y. Miyake, M. Roth, K. Tanaka, C. Weissman and W. Yerazunis. **Computer Vision for Interactive Computer Graphics.** *IEEE Computer Graphics and Applications, Vol. 18, Num 3, pages 42-53, May-June 1998.*

39. W.T. Freeman and C.D. Weissman. **Television Control by Hand Gestures.** *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 179-183, June 1995.*

40. D.M. Gavrila. **The Visual Analysis of Human Movement: A Survey.** *Computer Vision and Image Understanding, 73(1): 82-- 98, Jan. 1999.*

41. D.M. Gavrila and L.S. Davis. **Towards 3D Model-Based Tracking and Recognition of Human Movement: A Multi-View Approach.** *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 272-277, June 1995.*

42. H.P. Graf, E. Cosatto, D. Gibbon, M. Kocheisen and E. Petajan. **Multi-Modal System for Locating Heads and Faces.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, pp. 88-93, Oct. 1996.*

43. S. Grange, E. Casanova, T. Fong, and C. Baur. **Vision-Based sensor fusion for Human-Computer Interaction.** *IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, October, 2002.*

44. Djambazian Haig. **3D Tracking of Non-rigid Articulated Objects.** *Master Thesis.*

45. G. D. Hager. **Set-Based Estimation: Towards Task-Directed Sensing.** *Proc. of Electrotechnical Conf.,vol.2, pp. 1205-1208, 1991.*

46. C.V. Hardenberg and F. Berard. **Bare-Hand Human-Computer Interaction.** *Proc. of the ACM Workshop on Perceptive User Interfaces, Orlando, Florida, 2001.*

47. A.G. Hauptmann and P. McAvinney. **Gesture With Speech for Graphics Manipulation.** *Int'l J. Man-Machine Studies, vol. 38, pp. 231-249, 1993.*

48. T. Heap and D. Hogg. **Towards 3D Hand Tracking Using a Deformable Model.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, Vt., pp. 140-145, Oct. 1996.*

49. T. Heap and D. Hogg. **Wormholes in Shape Space: Tracking through Discontinuous Changes in Shape.** *6th International Conference on Computer Vision (1998), pp. 344-349. 17*

50. C. Hummels and P.J. Stappers. **Meaningful Gestures for Human Computer Interaction: Beyond Hand Postures.** *Proc. of the Third IEEE International Conference on Automatic Face & Gesture Recognition (FG'98), 1998.*

51. M. Isard and A. Blake **ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework.** *Proc. European Conf. Comput. Vision, vol. 1, 1998, pp. 767–781.*

52. K. Ishibuchi, H. Takemura, and F. Kishino. **Real Time Hand Gesture Recognition Using 3D Prediction Model.** *Proc. 1993 Int'l Conf. Systems, Man, and Cybernetics, Le Touquet, France, pp. 324-328, Oct. 17-20, 1993.*

53. Sankar Jayaram, Yong Wang, and Uma Jayarma. **A Virtual Assembly Design Environment.** *IEEE Virtual Reality Conference.*

54. C. Jennings. **Robust finger tracking with multiple cameras Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems.** *1999. Proceedings. International Workshop on, 26-27 Sept. 1999 Page(s): 152 –160.*

55. M.H. Jeong, Y. Kuno and N. Shimada. **Two-Hand Gesture Recognition using Coupled Switching Linear Model.** *Proc. of Pattern Recognition, 2002, vol.3, pp. 529 –532.*

56. J. Lin, Y. Wu and T.S. Huang. **Modeling the Constraints of Human Hand Motion.** *Proc. of 5th Annual Federated Laboratory Symposium, Maryland, March, 2001.*

57. S.X. Ju, M.J. Black and Y. Yacoob. **Cardboard People: A Parameterized Model of Articulated Image Motion.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, Vt., pp. 38-43, Oct. 1996.*

58. I.A. Kakadiaris, D. Metaxas, and R. Bajcsy. **Active Part-Decomposition, Shape and Motion Estimation of Articulated Objects: A Physics-Based Approach.** *Proc. IEEE C.S. Conf. Computer Vision and Pattern Recognition, pp. 980-984, 1994.*

59. S.B. Kang and K. Ikeuchi. **A Grasp Abstraction Hierarchy for Recognition of Grasping Tasks from Observation.** *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93), vol.1, pp. 621 – 628, 1993.*

60. M.N. Kaynak, Q. Zhi, A.D. Cheok, K. Sengupta and K.C. Chung. **Audio-Visual Modeling for Bimodal Speech Recognition.** *Systems, Man, and Cybernetics, 2001 IEEE International Conf., vol.1, pp. 181-186, 2001.*

61. C. Kervrann and F. Heitz. **Statistical Model-Based Segmentation of Deformable Motion.** *Proc. of the 3rd IEEE International Conf. on Image Processing (ICIP '96), Lausanne, Switzerland, pp. 937—940, 1996.*

62. Sang-Hoon Kim; Hyoung-Gon Kim. **Face detection using multi-modal information.** *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, 28-30 March 2000, Page(s): 14 –19.*

63. Ig-Jae Kim, Shwan Lee, Ahn, S.C, Yong-Moo Kwon, Hyoung-Gon Kim. **3D tracking of multi-objects using color and stereo for HCI.** *Image Processing, 2001. Proceedings. 2001 International Conference on , Volume: 3 , 7-10 Oct. 2001, Page(s): 278 -281 vol.3.*

64. R. Kjeldsen and J. Kender. **Toward the Use of Gesture in Traditional User Interfaces.** *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition, pp. 151--156, Killington, 1996.*

65. R. Kjeldsen and J. Kender. **Finding Skin in Color Images.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, pp. 312-317, 1996.*

66. R. Koch. **Dynamic 3D Scene Analysis Through Synthetic Feedback Control.** *IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 6, pp. 556-568, 1993.*

67. N. Krahnstoever, S. Kettebekov, M. Yeasin and R. Sharma. **A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays.** *Proc. of Fourth Intl. Conference on Multimodal Interfaces (ICMI 2002), USA.*

68. M.W. Krueger. **An Architecture for Artificial Realities.** *Thirty-Seventh IEEE Computer Society International Conf., Digest of Papers. , pp. 462 –465, 1992.*

69. M.W. Krueger. **Automating Virtual Reality.** *Computer Graphics and Applications, IEEE, vol.15 Issue: 1, pp. 9 –11, 1995.*

70. J.J. Kuch and T.S. Huang. **Human Computer Interaction via the Human Hand: A Hand Model.** *Signals, Systems and Computers, 1994. Conf. Record of the Twenty-Eighth Asilomar Conf., volum 2, pp. 1252 –1256, 1994.*

71. J.J. Kuch and T.S. Huang. **Vision-Based Hand Modeling and Tracking for Teleconferencing and Telecollaboration.** *Proc. of Fifth International Conference pp. 666 –671, 1995.*

72. Y. Kuno, M. Sakamoto, K. Sakata and Y. Shirai. **Vision-Based Human Computer Interface With User Centered Frame.** *Proc. IROS'94, 1994.*

73. T. Kurata, T. Okuma, M. Kourogi, K. Sakaue. **The Hand Mouse: GMM hand-color classification and mean shift tracking.** *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on, 13 July 2001 Page(s): 119 -124.*

74. Lae Kyoung Lee, Sungshin Kim, Young-Kiu Choi, Man Hyung Lee. **Recognition of hand gesture to human-computer interaction.** *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Conference of the IEEE, Volume: 3, 22-28 Oct. 2000, Page(s): 2117 -2122 vol.3.*

75. A. Lanitis, C.J. Taylor, T.F. Cootes, and T. Ahmed. **Automatic Interpretation of Human Faces and Hand Gestures Using Flexible Models.** *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 98-103, June 1995.*

76. F. Lathuiliere and J.Y. Herve. **Visual Tracking of Hand Posture With Occlusion Handling.** *Pattern Recognition, 2000. Proc. of 15th International Conf., vol.3, pp. 1129 –1133, 2000.*

77. Alison Lee, Kevin Schlueter, Andreas Girgensohn. **Sensing Activity in Video Images.** *In CHI 97 Extended Abstracts, ACM Press, 1997, pp. 319-320., March 22, 1997*

78. J. Lee and T.L. Kunii. **Model-Based Analysis of Hand Posture.** *IEEE Computer Graphics and Applications, pp. 77-86, Sept. 1995.*

79. C. Maggioni. **A Novel Gestural Input Device for Virtual Reality.** *1993 IEEE Annual Virtual Reality Int'l Symp., pp. 118-124, IEEE, 1993.*

80. Sebastien Marcel and Samy Bengio. **Improving Face Verification using Skin Color Information.** *Proceedings of the 16th International Conference on Pattern Recognition.*

81. J. Martin, V. Devin and J.L. Crowley. **Active Hand Tracking.** *IEEE Third International Conference on Automatic Face and Gesture Recognition, FG '98, Japan, 1998.*

82. J. Martin, D. Hall, and J. L. Crowley. **Statistical Gesture Recognition through Modelling of Parameter Trajectories.** *Third Gesture Workshop, France, March 17, 1999.*

83. T.B. Moeslund and E. Granum. **Multiple Cues used in Model-Based Human Motion Capture.** *In The fourth International Conference on Automatic Face and Gesture Recognition, Grenoble, France.*

84. T. Moeslund. **Computer vision-based human motion capture -- a survey.** *University of Aalborg Technical Report LIA 99-02, March 1999.*

85. B. Moghaddam and A. Pentland. **A Subspace Method for Maximum Likelihood Target Detection.** *IEEE International Conf. on Image Processing, Washington DC, 1995.*

86. Y. Moses, D. Reynard and A. Blake **Determining Facial Expressions in Real Time.** *Proc. of Fifth International Conf. on Computer Vision, Cambridge, MA, 1995.*

87. H. Nishino, K. Utsumiya, D. Kuraoka and K. Yoshioka. **Interactive Two-Handed Gesture Interface in 3D Virtual Environments.** *Proc. of the ACM symp. on Virtual reality software and technology, Switzerland, pp. 1 – 8, 1997.*

88. K. Nummiaro, E. Koller-Meier, L. Van Gool. **Object Tracking with an Adaptive Color-Based Particle Filter.** *Symposium for Pattern Recognition of the DAGM (2002) 353 -- 360.*

89. R. O'Hagan. **Finger Track- A Robust and Real-Time Gesture Interface.** *Australian Joint Conference on Artificial Intelligence, Perth, 1997.*

90. V. I. Pavlovic, R. Sharma and T.S. Huang. **Visual interpretation of hand gestures for human-computer interaction: a review.** *Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume: 19 Issue: 7, July 1997, Page(s): 677 – 695.*

91. V.I. Pavlovic, R. Sharma and T.S. Huang. **Gestural Interface to a Visual Computing Environment for Molecular Biologists.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, Vt., pp. 30-35, Oct. 1996.*

92. D.L. Quam. **Gesture Recognition With a DataGlove.** *Proc. 1990 IEEE National Aerospace and Electronics Conf., vol. 2, 1990.*

93. Huynh-Thu, Quan; M. Meguro, M. Kaneko. **Skin-color extraction in images with complex background and varying illumination.** *Applications of Computer Vision, 2002. (WACV 2002). Proceedings. Sixth IEEE Workshop on, 3-4 Dec. 2002. Page(s): 280 -285*

94. F.K.H. Quek. **Unencumbered Gestural Interaction.** *IEEE Multimedia. 36-47, Winter, (1996).*

95. F.K.H. Quek. **Eyes in the Interface.** *Image and Vision Computing, vol. 13, Aug. 1995.*

96. F. Quek, D. McNeilly, R. Bryll, C. Kirbas, H. Arslan, K.E. McCulloughy, N. Furuyamay and R. Ansari. **Gesture, Speech, and Gaze Cues for Discourse Segmentation.** *Computer Vision and Pattern Recognition (CVPR'00)-Volume 2.*

97. F.K.H. Quek, T. Mysliwiec and M. Zhao. **Finger Mouse: A Freehand Pointing Interface**. *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 372-377, June 1995.*

98. L.R. Rabiner. **A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition**. *Proc. IEEE, vol. 77, pp. 257-286, Feb. 1989.*

99. J.M. Rehg and T. Kanade. **DigitEyes: Vision-Based Human Hand Tracking.** *Technical Report CMU-CS-93-220, School of Computer Science, Carnegie Mellon Univ., 1993.*

100. J.M. Rehg and T. Kanade. **Model-Based Tracking of Self-Occluding Articulated Objects**. *Proc. IEEE Int'l Conf. Computer Vision, Cambridge, Mass., pp. 612-617, June 20-23 1995.*

101. J. Rittscher and A. Blake. **Classification of Human Body Motion.** *ICCV (1).*

102. P.L. Rosin and T. Ellis. **Image difference threshold strategies and shadow detection.** *In British Machine Vision Conf., pages 347-356, 1995.*

103. V. Sa, C. Malerczyk and M. Schnaider. **Vision-Based Interaction within a Multimodal Framework.** *10th Portuguese Computer Graphics Meeting, ISCTE, Lisbon, 2001.*

104. D. Saxe and P. Foulds. **Toward robust skin identification in video images.** *In Second International Conference on Automatic Face and Gesture Recognition, Killington, VT, 1996.*

105. J. Schlenzig, E. Hunter, and R. Jain. **Vision-Based Hand Gesture Interpretation Using Recursive Estimation.** *Proc. 28th Asilomar Conf. Signals, Systems, and Computer, 1994.*

106. J. Schlenzig, E. Hunter, and R. Jain. **Recursive Identification of Gesture Inputs Using Hidden Markov Models.** *Proc. Second IEEE Workshop on Applications of Computer Vision, Sarasota, Fla., pp. 187-194, Dec. 5-7, 1994.*

107. J. Segen and S. Kumar. **GestureVR: Vision-Based 3D Hand Interface for Spatial Interaction.** *Proc. of ACM Multimedia Conf., Briston, England, 1998.*

108. J. Segen and S. Kumar. **Fast and Accurate 3D Gesture Recognition Interface.** *Pattern Recognition, 1998. Proc. of Fourteenth International Conf., vol.1, pp. 86 – 91, 1998.*

109. M.S. Sellberg and M.J. Vanderploeg. **Virtual Human: A Computer Graphics Model for Biomechanical Simulations and Computer-Aided Instruction.** *Engineering in Medicine and Biology Society, 1994. Proc. of the 16th Annual International Conf. of the IEEE, pp. 329-330, vol.1, 1994.*

110. R. Sharma. **Two-Hand Gesture Recognition using Coupled Switching Linear Model.** *16th International Con. on Pattern Recognition (ICPR'02) Volume 1.*

111. R. Sharma, T.S. Huang, and V.I. Pavlovic. **Toward Multimodal Human–Computer Interface.** *Proc. of the IEEE, vol. 86, no. 5, May 1998.*

112. R. Sharma, T.S. Huang, V.I. Pavlovic, Y. Zhao, Z. Lo, S. Chu, K. Schulten, A. Dalke, J. Phillips, M. Zeller and W. Humphrey. **Speech/Gesture Interface to a Visual Computing Environment for Molecular Biologists.** *Proc. Int'l Conf. Pattern Recognition, 1996.*

113. R. Sharma, M. Zeller, V.I. Pavlovic, T.S. Huang, Z. Lo, S. Chu, Y. Zhao, J.C. Phillips and K. Schulten. **Speech/gesture interface to a visual-computing environment.** *Computer Graphics and Applications, IEEE, Vol.20, Issue: 2,pp. 29 –37, 2000.*

114. Y. Shiga, H. Ebine, M. Ikeda, O. Nakamura. **Human face extraction based on color and moving information and the recognition of expressions.** *Electrical and Computer Engineering, 2000 Canadian Conference on, Volume: 2, 7-10 March 2000, Page(s): 1100 -1108 vol.2*

115. N. Shimada, K. Kimura, Y.Shirai and Y. Kuno. **Hand Posture Estimation by Combining 2-D Appearance-based and 3-D Model-based Approaches.** *Pattern Recognition, 2000. Proc. of 15th International Conf., vol.3, pp. 705 –708, 2000.*

116. Lindsay I. Smith. **A tutorial on Principal Components Analysis.** http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

117. Jung Soh, Ho-Sub Yoon, Min Wang, Byung-Woo Min. **Locating hands in complex images using color analysis.** *Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'., 1997 IEEE International Conference on , Volume: 3 , 12-15 Oct. 1997, Page(s): 2142 -2146 vol.3*

118. T.E. Starner and A. Pentland. **Real-Time American Sign Language Recognition from Video Using Hidden Markov Models.** *Perceptual Computing Section Technical Report No. 375, MIT Media Lab, Cambridge, MA, 1996.*

119. B. Stenger, P.R.S. Mendonca, and R. Cipolla. **Model-Based 3D Tracking of an Articulated Hand.** *Computer Vision and Pattern Recognition (CVPR'01) - Volume 2 December 08 - 14, 2001 Kauai, Hawaii. p. 310*

120. D.J. Sturman and D. Zeltzer. **A Survey of Glove-Based Input.** *IEEE Computer Graphics and Applications, vol. 14, pp. 30-39, Jan. 1994.*

121. H. Sun, X. Yuan, G. Baciu and Y. Gu. Direct **Virtual-hand Interface in Robot Assembly Programming.** *IEEE TRANSACTION ON SYSTEMS, MAN, AND CYBERNETICS--PART C: APPLICANTIONS AND REVIEWS, Vol. 33, No. 2, 2003.*

122. D.L. Swets and J. Weng. **Using Discriminant Eigenfeatures for Image Retrieval.** *IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 8, pp. 831-836, 1996.*

123. D.O. Tanguay. **Hidden Markov Models for Gesture Recognition.** *Master Thesis in Department of Electrical Engineering and Computer Science in Massachusetts Institute of Technology.*

124. K.A. Tarabanis, P.K. Allen, and R.Y. Tsai. **A Survey of Sensor Planning in Computer Vision.** *IEEE Trans. Robotics and Automation, vol. 11, pp. 86-104, 1995.*

125. A. Torige and T. Kono. **Human-Interface by Recognition of Human Gestures With Image Processing. Recognition of Gesture to Specify Moving Directions.** *IEEE Int'l Workshop on Robot and Human Communication, pp. 105-110, 1992.*

126. S. Tsuruoka, A. Kinoshita, T. Wakabayashi, Y. Miyake, M. Ishida. **Extraction of Hand Region and Specification of finger Tips from Color Image.** *1997 International Conference on Virtual Systems and MultiMedia, September 10 - 12, 1997,Geneva, SWITZERLAND, p.206*

127. C. Uras and A. Verri. **On the Recognition of the Alphabet of the Sign Language through Size Functions.** *Proc. IAPR, pp. 334--338, Jerusalem, Israel, 1994.*

128. R. Vaillant, C. Monrocq and Y.L. Chun. **An Original Approach for the Localization Objects in Images.** *IEE Proc. on Vision, Image, and Signal Processing, vol. 141, no. 4, 1994.*

129. V. Vezhnevets, V. Sazonov, A. Andreeva. **A Survey on Pixel-Based Skin Color Detection Techniques.** *Proc. Graphicon-2003, pp. 85-92, Moscow, Russia, September 2003.*

130. C. Wang and D.J. Cannon. **A Virtual End-Effector Pointing System in Point-and-Direct Robotics for Inspection of Surface Flaws Using a Neural Network-Based Skeleton Transform.** *Proc. IEEE Int'l Conf. Robotics and Automation, vol. 3, pp. 784-789, May 1993.*

131. Watson, Richard. **A Survey of Gesture Recognition Techniques.** *Technical Report TCD-CS-93-11, Department of Computer Science, Trinity College Dublin, 1993.*

132. I. Weiss and M. Ray. **Model-Based Recognition of 3D Objects from Single Images.** *Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.23 Issue: 2, pp.116 –128, 2001.*

133. A.D. Wilson and A.F. Bobick. **Configuration States for the Representation and Recognition of Gesture.** *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, pp. 129-134, June 1995.*

134. A.D. Wilson and A.F. Bobick. **Recovering the Temporal Structure of Natural Gestures.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, Vt., pp. 66-71, Oct. 1996.*

135. C. Wren, A. Azarbayejani, T. Darrell and A. Pentland. **Pfinder: Real-Time Tracking of the Human Body.** *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, Vt., pp. 51-56, Oct. 1996.*

136. P. Woodland. **Speech Recognition.** *Speech and Language Engineering - State of the Art (Ref. No. 1998/499), IEE Colloquium on, 19 Nov. 1998.*

137. A. Wu, M. Shah, N. Da Vitoria Lobo. **A virtual 3D blackboard: 3D finger tracking using a single camera.** *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on , 28-30 March 2000 Page(s): 536 –543*

138. Ying Wu, T.S. Huang. **Non-stationary color tracking for vision-based human-computer interaction.** *Neural Networks, IEEE Transactions on, Volume: 13 Issue: 4th, July 2002, Page(s): 948 –960*

139. Y. Wu and T.S. Huang. **Robust visual tracking by co-inference learning.** *Proc. IEEE Int. Conf. Computer Vision, vol. II, Vancouver, July 2001, pp. 26–33.*

140. R.W.I. Yarger and F.K.H. Quek. **Surface Parameterization in Volumetric Images for Feature Classification.** *Bio-Informatics and Biomedical Engineering, 2000. Proc. of IEEE International Symposium, pp. 297 –303, 2000.*

141. S. Yonemoto and A. Matsumoto. **A Real-time Motion Capture System with Multiple Camera Fusion.** *Image Analysis and Processing, 1999. Proc. of International Conf., pp. 600 –605, 1999.*

142. Q. Yu and D. Terzopoulos. **Synthetic Motion Capture for Interactive Virtual Worlds.** *Proc. of Computer Animation, IEEE Computer Society Press, pp. 2-10, 1998.*

143. X. Yuan. **A Mechanism of Automatic 3D Object Modeling.** *IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol. 17, No. 3, 1995.*

144. X. Yuan and S. X. Yang. **Virtual Assembly With Biologically Inspired Intelligence.** *IEEE TRANSACTION ON SYSTEMS, MAN, AND CYBERNETICS--PART C: APPLICANTIONS AND REVIEWS, Vol. 33, No. 2, 2003.*

145. Xiaobu Yuan, Simon X. Yuang. **Intelligent Interface Design in Virtual Assembly.** *Proceedings of 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation July 2001, Canada.*

146. Zhu, X., Yang, J. and Waibel. **A. Segmenting Hands of Arbitrary Color.** *International Conference on Automatic Face and Gesture Recognition, Grenoble, 2000.*

147. B. D. Zarit, B. J. Super, and F. K. H. Quek. **Comparison of five color models in skin pixel classification.** *In Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, pages 58-63, Kerkyra, Greece, September 1999. IEEE.*

VITA AUCTORIS

NAME: Jiangnan Lu

PLACE OF BIRTH: Jilin, China

YEAR OF BIRTH: 1970

EDUCATION : Beijing Polytechnic University, Beijing China
1988 – 1993 B.Sc.

University of Windsor, Windsor, Ontario
2001 – 2004 M.Sc.