REAL-TIME IMMERSIVE HUMAN-COMPUTER INTERACTION BASED ON TRACKING AND RECOGNITION OF DYNAMIC HAND GESTURES

BY

Gan Lu

BEng (Hons)

Thesis submitted to the University of Central Lancashire
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

August 2011

The work presented in this thesis was carried out in the Applied Digital Signal and Image Processing (ADSIP) Research Centre, School of Computing, Engineering and Physical Sciences, University of Central Lancashire, Preston, England.

Declaration

I declare that while registered with the University of Central Lancashire for the degree of Doctor of Philosophy I have not been a registered candidate or enrolled student for another award of the University of Central Lancashire for any other academic of professional institution during the research programme. No portion of the work referred to in this thesis has been submitted in support of any application for another degree or qualification of any other University or Institution of learning.

Signed ________________________

ABSTRACT

With fast developing and ever growing use of computer based technologies, human-computer interaction (HCI) plays an increasingly pivotal role. In virtual reality (VR), HCI technologies provide not only a better understanding of three-dimensional shapes and spaces, but also sensory immersion and physical interaction. With the hand based HCI being a key HCI modality for object manipulation and gesture based communication, challenges are presented to provide users a natural, intuitive, effortless, precise, and real-time method for HCI based on dynamic hand gestures, due to the complexity of hand postures formed by multiple joints with high degrees-of-freedom, the speed of hand movements with highly variable trajectories and rapid direction changes, and the precision required for interaction between hands and objects in the virtual world.

Presented in this thesis is the design and development of a novel real-time HCI system based on a unique combination of a pair of data gloves based on fibre-optic curvature sensors to acquire finger joint angles, a hybrid tracking system based on inertia and ultrasound to capture hand position and orientation, and a stereoscopic display system to provide an immersive visual feedback. The potential and effectiveness of the proposed system is demonstrated through a number of applications, namely, hand gesture based virtual object manipulation and visualisation, hand gesture based direct sign writing, and hand gesture based finger spelling.

For virtual object manipulation and visualisation, the system is shown to allow a user to select, translate, rotate, scale, release and visualise virtual objects (presented using graphics and volume data) in three-dimensional space using natural hand gestures in real-time. For direct sign writing, the system is shown to be able to display immediately the corresponding SignWriting symbols signed by a user using three different signing sequences and a range of complex hand gestures, which consist of various combinations of hand postures (with each finger open, half-bent, closed, adduction and abduction), eight hand orientations in horizontal/vertical plans, three palm facing directions, and various hand movements (which can have eight directions in horizontal/vertical plans, and can be repetitive, straight/curve, clockwise/anti-clockwise). The development includes a special visual interface to give not only a stereoscopic view of hand gestures and movements, but also a structured visual feedback for each stage of the signing sequence. An excellent basis is therefore formed to develop a full HCI based on all human

gestures by integrating the proposed system with facial expression and body posture recognition methods. Furthermore, for finger spelling, the system is shown to be able to recognise five vowels signed by two hands using the British Sign Language in real-time.

Contents

Chapter 3 SYSTEM DEVELOPMENT

Chapter 4 VIRTUAL OBJECT MANIPULATION

Chapter 5 DIRECT SIGN WRITING

Chapter 6 *CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORKS*

References

ACKNOWLEDGEMENTS

It has been a long journey to the conclusion of my PhD study. Along the way many people graciously helped and cared for me, to them I am most thankful.

Firstly, it is a genuine pleasure to express my most sincere gratitude to my Director of Studies, Professor Lik-Kwan Shark, who supported me in so many ways. His professional supervision and knowledge guided me through this research and trained me. Through these years, not only has he imparted to me knowledge for this research, but also the logical and scientific mindset. His support has enabled me to complete my PhD study at the University of Central Lancashire.

Thanks also to my second supervisor, Dr. Geoff Hall and my third supervisor, Professor Ulrike Zeshan, for helping me through all stages of this research project. Their inspiration and technical support formed the basis of the research.

Others who deserve special thanks include: Dr. L. Xiao for her general guidance; Dr. X. Chen and Dr. C. Zhang for their technical guidance; Mr. W. Duan, Dr. D. Xu, Y. Li, Dr. J. Yang and Mr. Kent, for evaluating and commenting on the developed system; Colleagues Dr. D. George, Miss X. Guan, Mr. J. Chang, Dr. S. Volety, Dr. K. Nag and Dr. M. Sha, for the friendly and enjoyable atmosphere they created.

As an overseas student, the kindness and care from friends in the UK helped me to overcome any home sickness. Special thanks to the Good's family and Miss Chou, who helped to evaluate the developed system, as well as correcting my spoken English and treating me as a family member. Their loving care encouraged me to hold firmly to my goal. Also thanks to the families of Dr. Z. Wu, the Towers, Dr. P. Marston and the Meiers, for taking care of me; the Christian brothers and sisters in church who prayed for me.

With my great gratitude, I should like to thank my family and my friends in China, who spurred me to strain towards the goal. Without their unfailing love and support I would have proved much more difficult to get through my research.

Finally, I acknowledge the wonderful Divine providence which has been sustaining, guiding and empowering me to accomplish the task. This work is dedicated to Him. May His name be glorified now and forever!

GLOSSARY OF ABBREVIATIONS

Abbreviation	Definition
2D	Two-Dimension
3D	Three-Dimension
API	Application Programming Interface
APF	Annealed Particle Filter
ASL	American Sign Language
BSL	British Sign Language
CONDENSATION	Conditional Density Propagation
CPU	Central Processing Unit
CT	Computerised Tomography
DAS	Directory Assistance Service
DC	Direct Current
DHM	Dexterous HandMaster
DTW	Dynamic Time Warping
DOF	Degrees-of-Freedom
DSW	Direct Sign Writing
DVI	Digital Visual Interface
EEG	Electroencephalograph
EMG	Electromyography
FSD	Fourier Shape Descriptor
GPU	Graphic Processing Unit
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
IMU	Inertial Micro Unit
ISWA	International SignWriting Alphabet
MCMC	Markov Chain Monte Carlo
MEMS	Micro-Electro-Mechanical Systems
MFC	Microsoft Foundation Class
MMHCI	Multimodal Human-Computer Interaction
PCA	Principal Component Analysis
PF	Pagefile
RAM	Random Access Memory

SVM	Support Vector Machines
SW	SignWriting
UK	United Kingdom
UKF	Unscented Kalman Filter
VR	Virtual Reality
URU	Ultrasonic Receiving Unit
U.S.A	United States of America

Chapter 1

INTRODUCTION

1.1 INTRODUCTION TO THE RESEARCH

Since the concept of human-computer interaction (HCI) emerged in the later 50’s, the ever growing development in the computer world enables HCI to play a crucial role in human’s daily life [1]. Nowadays, HCI is a discipline of designing, evaluating, and implementing interactive computer systems for human use and includes the study of major phenomena surrounding them [2]. Its applications are wide and cover the areas of computer graphics, software engineering, human factors, psychology, etc [1-4]. In addition, as described in [5, 6], for virtual reality (VR) or virtual immersive environment, HCI technologies provide a better understanding of three-dimensional (3D) shapes and spaces, and have enabled VR to be practically used in areas such as industrial design, data visualisation, training, and others. However, due to the high computational cost associated with 3D, the complexity of human and object dynamic behaviour in 4D, as well as the difficulty for precise interaction in the virtual world [6], challenges are presented to provide users a real-time, natural, intuitive, effortless, and precise method for HCI.

Analysis of current hand based HCI techniques, particularly associated with VR, suggests that there are two dominant methods which can be classified as indirect and direct methods [7-10]. The former includes the use of keyboard based control, mouse based 3D widgets, and hand-held input devices like wireless 3D wands, to simulate the movement of the hand to control the virtual environment, which makes it awkward for HCI and has high cognitive load, because they are not natural and intuitive. In contrast, the latter is more intuitive and effective, which is based on the use of natural hand gestures, despite it faces challenges in terms of complexity of the hand configuration with 27 degrees-of-freedom (DOF) for just one hand [11], as well as the computational cost for hand movement tracking, and therefore is adopted in this research.

This research is concerned with the development of a real-time immersive HCI system based on dynamic hand gestures. Started with the literature review of currently available, especially the hand based, HCI techniques, presented in this thesis is the development of a novel HCI system based on tracking and recognition of dynamic hand gestures. Based on the constructed system platform, performance evaluation demonstrated the system is able to track and recognise the dynamic hand gestures in real-time as well as to enable a user to visualise them in the stereoscopic mode. To evaluate the system's usability performance, a graphic based virtual cube and a real CT (Computerised Tomography) medical image based volume object, have been created for immersive virtual object manipulation, and the system showed that the user can interact with them based on natural hand gestures directly. In terms of speed, the system can operate at 54 frames per second in the worst case with a latency time of approximately 94 ms by using a PC with 3 GHz CPU (Central Processing Unit). Particularly, based on the built platform, algorithms has been developed in terms of dynamic hand gesture and movement recognition, enables it being a unique system to perform direct sign writing (DSW) and finger spelling. System performance has been evaluated both systematically and holistically, where results showed that the system is able to produce SignWriting (SW) symbols based on various dynamic hand gestures and movements, and to recognise five vowel-letters for the British fingerspelling signs. The work has led to two conference papers published and one journal paper accepted (see Appendix A).

1.2 OBJECTIVES OF THE RESEARCH

The main aim of this research project is to develop a real-time immersive HCI system based on tracking and recognition of dynamic hand gestures. The specific objectives of this research are:

1) *To design and implement a HCI system to enable real-time tracking and recognition of dynamic hand gestures and movements in 3D.*

2) *To design and construct an immersive and interactive VR environment for virtual object visualisation and manipulation based on hand gestures and positions.*

3) *To design and develop a DSW system for sign input based on dynamic and complex hand gestures made using various combinations of hand postures, orientations and movements.*

1.3 THESIS ORGANISATION

The previous sections of this chapter gave a brief outline of the challenges in the direct hand based HCI, and the objectives of the research were stated.

Chapter 2 presents the literature reviews of the hand based HCI techniques. It starts with a review of various HCI techniques, and focuses on the hand modality for HCI. Furthermore, through a survey of the techniques for tracking and recognition of hand gestures and movements, the system prototype is proposed.

Chapter 3 presents the development of the system's hardware and software. Starting with the investigation of each sub-system, it introduces the hardware integration of the sub-systems as well as the software development for the integrated system. Performance evaluation of the developed system is also presented.

Chapter 4 presents the development and implementation of virtual object manipulation in stereoscopic 3D space based on the system described in Chapter 3 as well as its evaluation results including the system's usability and performance.

Chapter 5 contains the development of a novel DSW system. The DSW concept is introduced and the implementation details are presented. A number of tests are presented to demonstrate the functionality and capability of the system developed.

The final chapter highlights the original contributions of this research, and makes recommendations for future works in this field.

Chapter 2

LITERATURE REVIEW AND PROPOSED APPROACH

2.1 INTRODUCTION

Presented in this chapter are the literature reviews of hand based HCI techniques. It starts with an overview of various HCI techniques in Section 2.2. With the hand modality for HCI as the research focus, the hand anatomy is introduced in Section 2.3. This is followed by a survey of the techniques for capture and recognition of hand postures in Sections 2.4 and 2.5, as well as tracking and recognition of hand movements in Sections 2.6 and 2.7, respectively. Based on the literature reviews, an approach for this research is proposed in Section 2.8 with concluding remarks.

2.2 HUMAN-COMPUTER INTERACTION TECHNIQUES

With the ever increasing use of computerised machines in society, HCI has attracted tremendous interests in recent scientific studies, for it is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use, as well as their influence to each other [12-14]. Traditional HCI applications utilise indirect function-like metaphors such as keyboard based control, mouse based 3D widgets, or hand-held input devices like 3D wands. Compared to the normal human communication manner, these metaphors are non-ergonomic because they are not natural and intuitive. As the human forms the core of any HCI system, and thanks to the recent advances of technology, researches have attempted to capture and process possible physiological signals from every human body part directly to enable natural, intuitive, and immersive interaction for various applications. This section aims to give a brief review of current available techniques for direct HCI.

Early HCI techniques mainly use sight, sound and hand [13-15]. The use of sight for HCI includes gaze detection and tracking, with the gaze defined as the direction to which the eyes are pointing in space. One of the first eye based interaction attempts is a gaze based information display made in 1990, which replaces the mouse by tracking the gaze direction as well as displaying its selected information [16]. Although this method of driving computers was unable to register button clicks, the dwell time was proposed as a possible solution, and it aroused a new dimension of HCI for handicapped users. In the same year, Starker and Bolt applied gaze-tracking to identify which item on the screen attracts the user most based on the eye movements and its fixation results in the identified item to be zoomed-in for a closer look [17]. Another gaze based HCI is eye-typing, which is normally implemented by placing a virtual keyboard on the screen, and by tracking the user's gaze direction and behaviour to input the letter that the user is focusing on. The eye-typing technique could be used by the handicapped users, and the advances about it can be found in [18, 19]. Also, Tanriverdi and Jacob used gaze as a means of virtual object interaction [18, 19]. Compared to the hand based object interaction in virtual environments, eye based interaction has been claimed to be significantly faster but has greater difficulty to recall spatial information of the virtual objects with which it is interacting. More gaze based HCIs can also been found in [20-22]. Overall speaking, although gaze interaction has some advantages, such as faster pointing, it has difficulties in terms of low gaze measurement accuracy and drifting problem associated with the eye trackers due to fast eye movement, where the measured point of regard gradually falls off from the actual point of gaze [23].

Since sound or audio is one of the most natural modality methods for communication among human beings, it also plays a key role in the development of a natural interface to enhance human-machine communication. Its applications in HCI can mainly refer to speaker and speech recognition [18, 19, 24, 25]. The former is a process of automatically recognising who is speaking on the basis of individual utterances [26], thereby enabling the HCI system to verify their identities to provide control access [26-32]. The latter is a process of converting or interpreting a spoken speech into texts or symbols which the machine can understand, and has wide applications in HCI to make interaction more direct and intuitive [33, 34]. Example applications include virtual environment and object manipulation [35, 36], voice-activated remote machine operation [37], speech-driven menu for Directory Assistance Service (DAS) [34], and device control for handicapped persons [38]. Generally speaking, speech modality presents a natural and intuitive means for direct HCI. However, because of the complex phoneme articulation, large vocabulary, speaker's timbre and manner variability, context dependence, background noise, etc., speech recognition faces great challenges and attracts research of various processing methods such as, Dynamic time warping (DTW), hidden Markov model (HMM) networks, and neural networks. A more detailed review can be found in [39, 40].

Due to naturalness, intuitiveness and effectiveness, hand gestures form one of the most common and significant modes in object manipulation and human communication [41, 42]. As described in [43, 44], hand gestures provide a vital means for interacting and manipulating daily objects as well as an expressive means for interactions among people. In the context of HCI, recognition of hand gestures has wide applications for design and manufacturing, information visualisation, robotics, art and entertainment, sign language understanding, medicine, and health care. Tremendous research has been conducted in the area of hand gestures for HCI during the past few decades, and further discussions of various developed techniques will be given in the next section.

Apart from these three HCI techniques, which are based on sight, sound and hand, other new modalities have also been explored recently to enrich HCI. Examples include facial expressions [45], body movements [21], electromyography (EMG) [46], electroencephalograph (EEG) [47], and even human emotion [48]. In particular, EEG is a new method for HCI which interfaces directly with the human brain. Through the sensor monitoring the brain physical processes that correspond to certain forms of thought, users explicitly manipulate their brain activity to produce signals for computer or device control. This may provide a fantastic dimension of HCI in the future. Furthermore, a new stream has been formed recently which combines multiple modalities together to empower more advanced HCI that is called multimodal HCI (MMHCI) [21]. In MMHCI, modalities are interconnected and interdependent with each other, which enable a robust and efficient interaction and it is likely to become a widespread research topic.

Although various HCI techniques are available, the hand based modality provides a more natural, intuitive and immersive interaction in several hand-orientated HCI applications such as virtual object manipulation and direct sign language recognition, which are the focuses of this research. Moreover, by maturing the hand based HCI, it should provide a better basis for it to be included as a main modality in a large-scale MMHCI system in the future.

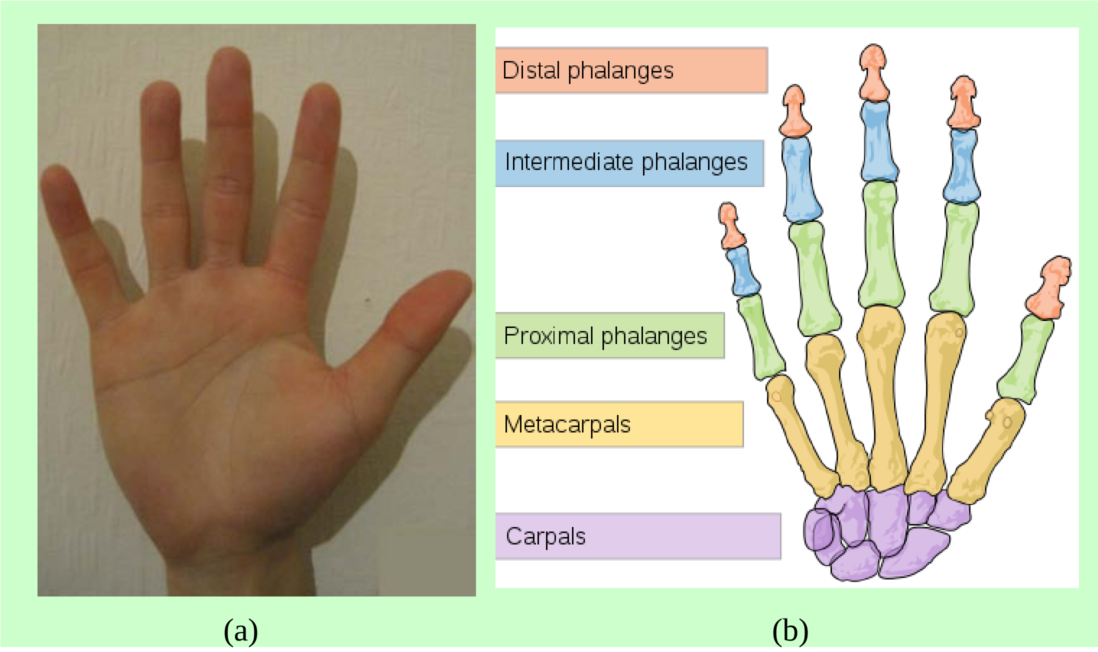
2.3 HAND ANATOMY

Due to the high DOF of the hand, tracking and recognition of hand gestures present a challenging task. As shown in Fig. 2-1a, the human hand consists of a broad palm (metacarpus) with 5 digits, attached to the forearm by the wrist joint (carpus) [49, 50]. The folding of the four fingers over the palm enables the grasping of objects. Starting from the one closest to the thumb, each finger has a colloquial name of:

- Index finger, pointer finger, or forefinger;
- Middle finger;
- Ring finger; and
- Little finger, small finger or pinky finger.

Furthermore, the human hand has 27 bones, including eight bones for the carpus or wrist, five for the metacarpals or palm, and fourteen for the digital bones that are the fingers and thumb, as shown in Fig. 2-1b. In particular, the fourteen digital bones, also called phalanges, are distributed as two in the thumb (the thumb has no middle phalanx) and three in each of the four fingers. Start from the position of the nail, these are the distal phalanx, the middle phalanx, and the proximal phalanx. Moreover, the four proximal phalanges of the fingers, proximal phalanx and metacarpal bone of the thumb and can perform bending (pitch) and adduction/abduction (yaw), but other phalanges can only perform bending. According to the hand anatomy, Rehg and Kanade [51] proposed a hand kinematic structure which contains 27 DOF for just one hand, as shown in Fig. 2-2, where each finger has 4 DOF, the thumb has 5 DOF, and the wrist has 6 DOF for its rigid global hand motion [11, 51, 52].

Therefore, due to the high DOF of the hand, challenges are presented for the hand based HCI to track and recognise dynamic hand gestures, which include hand postures and their movements. A general review of the available techniques for capture and recognition of hand postures as well as tracking and recognition of hand movements are conducted in the following sections.



(a) (b)

Figure 2-1. (a) hand and (b) its anatomy (from [53]).

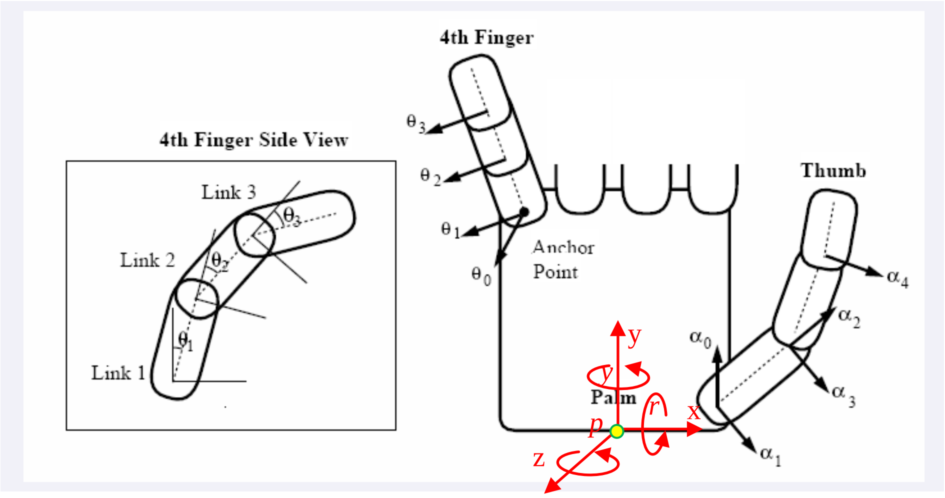


Figure 2-2. Kinematic hand model with arrows illustrating the joint axes for each joint hinge in the chain (modified from [11, 51, 52]).

2.4 HAND POSTURE CAPTURE

According to the literature review, the capture of hand posture data can be done by vision based and glove based approaches.

2.4.1 Vision Based Hand Posture Capture

Vision based posture capture is based on the use of video cameras to acquire hand images and processing of hand images to extract hand configuration data. Two approaches are presented in the following based on the number of video frames used to derive hand posture. The first one is vision based hand configuration extraction which uses the information contained in individual video frame only, and the second one is vision based hand posture tracking which uses the coherent information contained in neighbouring video frames.

A. Vision based hand configuration extraction

One of the most commonly used hand configuration data derived from a video frame containing hands is their silhouettes. However, there are some challenges to yield the hand silhouette from hand images [54]. Like other vision based methods, it has high computational cost due to a large number of image pixels to be processed, and the problem of self occlusion due to restricted camera views and complex hand gestures made by two hands with high DOF. A more specific problem in this method is hand image segmentation based on skin colour detection to extract the hand silhouette, because it requires no other skin-like objects and background, and the shape accuracy is susceptible to the lighting conditions [41, 55, 56].

Some algorithms have been attempted to address the difficulty associated with hand image segmentation, for example, in [57], Bretzner et al. developed an approach to combine silhouette and colour cues in a hierarchical object model for image segmentation, and used multiple scales of colour image features and particle filtering for recognition.

Although performing feature detection in colour space improves the robustness of the system when it operates in a poor contrast image, this approach was only demonstrated successfully with no other skin coloured objects present in the scene. In [58], it segments the hand silhouette based on the knowledge of its position in the image. However, the determination of the hand position is not necessary easy in many situations. Attempts have also been made to combine other available features to achieve a better hand segmentation. In [59], Coogan et al. proposed a unified system for segmentation and tracking of the face and hands, where skin detection is accomplished by combining 3 features: colour, motion and position. No assessment of the tracking performance has been done according to this paper, but an obvious challenge associated with this method is the increased computational cost.

To reduce the computational cost, other hand configuration data have been produced from hand images. One is the hand contour, which can be extracted easily from the silhouette of the segmented hand region. It simplifies the analysis of images by drastically reducing the amount of data to be processed and provides the useful structural information of hand boundaries [60]. In the sign language recognition system described in [61], Munib et al. proposed a contour detection method based on the ‘Canny edge detection’ technique [60], which uses an adaptive threshold with hysteresis to eliminate the streaking of edge contours and to handle images with different signal-to-noise ratios. From the results, this technique is shown to generate good edge detection performance for hand configuration extraction and the system is able to recognise American Sign Language (ASL) hand gestures with an accuracy around 98.5% for the training data and 80% for the testing data. However, this system still faces difficulty in dealing with a non-uniformed background, and is not able to operate in real-time. To improve it, Eigenvector has also been applied in contour extraction, which is a small set of basis image feature vectors extracted from the high-dimensional (or high-ordered) contour points which are attained from the raw image, with the redundant feature vectors being discarded. One possibility is to adopt the PCA (principal component analysis) algorithm [54]. By seeking an orthogonal basis across a low-dimensional subspace, PCA yields the principal components, which contain most of the variances of the image. In reverse, the original image can be restored through a linear combination of the principal components or basis vectors, where the vector coefficients are the result of projecting the image onto the respective basis vectors. Furthermore, the Fourier shape descriptor (FSD) has also been applied to represent the obtained contour, which is a closed boundary, via the frequency domain [62, 63]. By matching the features, which are the derived Fourier transform coefficients, with the stored feature templates, the hand shape can be identified. Although the FSD approach dramatically reduces the computational cost, the large gesture vocabulary (e.g. ASL) poses a problem for collecting an adequate training set to build the templates and may lose compactness in the subspace required for efficient processing [64]. Moreover, the attained templates may vary due to the different camera view angles from which they are taken.

B. Vision based hand posture tracking

Vision based hand posture tracking uses the relationship of hand configurations in neighbouring video frames to overcome unreliable hand segmentation and occlusion, and it can be categorised into two methods, the appearance based and hand model based methods [65].

Appearance based tracking methods

The appearance based tracking methods directly use hand configurations extracted from 2D video images, to find the best matching regions of hand from consecutive frames and to establish coherent relations between frames. Compared to the hand model based methods, it has advantages in terms of algorithm complexity and computational cost [66]. However, due to the camera view angle and high degrees of freedom of the hand, the obtained hand images and hand configurations may be partially occluded. As a result, the hand configuration on the current video frame may be significantly different from the previous one or even incomplete causing a failure in the hand matching process. To address the occlusion problem, although different techniques have been proposed, such as region-based, active-contour based, feature-based and regression-based methods [67-70], none of them yields a satisfactory solution.

Hand model based tracking methods

The severe occlusion problem associated with the appearance based tracking methods could be reduced to some extent by adopting the hand model based methods. These methods rely on the use of a 3D kinematic hand model with sufficiently high degrees of freedom, where the modelling of the hand is through finding the best match between the hand posture extracted from the image and the 2D posture appearances projected on to the image plane by the candidate 3D hand models. Compared to the appearance based method, with the constraints of hand model configuration, it has better pose-estimation performance for the occluded hand posture.

The earliest model based hand tracking system is the DigitEyes system proposed by Rehg and Kanade in 1994 [50]. The algorithm consists of two steps: feature extraction and state estimation. Upon obtaining the model parameters from the previous video frame, the parameters are then tested against the current video frame, and are updated to decrease the mis-correspondence. Subsequently, the updated parameters are used as the initial parameters for the next video frame, and the whole procedure is repeated thereafter. The system was shown to be able to recover the state of a 27-DOF hand model from ordinary gray-scale video at a speed of 10 Hz. However, this system has several disadvantages that prevent it from practical use. Firstly, a rapid hand posture change results in loss of tracking. Secondly, the feature extraction and state estimation process is sensitive to noise. Finally, although self-occlusion is reduced compared to the appearance based method, it still exists.

Another 3D hand model tracking method based on a deformable point distribution was proposed by Heap and Hogg in 1996 and is defined by a surface mesh which is built based on the statistical analysis of modelled hand examples [71]. The result shows that it provides a compact and accurate model for the range of legal hand shapes, together with real-time tracking performance. However, the inherent occlusion problem is still unresolved.

In 2001, Stenger et al. introduced a hand model built using 27-DOF truncated quadrics for tracking (see Fig. 2-3), which is used to yield a 2D hand contour for comparison with the hand image data [72]. The comparison employs an Unscented Kalman Filter (UKF) to estimate and update the current motion and configuration parameters of the hand model. This minimises the geometric error between the model profiles and hand image contours. By using a single camera, the result shows that the system is able to track 7-DOF of the hand motion, namely, the global hand motion (6-DOF) and thumb bending (1-DOF), with all other joints being kept fixed. The drawback is the computational complexity which grows linearly with the number of cameras, and is not suitable for real-time application.

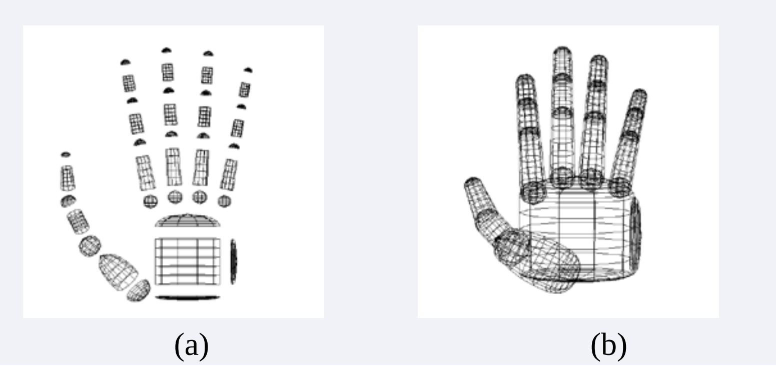


Figure 2-3. 27-DOF hand model constructed from 39 truncated quadrics: (a) front view and (b) exploded view (from [72]).

In 2006, Stenger et al. proposed another model based hand tracker that combines hierarchical detection and Bayesian filtering [73]. The essence of the tracker is a tree-based filter, which approximates the optimal Bayesian filtering equations. This filter presents a tree-based posterior distribution, where the nodes of the tree correspond to templates generated from a 3D model and the leaves define a partition of the state space with piecewise constant density. The advantage of this distribution is that the regions with low probability template candidates are rapidly discarded in a hierarchical search, which accelerates the processing speed, and the distribution can be approximated to arbitrary precision. However, it has a problem that certain independent assumptions must be made prior to the probabilistic distribution, which makes it impractical.

To summarise, hand model based tracking methods offer a way for complete modelling of all hand postures which reduced the occlusion problem of the appearance based tracking to some degree [74]. However, it is subject to several problems including:

- Susceptible to hand variation and deformation with model template re-calibration generally required for individuals;
- High complexity for the hand model building process;
- High computational cost due to a large hand model database to cover all hand shapes under different views; and
- Inherent self-occlusion.

2.4.2 Glove Based Hand Posture Capture

From the previous section, it can be seen that the approach of vision based hand posture capture has some inherent difficulties. These difficulties can be overcome by wearing a pair of data gloves with sensors to capture finger and joint information at the expense of introducing a small inconvenience to the user [75, 76]. Furthermore, since the data gloves usually provide sufficient hand degrees of freedom data and high sampling rate, it has no occlusion problem and hand posture tracking can be done directly without posture estimation. Various gloves have been explored over the last few decades. An introduction to these gloves is given in the following roughly based on a chronological order.

The first data glove, Sayre glove, was introduced by Defanti and Sandin in 1977, at the University of Illinois, Chicago [77]. This inexpensive and light-weight glove monitors hand motions via the flexible tubes (not fibre optics) with a light source at one end and a photocell at the other. The amount of light passing between its source and the photocell decreases when the tube is bent by finger bending. By correlating the voltage from the photocell with finger bending, the hand finger movements are monitored.

In the early 1980s, researchers at the MIT Media Lab developed a LED-studded glove for a body tracking system [78]. A camera is used to track the movements of the LEDs to capture the movements of the hand fingers.

In 1983, a glove designed by Grimes from Bell Telephone Laboratories was developed for translating discrete hand positions into electrical signals representing alpha-numeric characters [79]. Sensors were sewn on the glove to detect the bending of finger joints, contacts between various hand portions, movements of the hand with respect to the floor, as well as twisting and bending of the wrist. This glove is capable of recognising 80 unique combinations of sensor readings to produce a subset of the 96 printable ASCII characters. Although the initial works demonstrate its potential, this glove was never commercialised.

Another optical sensor based data glove was inspired by the invention of the optical flex sensor, which is a patent filed by Zimmerman in 1982 [80]. This optical sensor consists of a flexible tube with a reflective interior wall, a light source placed within one end of the flexible tube, and a photosensitive detector placed within the other end. The sensor detects a combination of direct light rays and reflected rays when the flexible tube is bent. Based on such sensor, in 1987, Zimmerman and his colleagues developed a glove that monitored 10 finger joints, with the optical fibres ran along the backs of each finger, and mounted on the user's hand by a lightweight Lycra glove [81]. As the finger flexion bends the fibre, it results in attenuation of the light received by the sensor, and the joint angles are determined by a processor analysing the signal strength received from the fibre, based on the calibration for each user. A 3D magnetic tracker with 6-DOF is also attached to the back of the hand to track the position and orientation of the palm. This data glove was commercialised by VPL Research at a reasonable cost, which is called the VPL data glove. Despite its light-weight, real-time operation, and does not rely on line-of-sight observation, it suffers from low accuracy and insufficient capturing speed [82].

Another commercialised glove developed in the 1980s is the Dexterous HandMaster (DHM) [76]. It is an exoskeleton-like device worn on the hand and fingers using Hall-effect sensors as potentiometers at the joints, by which the bending of the three joints of each finger and thumb as well as abductions are accurately measured. Its highly accurate sensors make it an excellent tool for fine work or clinical analysis, even though it is cumbersome to put on and take off. The drawback is it is unstable when the hand is shaken or moves rapidly.

Inspired by the VPL data glove, the Mattel toy company manufactured in 1989 a well-know glove peripheral with lower cost for the Nintendo video games, the Power Glove [76]. Resistive-ink flex sensors, embedded in the plastic on the back of the fingers, register overall bending of the thumb and fingers, with two bits of precision per finger. Mounted on the back of the hand are the acoustic trackers that locate the glove in space with respect to a companion unit mounted on the television monitor. However, this data glove is not accurate enough due to its limitation of the A/D converters used to convert the finger bending to digit signals Glove, and is no longer available on the market [76].

Kramer and Leifer developed the CyberGlove at Stanford University as part of their work to translate ASL into spoken English in 1989 [83]. This was then manufactured by the Immersion Corporation. It consists of a custom-made cloth glove with up to 22 thin foil strain gauges sewn into the fabric to sense finger and wrist bending. The glove performance is smooth and stable, having resolutions within a single degree of bending [76]. Moreover, it is comfortable and has an acceptable accuracy for complex gestural work or object manipulations. In addition to the CyberGlove, the Immersion Corporation also developed three other data glove products, which are considered as haptic data gloves [84]. They are the Cyber Touch, which provides vibration to each finger to simulate the touch sensation when it touches the virtual object; the CyberGrasp, which has a force feedback system enabling it to feel the size and shape of computer-generated 3D objects in a simulated virtual world; and the CyberForce device, which is a force feedback armature that not only conveys realistic ground forces to the hand, but also provides 6-DOF position tracking with translations and rotations of the hand in 3D space. Along with the CyberGrasp system, the CyberForce system can realistically measure the user’s hand motion.

In 2002, the Essential Reality Company developed the P5, a low-cost data glove [85]. It works by means of infrared sensors, where the receiver attached to the computer receives the signals from the hand and measures the hand movements in 3D space. The glove is also able to interpret the bending of the finger by the optical sensors running along the back of the fingers. However, due to the lack of compatible software, the development has since then been discontinued.

Recently, selections of data gloves with a broad range of features are available on the market. The pinch glove, developed by Fakespace, is based on the contact made by fingers, using conductive patches in the glove [86]. With the requirement of fingers touching each other to make an electrical contact, the recognisable gestures are not necessarily natural, and the number of identifiable gestures is limited. Other data gloves include the 5DT data glove and the DG5-Vhand, with the former having optical sensors attached along the fingers to generate finger-bend data, and the latter incorporating a three axes accelerometer to enable it to measure the hand acceleration in 3D. All of these data gloves suffer from the low accuracy problem [87, 88].

The most sophisticated and accurate data gloves at the present time are the CyberGlove II developed by Immersion, and the ShapeHand by Measurand [89-91]. They are wireless and capable of measuring finger bending and adduction. Moreover, they can be easily integrated with the full-body motion capture system. The CyberGlove II provides 22 high-accuracy joint-angle measurements by using a proprietary resistive bend-sensing technology to transform hand and finger motions into real-time digital joint-angle data. In contrast, the ShapeHand data glove is based on flexible tapes embedded with 40 fibre-optic curvature sensors arranged to sense bend and twist along the length of each tape. By attaching the tapes to run along each finger with one end at the finger tip and the other end fed to a small data acquisition box at wrist, the gesture movements of fingers introduce deformation of the tapes, the bend and twist measured at each sensor location with respect to the end of the tape at wrist enables relative positions and orientations of each finger joint to be determined. Furthermore, its ambidextrous and independent glove design enables the ShapeHand data glove to suit different hand sizes and easy to maintain, as shown in Fig. 2-4.

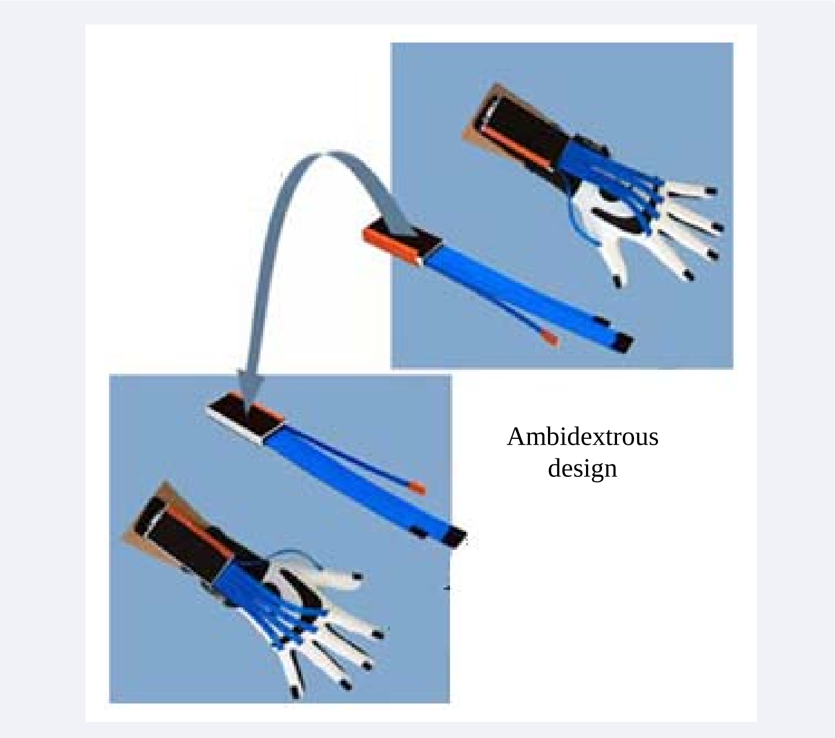


Figure 2-4. ShapeHand data gloves (modified from [89]).

Since the ShapeHand data glove is one of the most accurate data gloves at the present time, its performance has been accessed and evaluated (see Chapter 3), which proves such data glove is suitable for the purpose of this project, hence has been adopted in this project for the hand posture data capture.

2.5 HAND POSTURE RECOGNITION

From the hand posture captured, the useful hand features are extracted and classified to achieve hand posture recognition. Different hand feature extraction and classification techniques are available as discussed in the following.

2.5.1 Hand Posture Feature Extraction

A wide range of hand features has been derived from hand configuration data for hand posture recognition [92-94]. Lengths and widths of fingers, palm and hand have been considered in some research [95-97]. For example, in [95], it uses the lengths of finger and palm to correct the hand posture classification error, and in [97], it uses the lengths of finger, palm and hand from an obtained hand gesture image for recognition of digits made in sign languages. Although different attempts have been made to adopt the length and width measurements as hand gesture recognition features, they have been proved to be more suitable for the personal identification applications [98, 99].

Some research tried to use the fingertip positions as features for hand posture recognition. In [100], Hsiao et al. argues that the positions of fingertips and the centre of the palm are the most important hand posture features. They proposed an algorithm to estimate the position of each fingertip with respect to palm and to recognise hand postures based on finger-palm distance vectors. However, it seems that the posture recognition based on fingertip positions is more applicable for object manipulation and machine operation rather than for the communication purpose, such as, sign languages [101-104].

On the other hand, the hand joints angles and the angles between two fingers are more suitable for recognition of hand postures in sign languages [105-108]. A typical example is the DigitEyes hand tracking system, which considered the finger phalange lengths and joint angles as features for hand modelling [109]. The advantage of using the joint angle features is that it is more invariant to the size and position of the hand [94].

Apart from these, most research works choose to consider the hand holistically, which use the appearance of the hand, such as the shape and contour, as features for hand posture recognition [110, 111]. The general concept of such approach is to find the best match between the input hand shape and the hand shape templates. The difficulty associated with this approach comes from the large variance of the hand sizes and orientations.

2.5.2 Hand Posture Feature Classification

Based on the literature review, there are three commonly used feature classification methods for hand posture recognition, Fuzzy systems, HMM networks and Neural Networks.

The Fuzzy theory was introduced by Zadeh in 1965 [112]. Since it deals with the ambiguous inputs adaptively and produces approximate results rather than exact ones, it has been applied to the usually imprecise hand features [113]. Early attempts on the use of fuzzy systems for hand posture recognition were reported by Holden et al., where fuzzy expert systems were used to classify finger bending into three states, slightly bent, greatly bent and completely closed, for recognition of hand gestures in Australasian sign language [114, 115]. The system evaluation was conducted based on a dictionary of 21 signs and the system performance was shown to be significantly better than the system without the fuzzy method. In [116], Su presented a fuzzy rule-based approach for spatio-temporal hand gesture recognition. For each posture input, it is tested by the fuzzy rules prior to the comparison with the hand posture templates, where the templates are selected based on the hyper-rectangular composite neural networks. The system has been tested against a database with 90 sign words consisting of 34 basic hand shapes as shown in Fig. 2-5, and the author declared a recognition rate of 94.1%. Another example is Bedregal's hand gesture recognition system for Brazilian Sign Language, where the interval fuzzy rule based method is used to classify the hand joint data acquired from the data gloves [117]. Through these literatures, it can be seen that the Fuzzy system may be considered as a possible approach with good performance for hand posture recognition, because it is capable to deal with the uncertainty in the hand measurement parameters.

Other feature classification methods for hand posture recognition are largely based on the HMM and Neural Networks, which matches the extracted features with the pre-modelled feature templates [92, 93, 118, 119]. Since these techniques are also applied widely for movement classification, they will be discussed together in the section of movement classification.

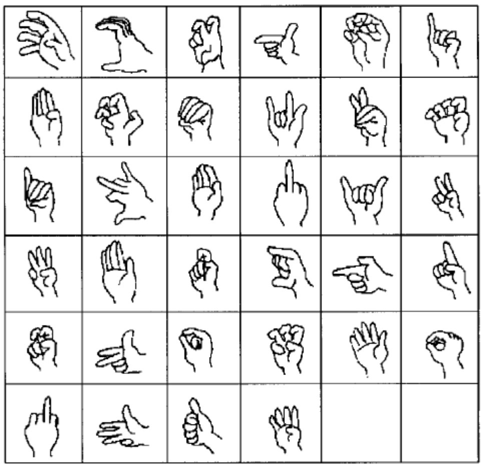


Figure 2-5. Thirty-four hand shapes implemented in [116].

2.6 HAND MOVEMENT TRACKING

For hand movement tracking, it can also be done by vision based or sensor based approaches.

2.6.1 Vision Based Hand Movement Tracking

For the vision based hand movement tracking approach, video cameras are often used to capture a sequence of images containing hand movements over a period of time, and the hand trajectory is retrieved according to the hand positions obtained from each image. This approach can also be further divided into appearance based and model based methods.

A. Appearance based movement tracking

For the appearance based movement tracking methods, the derivation of an object movement trajectory in a scene is through the analysis of the motion of the features or brightness patterns associated with the object in the image sequence [120, 121]. The simplest approach is to track the movement of the segmented object centre. A more sophisticated approach was proposed by Bobick and Davis. It computes the motion energy images through a series of low resolution images, wherein it tracks the centres of the human's body parts and results in the spatial movement information of the body parts to be obtained [122]. This is then followed by matching the movement trajectory against stored action energy image models to recognise the movement. In some other articles of Bobick and Davis, they generated a binary motion-energy image to represent the place where the motion occurred, and a motion-history image that is a grey-scale image with the intensity proportional to the recentness of the movement. The human movement is determined through the comparison with the predefined temporal templates [123, 124]. The result showed that the system worked at a low speed of 9 Hz and failed when two people presented in one scene due to the occlusion from each other. Also, it cannot deal with non-specified body parts. In [125], Kolsh and Truk used a set of KLT (stands for the initials of the authors) feature trackers to track the motion of a hand. The features are taken from the bounded skin colour blob of the hand and its centre position is determined by the median feature. The system has achieved a detection rate of 92.23%. In [126], a kernel-based tracking technique associated with a Kalman filter has been used to track objects in video images. By spatially masking a target with an isotropic kernel, the spatially smooth similarity function can be defined and localisation of the target is achieved by searching the basin of attraction of this function. Although the concept of tracking the centre of the segmented object is simple, it is sensitive to background noise.

In [127], Freeman et al. used the x-y image pixel movements and orientation histograms to extract human movement cues, and in [128], Polana and Nelson used a series of grid based tiles with movement features in each tile mapped into a vector to describe the state of movement at a time. However, both of them are subject to camera view variance. Regressive tracking methods have been investigated. For example, Heisele et al. presents an algorithm for tracking moving objects in a sequence of coloured images taken from a non-stationary camera [129]. For each image frame, an object is determined by a divisive clustering algorithm that specifies its colour and position. By adaptively renewing the clusters of the objects using a parallel k-means clustering algorithm, the centroids of the clusters are tracked.

Another attempt for appearance based movement tracking is based on computing the optical flow or the 2D field of instantaneous velocities of brightness values in the image plane [130-132]. Although this method is independent to camera motion, it is computational expensive, very sensitive to noise and lighting condition and requires the motion to be smooth and continuous thus implying a high rate of image acquisition. Still, it faces the challenge of tracking occlusion [67, 120, 130].

Therefore, although the appearance based movement tracking is conceptually simple, it has challenges in terms of computational load, sensitive to lighting, background noise, shadow and reflection, and different view perspectives. Moreover, the occlusion problem is extremely severe especially for movement tracking of multiple objects in a scene [133]. Users are often required to wear coloured suit or gloves to avoid these ambiguities [123, 134].

B. Model based movement tracking

To alleviate the problems associated with the appearance based movement tracking methods, especially occlusion and background noise, the other approach for vision based movement tracking is to use a 2D or 3D model.

2D model based movement tracking

For the 2D model based methods, the modelling is typically based on image projection of objects or parts of objects and the tracking is based on object position displacement. In the work of [135], Ju et al. defined a 'cardboard person model' based on parameterised models of optical flow, where a person's limbs are represented by a set of connected planar patches. Although it is able to track the walking movement, the orientation of the walker is assumed to remain unchanged. Bregler et al. tested several 2D object modelling methods, and finally used the expectation-maximisation method to estimate and constrain the pose with simple kinematics [136]. The result showed that it could only recognise non-trivial gestures. Lu et al. proposed a layered deformable model to recover human body parts and their movements in gait analysis, where 22 parameters has been used to model the body parts, and are categorised as four layers to allow limb deformation [137]. The model has been tested on 10,005 frames from 285 gait sequences captured under various conditions with the mean-shift algorithm used for tracking, and yields an average error rate of 7% in recovering the parameters of human limbs from their extracted silhouettes for human body pose determination. Ronfard et al. tracked people in static video frames using learned models of both the appearance and geometry of body parts [138]. The learning of articulated body planes uses Relevance Vector Machines which is a SVM-like (Support Vector Machines) classifier and offers a well-founded probabilistic interpretation and improved sparsity for reduced computation. The drawback is that the video frames must be static and the detection rate varies from 36% to 85%.

Segmenting and tracking multiple humans are difficult in complex situations due to the extended occlusion, where a single blob may contain multiple subjects due to their proximity and/or the shadows and reflections produced by the moving objects. Zhao et al. proposed a 2D model based tracking method where the subject hypotheses are generated using an elliptic shape model from multiple camera-views whereby its shadow models can be predicted [139]. These are then tracked by a Kalman filter. The results show that over 95% of human in the testing frames are correctly detected and tracked. Although the occlusion problem has been handled well, such a simple model cannot be casted in fine-grain objects, e.g., hands. Also, the multiple camera requirements make its implementation impractical. In their latter article [140], an attempt has been made to track multiple humans in a crowded scene using a single stationary camera. A Bayesian framework for multi-object tracking including a colour-based joint likelihood was used for simultaneous detection and tracking, and the MCMC-based (Markov Chain Monte Carlo) method was used to compute the optimal solution. Authors declared promising results, however, it did not solve the tracking difficulty among different object classes, and an assumption is made that the object shape under tracking must be relatively invariant. Incidentally, the paper points out another commonly used tracking filter, the particle filter, which scales poorly as the dimensionality increases because it keeps a non-parametric distribution of the joint state probability.

A difficulty associated with vision based motion tracking is that it is difficult to estimate body poses if the body movement and the camera viewing direction are coincident. In such views, the image variation caused by the motion is very small, typically much smaller than the image noise, which prevents it from successful tracking. To tackle this problem, generative modelling has been applied, which learns the model from the joint probability of both the inputs data and the estimated body pose, and then picks the most likely body pose estimation [141]. For example, the input data of the appearance configurations, e.g., appearances from different view angles, can be predicated by giving a body pose hypothesis from the subject model or calibrated camera to generate the joint probability [142, 143]. Jaeggli et al. proposed a generative model for the relationship between a body pose and its image appearance using a sparse kernel regressor, where they compare binary PCA and distance transforms to encode the appearance and this is followed by an estimation of the globally optimal trajectory through the entire sequence [142]. The tested movement is a combination of walking and running. Also in [143], Lee and Elgammal introduced a parameterised generative function where for an observed motion, contours of different body configurations, viewpoints, and shape styles have been generated. This is followed by the recovery of 3D configuration which is achieved by searching for the best matching of the models according to visual observation. Although the generative modelling method yields good results in simple human motion tracking, it is difficult to handle complex human motions due to its high dimensionality.

To reduce the computational cost for articulated motion modelling with a large dimensionality, another attempt made in [144] is to analyse and model subparts locally with the constraints of different subparts reinforced by a Markov network, which is a generative model that characterises the dynamics and the image observations of each individual subparts as well as the motion constraints among different subparts. In the most challenging condition, a 10-part articulated body can only be tracked at 0.56 frames/second. Moreover, the occlusion problem is very severe.

The temporal-flow 2D models are also used for motion tracking. Yacoob and Davis proposed a temporal-flow model which is presented as a set of orthogonal temporal-flow bases that are learned using PCA of instantaneous flow measurements [145]. Spatial constraints on the temporal-flow are also developed to model the motion of regions in rigid and articulated motion. The author claims the results are accurate, albeit tracking of some body parts failed.

3D model based movement tracking

Unlike the 2D methods, which are not able to recover the actual object position in 3D space, the 3D model based movement tracking methods aims to continuously recovering all 6-DOF data of an object, namely, the its position and orientation data [146].

Earlier attempts were made using a kinematic model for articulated body parts. Rehg and Kanade introduced a 3D hand kinematic model to predict occlusions, where the index and middle fingers are tracked using a 9-DOF kinematic model [147]. The weakness found through the results is that the tracking cannot be done in real-time. In [148], a hierarchical model has been proposed to track human body in monocular video sequences, where the kinematics is encoded using Hierarchical PCA, and dynamics are encoded using HMM networks. Although the results show that it can recover 3D or 2D skeletons from 2D images, the error rate increases drastically with the increase of ambiguity degree of the front view. In the research by Sahheen et al., they defined that 3D model based tracking usually relies on filtering, local optimisation, or global optimisation, where the global optimisation can be further divided in to single hypothesis and multiple hypothesis optimisation [149]. They compared these underlying mathematical models and evaluated the performance of one representative algorithm for each class as well as comparing several likelihoods and parameter settings. A conclusion is made that multiple hypothesis optimisation performs better than single hypothesis optimisation and global optimisation is better than filtering or local optimisation for markerless human motion capture with 3D models. However, all of them suffer from low accuracy and low tracking rate.

Motion filters have been used in many 3D tracking methods to smooth the pose estimation and to provide predictions which improve the reliability of the tracker. Stochastic tracking frameworks such as CONDENSATION (Conditional Density Propagation) and particle filtering are applied to handle the condition of complex non-Gaussian probability density functions e.g., kinematic singularity and joint endstops [150]. However, these filters are ineffective because the number of required particles is exponentially proportional to the dimensionality and evaluation of the likelihood is computational expensive [121, 151]. Davison et al. addressed these problems by developing an Annealed Particle Filter (APF) which uses a weighting function to approximate the likelihood [152]. Despite it allows the use of larger particle distributions with less computational effort, which improves the tracking results, many particles are wasted in randomly generated configurations.

Hence, despite efforts of using various 3D modelling techniques to improve vision based tracking, particularly in the area of occlusion and background noise, they are not able to resolve these difficulties completely and also increase the computation expense in terms of the model construction.

2.6.2 Sensor Based Hand Movement Tracking

Although vision based hand movement tracking is uncumbersome and unintrusive, it has disadvantages in terms of low accuracy, high computational load, and high sensitivity to various background noise and camera-view variance, as well as its inherent occlusion problem. The inherent weaknesses of the vision based approach, especially the image segmentation, camera-view variance and occlusion problems, can be reduced by the sensor based method to some degree, at the expense of introducing a little inconvenience for sensor mounting. While a large number of commercialised tracking sensors are available on the market, the research conducted in this report adopted the InterSense IS-900 tracking devices to track the hand and head movement of users due to its advantages in terms of resolution, accuracy, update frequency, latency, tracking range and cost. More details of such tracking devices will be given in Chapter 3.

2.7 HAND MOVEMENT RECOGNITION

Similar to hand posture recognition, hand movement recognition has also attracted substantial research due to its wide applications e.g., surveillance, medical studies and rehabilitation, robotics, video indexing, and animation for film and games [153]. The process also consists of feature extraction and classification, and they will be discussed as follows.

2.7.1 Movement Feature Extraction

Literature review reveals that movement feature extraction can be conducted based on the regularity features or spatio-temporal features. Compared to the latter, the feature extraction based on the movement regularity has advantages in terms of speed as well as robustness for its view and time invariance, and the commonly used methods are the Fourier analysis and autocorrelation techniques.

Fourier analysis is a powerful tool which has been applied in multiple disciplines. In movement recognition, one of its techniques, the Fourier transform, has been explored for repetitive movement feature extraction [154-158]. A standard procedure for the Fourier transform applied in movement recognition starts with the removal of the DC (Direct Current) component and followed by the search for the fundamental frequency. With the fundamental frequency and the span length of the movement known, the movement feature of a repetitive movement, i.e., the number of peaks and valleys, can be easily extracted. Although Fourier analysis is simple and effective, its results only reveal the frequency of the repetitive movements, and it is not possible to retrieve the features without the knowledge of the movement length. Also, the repeating manner of the movement must behave in a very similar way; otherwise, multiple frequencies will appear. Furthermore, the sampling frequency must be at least double of the highest frequency of the periodic motion to avoid false results [159].

Fourier series has also been investigated for movement feature extraction. Since the Fourier series conventionally targets to the analysis of periodic waveform [160, 161], early attempts have been limited to the analysis of periodic movements, e.g., walking and running [162, 163]. However, Cosgriff [164] proposed that a closed shape contour, which is not necessary repetitive, can be recognised as a periodic waveform, since a starting point to count through the contour can also be considered as the starting point for the next counting-through process [165-167]. By mapping the contour pixels to complex coordinates and applying the Fourier analysis with the contour length normalised to 2π, the Fourier coefficients obtained are known as the Fourier descriptor and a shape contour can be uniquely identified according to the distribution of the Fourier coefficients. Due to its frequency characteristics, the Fourier descriptors have the following properties:

- Translation: the translation of a contour is equivalent to add a constant to its DC component with the DC component value indicating the centre position of the contour;

- Rotation: the rotation of a contour about its origin by angle θ in the complex plane is equivalent to multiply its Fourier descriptor by $e^{i\theta}$; and

- Scaling: the scaling of a contour is equivalent to multiply its Fourier descriptor by the same constant.

In practical applications, a movement trajectory is not necessary a close curve, where an alternative is to dilate the recorded movement trajectory first, and then extract its boundary to obtain its Fourier descriptor [168]. However, similar to the application for hand configuration extraction discussed in Section 2.4.1, although this approach has some advantages, it has drawbacks in losing fine-grain features and demanding a large trajectory templates for high precision movement classifications. Additionally, in [169], the authors use the Fourier descriptor to analysis the human's gait by considering the human silhouette contour and gait as periodic.

Although the FSD approach dramatically reduces the computational cost, the large gesture vocabulary (e.g. ASL) poses a problem for collecting an adequate training set to build the templates and may lose compactness in the subspace required for efficient processing [64]. Moreover, the attained templates may vary due to the different camera view angles they are taken from

Autocorrelation techniques have also been applied for movement feature extraction which is to find the cues of movement similarities, that is, its periodicity [157, 159, 170, 171]. However, the drawback associating with this method is that it only yields good results for pure periodic movements.

Therefore, although the movement analysis approaches based on the regularity features are fast and robust in terms of view and time invariance, they are more applicable to periodic movement feature extraction and recognition, such as human walking gait, cycling movement, rotating and swinging [163, 172].

On the other hand, the movement analysis based on spatio-temporal features analyses the movement shape information over time, and the commonly extracted curve features are curvature, torsion, velocity and acceleration [173-177]. These features can be used to segment a movement into different intervals according to its significant changes in order to do further classification. In [174], Faria and Dias used the features of curvature and hand orientation to segment hand movements, histogram techniques to build movements database and Bayesian techniques to classify the movements. In [175], Chen and Chang decomposed the whole object trajectory into sub-trajectories based on the features of acceleration, velocity and arclength, thereby enabling each sub-trajectory to be modelled and matched with templates respectively. The curvature is an important feature to describe a movement. In [178], the curvature is defined as the reciprocal of its radius and expressed as:

$$k_1 = 4\frac{\sqrt{s(s-a)(s-b)(s-c)}}{abc} \tag{2.1}$$

where

$$s = \frac{(a+b+c)}{2} \tag{2.2}$$

and *a*, *b* and *c* denote the side lengths of the triangle linking three neighbouring points on the curve as shown in Fig. 2-6.

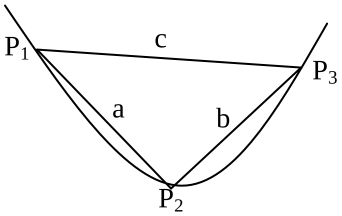


Figure 2-6. Curvature based on triangle linking three points on a curve.

Although this method is simple, the curvature results depend on the distances between the three neighbouring points selected for computation.

Furthermore, two other commonly used 2D curvature computation methods have also been attempted for feature extraction which are shown in equations (2.3) [179, 180] and (2.4) [181].

$$k_2 = \frac{x'(t)y''(t)-x''(t)y'(t)}{(\sqrt{x'(t)^2+y'(t)^2})^3} \tag{2.3}$$

$$k_3 = \frac{\sqrt{y''(t)^2+x''(t)^2+(x'(t)y''(t)-x''(t)y'(t))^2}}{(\sqrt{x'(t)^2+y'(t)^2+1})^3} \tag{2.4}$$

where $x'(t)$, $y'(t)$ and $x''(t)$, $y''(t)$ are the first and second derivative of the displacement along the x and y axes denoted by $x(t)$ and $y(t)$.

These two methods can be extended to 3D, where the torsion feature on the third dimension can also be computed in a similar manner [173]. In general, these methods provide good results in terms of the changes in curvature. However, a drawback associated with them is that they are sensitive to the movement jitters due to the use of derivative. This problem could be solved by applying some low-pass or median filters. Through the literature review, it can be seen that although many movement features can be used for movement feature recognition, a thorough consideration should been taken to extract the suitable features for movement recognition and not to jeopardise the system's real-time performance. In this project, some other simple movement features have been investigated for movement recognition, including movement length, height, etc., which will be discussed in the following chapters.

2.7.2 Movement Feature Classification

For hand movement feature classification, the common approach is to match the extracted features with the pre-modelled feature templates. Many techniques have been explored for template modelling. The HMM is one of the most broadly used techniques, which is a kind of stochastic machine that statistically generates a mixture model with the variables related through a Markov process rather than independent of each other [182]. The implementation generally starts with training of the model based on a sufficient number of observations gathered from identical movements. This is followed by classification to find the best match between the input candidate movement and the movement model. A variety of HMM models has been used for movement recognition [179, 183-189]. Another well-studied network model is the neural network, which can be self-trained through weight adjustment of the connections between neural states [190-192]. Bayesian networks have also been investigated to classify the hand motion trajectory from obtained observations [193-196]. However, overall speaking, although template modelling of the movements is more accurate and robust in movement recognition, it is a computational expensive and time-consuming process due to the requirement of huge observations to train the network, the mathematic complexity for the state and variable computation, as well as the iteration requirement for network model evolution.

Another feature classification technique is the Dynamic Time Warping (DTW), which is a dynamic programming matching technique. It finds an optimal match between two sequences of feature vectors, i.e., the input and template movements, by stretching and compressing them. Therefore, a movement can be recognised according to its shape which is of most importance regardless of its size. Compare to the statistical network based method, this is better in flexibility, conceptual simplicity and time effectiveness [67, 173, 197, 198]. However, it is sensitive to movement jitters. Similarly, some other curve fitting techniques, such as least square curve fitting [199, 200], B-spline curve fitting [201] and Bezier curve fitting [202], have been investigated and applied in movement characterisation, e.g., estimating the polynomial approximation of movements [203]. The drawback associated with these curve fitting techniques is that the inaccuracy in shape representation is inevitable. For example, if more than two data points are available, a curve polynomial may fit the data better than a line regardless whether these data are collinear.

Therefore, although different techniques have been attempted for hand movement feature classification, all of them have advantages and disadvantages compared to one another. A thoughtful consideration should be taken to find a suitable method of movement classification for this project to recognise the hand movements.

2.8 CONCLUDING REMARKS AND PROPOSED APPROACH

This chapter presents the literature review of the current available techniques for HCI. With the hand based HCI being a key HCI modality which provides a natural, intuitive and immersive interaction in several hand-orientated HCI applications such as virtual object manipulation and direct sign language recognition, it becomes the focus of this project. Also, in order to avoid the potential problems associated with the vision based hand posture capture methods, e.g., high computational cost, low accuracy and self-occlusion, the glove based approach has been adopted for the system development, where the ShapeHand glove has been used due to its satisfied hand data capturing accuracy.

Since the ShapeHand data glove merely captures the relative positions and orientations of each finger joint with respect to the wrist by measuring the bend and twist at each sensor, an IS-900 wireless inertial and ultrasonic sensor based tracking system from InterSense [204, 205], is used to provide the required hand position and orientation data, as well as the head position and orientation data of the user in order to generate correct views, due to its advantages in terms of accuracy, computational load, performance in occluded situation, etc., compared with the vision based movement tracking systems. Furthermore, a large stereoscopic display screen with two stereoscopic back projections is used to create an immersive virtual environment for visual feedback.

As shown in Fig. 2-7, driven by a computer, the proposed system consists of a pair of the ShapeHand data gloves to be worn by the user for acquisition of hand posture data, a pair of InterSense hand trackers to be attached to the user wrists to collect the position and orientation data of hands in 3D, an InterSense head tracker to be worn by the user to collect the position and orientation data of head in 3D, and a large stereoscopic display system for immersive visualisation. Presented in the following chapters are the system and algorithm development to enable a user to perform virtual object manipulation and DSW through dynamic hand gestures in an immersive environment.

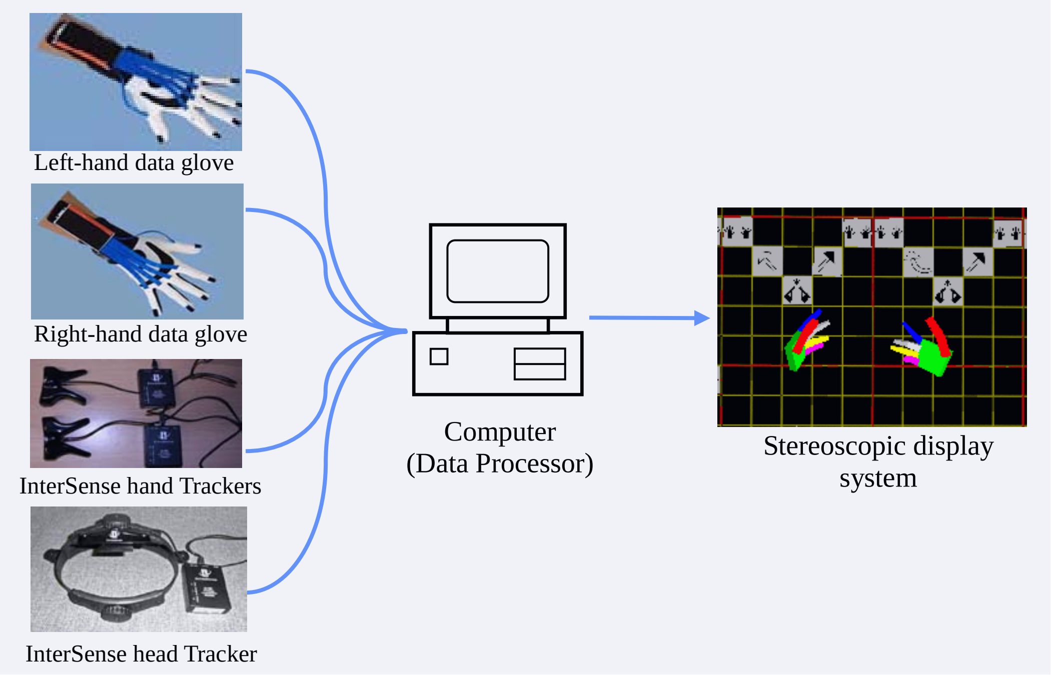


Figure 2-7. Illustration of the proposed system.

Chapter 3

SYSTEM DEVELOPMENT

3.1 INTRODUCTION

This chapter presents the work related to the hardware and software development of the proposed system. It begins with the investigation of each sub-system, namely, ShapeHand data glove system in Section 3.2, InterSense tracking system in Section 3.3 and stereoscopic display system in Section 3.4. For each of them, the specification, principle of working, system setup and performance evaluation will be presented. This is followed by the hardware integration presented in Section 3.5 and software development presented in Section 3.6. Finally, Section 3.7 presents the performance evaluation of the developed system, and Section 3.8 gives some concluding remarks.

3.2 SHAPEHAND DATA GLOVE SYSTEM

3.2.1 Introduction of ShapeHand System

As shown in Fig. 3-1, the ShapeHand data glove from the Measurand ShapeWrap system series is a portable, light weight and wireless hand motion capture system [206]. It is based on five flexible rubber tapes embedded with multiple fibre-optic curvature sensors arranged to sense bend and twist along the length of each tape. By wearing a leather glove to wrap the tapes to run along each finger with one end at the finger tip and the other end feeding to a small data acquisition box at wrist, the movements of fingers introduce deformation to the tapes. As a result, the bend and twist measured at each sensor location with respect to the end of the tape at the wrist enables relative position and orientation of each finger joint to be determined [207, 208].

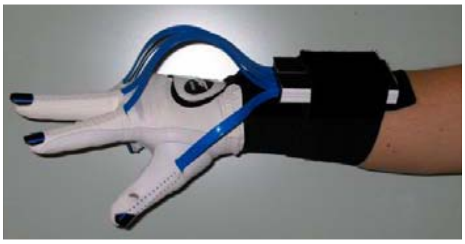


Figure 3-1. ShapeHand data glove.

To gain a better understanding of the ShapeHand data gloves, the specifications and the operation principle of the ShapeTape were investigated. The ShapeTape has a width of 1.0 cm and thickness of 0.1 cm with different lengths for each hand finger, where a thin array of fiber optic curvature sensors is laminated on a ribbon substrate. Each sensor provides a single output proportional to its curvature according to the light intensity passing through, and the curvature contains two degrees of freedom of bend (pitch and yaw angles) and one degrees of freedom of twist (roll angle) components. Since the sensor positions in each ribbon are known with respect to the ribbon starting location at the wrist, the resulting signals of the sensors enable the position and orientation information (6-DOF) to be obtained for any location along the ribbon. Since the spacing between two neighbouring sensors along the tape is not necessary uniform, interpolation is needed to enable equal distance curvature measurement to be computed at a fine resolution along the tape. Therefore, the ShapeTape measurement resolution is limited by the interpolation segment length. Furthermore, the error may accumulate along the length of the tape, because the light intensity sensed by the last sensor depends on the light passing through previous sensors [90].

To assess the significance of the ShapeTape error, some engineering experiments have been conducted by others. A test reported by [209] compares the manual and the ShapeTape based position measurements of a point. Another test reported by [210] includes two assessments, where the former was based on the angular difference between the line going through the user’s eye to the real position of the point on a pre-calibrated grid and the line going through the user’s eye to the measured position of the ShapeTape tip that was held by the user and pointed to the point positioning on the grid (see Fig. 3-2); and the latter was based on recording the end segment position of the tape at the start, followed by projecting the points onto a 2D plane and verifying the movement by a Graffiti gesture recogniser [211]. The results from [209] showed that the ShapeTape has the position errors exceeding 6%, and the results from [210] showed that the angular pointing errors could be as high as 10^{o}, but the ShapeTape demonstrated to be accurate enough for users to enter text using a Graffiti-based gesture recogniser. From these results, they suggest that the ShapeTape is not suitable for HCI requiring a high degree of accuracy. However, it is capable of reproducing the qualitative motion performed by the user, and thus could be used to support 3D gesture-based interaction where absolute accuracy is of limited importance. Therefore, the ShapeTape-based data glove, ShapeHand, is considered as an appropriate tool for hand gesture based virtual object manipulation and the DSW in this project.

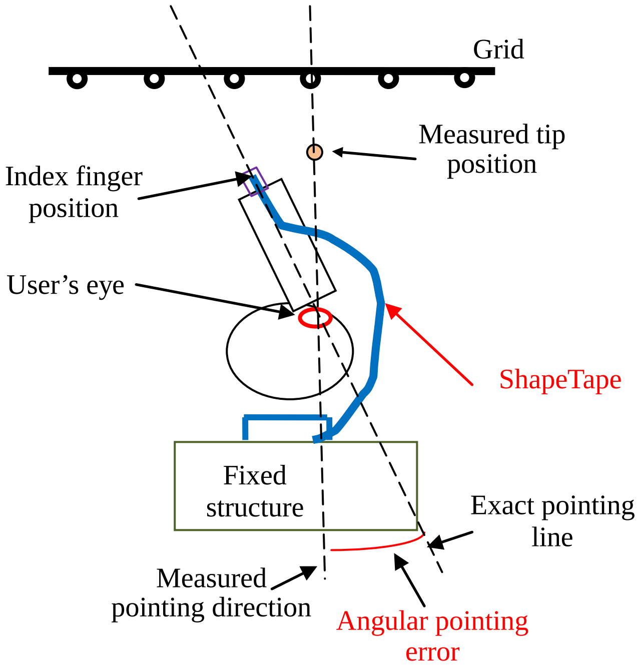


Figure 3-2. ShapeTape angular pointing error measurement (modified from [210]).

3.2.2 ShapeHand System Setup

Fig. 3-3 illustrates the setup of the ShapeHand system. Driven by a computer, the system is an integration of three sub-modules, namely, the input module to collect the hand data from the ShapeHand data gloves, the data communication module to provide the link between the computer and the ShapeHand data gloves, and the data output module. For the input module, a pair of ShapeHand data gloves is used. Hand gestures formed by movements of fingers introduce deformation to the tapes, and the bend and twist measured at each sensor location with respect to the end of the tape at the wrist enable relative position and orientation of each finger joint to be determined. These measurements are sent to the computer through the data communication module. This module consists of a wireless router and a ShapeHand data concentrator which contains a wireless network card, 11 ports for data collection from the ShapeTapes and a port connecting to the computer for the setup initialisation. With all the raw hand data from the gloves gathered, they will be transmitted via the wireless network card to the wireless router which is connected to an Ethernet port of the computer for processing. Finally, the processed data from the computer is sent to the data output module in a format compatible with the output device for display.

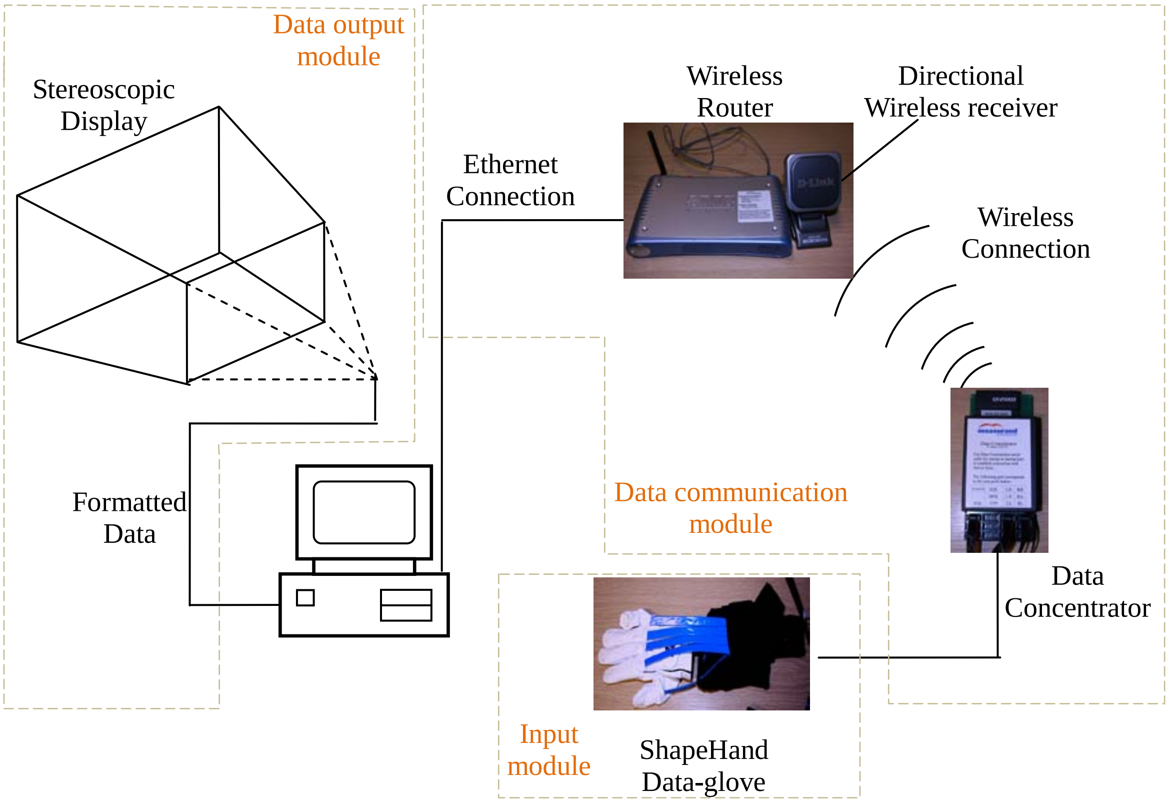


Figure 3-3. ShapeHand system setup.

3.2.3 ShapeHand System Performance Evaluation

For evaluation of the ShapeHand system performance, the ShapeRecorder API interface provided by the Measurand Company has been used. ShapeRecorder is able to display, capture, and export motion capture data from all the ShapeWrap devices. It can also be used for viewing and saving data from individual ShapeTape. Some basic instructions for the ShapeHand system are listed in Table 3.1, which perform the system initialisation, communication link setting up, data collection and application termination, respectively. During the evaluation, the user is required to wear the ShapeHand system as shown in Fig. 3-4, where two ShapeHand data gloves are worn on the user's hands and the data concentrator is attached onto the forebody of the user with the help of Thoracic Harness. Moreover, the system could operate in the wireless mode after setting up the connection between the data concentrator and computer, thereby enabling the user to wander around freely.

Table 3.1: API commands for ShapeHand system

Commands	**Functions**
Initialisation()	ShapeHand data initialisation
ConnectToDataCollector()	Connect to the data concentrator
GetPortInfo(SERIAL_DEVICE)	Identify and setup link to the detected tape
GetHandData()	Collect the hand data from data gloves
Shutdown()	Terminate the application and free resources

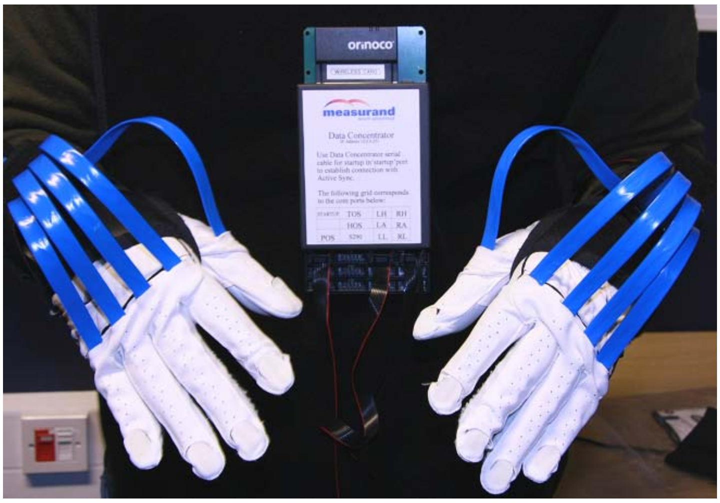
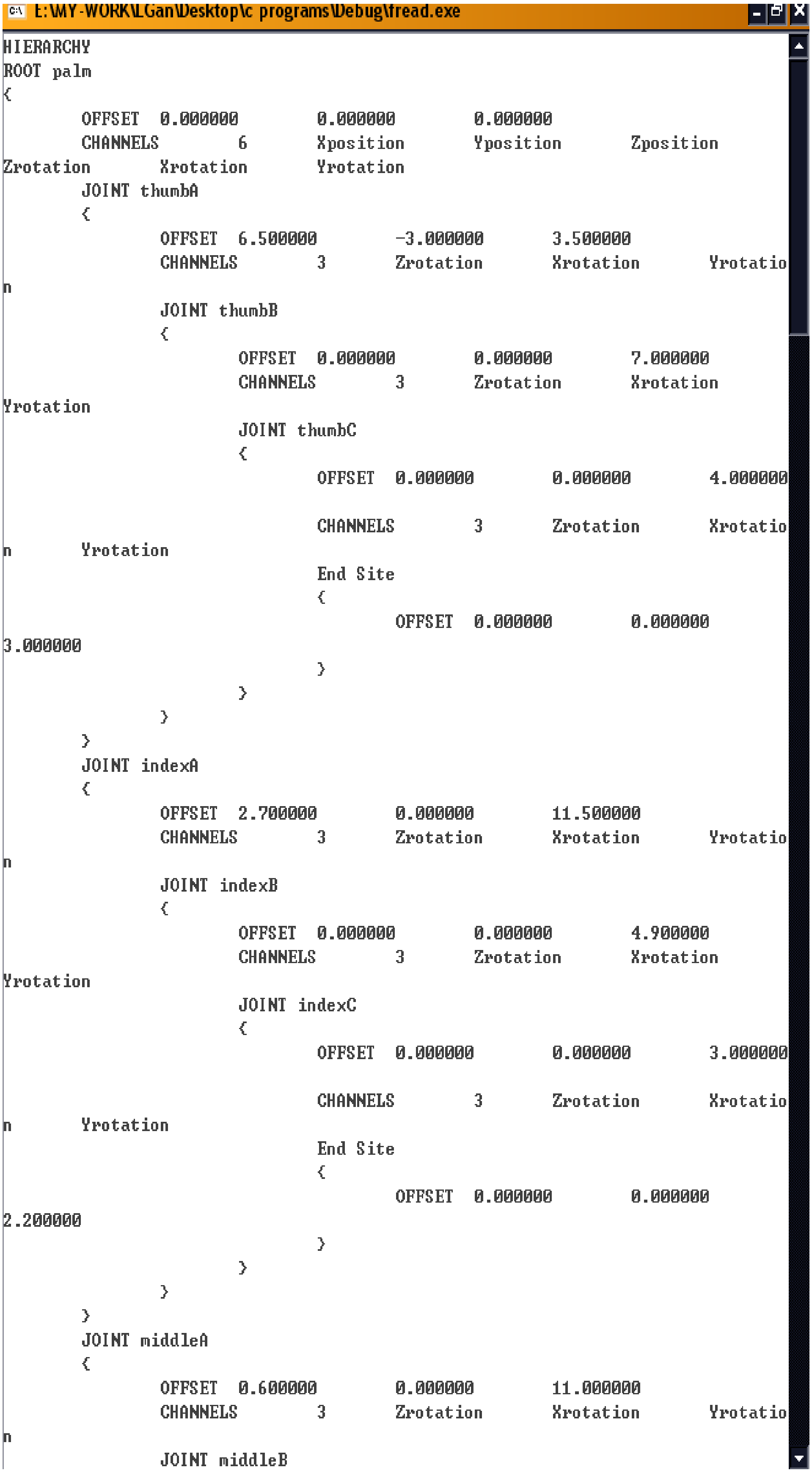


Figure 3-4. User equipped with ShapeHand system.

The outputs from ShapeHand are the position and orientation parameters of each finger joint. Two types of output are available from the ShapeRecorder software with one in numeric data formats (with file name extensions of ‘***.C3D***’,’***.BVH***’ and ‘***.txt***’), and the other one in a graphic format. As an example, a recorded ’***.BVH***’ file and a graphic output are presented in the following.

The ‘***.BVH***’ data file shown in Fig. 3-5 is decoded using a C/C++ program. From this figure, it is seen to contain five frames of hand motion data at the frame rate of 2.6 ms per frame. The output shows that the hand data have been recorded in a hierarchical structure which is based on the hand kinematic joint system [210]. Based on the hand joint structure and their offsets described in the ‘ROOT’ section (see Fig. 3-5(a)), which are the lengths of the hand phalanges, the ‘MOTION’ section (see Fig. 3-5(c)) records the position and rotation parameters of the joints corresponding to the channel IDs stated in the ‘ROOT’ section. For example, the first six data recorded in the ‘MOTION’ section in the line below the line of ‘Frame Time’ are corresponding to the channel IDs of ‘Xposition’, ‘Yposition’, ‘Zposition’, ‘Zrotation’, ‘Xrotation’ and ‘Yrotation’ in the line above the line of ‘JOINT thumbA’ in the ‘ROOT’ section, respectively,. It can be seen that according to the channel and Joint IDs, the ‘MOTION’ data structure starts with the three position and three rotation data of the hand base at the recording time, where the position normally remains as zero for it is the centre of the hand local coordinate system. It is then followed by 5 x 3 (5 finger digits, where each digit has three joints) groups of three-parameter direction data for each hand joint starting from the proximal joint of the thumb to the distal joint of the little finger.

Based on the recorded ’***.BVH***’ file, the hand posture can be reconstructed as shown in Fig. 3-6, where the hand model has been rotated -90^0 with respect to the x-axis (denoted by X_S) compared to the conventional right hand coordinate system in order to yield a frontal view. It can be seen that the x-axis is towards the side of the narrow flat tape (in the direction across the palm), the y-axis (denoted by Y_S) is perpendicular to the back of the narrow flat tape (perpendicular to the back of the palm), and the z-axis (denoted by Z_S) runs along the length of the narrow flat tape (along each finger towards the finger tip), which forms a right hand coordinate. Furthermore, the origin is fixed at the bottom of the palm in the middle of the wrist.

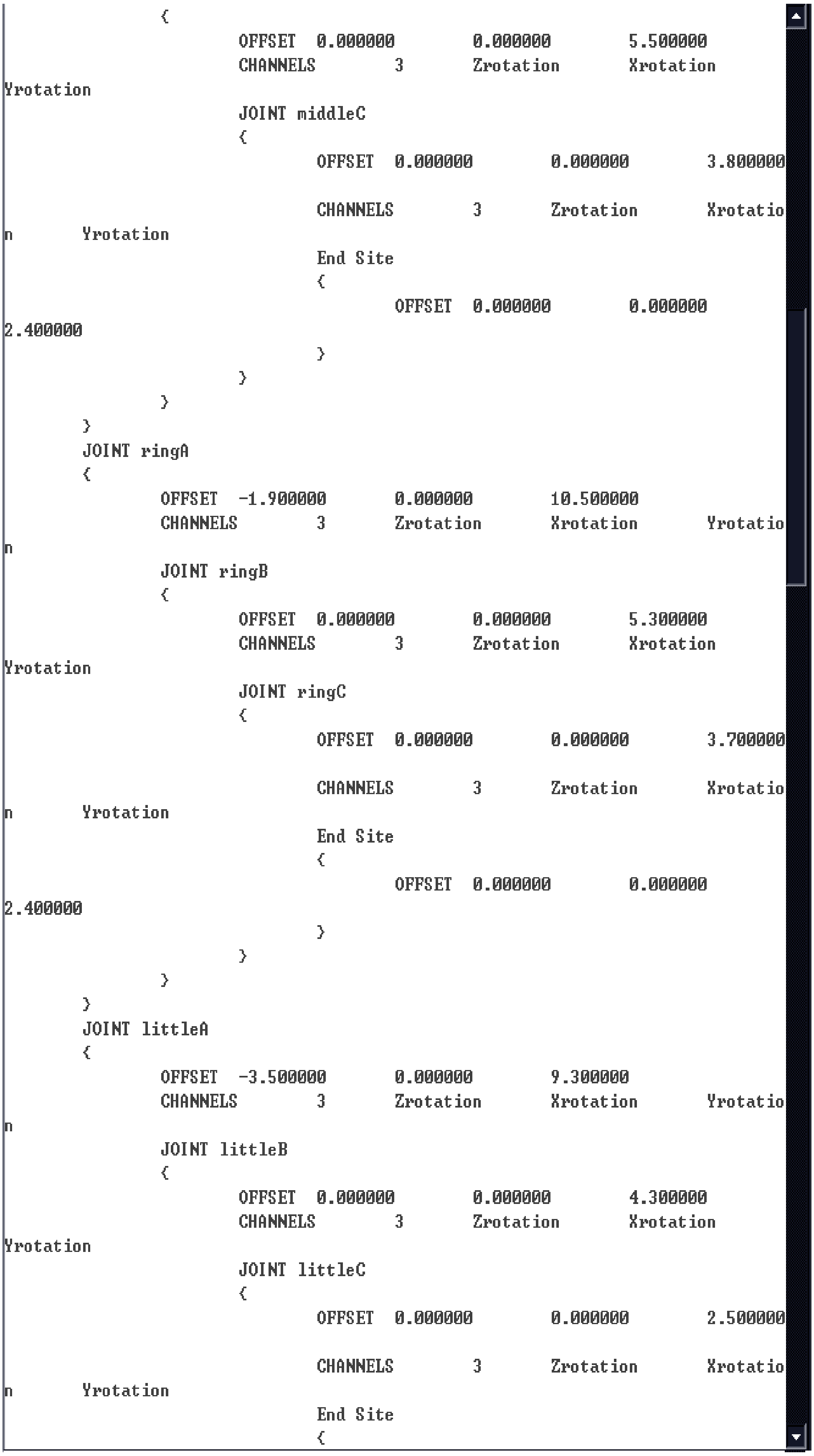


```
HIERARCHY
ROOT palm
{
        OFFSET  0.000000        0.000000        0.000000
        CHANNELS        6       Xposition       Yposition       Zposition
Zrotation       Xrotation       Yrotation
        JOINT thumbA
        {
                OFFSET  6.500000        -3.000000       3.500000
                CHANNELS        3       Zrotation       Xrotation       Yrotatio
n
                JOINT thumbB
                {
                        OFFSET  0.000000        0.000000        7.000000
                        CHANNELS        3       Zrotation       Xrotation
Yrotation
                        JOINT thumbC
                        {
                                OFFSET  0.000000        0.000000        4.000000

                                CHANNELS        3       Zrotation       Xrotatio
n       Yrotation
                                End Site
                                {
                                        OFFSET  0.000000        0.000000
3.000000
                                }
                        }
                }
        }
        JOINT indexA
        {
                OFFSET  2.700000        0.000000        11.500000
                CHANNELS        3       Zrotation       Xrotation       Yrotatio
n
                JOINT indexB
                {
                        OFFSET  0.000000        0.000000        4.900000
                        CHANNELS        3       Zrotation       Xrotation
Yrotation
                        JOINT indexC
                        {
                                OFFSET  0.000000        0.000000        3.000000

                                CHANNELS        3       Zrotation       Xrotatio
n       Yrotation
                                End Site
                                {
                                        OFFSET  0.000000        0.000000
2.200000
                                }
                        }
                }
        }
        JOINT middleA
        {
                OFFSET  0.600000        0.000000        11.000000
                CHANNELS        3       Zrotation       Xrotation       Yrotatio
n
                JOINT middleB
```

(a)



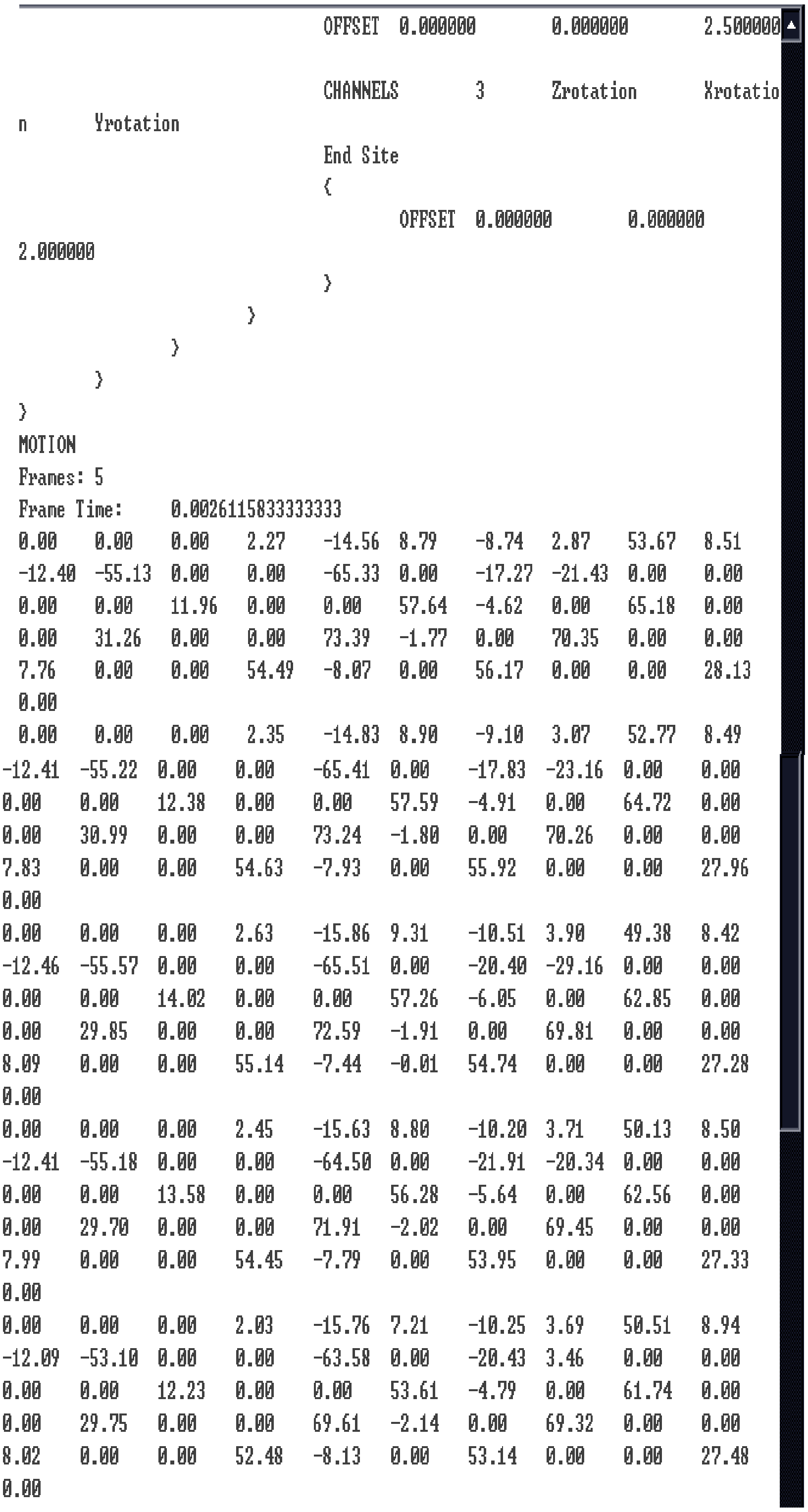
```
                {
                        OFFSET  0.000000        0.000000        5.500000
                        CHANNELS        3       Zrotation       Xrotation
Yrotation
                        JOINT middleC
                        {
                                OFFSET  0.000000        0.000000        3.800000

                                CHANNELS        3       Zrotation       Xrotatio
n       Yrotation
                                End Site
                                {
                                        OFFSET  0.000000        0.000000
2.400000
                                }
                        }
                }
        }
        JOINT ringA
        {
                OFFSET  -1.900000       0.000000        10.500000
                CHANNELS        3       Zrotation       Xrotation       Yrotatio
n
                JOINT ringB
                {
                        OFFSET  0.000000        0.000000        5.300000
                        CHANNELS        3       Zrotation       Xrotation
Yrotation
                        JOINT ringC
                        {
                                OFFSET  0.000000        0.000000        3.700000

                                CHANNELS        3       Zrotation       Xrotatio
n       Yrotation
                                End Site
                                {
                                        OFFSET  0.000000        0.000000
2.400000
                                }
                        }
                }
        }
        JOINT littleA
        {
                OFFSET  -3.500000       0.000000        9.300000
                CHANNELS        3       Zrotation       Xrotation       Yrotatio
n
                JOINT littleB
                {
                        OFFSET  0.000000        0.000000        4.300000
                        CHANNELS        3       Zrotation       Xrotation
Yrotation
                        JOINT littleC
                        {
                                OFFSET  0.000000        0.000000        2.500000

                                CHANNELS        3       Zrotation       Xrotatio
n       Yrotation
                                End Site
                                {
```

(b)



```
                                OFFSET  0.000000        0.000000        2.500000

                                CHANNELS        3       Zrotation       Xrotatio
n       Yrotation
                                End Site
                                {
                                        OFFSET  0.000000        0.000000
2.000000
                                }
                        }
                }
        }
}
MOTION
Frames: 5
Frame Time:     0.0026115833333333
0.00    0.00    0.00    2.27    -14.56  8.79    -8.74   2.87    53.67   8.51
-12.40  -55.13  0.00    0.00    -65.33  0.00    -17.27  -21.43  0.00    0.00
0.00    0.00    11.96   0.00    0.00    57.64   -4.62   0.00    65.18   0.00
0.00    31.26   0.00    0.00    73.39   -1.77   0.00    70.35   0.00    0.00
7.76    0.00    0.00    54.49   -8.07   0.00    56.17   0.00    0.00    28.13
0.00
0.00    0.00    0.00    2.35    -14.83  8.90    -9.10   3.07    52.77   8.49
-12.41  -55.22  0.00    0.00    -65.41  0.00    -17.83  -23.16  0.00    0.00
0.00    0.00    12.38   0.00    0.00    57.59   -4.91   0.00    64.72   0.00
0.00    30.99   0.00    0.00    73.24   -1.80   0.00    70.26   0.00    0.00
7.83    0.00    0.00    54.63   -7.93   0.00    55.92   0.00    0.00    27.96
0.00
0.00    0.00    0.00    2.63    -15.86  9.31    -10.51  3.90    49.38   8.42
-12.46  -55.57  0.00    0.00    -65.51  0.00    -20.40  -29.16  0.00    0.00
0.00    0.00    14.02   0.00    0.00    57.26   -6.05   0.00    62.85   0.00
0.00    29.85   0.00    0.00    72.59   -1.91   0.00    69.81   0.00    0.00
8.09    0.00    0.00    55.14   -7.44   -0.01   54.74   0.00    0.00    27.28
0.00
0.00    0.00    0.00    2.45    -15.63  8.80    -10.20  3.71    50.13   8.50
-12.41  -55.18  0.00    0.00    -64.50  0.00    -21.91  -20.34  0.00    0.00
0.00    0.00    13.58   0.00    0.00    56.28   -5.64   0.00    62.56   0.00
0.00    29.70   0.00    0.00    71.91   -2.02   0.00    69.45   0.00    0.00
7.99    0.00    0.00    54.45   -7.79   0.00    53.95   0.00    0.00    27.33
0.00
0.00    0.00    0.00    2.03    -15.76  7.21    -10.25  3.69    50.51   8.94
-12.09  -53.10  0.00    0.00    -63.58  0.00    -20.43  3.46    0.00    0.00
0.00    0.00    12.23   0.00    0.00    53.61   -4.79   0.00    61.74   0.00
0.00    29.75   0.00    0.00    69.61   -2.14   0.00    69.32   0.00    0.00
8.02    0.00    0.00    52.48   -8.13   0.00    53.14   0.00    0.00    27.48
0.00
```

(c)

Figure 3-5. ‘***.BVH***’ file output from ShapeRecorder.

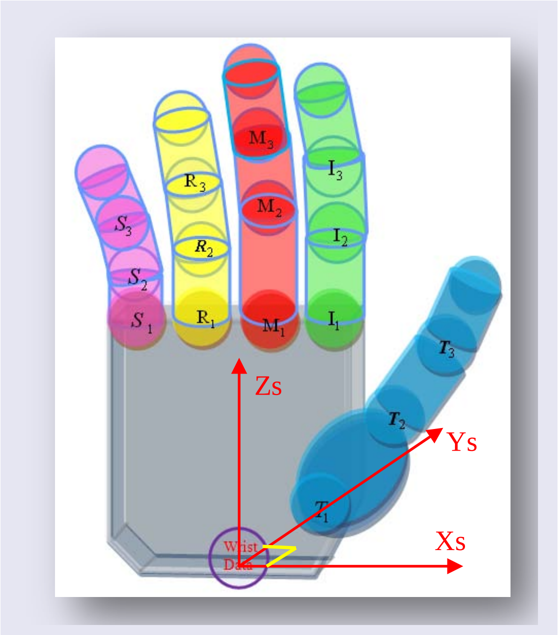
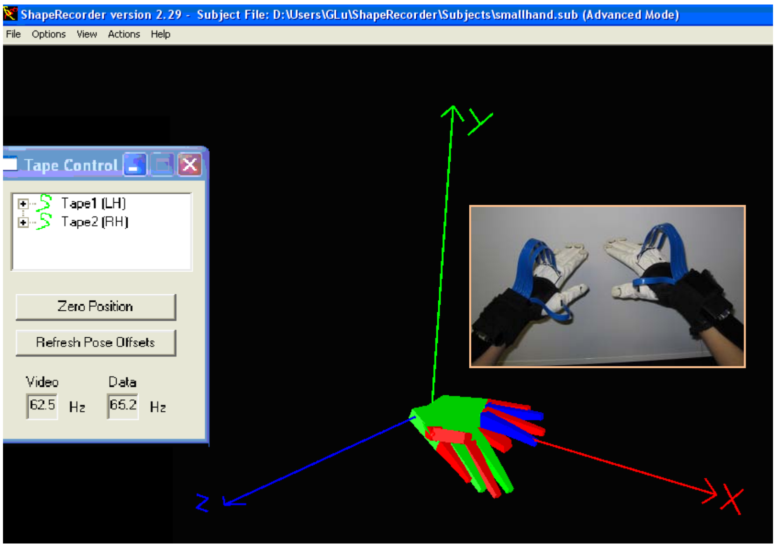


Figure 3-6. Kinematic hand model where T_1, T_2 and T_3 correspond to JOINT thumbA, JOINT thumbB and JOINT thumbC in the ‘***.BVH***’ file. Similarly, I_1, I_2 and I_3 to JOINT indexA, JOINT indexB and JOINT indexC; M_1, M_2 and M_3 to JOINT middleA, JOINT middleB and JOINT middleC; R_1, R_2 and R_3 to JOINT ringA, JOINT ringB and JOIINT ringC; S_1, S_2 and S_3 to JOINT littleA, JOINT littleB and JOINT littleC.

Furthermore, the graphic output of ShapeRecorder provides an animated approach to display the user’s hand configuration. As illustrated in Fig. 3-7, although the ShapeRecorder graphic output follows the hand movements of both hands, it does not provide the global positions and orientations of the two wrists. Furthermore, it can be seen that it has a video update rate of 62.5 Hz and a data transfer rate of 65.2 Hz. Incidentally, to obtain good tracking results of the hand, a thorough pre-calibration and a suitable glove size for the user’s hand are required, and more information can be found in the documentation provided by Measurand [206].



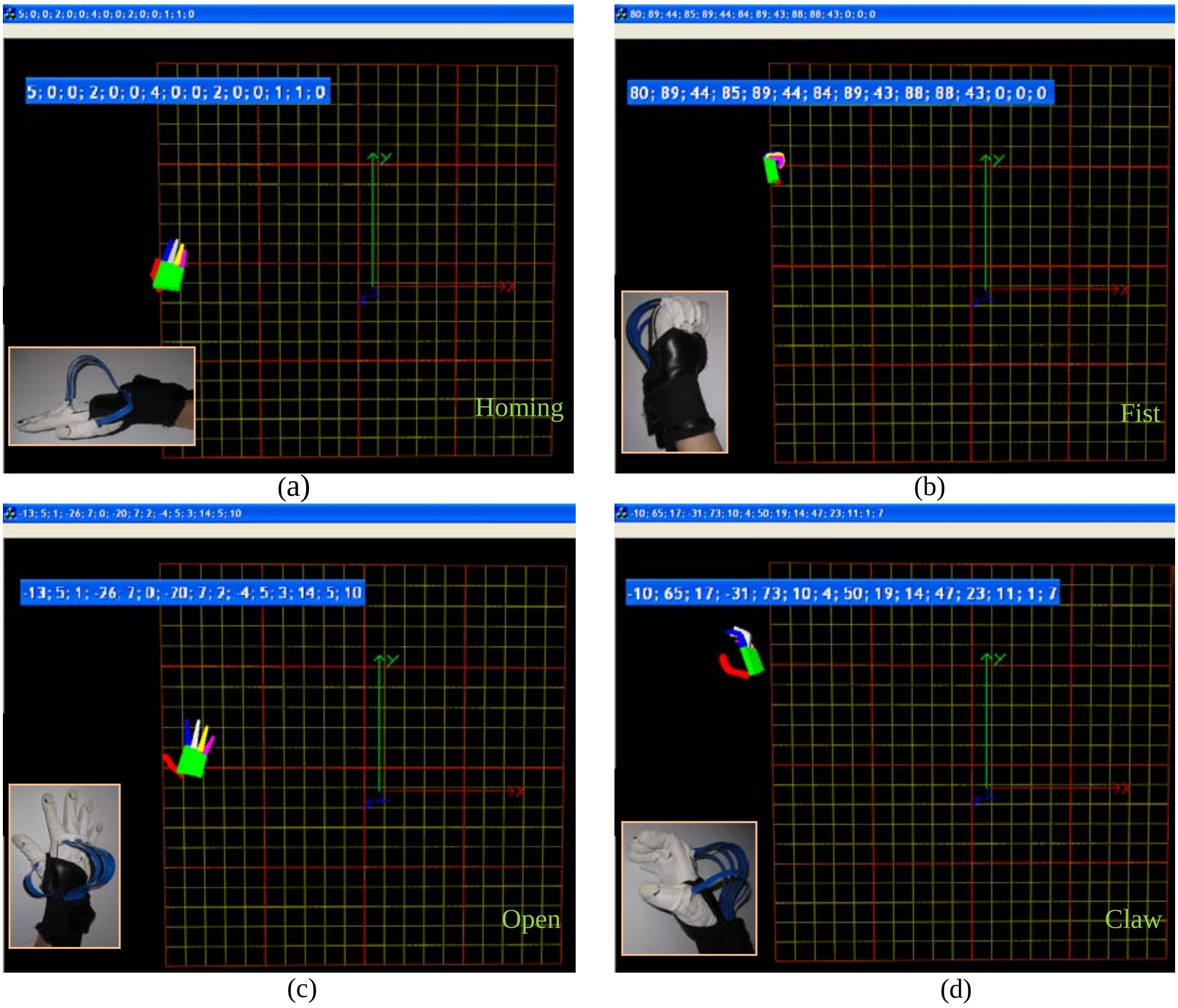
ShapeRecorder version 2.29 - Subject File: D:\Users\GLu\ShapeRecorder\Subjects\smallhand.sub (Advanced Mode)

Tape1 (LH)
Tape2 (RH)

Video 62.5 Hz Data 65.2 Hz

Figure 3-7. ShapeRecorder output for two hand gestures shown in the photograph.

Further testing has also been conducted to evaluate the hand joint angle measurement. By making the hand postures of ‘Homing’, ‘Open’, ‘Fist’ and ‘Claw’, with the hand joint angles programmed to be shown on the tool bar (see Fig. 3-8), the accuracy of the data glove measurement can be assessed. From Fig. 3-8, for the ‘homing’ hand posture, all four finger joint angles are seen to be approximately zero degree; for the ‘Fist’ hand posture, the pitch angles of all four finger joints are around their maximum values with the angles of the proximal and intermediate phalanges close to 90 degrees and the distal phalanges close to 45 degrees; for the ‘open’ hand posture, the pitch angles of all four fingers are close to 0 degree and the yaw angles between the four figures are at least 7 degrees; and for the ‘claw’ hand posture, the pitch angles are around the middle values between zero degree and their maximum values.



(a)

```
5; 0; 0; 2; 0; 0; 4; 0; 0; 2; 0; 0; 1; 1; 0
```

(b)

```
80; 89; 44; 85; 89; 44; 84; 89; 43; 88; 88; 43; 0; 0; 0
```

(c)

```
-13; 5; 1; -26; 7; 0; -20; 7; 2; -4; 5; 3; 14; 5; 10
```

(d)

```
-10; 65; 17; -31; 73; 10; 4; 50; 19; 14; 47; 23; 11; 1; 7
```

Figure 3-8. Joint angles for hand postures of (a) homing, (b) fist, (c) open, and (d) Claw, with the first 4x3 numbers on the data bar showing the pitch angles of each phalanx (starting from the proximal phalanx) on each finger (starting from the index finger) and the last three numbers showing the yaw angles between four fingers.

3.3 INTERSENSE TRACKING SYSTEM

3.3.1 Introduction of InterSense System

Since the absolute hand position and orientation data in 3D space are not provided by the ShapeHand data gloves, sensors are therefore required to track hand wrists position. Available commercialised tracking sensors include magnetic sensors, optical sensors, acoustic sensors and mechanical sensors, and each of them comes with certain problems, e.g., susceptibility to interference, line-of-sight restrictions, jitter, latency, small range and high cost [212-214]. A better alternative is the inertial based tracking sensor which utilises the gyroscopes and accelerometers to measure the rotation and rate of acceleration, respectively, although it is unable to provide the global position and orientation tracking and introduced the tracking drifting problem. This sensor technology offers several potential advantages which include [212, 215]:

- Immunity to all forms of interference;
- No line-of-sight problem;
- Good resolution/negligible jitter over entire range;
- Low latency; and
- Unrestricted range.

One of the commercially available inertial sensors is the InertiaCube, which was developed by InterSense [204]. The InertiaCube is a monolithic part based on the MEMS (micro-electro-mechanical systems) technology involving no spinning wheels which might generate noise, inertial force and mechanical failures [216]. A typical InertiaCube is capable of simultaneously measuring nine physical properties, namely, angular rates measured by gyroscopes, linear accelerations measured by accelerometers and magnetic field components measured by compass, along all three axes. Furthermore, micro-miniature vibrating elements are normally employed to measure all the angular rate components and linear accelerations, with integral electronics and solid-state magnetometers. However, for a hybrid inertial tracker, which may be coupled with other alternative external sensing techniques, such as ultrasonic or optical technique, it does not necessary contain the magnetic measurement resulting in only six physical properties measured. A 6-DOF InertiaCube and its principal functional structure are illustrated in Fig. 3-9, where a gyroscope and a linear accelerometer are attached to each orthogonal axis of the sensor body.

For the gyroscope, Fig. 3-10 illustrates its underlying conventional physical principle, where the tines of the tuning fork may be driven by an electrostatic, electromagnetic or piezoelectric force to oscillate in the plane of the fork. When the entire fork is rotated about its axis, the tines experience a Coriolis force, $F = \omega \times v$, resulting in the tines vibrating perpendicularly to the plane of the fork. The amplitude of this out-of-plane vibration is proportional to the input angular rate, and subsequently sensed by capacitive or inductive or piezoelectric means to measure the angular rate [217].

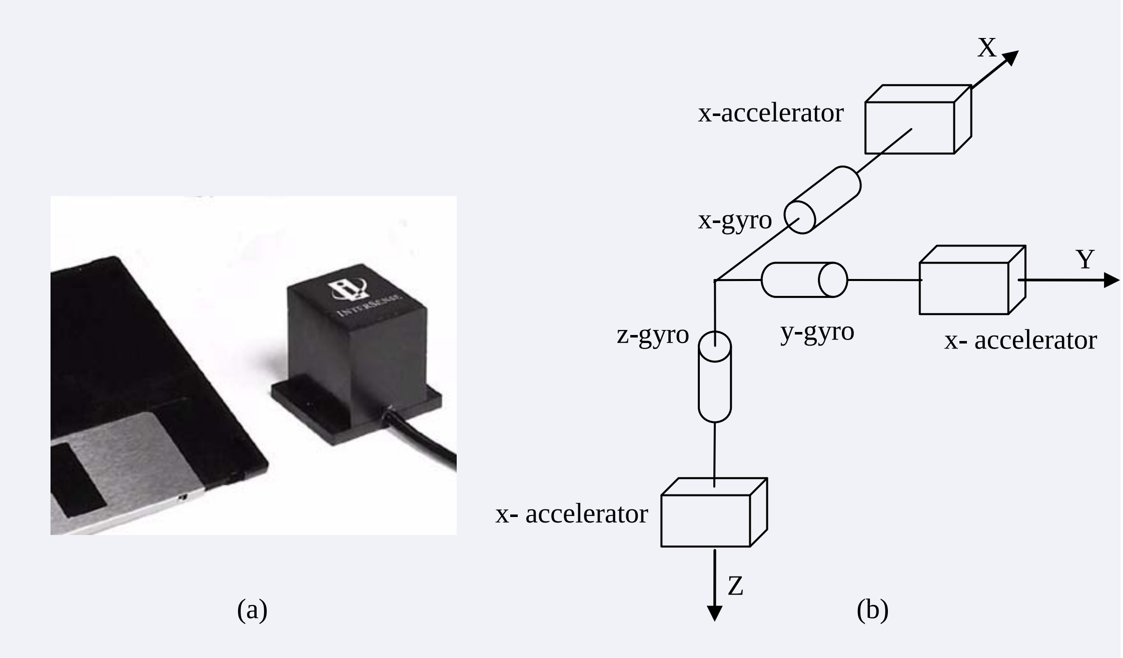


Figure 3-9. (a) InertiaCube and (b) its principal functional structure (modified from [218]).

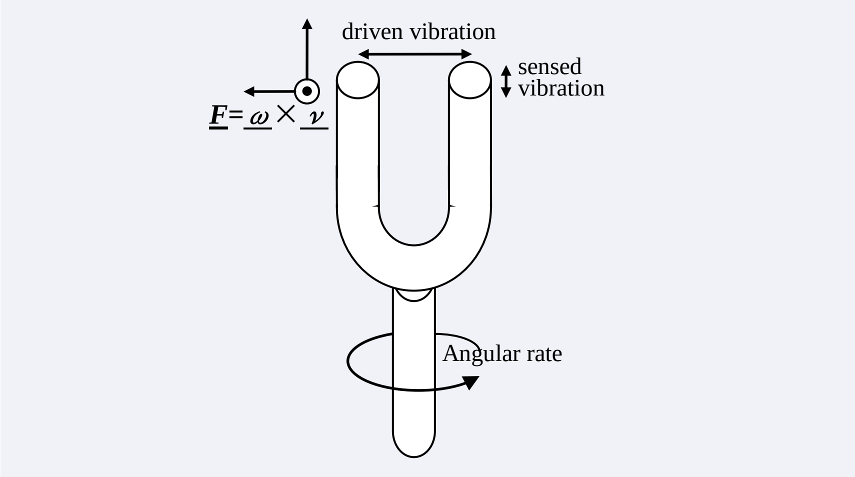


Figure 3-10. Basic physical principle of a gyroscope (modified from [217]).

The processing flowchart of the InertiaCube is illustrated in Fig. 3-11. Through the Coriolis force *F*, the gyroscope determining the instantaneous orientation of the body. Together with the local body acceleration measured by the accelerometers, the acceleration in the earth coordinates, *a*, can be determined after the coordinate transformation. Followed by the subtraction of the gravitational acceleration to cancel the gravity effect to yield $a^{'}$, the relative position of the InertiaCube in 3D is obtained by performing the integration of $a^{'}$ twice [218].

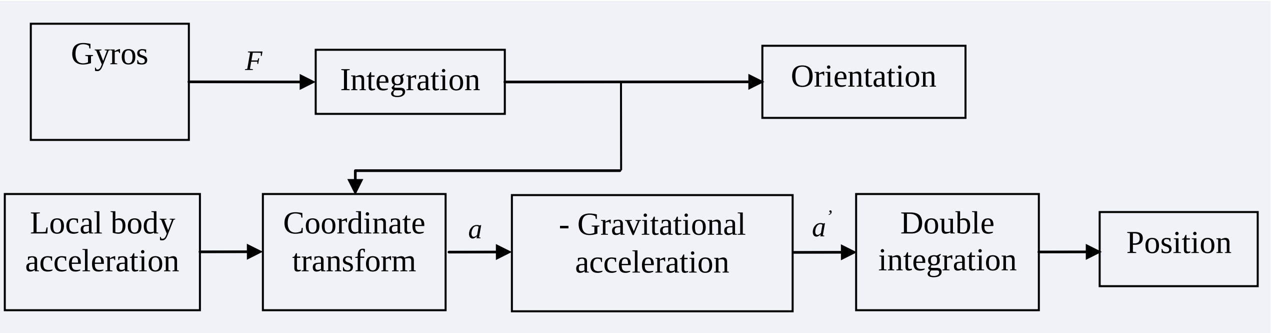


Figure 3-11. InertiaCube processing flowchart (modified from [218]).

Since the InertiaCube sensor can compute only the relative rotation and displacement information with respect to its initial position, a reference frame is required to obtain the absolute 6-DOF information. Additionally, the computation of the linear displacements, which is based on the double integration of the acceleration data, is inherently sensitive to any minor offset in the measured signals. Together with the gyroscope biases that are the undesired output produced by a gyroscope at rest, these lead to a significant drift in the computed positions and orientations. One method to overcome these problems is to periodically combine information from an additional position sensor, such as the ultrasonic or optical sensor, to establish the sensor’s initial position and orientation as well as to correct the drifting error. The ultrasonic sensor is a good choice to maintain the best features of both types of sensors, since the low-accuracy real-time measurements from the InertiaCube sensor may be coupled with a high-accuracy but relatively slow or intermittent measurements to reduce its drifting error.

Currently, the InterSense Company produces an IS-900 sensor family which integrates the InertiaCube sensor technology with the ultrasonic sensing technology. For example, for the MiniTrax Hand Tracker shown in Fig. 3-12a, the angular rates measured by the gyroscope are used to obtain its relative or local orientation (yaw, pitch, and roll), and the linear accelerations measured by the accelerometer are transformed into a reference coordinate frame and double integrated to keep tracking of the changes in its local position (*x*, *y*, and *z*). Furthermore, the absolute position of the hand tracker is computed from the ultrasonic range measurements which are made with respect to an array of the SoniDiscs (SoniStrips) positioned over the required tracking area (see Fig. 3-12b). A command from the IS-900 Processor triggers a SoniDisc transmitter in the SoniStrips to send a 40k Hz ultrasonic pulse at a time with its unique trigger code. When the ultrasonic pulse and its trigger code is received by the microphones mounted on the tracker (see Fig. 3-12a), the time-of-flight measurement will be obtained. Thus, according to the speed of sound (which is calculated based on the measured ambient temperature), the range measurements of the tracker can be obtained. These measurements are then fed into an advanced Kalman filter to adjust the position predicted by the tracker's inertial sensor and to prevent the position and orientation from drifting.

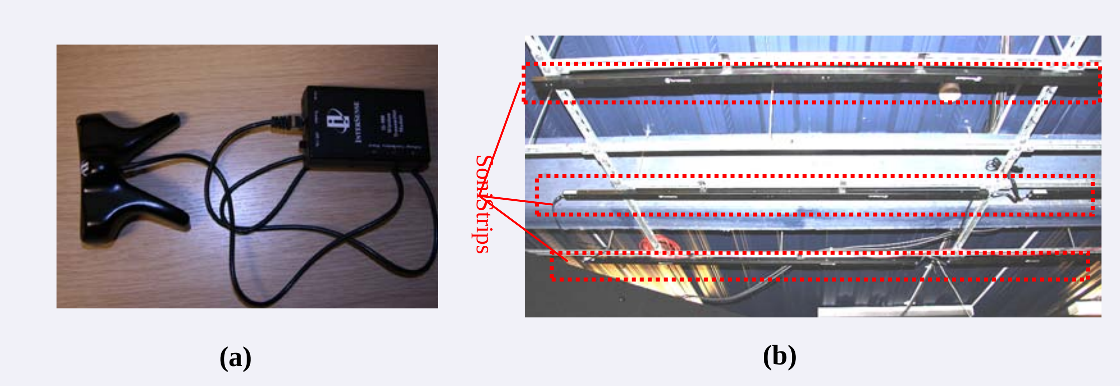


Figure 3-12. (a) IS-900 MiniTrax Hand Tracker and (b) SoniStrips.

To provide a comparison with other tracking sensors, some critical specifications of various position tracking sensors are listed in Table 3.2 [219-223]. From Table 3.2, the InterSense ultrasonic-inertial tracking sensor is seen to have advantages in terms of resolution, accuracy, update frequency, latency, tracking range and cost, despite of its inherent jitter/drift deficiency. For the work presented in this thesis, a pair of InterSense IS-900 MiniTrax Hand Tracker devices, which uses the combined ultrasonic and inertial sensing technique, is used to provide the required position and orientation data of both hand wrists. Furthermore, another IS-900 MiniTrax Head tracker device is used to provide the head position and orientation data in order to generate a correct view to the user.

Table 3.2: Specification comparison among different tracking sensors [219-223]

	Magnetic	Ultrasonic	Optical	Inertial	Mechanical	InterSense (Ultra-Inertial)
DOF	6	6	3/6	3/6	6	6
Resolution: Position	0.5mm (30.5cm) ±50cm (3m)	0.5cm	0.2-7mm	N.A.	0.5mm	0.75 mm (1.5mm wireless)
Resolution: Orientation	0.1^0 (30.5cm) ±17^0 (3m)	5^0	0.01^0	0.02^0	0.1^0	0.05º (0.10º Wireless)
Accuracy: Position	1.8mm	5cm	0.4mm	N.A.	1mm	2.0-3.0 mm (3.0-5.0mm Wireless)
Accuracy: Orientation	0.5^0	5^0	0.02^0	3^0	0.5^0	0.25º-0.5º (0.5º-1ºWireless)
Update Frequency	120Hz	20-50Hz	60-600Hz	1800-500Hz	70Hz	180Hz (120 Hz Wireless)
Latency	4-20ms	60ms	1-60ms	2ms	1ms	4ms
Jitter	low	high	low-medium	low	low-medium	medium
Range	3m	10m	0.5-3m	360^0 all axes	1-2m	4-20m^2
Drift	medium	medium	low	high	low-medium	medium
Cost	£9780	£815	£1.6k-81.5k	£1600	£2425-155k	£1,805

3.3.2 InterSense System Setup

The setup of the InterSense tracking system is illustrated in Fig. 3-13. Similar to the ShapeHand system, the system is also an integration of three sub-modules driven by a computer. The sub-modules are the input module from InterSense tracker devices, the data communication module, and the data output module.

For the input module, the IS-900 SoniStrips containing ultrasonic emitters are mounted on the ceiling, which transmit ultrasonic pulse upon receiving address signals from the IS-900 processor. The InterSense tracker devices, e.g., the MiniTrax hand tracker and the MiniTrax head tracker, perform time-of-flight range measurement based on the ultrasonic pulse received through the URM (Ultrasonic Receiving Unit), and combines them with its relative position and orientation data measured by the internal IMU (Inertial Micro Unit) to provide its absolute global position data.

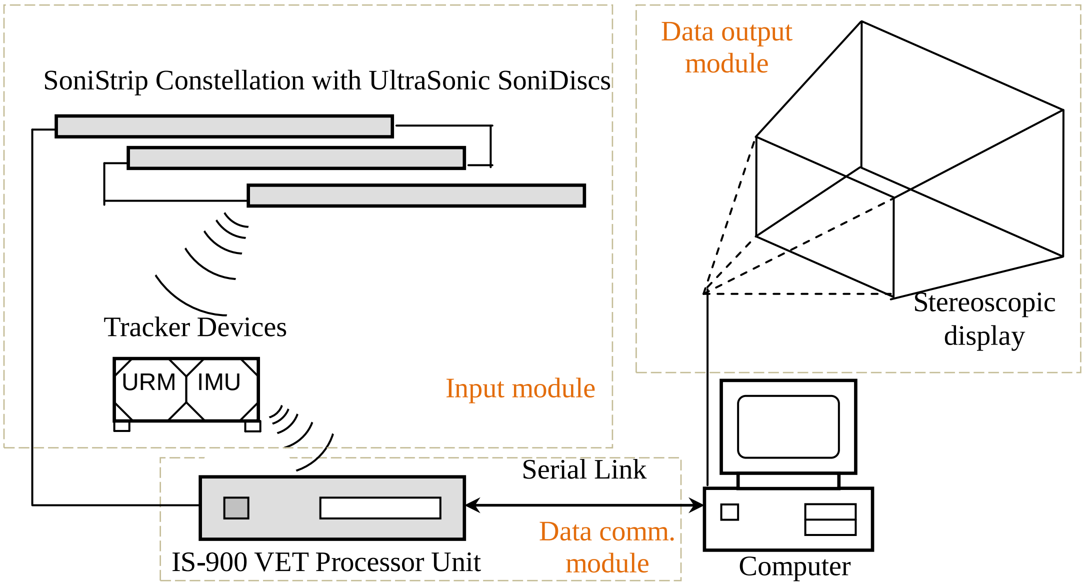


Figure 3-13. InterSense system setup.

For the data communication module, each InterSense tracker device transmits its position and orientation data to the corresponding MiniTrax receiver which is connected to the IS-900 processor unit, wirelessly. Through a serial link, the data collected by the IS-900 processor unit are subsequently transferred to the computer for processing and to the data output module after processing.

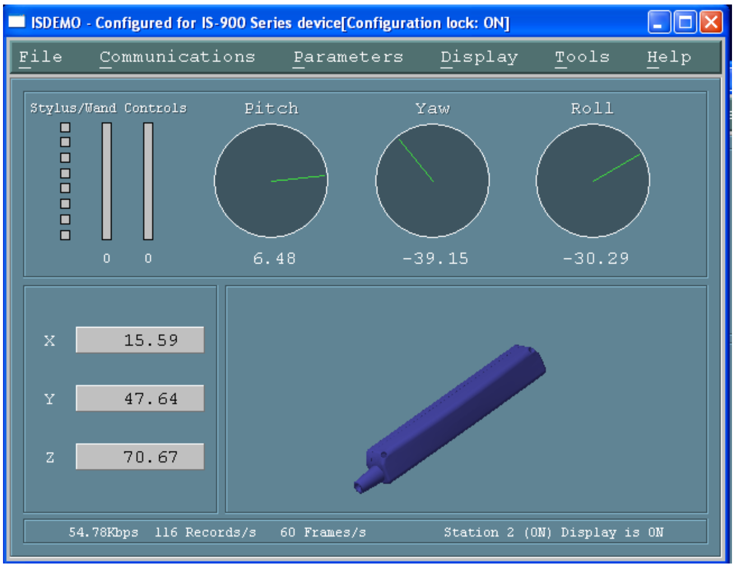
3.3.3 InterSense System Performance Evaluation

The IS-900 VETracker Processor of the InterSense tracking system used in this project is capable of tracking eight trackers at once. The InterSense Company supplies an interface that is able to provide either a graphic data output or a ***'.text'*** file output, which contains the position and orientation information acquired from the tracked devices. Some software routines with the corresponding commands for the InterSense tracking system are listed in Table 3.3, which perform localisation of the tracker device, data acquisition, and application termination, respectively.

A graphic data output is illustrated in Fig. 3-14 as an example to show the user holding, translating and rotating a MiniTrax head tracker. As shown in Fig. 3-14, the *x*, *y*, and *z* position data of the tracker are displayed at the bottom left panel in the numerical form, with the orientation information, which are the *Pitch*, *Yaw* and *Roll*, displaying in the top panel in both numerical and meter forms. Moreover, the bottom right panel contains a graphic display showing the tracker movement and the tracking rates (54.78 kbps for the data transfer rate, 116 records/s for tracking, and 60 frames/s for the updating frequency), in real-time.

Table 3.3: API commands for InterSense system

Commands	Functions
ISD_OpenTracker()	Locate the InterSense tracker to the specified serial port.
ISD_GetData()	Acquire data from InterSense tracing devices
ISD_CloseTracker()	Terminate application and free resources.



Pitch 6.48

Yaw -39.15

Roll -30.29

X 15.59

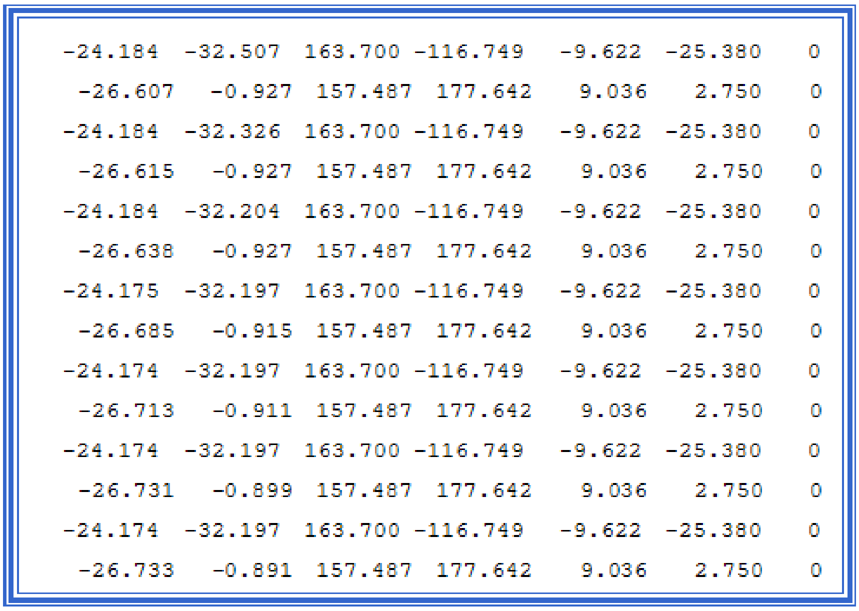
Y 47.64

Z 70.67

54.78Kbps 116 Records/s 60 Frames/s Station 2 (ON) Display is ON

Figure 3-14. Graphic data output from InterSense.

An example of the '***.txt***' file output from the InterSense system is given in Fig. 3-15. As shown in the figure, the first three numbers in each line correspond to *x*, *y* and *z* position data, respectively, and the following three numbers correspond to *roll*, *pitch*, and *yaw*, respectively. The number *0* at the end of each line indicates the tracking station number, which is the first tracker at the time.



```
-24.184  -32.507  163.700 -116.749   -9.622  -25.380    0
  -26.607   -0.927  157.487  177.642    9.036    2.750    0
-24.184  -32.326  163.700 -116.749   -9.622  -25.380    0
  -26.615   -0.927  157.487  177.642    9.036    2.750    0
-24.184  -32.204  163.700 -116.749   -9.622  -25.380    0
  -26.638   -0.927  157.487  177.642    9.036    2.750    0
-24.175  -32.197  163.700 -116.749   -9.622  -25.380    0
  -26.685   -0.915  157.487  177.642    9.036    2.750    0
-24.174  -32.197  163.700 -116.749   -9.622  -25.380    0
  -26.713   -0.911  157.487  177.642    9.036    2.750    0
-24.174  -32.197  163.700 -116.749   -9.622  -25.380    0
  -26.731   -0.899  157.487  177.642    9.036    2.750    0
-24.174  -32.197  163.700 -116.749   -9.622  -25.380    0
  -26.733   -0.891  157.487  177.642    9.036    2.750    0
```

Figure 3-15. '***.txt***' file output example of the InterSense system.

Testing has also been conducted to measure the tracking coverage limitation of the InterSense system. By moving the InterSense tracker up and down, as well as close to and away from the SoniStrip constellation, the tracking quality rate indicating the tracking quality of the trackers is shown to have the highest value, that is approximating to its full tracking quality value of 15 when it is placed no lower than 1.5m under the constellation; and the tracking rate drops down to 10 when the tracker is placed on the floor which is 2.84m below the constellation. Furthermore, if the tracker is placed 1m away from the constellation's direct coverage volume, the tracking rate drops down to 5.

To confirm the adequacy of the InterSense speed performance, a test has also been conducted to check its stability for speedy movement. A hand model has been built through OpenGL programming with the InterSense position data providing the hand position, and a video camera is used to record the movements of both the user's hand and the virtual hand model on the screen. When the user moves the hand tracker at its maximum speed of approximate 2 m/s, the hand model shown on the screen was seen to be able to follow the tracker's movement closely. However, the hand model was seen to drift away if the microphone in the hand tracker is covered by the hand causing interference to signal transmission.

According to its performance, the InterSense tracking system is shown to be able to provide the 6-DOF data, with a good tracking rate of 116 records per second. Together with the specifications presented in previous section as well as the tests of its tracking coverage limitation and speed adequacy, the InterSense tracking system is demonstrated to be suitable for the tracking purpose in this project.

3.4 STEREOSCOPIC DISPLAY SYSTEM

3.4.1 Introduction of Stereoscopic Display System

Stereoscopy is a technique of creating a visual perception of depth to the user based on two slightly different projections of a scene onto the two eyes, and it has been accepted that the stereoscopic displays can provide many benefits to a user, examples include [224]:

- Perceiving depth of the displayed surface;
- Gaining the spatial localisation;
- Allowing concentration on different depth planes;
- Perception of structure in visually complex scenes;
- Improving perception of surface curvature and material types; and
- Improving motion judgment.

While many stereoscopic display products based on different techniques are now readily available in the marketplace, such as the polarised projection method, time-sequential method, Lenticular method and anaglyph method, the underlying technique is to present each of the person’s eyes with an image projected from a different perspective by coding and decoding the multiple stereoscopic views in the same light field through colour, polarisation, time, and/or spatial separation[224-228]. In this project, a large stereoscopic display using a polarised projection method has been adopted for visual output display.

The stereoscopy based on polarised projection normally employs two video projectors (see Fig. 3-16) to project two optically overlaid images on to a single screen with a slight disparity. Additionally, for each projection, it is polarised along a different direction through a polarizer before it is projected on to the screen. With the viewer wearing a pair of polarised glasses (see Fig. 3-17), it results in each eye seeing one projection only through the screen, and the disparity between the two projections creates an illusion of depth effect of the image. Two common types of polarisations are the linear and circular polarisations. The former polarises the input light linearly, e.g., using the wire-grid polariser to perpendicularly filter the incident beam to allow only the vertical or horizontal beam components to pass through. The latter creates circularly polarised light by selectively absorbing or passing clockwise and counter-clockwise circularly polarised light through a polariser. Compared to the former, the latter has an advantage of no false polarisation when the user viewing the projection image by rotating the view angle in the vertical direction and is used in the project. Furthermore, the stereoscopic display used in this project is using back projection instead of front projection to avoid the projection shadow caused by a user standing between the projector and the screen.

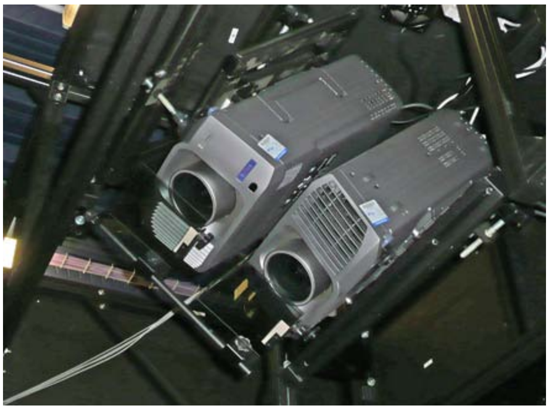


Figure 3-16. Epson PowerLite8800 projectors.

Figure 3-17. Polarised glasses.

3.4.2 Stereoscopic Display System Setup

The stereoscopic display used in the project has a size of 2.74 m x 2.06 m, with two back projectors operating in passive circular polarisation mode. These projectors are connected to the computer through two dual DVI (Digital Visual Interface) graphics card output ports, where the graphic card is the NVIDIA Quadro FX with 256MB memory.

The creation of a stereoscopic scene can be implemented through OpenGL programming with the hardware support of quad-buffered stereo rendering. Quad-buffering provides the left/right and front/back buffers for image storage. During image display execution, while the left-eye-image projector fetches the image from the left-front buffer and the right-eye-image projector fetches the identical image with a slight distance disparity from the right-front buffer, the back buffers update and store the next rendering images ready for display according to the displaying sequence in order to obtain a smooth display. To generate a correct distance disparity for the stereoscopic image, there are two geometry algorithms, namely, toed-in stereo and asymmetric frustum parallel axis projection stereo.

Although the toed-in stereo is quicker and easier compared to the latter, it has the side effect that the distortion exists in both left and right views due to the difference between the rendering and viewing planes [229, 230]. The asymmetric frustum parallel projection is able to correct this distortion or parallax and to put the rendering plane and viewing plane in the same orientation. As a result, the asymmetric frustum parallel axis projection stereo has been adopted in this project.

The illustration of the asymmetric frustum parallel axis projection stereo is shown in Fig. 3-18. With the view positions of two eyes parallel to each other and their view areas at the rendering plane at the focal point overlapped, two asymmetric views are produced. Furthermore, with the knowledge of the eye position and the projection area at the focal point, the visible region which is called the frustum in OpenGL can be computed, wherein the objects are seeable to the user. For OpenGL implementation, the computation of the frustum is based on the setting of the focal point, f_d, the visible area on the rendering plane, s, and the separation distance between two eyes, e, as shown in Fig. 3-18.

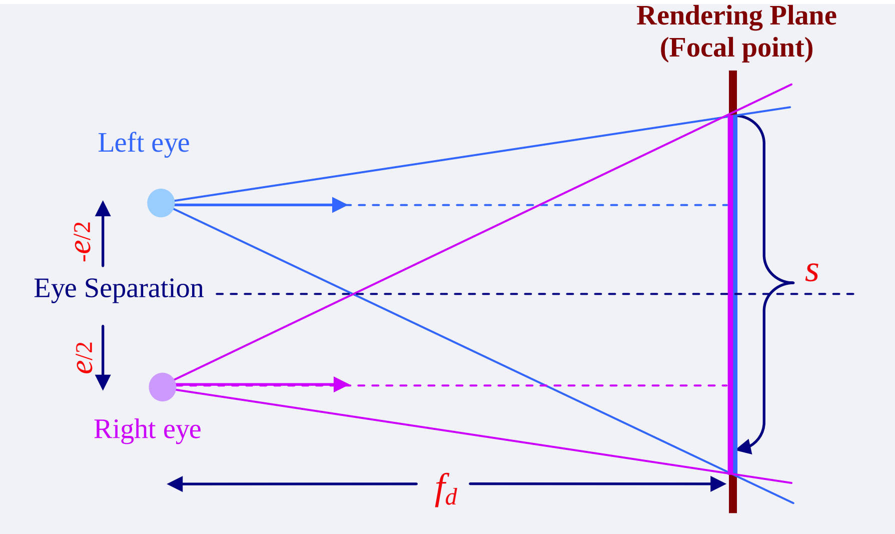


Figure 3-18. Asymmetric frustum parallel axis projection (modified from[230]).

Some key instructions to set up the stereoscopic view in OpenGL using the asymmetric frustum parallel axis projection stereo method are listed in Table 3.4.

Table 3.4: Commands for stereoscopic view

Commands	Functions
Aspect = W_screen/H_screen	Calculate the screen aspect ratio
glDrawBuffer(GL_BACK_LEFT)	Draw into back left buffer
glFrustum()	Setup the view frustum for the left/right eye
glTranslatef(-IOD/2, 0, 0)	Translate to cancel parallax for the left eye
Drawscene()	Draw the scene for rendering
glDrawBuffer(GL_BACK_RIGHT)	Draw into back right buffer
glTranslatef(IOD/2, 0, 0)	Translate to cancel parallax for the right eye
glutSwapBuffers()	Swap the front and back buffer

3.4.3 Stereoscopic Display System Performance Evaluation

To evaluate the stereoscopic effect of the system, a scene contains a graphics based cube has been created using the OpenGL programming through the computer, where the cube surfaces are generated using the quadrilateral mesh method, and is rendered onto the large stereoscopic screen for display.

For the display generated as shown in Fig. 3-19, without wearing the 3D glasses, the user sees a blurred scene of a virtual cube formed by two identical images with slightly disparity. By wearing the 3D glasses, the user sees only one cube image in each eye resulting in a perception of a 3D virtual cube floating in the scene with depth effect. Rotating the cube with the mouse and keyboard results in different sides of the cube being displayed to the user, whilst dragging the cube nearer results in it appearing larger to the user and dragging away results in the cube appearing smaller. Furthermore, if the cube is dragged beyond the left or right edge, or, too near or too far with respect to the screen, the cube will be partly visible or disappeared, for it is outside the view frustum. By standing in the middle of the large screen, the user has the sensation of being immersed in the scene.

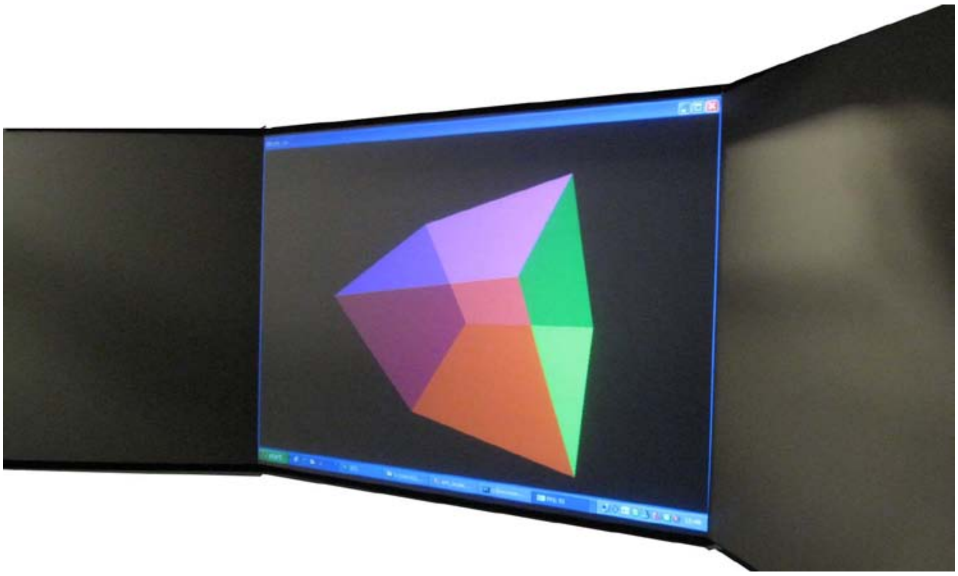


Figure 3-19. Stereoscopic display of a virtual cube.

3.5 HARDWARE INTEGRATION

The demonstration system developed for real-time HCI based on tracking and recognition of dynamic hand gestures is illustrated in Fig. 3-20 in a schematic diagram form. Driven by a desktop computer, the system is an integration of three subsystems, namely, ShapeHand data glove system used for gesture data acquisition, InterSense tracking system for head and hand position acquisition and stereoscopic display system for 3D visualisation.

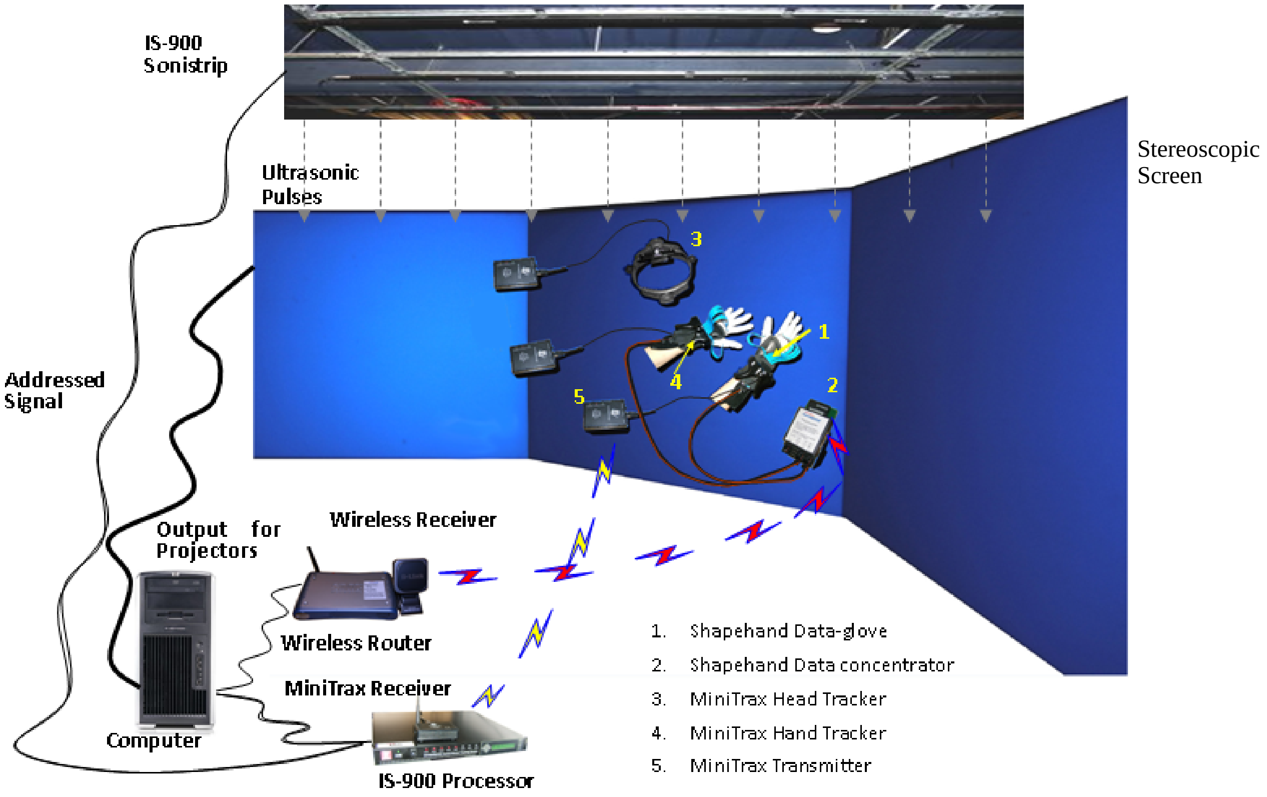


Figure 3-20. System hardware schematic diagram.

For gesture data acquisition, a pair of ShapeHand data gloves is used, which provides the relative position and orientation information of the finger joints and tips with respect to the hand wrist. These data are then transmitted wirelessly by the ShapeHand data concentrator to a wireless receiver/router, which is connected to the Ethernet port of the computer.

Since the absolute hand position and orientation data in 3D space are not provided by ShapeHand data gloves, the IS-900 wireless tracking system from InterSense is used to provide the required hand position data. The system is also used to provide the head position and orientation data in order to generate a correct view. As shown in Fig. 3-20, the IS-900 SoniStrips containing ultrasonic emitters are mounted on the ceiling with the direct coverage of 4.3 m x 1.35 m, which transmit ultrasonic pulses upon receiving addressed signals from the IS-900 processor that is connected to the serial port of the computer. Three MiniTrax tracking devices containing inertial sensors and ultrasonic receivers are used with two attached to the user's wrists and one attached to the user's head. Each MiniTrax tracking device performs time-of-flight range measurement based on the ultrasonic pulses received, and transmits wirelessly its position and orientation data to the corresponding MiniTrax receiver connected to the IS-900 processor.

For the stereoscopic display, the demonstration system uses a large stereoscopic screen with a size of 2.74 m x 2.06 m (shown as the middle screen in Fig. 3-20) placed at 0.98 m facing to the SoniStrips direct coverage area, and two back projectors operating in the passive circular polarisation mode are connected to the computer through two dual DVI graphics card output ports. The 3D objects with depth effect are seen by the user wearing a pair of light-weight polarised glasses.

The computer used in the demonstration system runs on Microsoft Windows XP, and is based on an Intel Xeon 3.06GHz CPU with a 2GB RAM (Random Access Memory) and NVIDIA Quadro FX 3000 Graphics Card with 256MB memory. As shown in Fig. 3-20, the gesture data from ShapeHand are fed to the computer by a wireless connection via a data concentrator, the position and orientation data from InterSense trackers are fed to the computer via a wireless tracking control unit (IS-900 Processor), and the computer outputs the processed data for stereoscopic display through a serial cable.

3.6 SOFTWARE DEVELOPMENT

3.6.1 Software Development Framework

The system software is implemented as a Windows XP based application using C++. To minimise the development time, the software utilises the MFC (Microsoft Foundation Classes) to build the user interface and control units. Fig. 3-21 shows the program initialisation flowchart (with steps labelled) to create the global object '*The App*' through the use of object class '*CAPI_TestApp*' and its parent class '*CWinApp*'. Through '*CAPI_TestApp*', three objects are created by the "*OnFileNew*()" and "*OnFileOpen*()" functions, namely, the Main frame object '*CMainFrame*', View object '*API_TestView'* through its parent class '*CView_OpenGL'*, and Document Object '*API_TestDoc*'. Furthermore, through the function of "*CreateMainFrame*()" in the class '*CView_OpenGL'*, two windows are created, namely, the Main frame window, and the View window. With the '*CMainFrame*' object controlling the output toolbar and menu layout for the Main frame window, a modified version of the standard Single Document View Model has been implemented to allow the input data to update the document object (containing the ShapeHand and InterSense data) [231], and the output display through the View window is treated as an individual "view" of the document object to update and display its contained data.

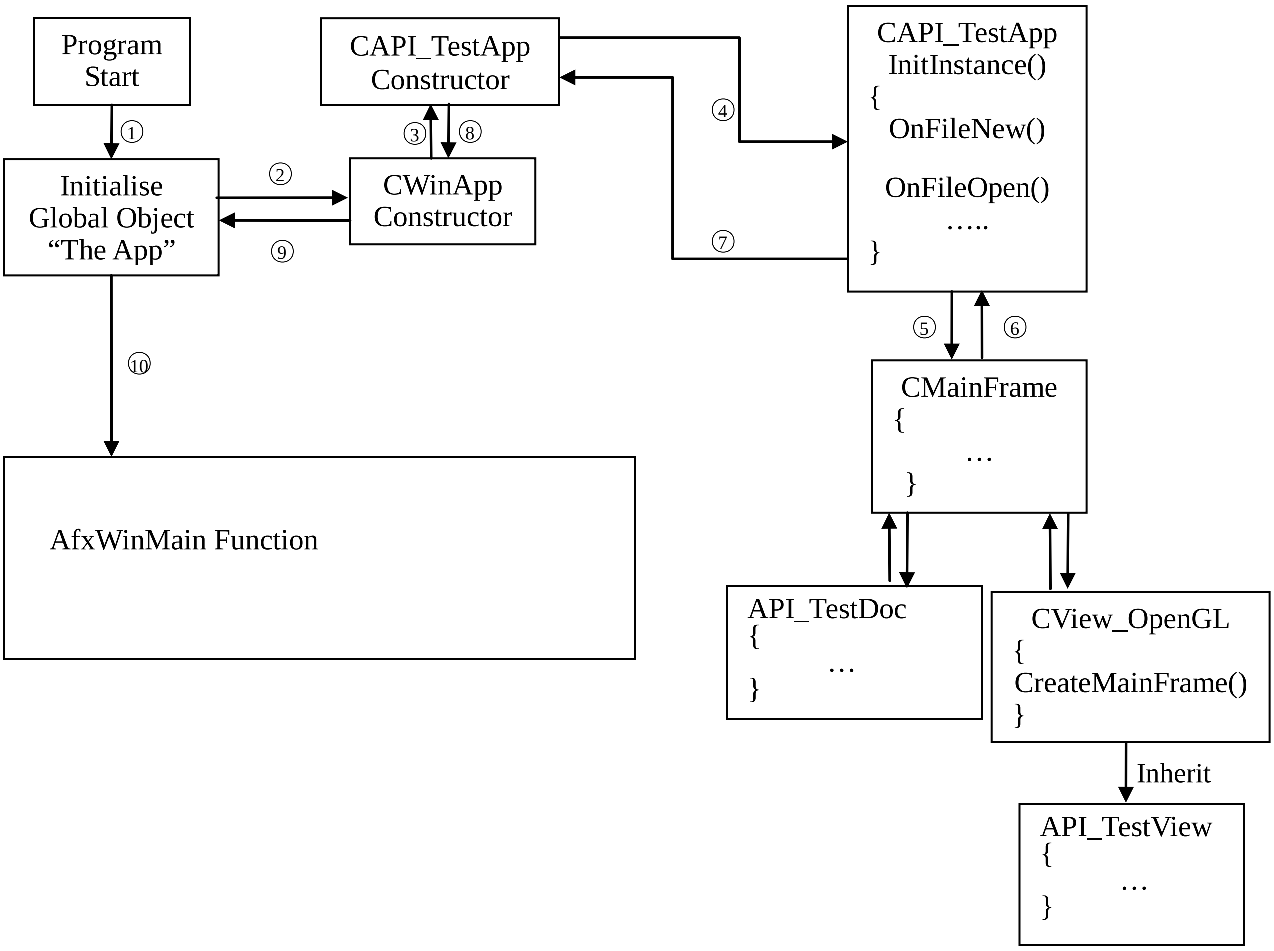


Figure 3-21. System program initialisation flowchart.

After all the initialisation, the program activates the main program through the "*AfxWinMain*()" function, and starts the message cycling by continuously checking any message from other program objects, e.g., the request of sensor data updating from the '*API_TestDoc*' object, through the functions of "*GetMessage*()" and "*PeekMessage*()" (see Fig. 3-22). If there is a message received, it will do the message mapping to generate the corresponding response to the corresponding object for processing; otherwise, the program will stay idle by executing the "*CWinApp::OnIdle*()" function. This process continues until it receives the '*WM_QUIT*' message to ask the program to terminate where the created windows will be destroyed.

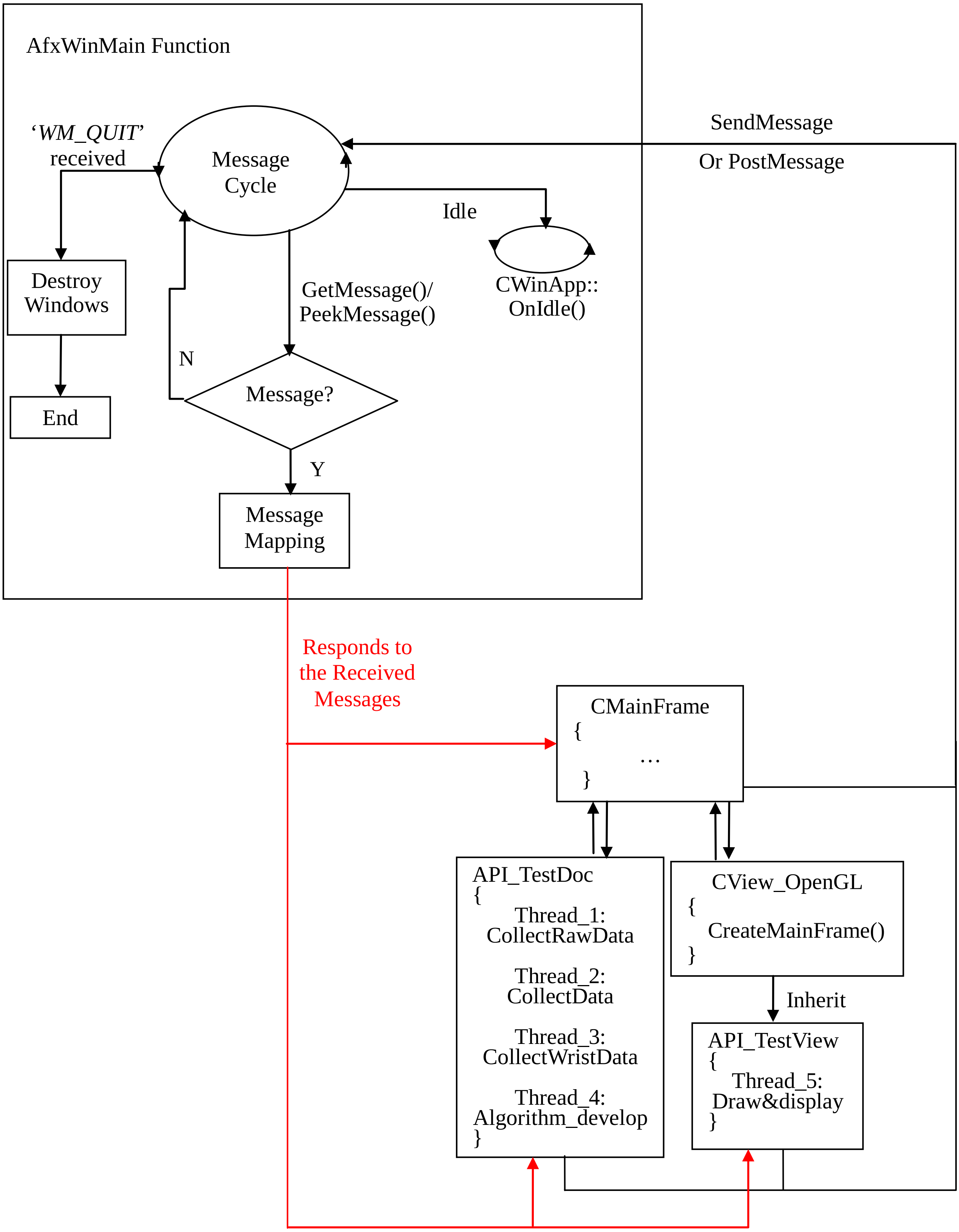

AfxWinMain Function
'WM_QUIT' received
Message Cycle
SendMessage
Or PostMessage
Idle
Destroy Windows
GetMessage()/ PeekMessage()
CWinApp:: OnIdle()
N
End
Message?
Y
Message Mapping
Responds to the Received Messages
CMainFrame { … }
API_TestDoc { Thread_1: CollectRawData Thread_2: CollectData Thread_3: CollectWristData Thread_4: Algorithm_develop }
CView_OpenGL { CreateMainFrame() }
Inherit
API_TestView { Thread_5: Draw&display }

Figure 3-22. System program running flowchart.

Furthermore, the software is implemented following a multi-thread approach to minimise the response time for HCI. There are five parallel program threads with two of them performing hand gesture data acquisition (denoted as *CollectRawData* in Fig. 3-22) and extraction from ShapeHand (denoted as *CollectData* in Fig. 3-22), and the other three performing position and orientation data acquisition from InterSense (denoted as *CollectWristData*in in Fig. 3-22), dynamic hand gesture recognition (denoted as *Algorithm_develop* in Fig. 3-22), and stereoscopic display (denoted as *Draw&display* in Fig. 3-22), respectively. The first four threads are programmed in the Document object '*API_TestDoc*' and the last one in the View object '*API_TestView*'. Each thread is assigned with a short amount of CPU time, the time slots, to run, and the scheduling of the threads is done in a round-robin manner with all threads having the same priority. Since the time slot is so short, all threads will appear as running simultaneously, thereby increasing the system's efficiency greatly [232]. On the other hand, since all threads have the right to access the common resources, a potential risk is introduced that many threads may access a resource at the same time. For example, an unwanted case would be a thread is about to use a number for calculation, but its assigned time slot is finished and it forced give up the execution right to the next thread to run, whereby when the first thread accesses the number again, it may received unwanted result. Therefore, the access-control among these threads is required. MFC provides the '*CSingleLock*' class for such purpose. Basically, this class provides the protection to a common-access resource by halting its usage to other threads when it is being accessed by one thread, thereby enabling the threads to run under the multi-thread mode without collision. Moreover, whilst executing the position data acquisition thread, no change of the received data during its allocated time slice will result in early switching to the next program thread. Some basic commands for the multi-thread access-control are listed in Table 3.5.

Table 3.5: Commands for multi-thread access control

Commands	Functions
CSingleLock sL(m_drawmutex);	Create a class object for multi-threaded access control
sL.Lock();	Protect the selected resources
sL.Unlock();	Give up the resource control right

The two program threads for acquisition and extraction of hand gesture data are implemented based on the ShapeHand API. The steps include initialisation of data collection, receiving data and checking the received data. The data acquisition program thread obtains raw data from a pair of wireless ShapeHand data glove via the Ethernet port of the computer. With the raw data obtained, the required positions and orientations of a digit joint are determined through the data extraction program thread. Main commands for these two threads are listed in Table 3.6.

Table 3.6: Commands for hand gesture data acquisition and extraction threads

Commands	Functions
ReceiveSampleBytes();	Sampling data from the ShapeHand tape sensors
ParseNextSample();	Parse the received data
setBuffer ();	Acquire the hand data and store them in the buffer
GetHandData()	Extract the hand data and rearrange them according to the hand data format

The program thread of position data acquisition is implemented based on the InterSense API. It performs data acquisition from the three MiniTrax tracking devices attached to the head and two wrists of the user via the serial port of the computer to gather their position and orientation data. Steps in this thread include data collection, and data updating if the received data are found to be different from the previously one. Main commands for this thread are listed in Table 3.7.

Table 3.7: Commands for head/hand position/orientation data acquisition thread

Commands	Functions
OpenTraker();	Connect to the InterSense Trackers
Get_Intersense_Data()	Collect data from the InterSense Trackers.
SetHeadPosition();	Set the InterSense Head tracker position as the user’s head position
SetWristPosition();	Set the InterSense Wrist tracker positions as the user’s wrist positions
CloseTraker();	Terminate the connection to the InterSense

The program thread of dynamic hand gesture recognition is based on the algorithms presented in the next section and the following chapters. Essentially, it involves data merging through coordinate transformations among different component coordinate systems (see Section 3.6.2), tracking and recognising a number of pre-specified hand gestures for the manipulation of displayed virtual objects (see Chapter 4), dynamic hand gestures recognition for DSW (see Chapter 5), as well as the recognition of a number of hand gestures for British Sign Language (BSL) fingerspelling (see Appendix B).

The final program thread of stereoscopic display is implemented based on OpenGL programming to provide 3D visual feedback to the user during the interaction. As discussed in Section 3.4.2, this is done by generating two identical views of the virtual objects and two hands with a slight disparity. As viewing through polarisation glasses results in each eye seeing only the view generated for it, it creates a visual immersion with depth impression. Steps in this program thread include the use of the head position data acquired to specify the viewing position and direction of the left and right eyes, configuration of the viewing frustum for each eye, stereo rendering to draw the left and right images of the 3D objects and hand models as well as the dynamic hand gesture recognition results for DSW and BSL fingerspelling by perspective projection.

3.6.2 Data Integration through Coordinate Transformation

Since different equipments in the proposed system using different coordinate systems for data acquisition, processing and display, coordinate transformations are required to bring different data sets into a unified coordinate system.

Fig. 3-23 illustrates the spatial relationships between different coordinate systems. The world coordinate system is defined to have the same orientation as the stereoscopic display. With the x-axis (denoted by X_w) pointing towards the right, the y-axis (denoted by Y_w) pointing upwards, and the z-axis (denoted by Z_w) pointing towards the viewer, this forms a right-handed coordinate system (see Fig. 3-23) with a positive rotation about the axis in the anticlockwise direction. Furthermore, the origin of the world coordinate system (denoted by O_w) is located at the middle of the screen along the x-axis (1.37 m away from the screen right edge), 1 m above the floor along the y-axis, and 1.9 m in front of the screen along the z-axis.

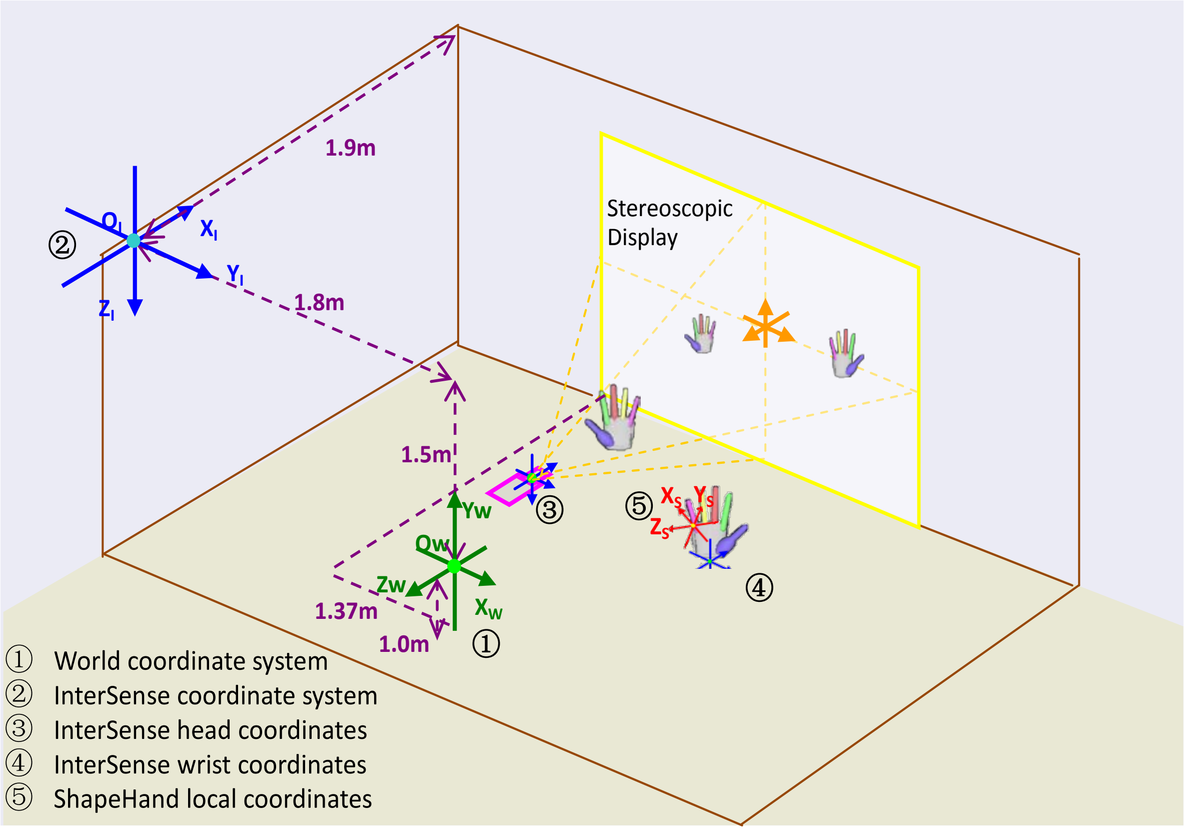

1.9m
②
O_I
X_I
Y_I
Z_I
1.8m
Stereoscopic Display
1.5m
Yw
③
Ow
Zw
Xw
1.37m
1.0m
①
⑤
X_S
Y_S
Z_S
④
① World coordinate system
② InterSense coordinate system
③ InterSense head coordinates
④ InterSense wrist coordinates
⑤ ShapeHand local coordinates

Figure 3-23. Coordinate systems.

For the InterSense system with its coordinate axes denoted by (X_I, Y_I, Z_I), the position data acquired for head and wrists are calibrated with respect to its origin denoted by O_I at (−1.8 m, 1.5 m, 0 m) in the world coordinate system as shown in Fig.3-23. Two rotation operations are required to align the orientations of the InterSense coordinate system with the orientations of the world coordinate system, namely, rotation of -90^{o} about the InterSense y-axis to make the new InterSense x-axis parallel to the world coordinate x-axis, and rotation of 90^{o} about the new InterSense x-axis to make the new InterSense y and z axes parallel to the world coordinate systems. If $\boldsymbol{i} = [x_i, y_i, z_i, 1]'$ denotes the homogeneous coordinates of a position in the InterSense coordinate system, then its corresponding homogeneous coordinates in the specified world coordinate system denoted by $\boldsymbol{i}_w = [x_{iw}, y_{iw}, z_{iw}, 1]'$ are given by

$$i_w = T^{I \to W} i \tag{3.1}$$

where $\boldsymbol{T}^{I \to W}$ is the matrix for geometric transformation from the InterSense coordinate system to the world coordinate system. Based on the geometric relationship between the two coordinate systems described above, $\boldsymbol{T}^{I\rightarrow W}$ is given by

$$T^{I\rightarrow W} = \begin{bmatrix} 0 & 1 & 0 & -1.8 \\ 0 & 0 & -1 & 1.5 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

For the ShapeHand system, the position data of each finger joint are acquired using the local ShapeTape coordinate system, as discussed in Section 3.2.3. With the wrist position and orientation data provided by the InterSense system, the finger joint position data need to be transformed from their local coordinate system to the InterSense coordinate system first and to the world coordinate system subsequently.

If $\boldsymbol{s} = [x_S, y_S, z_S, 1]'$ denotes the homogeneous coordinates of a position in the local ShapeHand coordinate system, then its corresponding homogeneous coordinates in the specified world coordinate system denoted by $\boldsymbol{s}_w = [x_{sw}, y_{sw}, z_{sw}, 1]'$ are given by

$$s_w = T^{I\rightarrow w} T^{S\rightarrow I} s \tag{3.3}$$

where $\boldsymbol{T}^{S\rightarrow I}$ is the matrix for geometric transformation from the ShapeHand coordinate system to the InterSense coordinate system. If the position and orientation data provided by the InterSense system are denoted by (x_i, y_i, z_i) and $(\alpha_i, \beta_i, \gamma_i)$, then $\boldsymbol{T}^{S\rightarrow I}$ is given by

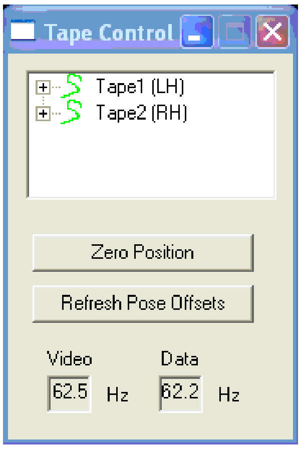
$$T^{S\rightarrow I} = \begin{bmatrix} c_\alpha s_\beta - s_\alpha c_\beta s_\gamma & c_\beta c_i & c_\alpha c_\beta s_\gamma + s_\alpha s_\beta & x_i \\ s_\alpha c_\gamma & -s_\gamma & c_\alpha c_\gamma & y_i \\ -s_\alpha s_\beta s_\gamma - c_\alpha c_\beta & s_\alpha c_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & z_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

where *c* and *s* denote *cos* and *sin* functions, with subscripts denoting the orientation angles from the InterSense system.

Therefore, with the InterSense head and wrist coordinates transformed to the world coordinate system by using equation (3.1), and the ShapeHand coordinates transformed to the world coordinate system by using equation (3.3), all the data can be unified to a common coordinate system.

3.7 SYSTEM PERFORMANCE EVALUATION

With the ShapeHand data glove system providing the hand posture information, three InterSense MiniTrax tracking devices providing the head and hand information, and a large stereoscopic display providing the output display, Fig. 3-24 shows the tracking performance of the system operating under the wireless mode, where the data transfer rate for the ShapeHand system is shown to be approximately 62 Hz, and the InterSense tracking quality shown on the IS-900 VETracker Processor is at least 11 out of the maximum of 15 for the three trackers operating simultaneously. By checking the memory usage (see Fig. 3-25), the PF (Pagefile, also referred to as the Windows swap file) usage is low and stable indicating that the system is under good memory control and has no memory leaking issue. Furthermore, it is noted that the usage of the CPU is being kept at a maximum value. This is due to running of the multi-threading program on a single-CPU processor, where the program keeps on occupying the usage of the CPU without pause.



Video 62.5 Hz

Data 62.2 Hz

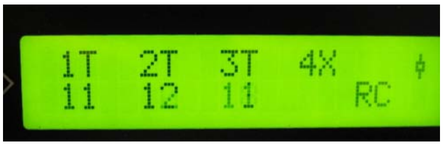
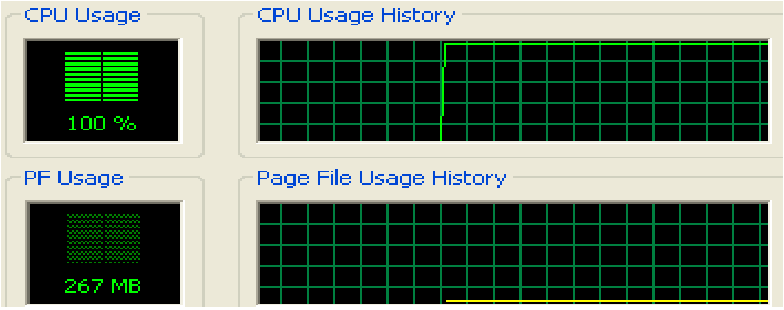


Figure 3-24. Video update rate and data transfer for ShapeHand system, and InterSense tracking rate.



CPU Usage 100 %

PF Usage 267 MB

Figure 3-25. CPU/memory usages for integrated system.

For the output display, the integrated system is able to capture hand gestures and movements made by the user in real-time, and to provide an immediate visual feedback by correctly displaying them based on two 3D hand models as shown in Fig. 3-26. Furthermore, the stereoscopic visual feedback could be seen by the user wearing a pair of light-weight polarised glasses. With the head tracking device providing the position and orientation of the user, the user can physically move around in front of the display screen with an impression of the 3D virtual hand models and a virtual coordinate system floating in space, whereby a forward movement causes the hands and the coordinate system to appear nearer with a bigger size, a backward movement causes them to appear further away and smaller, and a side movement or a side glance enables the user to view different sides of them.

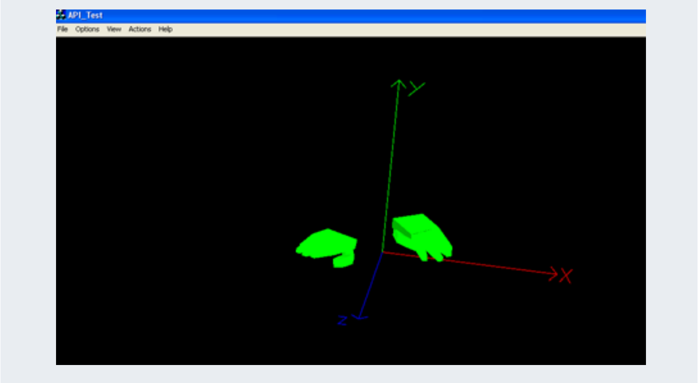
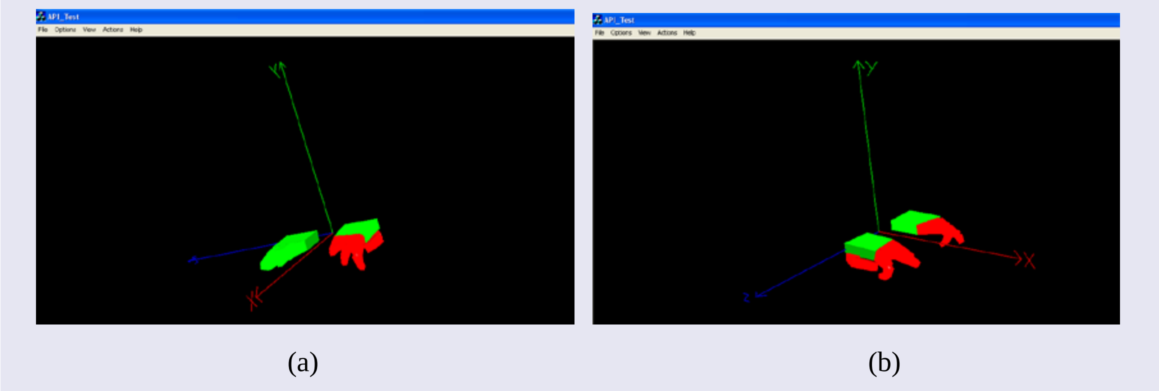


Figure 3-26. Integrated system output.

In order to test the system's usability, recognition of finger bending was implemented with the result shown in Fig. 3-27. By assigning a different colour to the sensed bending finger, the system is demonstrated to be able to recognise the bending gesture made by both hands immediately. This has been done by measuring the bending angle of the proximal phalanx of the index finger with respect to the metacarpal, and a bending angle over 20 degrees will result in the colour changing from green to red.


API_Test
File Options View Actions Help
Y
X
API_Test
File Options View Actions Help
Y
X
Z

(a)

(b)

Figure 3-27. Index finger bending detected (a) on one hand, and (b) on both hands.

To confirm the adequacy of the system speed for dynamic hand gesture recognition, a test was also performed to check the tracking speed against the maximum speed of finger movements. By wearing the ShapeHand data glove with the index finger open and closed repeatedly at its highest possible speed, the virtual hands on the stereoscopic display was found to follow the angular movements of the index finger at its maximum speed around 14 times per second.

Furthermore, to assess the system latency for dynamic hand movement recognition, a high speed video camera was used to record the hand movement made by a user wearing the ShapeHand data glove as well as the movement of the virtual hands appeared on the stereoscopic screen. From the video captured at 64 frames per second, with the hand opening, closing and moving repeatedly, the video analysis of the corresponding hand gestures showed a delay around 6 frames of the virtual hand movement with respect to the real hand movement, which is equivalent to a latency of approximately 94ms.

3.8 CONCLUDING REMARKS

This chapter presents the work of system development for a unique hand based HCI system. The investigation of three sub-systems, namely, ShapeHand system, InterSense system and stereoscopic display system, has been conducted. The specification, principle of working, system setup and system performance evaluation have been presented for each one of them, where significant tests have been done which demonstrates that these systems are well suited for the purpose of this project. Also, the hardware integration of these sub-systems has been described. Driven by a computer, the unique system integration enables acquisition of the hand gesture data through the ShapeHand system, acquisition of head and hand position and orientation data through the InterSense system, and output being displayed through a large stereoscopic screen.

For the software implementation, it is implemented as a Windows XP based application using C++, and the program structure is built as a standard Single Document View Model by using MFC. Following the multi-thread programming, five parallel program threads have been created in this software application, namely, two threads of data acquisition and extraction from ShapeHand, one thread for position and orientation data acquisition from InterSense, one thread for dynamic hand gesture recognition, and one thread for stereoscopic display. These five program threads are executed in parallel and in a round-robin manner with each thread assigned a slice of its CPU time, resulting in minimisation of response time for virtual immersive environment interaction. Furthermore, the data unification in the world coordinate system has been performed.

A significant amount of tests has been done to evaluate the system's performance. Results shown the developed system is able to work correctly under a global coordinate system. Particularly, with the knowable of a normal maximum finger tapping speed is around 7.5 times per second [233], the speed and latency evaluation demonstrates that the system is able to capture the hand gesture change at its maximum speed of approximately 14 times per second, and to capture the hand movement change with a latency of 94 ms.

Chapter 4

VIRTUAL OBJECT MANIPULATION

4.1 INTRODUCTION

To investigate the usability and performance of the system, this chapter reports the development and assessment of the system for virtual object manipulation. Immersive virtual object manipulation in 3D is a fundamental technique for virtual environment interaction [234-236]. A basic sequence can be considered as consisting of object selection at the start, followed by object manipulation which can be a combination of translation, rotation and scaling, and object release at the end [237]. These five basic gestures form the scope of object manipulation in this research. The implementation requires selection of a set of meaningful hand gestures as well as computation of the distances between hands and objects, which will be reported in Sections 4.2 and 4.3, respectively. System performance evaluation will be reported in Section 4.4, and the concluding remarks will be given in Section 4.5.

4.2 HAND GESTURE RECOGNITION FOR VIRTUAL OBJECT MANIPULATION

For natural interaction and user comfort, Fig. 4-1 shows three selected meaningful hand gestures for the five basic object manipulation operations, where the index finger pointing gesture shown in Fig. 4-1a is used for not only the selection of a virtual object but also for the object translation and rotation based on the position of the left or right index fingertip and the hand orientation; the hand open gesture shown in Fig. 4-1b is used for the release of a selected object; and the gesture of two moving hands with the ring and small fingers closed (gun gesture) shown in Fig. 4-1c is for object scaling.

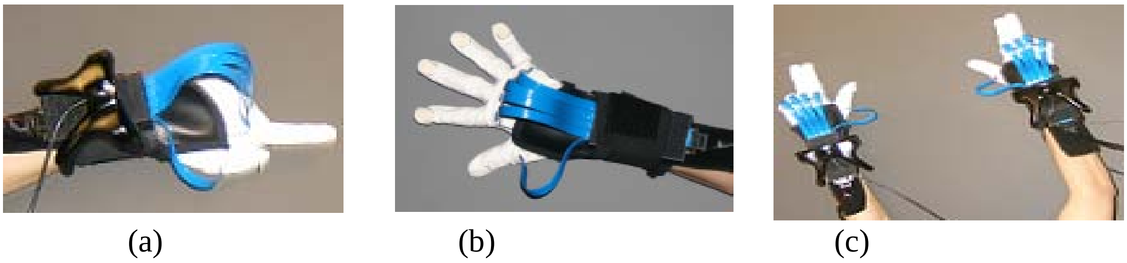

(a) (b) (c)

Figure 4-1. Hand gestures: (a) object selection, translation and rotation; (b) object release; and (c) object scaling.

The three selected gestures shown in Fig. 4-1 are seen to consist of a combination of bending down and extending thumb and fingers in each hand, which can be determined based on the pitch angle of the proximal phalanx of the thumb or finger with respect to the back of the hand. With the finger pitch angle calibrated to around 0° to correspond to a fully extended position (by hand opening) and around 90° to correspond to a fully bending down position (by hand closing), the selected hand gestures can be recognised by expressing them using the corresponding binary state based on a threshold of 45°. Let two hands be denoted by *H* with its binary state set to logic 0 for the left and logic 1 for the right, and let the thumb and four fingers on each hand be denoted by *T*, *I*, *M*, *R*, *S* with the binary state of each one set to logic 1 if its pitch angle, α_s^{pp}, measured by the ShapeHand data glove is less than 45°, and logic 0 otherwise. The index finger pointing gesture is then given by

$$(\overline{HT}I\overline{MRS}) \cup (H\overline{T}I\overline{MRS}) \tag{4.1}$$

the hand open gesture is given by

$$(\overline{H}TIMRS) \cup (HTIMRS) \tag{4.2}$$

and the two hand moving gesture for object scaling is given by

$$(\overline{HTIM}RS) \cap (H\overline{TIM}RS) \tag{4.3}$$

4.3 OBJECT DISTANCE COMPUTATION

According to Fig. 4-1, apart from the object release operation which requires only recognition of the corresponding hand gesture, other object manipulation operations require additional information of the object with respect to the user hands in terms of its location, orientation and volume.

In order to execute the object selection and manipulation operations using the selected gestures, the 3D position of the left and right index fingertips need to be determined and tracked. As a hinged joint, there is only one degree of freedom for the distal interphalangeal joint on the index finger [51]. With the distal phalanx length known through the measurement of the user's index finger, the index fingertip position can be determined based on the distal interphalangeal joint position and the distal phalanx bending angle with respect to the middle phalanx provided by the ShapeHand data glove, as illustrated in Fig. 4-2.

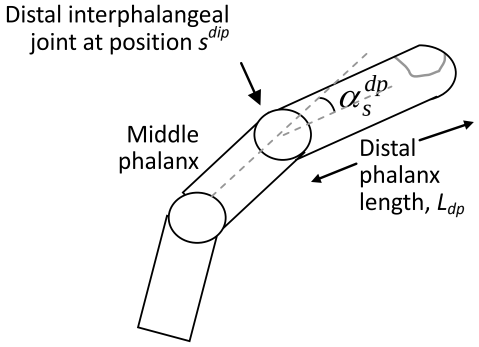

Distal interphalangeal joint at position s^{dip}
α_s^{dp}
Middle phalanx
Distal phalanx length, L_{dp}

Figure 4-2. Index finger model.

As shown in Fig. 4-2, if $\boldsymbol{s}^{dip,i} = [x_s^{dip,i}, y_s^{dip,i}, z_s^{dip,i}, 1]'$ denotes the homogeneous coordinates of the distal interphalangeal joint on the index finger and $\alpha_s^{dp,i}$ denotes the distal phalanx bending angle in the local ShapeHand coordinate system, then the homogeneous coordinates of the index fingertip position in the world coordinate system denoted by $\mathbf{s}_w^{tip,i} = [x_{sw}^{tip,}, y_{sw}^{tip,I}, z_{sw}^{tip,I}, 1]'$ are given by

$$\mathbf{s}_w^{tip,i} = \mathbf{T}^{I \to W} \mathbf{T}^{S \to I} \begin{bmatrix} 1 & 0 & 0 & \cos \alpha_s^{dp,i} L_{dp} \\ 0 & 1 & 0 & \sin \alpha_s^{dp,i} L_{dp} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{s}^{dip,i} \qquad (4.4)$$

where L_{dp} denotes the distal phalanx length.

For virtual object manipulation, let the virtual object to be manipulated be denoted by $\boldsymbol{o}$ centered at (o_x, o_y, o_z) in the world coordinate system, with orientation of (o_α, o_β, o_γ), and with its bounding box defined by the lengths of (L_x, L_y, L_z). Object selection requires not only the recognition of the index finger pointing gesture by using equation (4.1), but also the position of the index finger tip with respect to the object bounding box in 3D space. A virtual object is selected, if the index finger tip (left or right) touches anywhere in one of the side faces of the bounding box. Hence, upon recognition of the index finger pointing gesture, two more conditions need to be satisfied for a virtual object to be selected. One is based on the distances between the index finger tip and the side face centres of the object bounding box, and the other is based on the distances between the index finger tip and the side face planes of the object bounding box.

For the first condition, if $\boldsymbol{S}_m$ with m= 1, 2, …, 6, denote the six side planes of the virtual object bounding box, and $\boldsymbol{P}_{m,1} = [x_{Pm,1}, y_{Pm,1}, z_{Pm,1}]$ the coordinates of each side plane centre, then the distance between the pointing index finger tip and each side plane centre is given by

$$dist(\mathbf{s}_w^{tip,i}, P_{m,1}) = \sqrt{(x_{sw}^{tip,i} - x_{Pm,1})^2 + (y_{sw}^{tip,i} - y_{Pm,1})^2 + (z_{sw}^{tip,i} - z_{Pm,1})^2} \qquad (4.5)$$

For the second condition, three non-collinear points lying on each side of the bounding box are selected to represent each side plane. If these three points are denoted by $\boldsymbol{P}_{m,n}$ with m = 1, 2, …, 6 and n = 1, 2, 3, then the distance between the pointing index finger tip and each side plane of the bounding box is given by

$$dist(s_{sw}^{tip,i}, S_m) = \frac{\left|(a_m x_{sw}^{tip,i} + b_m y_{sw}^{tip,i} + c_m z_{sw}^{tip,i} + d_m)\right|}{\sqrt{{a_m}^2 + {b_m}^2 + {c_m}^2}} \qquad (4.6)$$

where a_m, b_m, c_m, and d_m are the coefficients of the plane equation for $\boldsymbol{S}_m$, and are obtained by solving the following equation

$$\text{a} = (y_{Pm,2} - y_{Pm,1})*(z_{Pm,3} - y_{Pm,1}) - (z_{Pm,2} - z_{Pm,1})*(y_{Pm,3} - y_{Pm,1});$$

$$\text{b} = (z_{Pm,2} - z_{Pm,1})*(x_{Pm,3} - x_{Pm,1}) - (x_{Pm,2} - x_{Pm,1})*(z_{Pm,3} - z_{Pm,1});$$

$$\text{c} = (x_{Pm,2} - x_{Pm,1})*(y_{Pm,3} - y_{Pm,1}) - (y_{Pm,2} - y_{Pm,1})*(x_{Pm,3} - x_{Pm,1});$$

$$\text{d} = (0-\text{a}* x_{Pm,1}) + \text{b}* y_{Pm,1} + \text{c}* z_{Pm,1}; \qquad (4.7)$$

In the implementation, the three non-collinear points selected for each side of the bounding box include the corresponding side plane centre, and a virtual object is selected when the following condition is true

$$\begin{aligned}&\left\{\left(\overline{H}\,\overline{T}I\overline{M}\,\overline{R}\,\overline{S}\right)\cup\left(H\overline{T}I\overline{M}\,\overline{R}\,\overline{S}\right)\right\}\\ &\cap\left\{dist\left(s_{sw}^{tip,i},P_{m,1}\right)\le\min(L_x,L_y,L_z)/2\right\}\cap\left\{dist\left(s_{sw}^{tip,i},S_m\right)=0\right\}\end{aligned} \tag{4.8}$$

In equation (4.8), the first two terms come from equation (4.1) and are used to confirm the index finger pointing gesture that can be made by the left and/or right hand, the third and fourth terms indicate the index finger tip touching a side face of the bounding box. When equation (4.8) is satisfied, the bounding box of the virtual object is highlighted to provide a visual feedback to the user, which will be discussed in Chapter 6.

For object translation and rotation followed by object selection, it was implemented by making the object centre follow the current 3D position of the index fingertip computed using equation (4.4) and the object 3D orientation to follow the current wrist orientation provided by InterSense. If both hands are making the index finger pointing gestures, the position of the selected object will follow the index finger tip with minimum distance to the object centre, where the distance is computed by

$$dist(\mathbf{s}_w^{tip,i},o)=\sqrt{(x_{sw,i}^{tip}-o_x)^2+(y_{sw,i}^{tip}-o_y)^2+(z_{sw,i}^{tip}-o_z)^2} \tag{4.9}$$

Since a user may use one pointing index finger to do object selection, translation and rotation with the other hand open, the object release operation is disabled if equation (4.8) is satisfied.

Object scaling requires not only recognition of the two hand gesture using equation (4.3) but also the positions of the left and right index fingertips with respect to the object bounding box in 3D space. Hence, the function to activate the scaling operation can be expressed by

$$\begin{aligned}&\left\{\left(\overline{H}TIM\overline{R}\,\overline{S}\right)\cap\left(HTIM\overline{R}\,\overline{S}\right)\right\}\\ &\cap\left\{dist\left(s_{sw,l}^{tip,i},P_{m,1}\right)\le\min(L_x,L_y,L_z)/2\right\}\cap\left\{dist\left(s_{sw,l}^{tip,i},S_m\right)=0\right\}\\ &\cap\left\{dist\left(s_{sw,r}^{tip,i},P_{m',1}\right)\le\min(L_x,L_y,L_z)/2\right\}\cap\left\{dist\left(s_{sw,r}^{tip,i},S_{m'}\right)=0\right\}\end{aligned} \tag{4.10}$$

where the first two terms come from equation (4.3) and are used to recognise the object scaling gesture, the middle two terms indicate the left index finger tip touching a side of the bounding box, and the last two terms indicated the right index finger tip touching the opposite side of the bounding box, with $m \neq m'$. When equation (4.10) is satisfied, the scaling operation is activated, and the virtual object size is enlarged or reduced uniformly in 3D by setting the length of the object bounding box equal to the distance between two index fingertips.

$$L = \max[abs(x_{w,l}^{tip,i} - x_{w,r}^{tip,i}), abs(y_{w,l}^{tip,i} - y_{w,r}^{tip,i}), abs(z_{w,l}^{tip,i} - z_{w,r}^{tipi})] \qquad (4.11)$$

4.4 RESULTS AND EVLAUTION OF VIRTUAL OBJECT MANIPULATION SYSTEM

To demonstrate the usability and evaluate the performance of the developed system in terms of virtual object manipulation, a scene with two virtual objects was created for immersive manipulation by the user (author) wearing a pair of wireless ShapeHand data gloves, a pair of InterSense wrist tracking devices and a head tracking device, as well as a pair of polarised glasses. The objects to be manipulated are a virtual cube and a medical CT volume. The 3D virtual cube is a simple six-colour cube, with an initial size of 80 x 80 x 80 mm^3. By assigning each surface a different colour, it makes it easy to visualise. On the other hand, the CT volume is rendered based on a real medical CT image data of a human skull, which is constructed by 256 x 256 x 256 voxels with the size initialised to 256 x 256 x 256 mm^3. The CT volume data is displayed by adopting a volume rendering technique.

Many visual effects are volumetric in nature, and these models assume that light is emitted, absorbed, and scattered by a large number of particles in the volume [238]. Volumetric data rendering is essential for medical applications that require visualisation of three-dimensional data sets, for it has long been recognised that computer-generated 3D visualisation provides an effective presentation of the anatomical data to the clinicians [239]. Different from the 2D primitive objects rendering, the volumetric models and data require a special 3D rendering technique which is called volume rendering for visualisation. Moreover, interactive volume rendering relies on the performance of modern graphics accelerators and appropriate volume rendering approaches for efficient data exploration and feature discovery. Two methods for volume rendering are the indirect and direct methods [240]. The former renders the volume data by extracting surfaces with equal values from the volume and rendering them as polygonal meshes, e.g., iso-surface volume rendering. The latter renders the volume directly as a block of data by slicing a volume in a back-to-front manner, the opacity and chromaticity of each voxel are then stored in an one-dimensional indexing table (often referred to as a transfer function) that transforms the scalar value at each voxel into a RGBA (red, green, blue and alpha) vector. Finally, the composed RGBA result is projected onto the corresponding pixel of the frame buffer. Such projection could be done through several rendering techniques such as volume ray-casting, shear wrap, and texture mapping [241]. Since, in scientific visualisation, direct volume rendering is used to generate high quality semi-transparent images with details, and provides more realistic, flexible visualisation of the interior anatomical structures [239, 242], 3D texture mapping of the direct volume rendering method was employed to render the CT volume data.

From the visual perspective of virtual object manipulation, the user is able to see the stereoscopic images of the cube and CT skull volume as well as his/her hands displayed through two projectors placed at the back of a large screen operating in passive circular polarisation mode. With the head tracking device providing the position and orientation of the user's head, the user can physically move around in front of the display screen with an impression of a 3D virtual cube and a 3D CT volume floating in space, whereby a forward movement causes objects to appear nearer and larger, a backward movement causes objects to appear further and smaller, and a side movement with a side glance via head rotation causes a different side of objects to appear. In addition, the 3D skull volume is shown inside a cubic frame and is rendered with semi-transparency, whereby the user is able to see through the entire 3D appearance of the skull with depth information as well as skeleton structure details.

From the interaction perspective of virtual object manipulation, the user is able to see his/her hands in 3D with respect to the virtual objects, as well as the gestures made. Furthermore, when the pointing index finger tip of the user reaches a side plane of a virtual object, the object bounding box is highlighted to provide a visual feedback of the object selected (highlighting the touched surface of the virtual cube as shown in Fig. 4-3, or the frame of the CT volume touched). The simplicity and intuitiveness of the hand gestures are seen to enable a new user to quickly handle and manipulate each object or both objects simultaneously, namely, pointing the index finger(s) to touch (select), drag, rotate the 3D cube or the CT volume, or both in 3D space as shown in Fig. 4-4, passing the selected object from one hand to another hand (from one pointing index finger to another pointing index finger), sliding two hands (with the ring and small finger closed) with respect to each other to enlarge and reduce the size of the selected object as shown in Fig. 4-5, and opening the hand(s) to detach from the selected object(s).

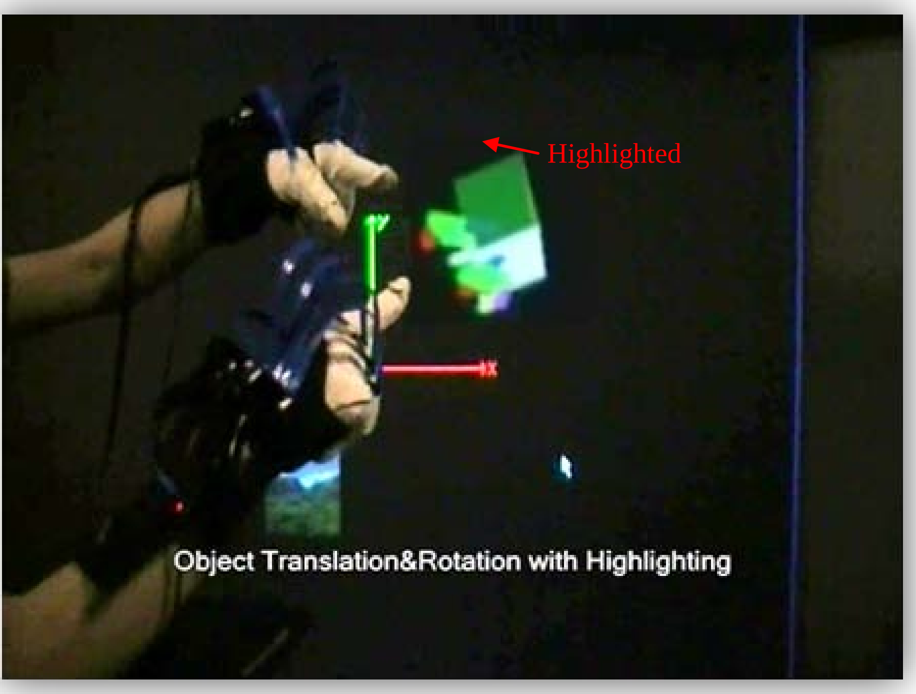
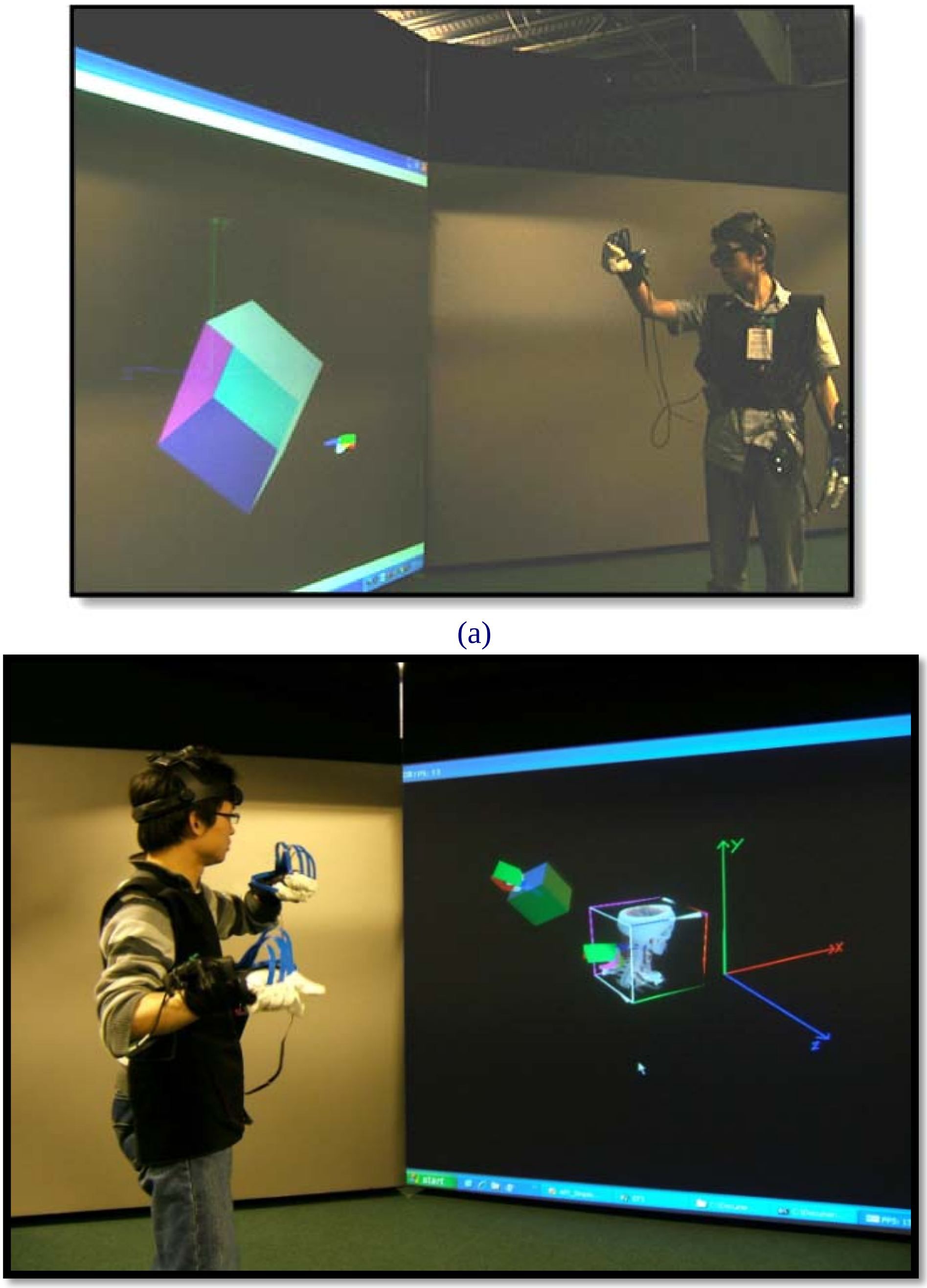


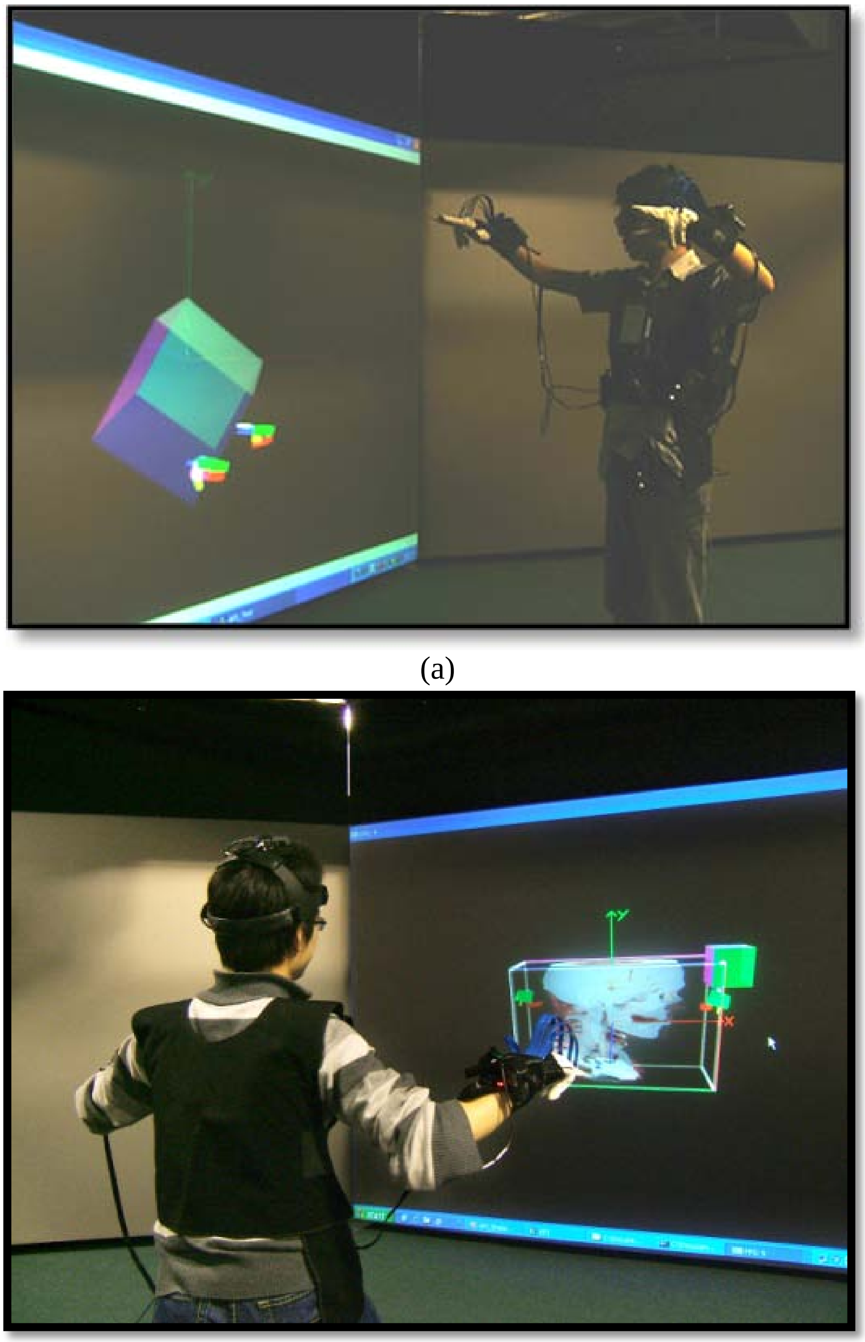
Figure 4-3. Object bounding box highlighted when selected.



(a)

(b)

Figure 4-4. User performing (a) translation and rotation of a cube; and (b) translation and rotation of a cube and a CT volume.



(a)

(b)

Figure 4-5. User performing (a) scaling of a cube; and (b) scaling of a CT volume.

Besides, the system is able to perform the required operations with certain deviation in the gestures made such as fingers not fully open and closed, and highly accurate recognition can be achieved by performing calibration of hand close and open gestures at the start, where the hand joints angles for the fully open and closed gestures will be recorded and on which the later hand gesture status judgment will be based.

As an example, Fig. 4-6 shows some of the data acquired from performing a short sequence of selection, translation and release of one of the virtual object in the scene, namely, (o_x, o_y, o_z) to show the 3D position variation of the virtual object centre using red, green and blue dotted lines, ($x_{sw,r}^{tip}$, $y_{sw,r}^{tip}$, $z_{sw,r}^{tip}$) to show the 3D position variation of the right hand index finger tip using red, green and blue solid lines, and α_s^{pp} to show the pitch angle of the proximal phalanx of the right hand middle finger in a black solid line. With an open hand gesture at the start of the sequence, it is seen from Fig. 4-6 that α_s^{pp} is around 0º, the user right hand moves in the horizontal plane as indicated by the changing coordinate values of $x_{sw,r}^{tip}$ and $z_{sw,r}^{tip}$ with $y_{sw,r}^{tip}$ roughly constant, and the virtual object is stationary at the origin of the world coordinate system as indicated by (o_x, o_y, o_z) = (0, 0, 0). Soon after the hand gesture changed into an index finger pointing gesture as indicated by the sharp rise of α_s^{pp} from 0º to around 70º due to the middle finger closed, the index finger tip is seen to approach the virtual object with the coordinate values of ($x_{sw,r}^{tip}$, $y_{sw,r}^{tip}$, $z_{sw,r}^{tip}$) moving towards (o_x, o_y, o_z). When the distance is computed to be sufficiently small by using equation (4.5), the virtual object is seen to be selected, and follows the index finger tip with the coordinate values of o_x following $x_{sw,r}^{tip}$, o_y following $y_{sw,r}^{tip}$, and o_z following $z_{sw,r}^{tip}$. Finally, the user hand opens to release the object as indicated by α_s^{pp} falling back to around 0º due to the opening of the middle finger, the virtual object is seen to stay at its final position with (o_x, o_y, o_z) fixed as the user hand moves away.

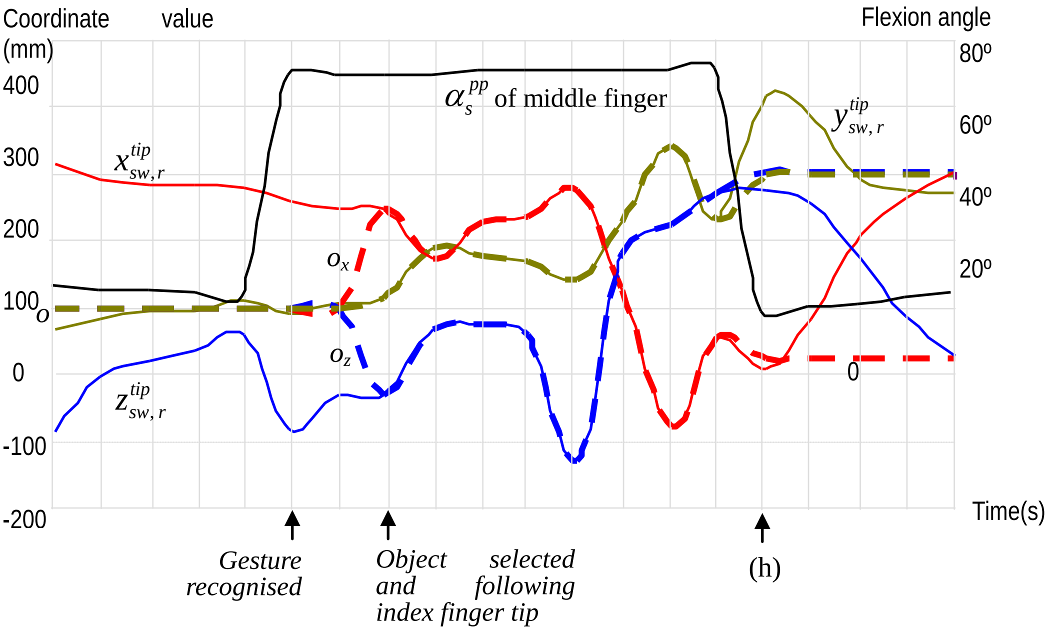


Figure 4-6. A sequence of dynamic gesture data.

A number of tests were also conducted to assess the real-time performance. For graphics based object manipulation, speed performance evaluation was based on the displayed graphic objects, which include two virtual hand models, a virtual cube, and other graphic text outputs, e.g., the axes and text. For data based object manipulation, the CT volume is also included. The evaluation was implemented by inserting a counter at the start of each program thread to record the number of times to run the thread per second.

For graphics based object manipulation, Fig. 4-7 shows a typical example listing the frequency of executing each program thread over a period of one minute with continuous hand movement in 3D space. From Fig. 4-7, the program threads of gesture recognition and InterSense data acquisition are seen to be relatively fast with relatively large fluctuations. Whilst the former is seen to be the fastest one with an average execution frequency of 136 times per second and the largest variation between the maximum of 171 times per second and the minimum of 102 times per second, the latter is the second fastest with an average execution frequency of 118 times per second and a variation between the maximum of 159 times per second and the minimum of 99 times per second. Very similar behaviour of the execution frequencies for the ShapeHand gesture data acquisition program thread and the stereoscopic display program thread are also seen from Fig. 4-7, with the former slightly faster at 62 times per second on average between the maximum of 65 times per second and the minimum of 59 times per second, the latter is at 60 times per second between the maximum of 64 times per second and the minimum at 54 times per second.

For data based object manipulation, which includes the CT volume, the stereoscopic display program thread was also found to be the slowest with the worst case execution frequency of 31 times per second. Hence, the speed of the system depends on the complexity and the number of the objects to be displayed, and is comparable with other related works [240, 242, 243].

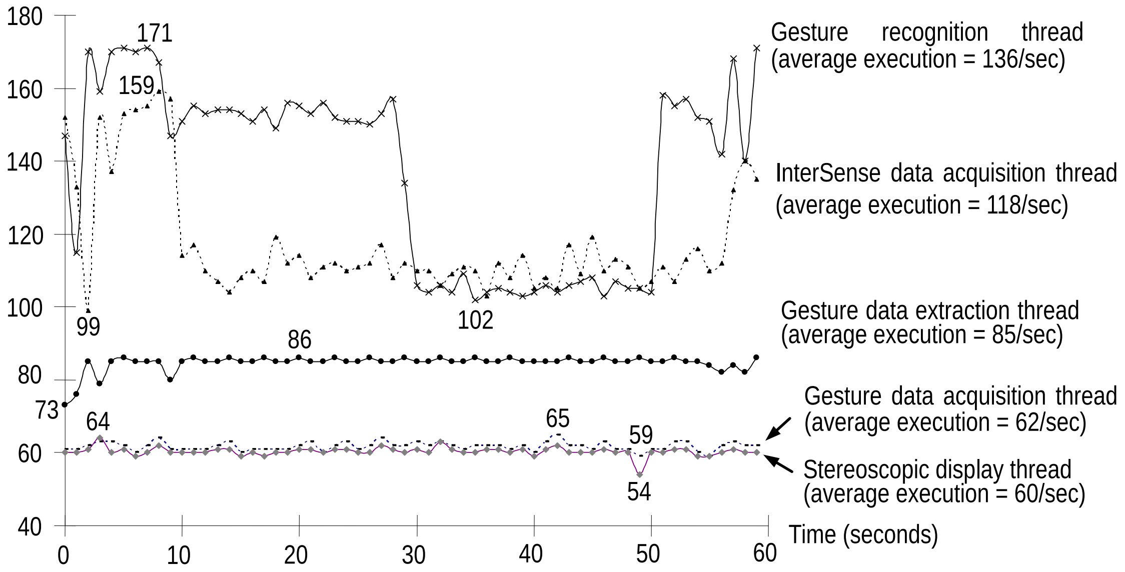


Figure 4-7. Thread execution frequency versus time.

4.5 CONCLUDING REMARKS

This chapter reports the development and assessment of virtual object manipulation to investigate the usability and performance of the proposed system. By using the designated hand gestures, namely, pointing hand gesture for object selection, translation and rotation, gun gesture for object scaling and open hand gesture for object release, each hand gesture can be recognised based on the hand joint angles. Together with the developed object distance computation rules, a user is able to manipulate objects in the virtual scene immersively using natural hand gestures. Two virtual objects have been used for performance evaluation, which are a graphic-based virtual cube and a real-image based CT medical data. Results showed that the user is able to use the designated hand gestures to manipulate these virtual objects effortlessly and naturally. Furthermore, speed tests have also been conducted. Results showed that the system is able to operate at a minimum of 54 frames per second when rendering a graphic cube and at a minimum of 31 frames per second when rendering a CT medical image data. Particularly, compared to other related works, the rendering and manipulation of the real-image based CT medical data demonstrated its usability for real applications [240, 242, 243].

Chapter 5

DIRECT SIGN WRITING

5.1 INTRODUCTION

There are several notation systems to enable a sign language communicated in a visual-gestural form to be transcribed into a written form [244], such as Stokoe Notation [245], Sutton SW [246] and HamNoSys [247]. The SW developed by Valerie Sutton in 1974 is a popular one among the deaf communities and was inspired from her dancing choreographic notation system called DanceWriting [248]. For hearing-impaired people, sign language is their only native language, and they have significant difficulties to learn and speak any other languages. Without sign notation systems, there is no means for hearing-impaired people to transcribe their own thought into a written form. In other words, a deaf person has severe barriers both in reading and writing. The invention of the SW notation system enables them to overcome these barriers by providing notations for reading and writing signs. Each sign in SW is represented by a sign-box containing a composition of basic standardised pictorial symbols to depict the hand configuration (hand shapes), body location, contacts, movements, and facial expressions [249]. Although deaf sign languages in the world are as many as spoken languages, the SW pictorial symbol-based system does not require any prior knowledge, and can be used to express any sign language [250]. Therefore, it has become a worldwide sign language notation system, whereby the hearing-impaired people can use their native sign language to transcribe their thoughts for knowledge disseminating, gaining education from reading SW books and communicating through internet.

Literature review shows a few related researches in SW. One converts the SW text into a virtual avatar animation [251]. Another in the University of Southern California adopts the vision-based approach, claiming recognition of 41 basic static hand gestures [252]. A useful and widespread tool for SW translation is called the SignWriter Keyboard [250]. This keyboard can switch between three keyboard modes: alphabet keyboard, SignWriter keyboard (see Fig. 5-1), and fingerspelling keyboard (see Fig. 5-2). The keyboard acts as a normal one in the alphabet keyboard mode. In the SignWriter keyboard mode, it contains all the commonly used symbol elements for SW that is identical for all countries, and the user can produce various SW notation outputs by combining different elements together to compose a certain SW notation symbol (sign-box). With the keyboard divided into five sections as shown in Fig. 5-1, facial expression symbols are in section one, body movement symbols in section two, hand movement symbols in section three, hand gesture symbols in section four, and special command symbols in section five. Unlike the SignWriter keyboard, the fingerspelling keyboard varies from one country to another with each country having its own alphabet fingerspelling sign gestures. For example, the fingerspelling keyboard for UK is shown in Fig. 5-2 and the U.S.A fingerspelling keyboard is shown in Fig. 5-3.

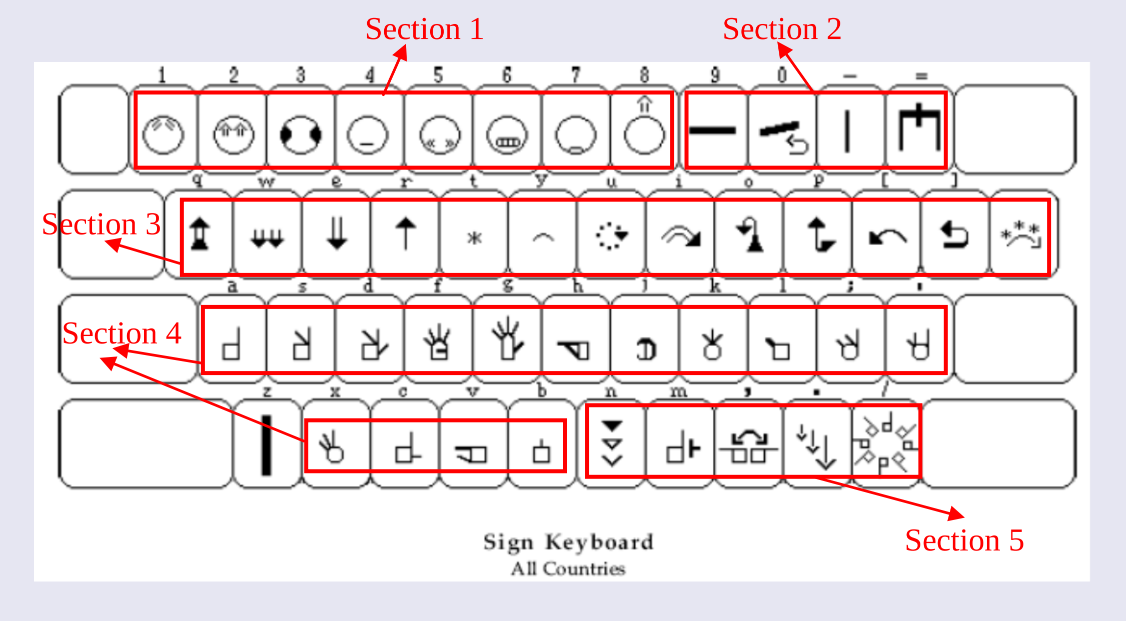


Figure 5-1. SignWriter Keyboard (modified from [250]).

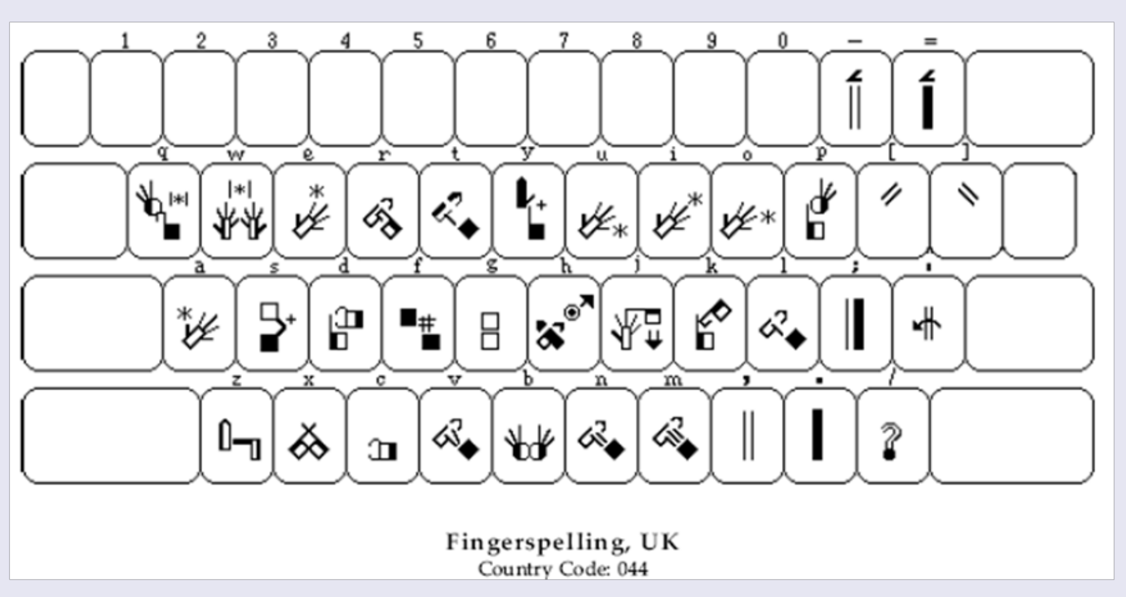


Figure 5-2. Fingerspelling Keyboard for UK (from [250]).

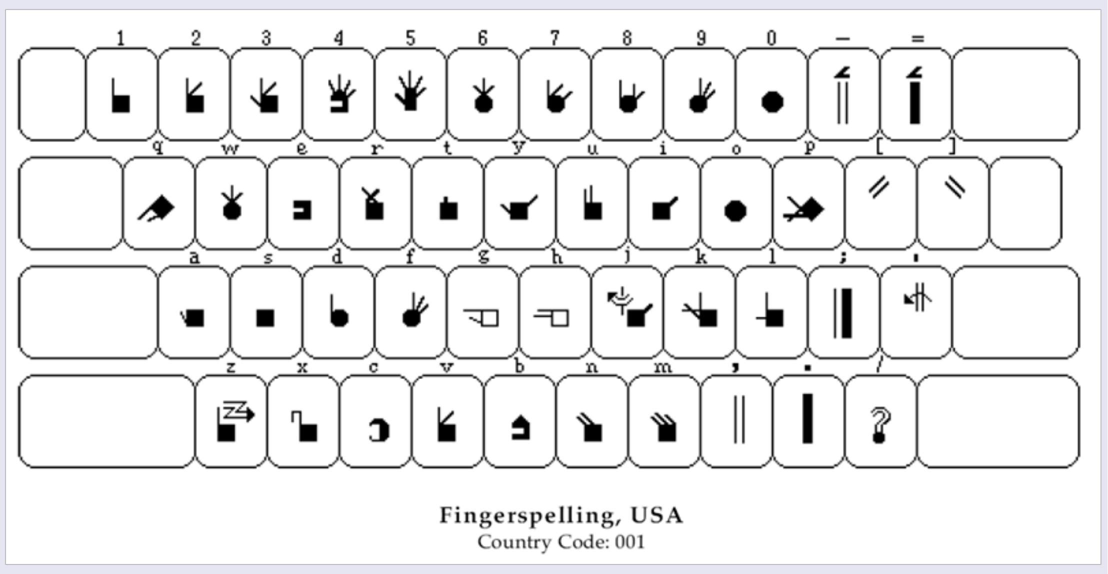


Figure 5-3. Fingerspelling Keyboard for U.S.A. (from [250]).

On one hand, the SignWriter keyboard based input provides a good solution for SW transcription; on the other hand, it is a time consuming process due to a large number of pictorial symbols for selection and a number of spatial manipulations (rotation and translation) of selected symbols required to compose a sign. According to International ISWA (SignWriting Alphabet) 2008, there are 30 pictorial symbol groups containing 639 basic pictorial symbols, where each basic symbol can have up to a maximum of 96 variations (up to 6 different fills and 16 rotations), which results in a large dictionary that currently contains a total of 35,023 valid symbols [253, 254]. Using the hand gesture of 'bright' shown in Fig. 5-4a as an example, it starts with two hands making 'triangle' gestures, which is followed by two hands making up-diagonal movement vertically, and both hands open at the end. In order to input this hand gesture symbol through the SignWriter keyboard, at least thirteen steps are required:

1) Press the 'h' key on the keyboard to yield the screen as shown in Fig. 5-4b;
2) Press the 'a' key to select the 'triangle' hand posture symbol for the left hand;
3) Select the hand symbol by pressing the 'Enter' key, and press the '/' key (corresponding to 'rotate' key on SignWriter keyboard) to rotate the hand symbol until it is in the right orientation;
4) Repeat steps 1 to 3 to input the same symbol for the right hand symbol;
5) Move the cursor to the left hand 'triangle' symbol, press the 'e' key and then press the 'a' key to select the left hand movement symbol (Fig. 5-4c);
6) Repeat step 3 to put the right hand movement symbol in the right orientation;

7) Move the cursor to the right hand ‘triangle symbol’ and repeat steps 5 and 6 to input in the same symbol for the right hand;

8) Select the right hand movement symbol and press the ‘n’ key (corresponding to ‘change’ key on SignWriter keyboard) to fill the movement arrow head as black, indicating it is a right left hand movement;

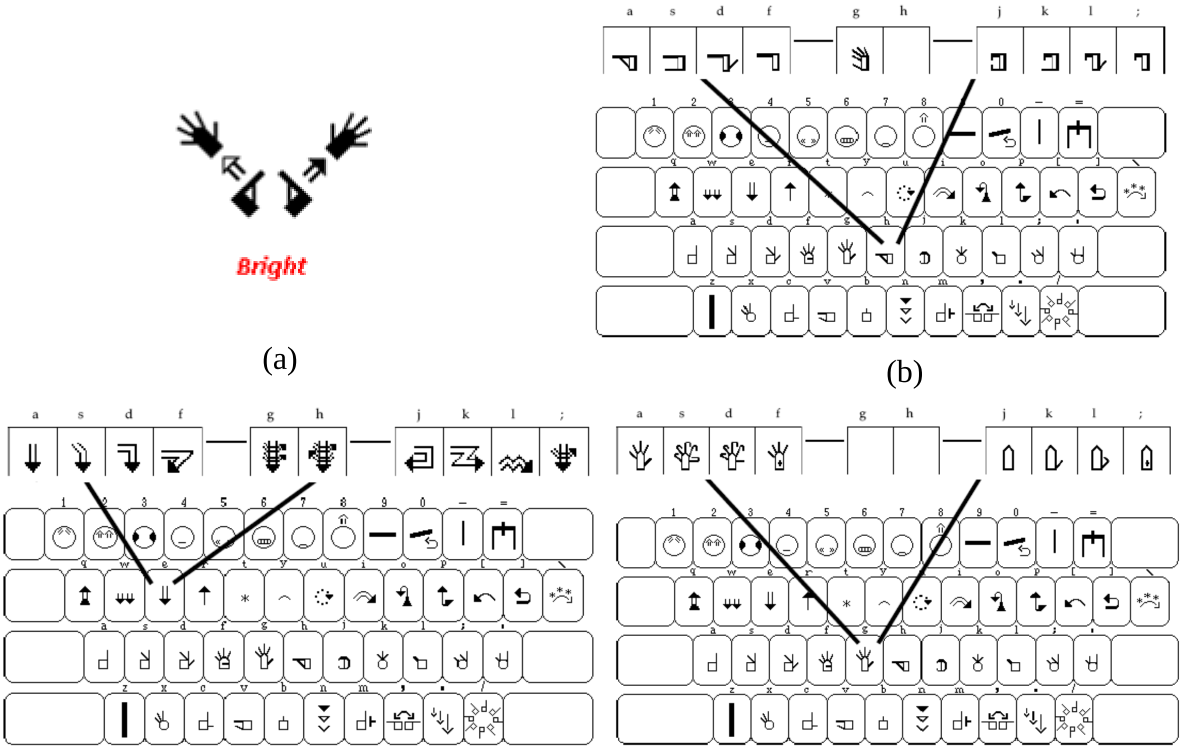
9) Repeat step 3 to put the left hand movement symbol in the right orientation;

10) Move the cursor to above the left hand movement symbol;

11) Press the ‘g’ key, and then press the ‘a’ key to select the ending hand gesture for the left hand (Fig. 5-4d);

12) Repeat step 3 to put the left hand ending gesture symbol in the right orientation; and

13) Repeat steps 10 to 12, to input the right hand ending gesture.



(a)

(b)

(c)

(d)

Figure 5-4. Steps to input the ‘bright’ gesture (from [250]).

Furthermore, an intermediate process is required for keyboard based SW input, since a user needs to transcribe the sign language into SW symbols in his/her mind prior to the selection of keyboard symbols. A challenge is therefore presented is to develop a DSW system to enable automatic transcription of articulated signs into corresponding sign-boxes in an electronic form without using keyboards. Moreover, as hand signs forming a core part of any sign language, the research focuses on automatic transcription of hand signs. For the proposed system, a pair of ShapeHand data gloves is worn by the signer to collect the hand joint data with 27-DOF for each hand, a pair of InterSense wrist trackers is attached to the signer wrist to collect the hand orientation and movement data, and a large stereoscopic display is used to display the virtual hand motion and their corresponding SW symbols through the developed DSW visualisation interface.

Although some works have also been performed to implement automatic recognition of BSL fingerspelling based on the developed system (see Appendix B), the work presented in this chapter focuses on DSW and is organised as follows. With the introduction of the SW transcription system presented in Section 5.1, it is followed by the algorithm developments of hand gesture recognition for DSW in Section 5.2, hand movement recognition for DSW in Section 5.3, as well as the 3D visualisation interface for DSW in Section 5.4. The results and evaluation of the developed system will be presented in Section 5.5. Finally, concluding remarks will be given in Section 5.6.

5.2 HAND GESTURE RECOGNITION FOR DIRECT SIGN WRITING

Through the study of ISWA [250], different hand postures are seen to be formed by different joint angles between two phalanges in 3D, namely, the pitch angles between two phalanges on a finger and the yaw angles between the proximal phalanges of two fingers (see Fig. 5-5). While the pitch angles can be classified into three possible states corresponding to finger closed, half-bent and open, the yaw angles can be classified into two states corresponding to adduction and abduction. Furthermore, each hand posture may be placed in the floor (x-z horizontal) plane or wall (x-y vertical) plane (see Fig. 5-6A), along eight directions in each plane (see Fig. 5-6B), and the orientation of a hand posture at each direction in either plane can have its palm facing three directions (see Fig. 5-6A). Thus, hand gesture recognition for DSW consists of hand posture and orientation recognition.

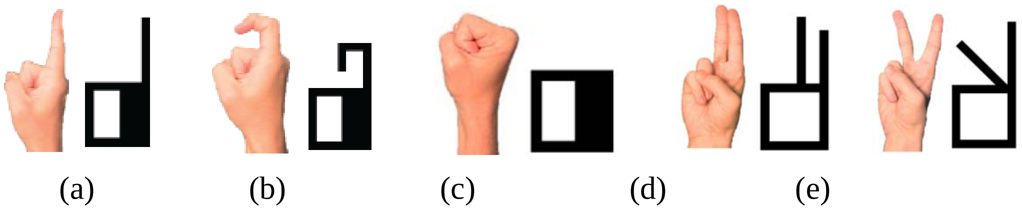


Figure 5-5. Hand postures and SW symbols of: (a) pointing; (b) half-pointing; (c) close; (d) close-two; and(e) open-two (from [250]).

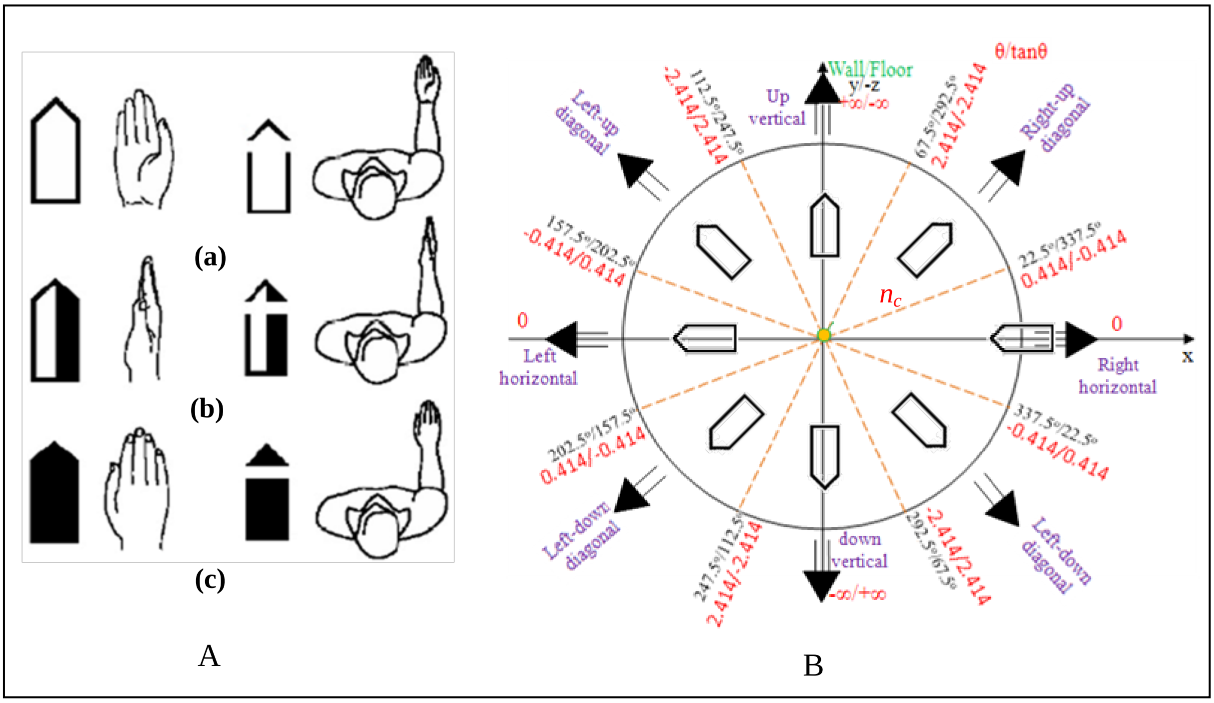


Figure 5-6. A: SW symbols showing palm facing in wall and floor planes (a); side facing in wall and floor planes (b); back facing in wall and floor planes (c); and B: Hand positions and movements along eight directions in x-z floor and x-y wall planes (modified from [250]) .

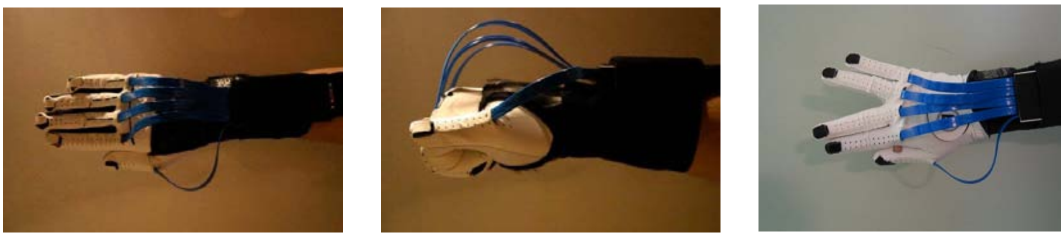
5.2.1 Hand Posture Recognition

Recognition of each hand posture is based on the 15 joint angles acquired by the ShapeHand data glove. Let the state of these joint angles of interest be denoted by

$$J_i = [\mathrm{p}_{\mathrm{I}_1}, \mathrm{p}_{\mathrm{I}_2}, \mathrm{p}_{\mathrm{I}_3}, \mathrm{p}_{\mathrm{M}_1}, \mathrm{p}_{\mathrm{M}_2}, \mathrm{p}_{\mathrm{M}_3}, \mathrm{p}_{\mathrm{R}_1}, \mathrm{p}_{\mathrm{R}_2}, \mathrm{p}_{\mathrm{R}_3}, \mathrm{p}_{\mathrm{S}_1}, \mathrm{p}_{\mathrm{S}_2}, \mathrm{p}_{\mathrm{S}_3}, \mathrm{Y}_{\mathrm{IM}}, \mathrm{Y}_{\mathrm{MR}}, \mathrm{Y}_{\mathrm{RS}}] \quad (5.1)$$

where *P* denotes the state of a joint pitch angle with three possible values of 0, 0.5 and 1 corresponding to interphalangeal joint close, half-bent, and open; *Y* denotes the state of a yaw angle between two fingers with two possible values of 0 and 1 corresponding to adduction and abduction; subscripts *I*, *M*, *R* and *S* denote the index, middle, ring and small fingers; and sub-subscripts 1, 2, and 3 denote the proximal, middle and distal interphalangeal joint. To ensure correct angle state classification of individual users, a hand calibration process is performed to acquire the range of angles. Using the three hand calibration postures shown in Fig. 5-7, pitch angles for each finger joint in fully open and fully close positions as well as yaw angles for two fingers in fully abduction and fully adduction positions are obtained. By using the maximum and minimum pitch angles obtained from each joint to normalize its range to 1, the pitch angle state of a joint is set to 0 if its normalized pitch angle is below 0.2; to 0.5 if its normalized pitch angle is between 0.2 and 0.8; and to 1 if its normalized pitch angle is above 0.8. Similarly, the yaw angle state is set to 0 if its normalized yaw angle is below 0.5; and to 1 otherwise (see Fig. 5-8). As an example, the joint angle state for the hand posture shown in Fig.5-5b is given by

$$J_i = [1, 0.5, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \qquad (5.2)$$



(a) (b) (c)

Figure 5-7. Hand calibration postures: (a) ‘homing’, (b) ‘fist’ and (c) ‘finger splay’.

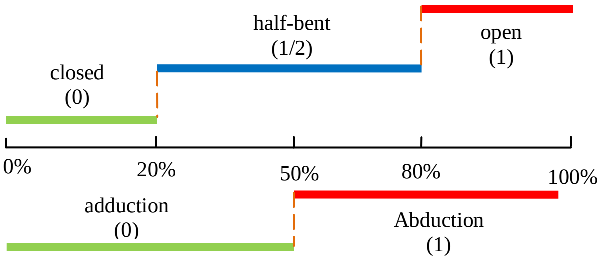


Figure 5-8. Hand joint angles’ scaling.

To determine an input hand posture made by a signer, the minimum difference (denoted by d_t) between the input joint angle state (denoted by J_i) and template joint angle states stored in a database (denoted by J_t) are computed using equation (5.3). If d_t is less than a specified threshold, then the input hand posture is recognised to be the same as the best template candidate, otherwise, it will be classified as an unknown posture.

$$d_t = \arg\min_t \left\| J_i - J_t \right\| \tag{5.3}$$

5.2.2 Hand Orientation Identification

Recognition of hand orientation is based on the use of the InterSense position and orientation data as well as the hand rotation data provided by ShapeHand data glove in its local coordinate system, where the origin is fixed at the bottom of the palm in the middle of the wrist (see Fig. 3-6). Two local hand planes are used to classify hand orientation in terms of parallel to the floor or wall plane, pointing in one of the 8 principal directions in each plane, and palm facing one of the 3 directions. One is the palm plane and the other is the wrist cross-section plane (see Fig. 5-9). Based on local ShapeHand coordinate system, the former is defined by three non-collinear points with the first point located at the centre of the palm bottom with the coordinates given by Pp_1 = (0, 0, d/2), where d denotes the thickness of the palm; the second point located at the centre of the palm with the coordinates given by Pp_2 = (0, l/2, d/2), where l denotes the palm length; and the third point located at the bottom and side of the palm with the coordinates given by Pp_3=(w/2, 0, d/2), where w denotes the wrist width. Similarly, the latter is defined by three non-collinear points with the coordinates given by Pb_1 = (0, 0, 0), Pb_2 = Pp_1, and Pb_3=Pp_3.

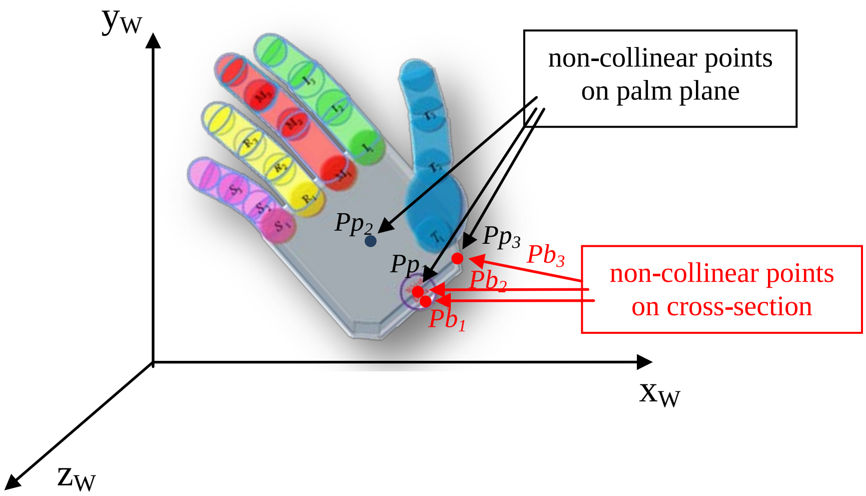

yW
non-collinear points on palm plane
Pp2
Pp3
Pb3
Pp
Pb2
non-collinear points on cross-section
Pb1
xW
zW

Figure 5-9. Palm plane and wrist cross-section for determination of hand orientation.

With the ShapeHand data glove providing the hand rotation data with respect to the ShapeHand local origin as well as the InterSense wrist trackers providing the position and orientation of each wrist with respect to the InterSense origin, the world coordinates of these two sets of local coordinates in the palm plane and the wrist cross-section can be obtained through the geometrical transformation matrix that describes the relationship between the coordinate systems by using equation (3.3). For example, the world coordinate of *Pb$_1$* is given by

$$\mathrm{S_w} = T^{I\to w}T^{S\to I}T^{H}\,\mathrm{Pb_1} \tag{5.4}$$

where $T^{S\to I}$, and $T^{I\to W}$ are given by equations (3.4) and (3.2), and T^{H} is given by

$$T^{H} = \begin{bmatrix} c_\alpha s_\beta - s_\alpha c_\beta s_\gamma & c_\beta c_\gamma & c_\alpha c_\beta s_\gamma + s_\alpha s_\beta & 0 \\ s_\alpha c_\gamma & -s_r & c_\alpha c_\gamma & 0 \\ -s_\alpha s_\beta s_\gamma - c_\alpha c_\beta & s_\alpha c_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.5}$$

where *c* and *s* denoting *cos* and *sin* functions with the subscripts *α*, *β* and *γ* denoting the hand rotation angles provided by the ShapeHand data glove.

Based on the world coordinates of the non-collinear points selected in the palm and cross-section planes computed using the above equations, two corresponding plane equations can be determined by using equation (4.7), whereby their normal vectors in the world coordinate system can be obtained. Let the plane equation of the wrist cross-section in the world coordinate system be expressed as $A_c(x\text{-}x_c)+B_c(y\text{-}y_c)+C_c(z\text{-}z_c)=0$, where $p_c = (x_c, y_c, z_c)$ is a point on the cross-section plane expressed in the world coordinate system, the non-zero normal vector of the cross-section plane is given by $\overset{\rightarrow}{n_c} = (A_c, B_c, C_c)$. Classification of the hand in the wall or floor plane is based on the angle between the normal vector of the wrist cross-section and the floor plane with the angle threshold set to 30°. With the three parameters of a normal vector having the relationships of $A_c^2 + B_c^2 + C_c^2 = 1$ in 3D, this threshold condition corresponds to $A_c/C_c = \tan 30^{o}$ and results in $B_c = 0.25$. In other words, the hand will be recognised as being placed in the wall plane if $B_c \geq 0.25$, and in the floor plane otherwise.

Furthermore, for classification of the hand pointing direction, the floor and wall planes are partitioned using 45° angular sectors centred at eight principal directions as shown in Fig. 5-6B. With the normal vector of the wrist cross-section denoted by $\overset{\rightarrow}{n_c}$, the hand pointing direction in each plane is determined by finding at which angular sector $\overset{\rightarrow}{n_c}$ points.

For the hand appeared in the wall plane, it is classified based on the following two possible cases: (i) $A_c = 0$ and $B_c \neq 0$; and (ii) $A_c \neq 0$ and $B_c \neq 0$. For the first case, the hand pointing direction is classified as pointing up if $B_c > 0$, and pointing down if $B_c < 0$. For the second case, the hand orientation angle is computed using

$$\tan n_c = B_c/A_c \tag{5.6}$$

While the sign of the result produced by equation (5.6) is used to identify the hand in the upper half (if it is positive) or lower half (if it is negative) of the wall plane, and the magnitude value is used to define the particular angular sector in the upper or lower half of the wall plane the hand points, according to the pre-computed limits of the tangent values for each angular sector as shown in Fig. 5-6B.

For the hand appeared in the floor plane, it is classified based on the following three possible cases: (i) $A_c = 0$ and $C_c \neq 0$; (ii) $A_c \neq 0$ and $C_c = 0$; and (iii) $A_c \neq 0$ and $C_c \neq 0$. For the first case, the hand pointing direction is classified as pointing backward if $C_c > 0$ and forward if $C_c < 0$. For the second case, the hand pointing direction is classified as pointing right if $A_c > 0$ and left if $A_c < 0$. For the third case, the hand pointing direction is computed using

$$\tan n_c = C_C / A_c \tag{5.7}$$

While the sign of the result produced by equation (5.7) is used to identify the hand in the back half (if it is positive) or front half (if it is negative) of the floor plane, and the magnitude value is used to define the particular angular sector in the front or back half of the floor plane the hand points, according to the pre-computed limits of tangent values for each angular sector as shown in Fig. 5-6B.

Similarly, let the hand palm plane equation be expressed as $A_p(x\text{-}x_p)+B_p(y\text{-}y_p)+C_p(z\text{-}z_p)=0$, where $p_p = (x_p, y_p, z_p)$ is a point on the palm plane expressed in the world coordinate system, the nonzero normal vector of the palm plane is given by $\vec{n}_p = (A_p, B_p, C_p)$. Classification of palm facing directions is based on the C_p parameter value for the wall plane and B_p parameter value for the floor plane. For the former, the values of C_p between -1 and -0.33 are set to correspond to back facing; between -0.33 and 0.33 for side facing; and between 0.33 and 1 for palm facing. Same threshold settings are used for the values of B_p to determine the palm facing direction in the floor plane.

5.2.3 Recognition Performance Evaluation

To evaluate the hand gesture recognition method developed for DSW, tests were conducted by the user wearing a pair of data gloves to provide the hand finger joint data and a pair of InterSense wrist trackers to provide the hand position and orientation data in 3D. To provide a visual feedback to the user making hand gestures with different postures and orientations, a stereoscopic screen is used to display the 3D hand models with the corresponding gestures as well as the corresponding SW hand gesture symbols.

For hand posture recognition evaluation, some representative postures have been selected, which cover various combinations of different finger bending and abduction possibilities, namely, 'pointing', 'half-pointing', 'close', 'close-two', 'open-two', 'homing', 'homing-claw', 'open-claw' and 'open', as shown in Fig. 5-10. Based on equation (5.1), a look-up table for the joint angles states of these hand postures has been constructed (see Table 5-1), where the number *x* corresponds to the state of '*don't care*'.

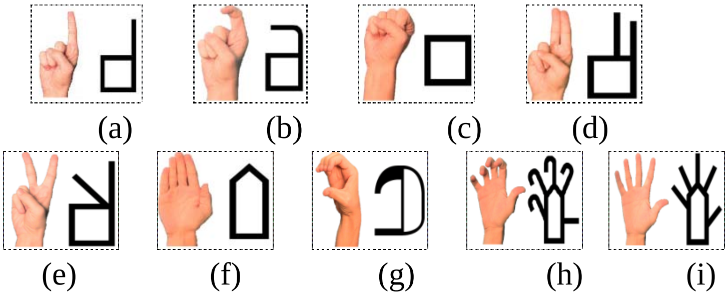


Figure 5-10. Hand postures and SW symbols of: (a) 'pointing', (b) 'half-pointing', (c) 'close', (d) 'close-two', (e) 'open-two', (f) 'homing', (g) 'homing-claw' (h) 'open-claw'', and (i) 'open' (modified from [250]).

Table 5-1: Look-up table for joint angles states

	p_{I_1}	p_{I_2}	p_{I_3}	p_{M_1}	p_{M_2}	p_{M_3}	p_{R_1}	p_{R_2}	p_{R_3}	p_{S_1}	p_{S_2}	p_{S_3}	Y_{IM}	Y_{MF}	Y_{RS}
point	1	1	1	0	0	0	0	0	0	0	0	0	x	x	x
Half -point	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	x	x	x
close	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x
Close -two	1	1	1	1	1	1	$\frac{1}{2}$,0	0	x	$\frac{1}{2}$,0	0	x	0	x	x
Open -two	1	1	1	1	1	1	$\frac{1}{2}$,0	0	x	$\frac{1}{2}$,0	0	x	x	x	x
homing	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
homing -claw	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
open -claw	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1
Open	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 5-11 shows the results obtained, where the system is seen to simultaneously display the virtual hand model corresponding to the gesture made by the user and the correct SW symbols in a small symbol display box.

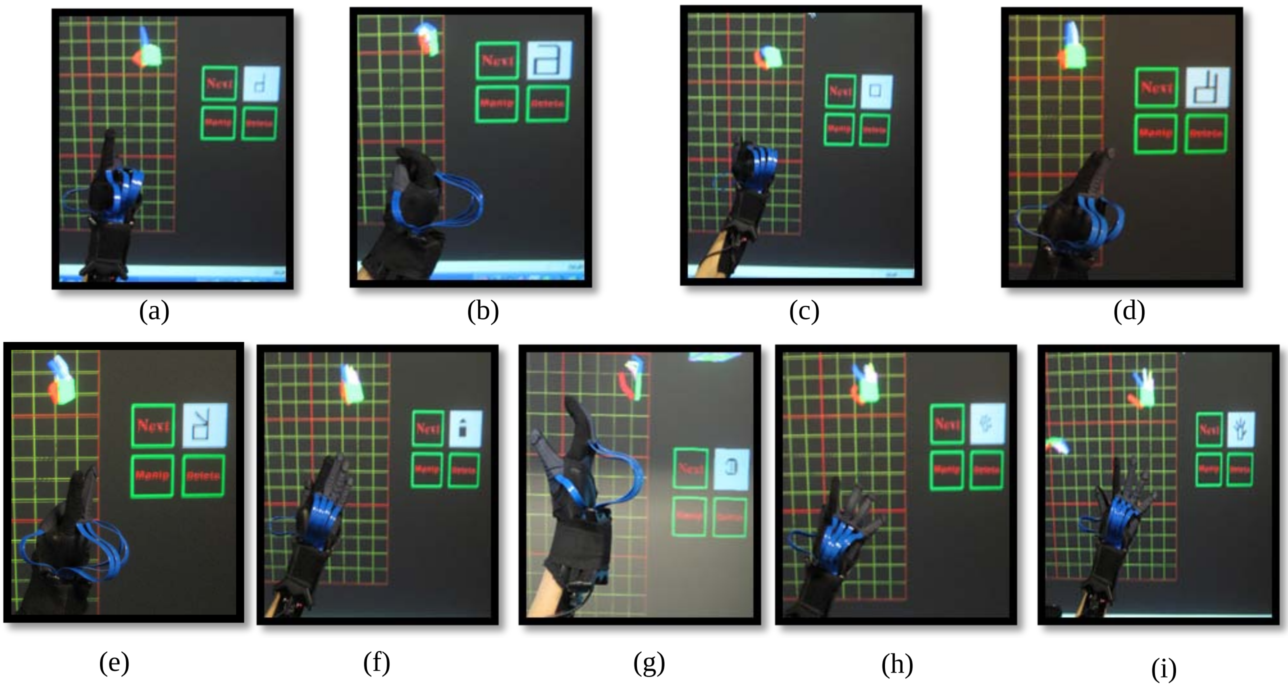
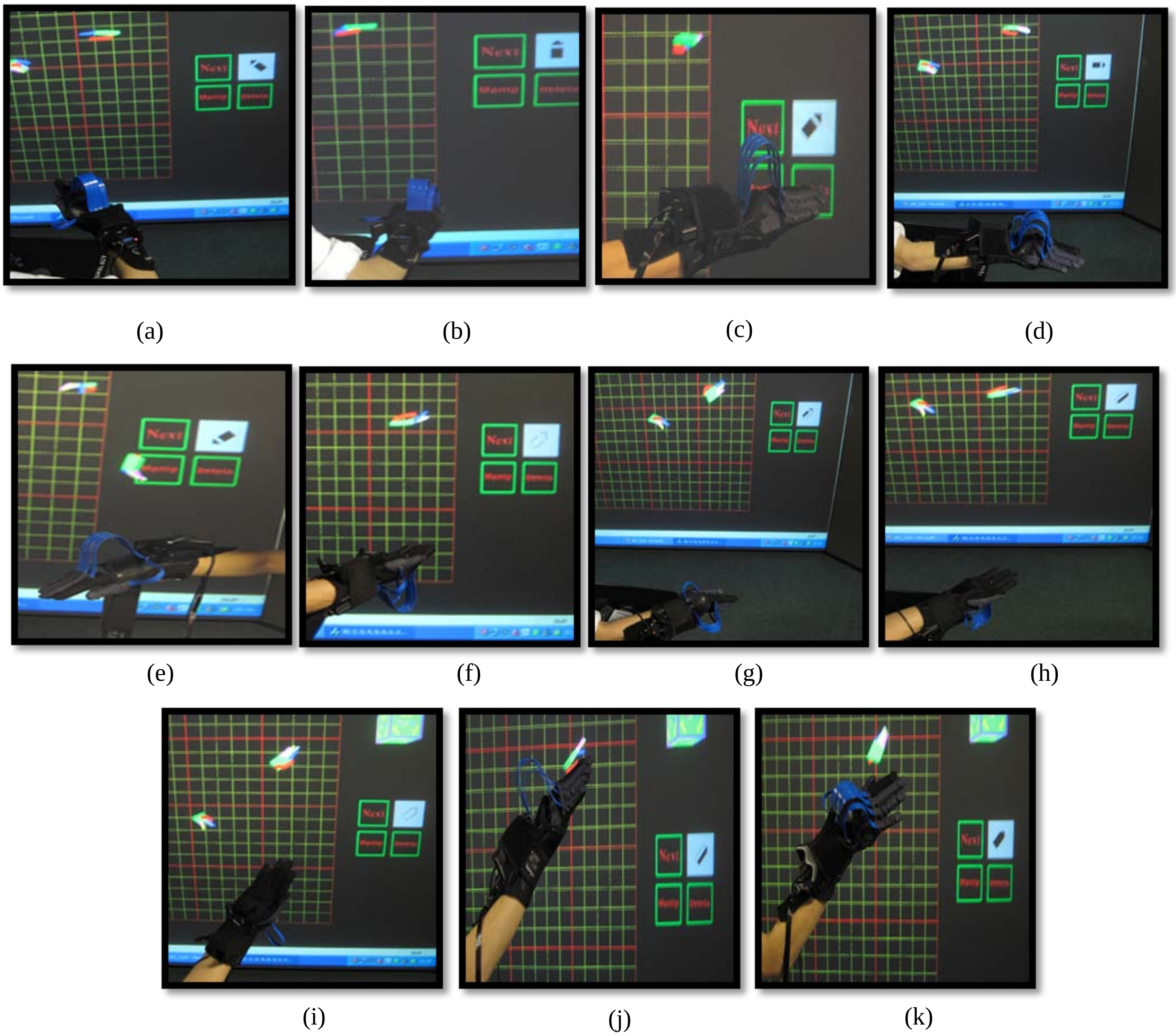


Figure 5-11. Right hand making postures of: (a) 'pointing', (b) 'half-pointing', (c) 'close', (d) 'close-two', (e) 'open-two', (f) 'homing', (g) 'homing-claw' (h) 'open-claw'', and (i) 'open'.

For hand orientation recognition evaluation, tests were conducted for hands placed at eight directions in both wall and floor planes with three palm facing directions. The results show that the system is able to identify the hand placing plane to be the floor plane if the hand is level and parallel to the floor, and the wall plane if the hand is tilted and at a sufficiently large angle with respect to the floor plane. Also, the system is able to identify the hand orientation at eight directions in each plane, where at each hand pointing direction the hand orientation can be further identified as one of the three palm facing directions. Some representative examples are shown in Fig. 5-12. While Figs. 5-12(a-e) show the hand being placed parallel to the floor plane at the directions of forward-left, forward, forward-right, right and left-backward, Figs. 5-12(f-h) demonstrate front, side and back palm facing directions for the forward-right hand direction, and Figs. 5-12(j-k) show the hands being placed at the up-right direction in the wall plane with front, side and back palm facing directions.



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

(e) (f) (g) (h)

(i) (j) (k)

Figure 5-12. Examples for hand orientation recognition.

5.3 HAND MOVEMENT RECOGNITION FOR DIRECT SIGN WRITING

5.3.1 Hand Movement Recognition

According to the SW lesson textbook, SW could transcribe four groups of hand movements for dynamic hand gestures, namely, straight movement, curved movement, axial movement and circular movement [255]. Since each group of hand movements has a maximum of 43 basic symbols and each basic symbol has a maximum of 96 variations (up to 6 different fills and 16 rotations), it results in a large dictionary that contains 16,512 valid hand movement symbols. As a first step to achieve the full hand movement recognition for DSW, the useful hand movements selected for recognition include straight movements on both wall/floor planes, curved movements on both wall/floor planes, and repeated movements.

As shown in Fig. 5-13, unfilled/filled arrow heads are used to indicate left/right hands, double-stem/single-stem arrows are used to indicate movements parallel to the wall/floor planes, and arrow orientations are used to indicate movement directions on either plane (with hand movements in each plane divided into eight principal directions). Also, arrows can be straight/curved to indicate movement paths, and duplicated/tripled for repeated movements.

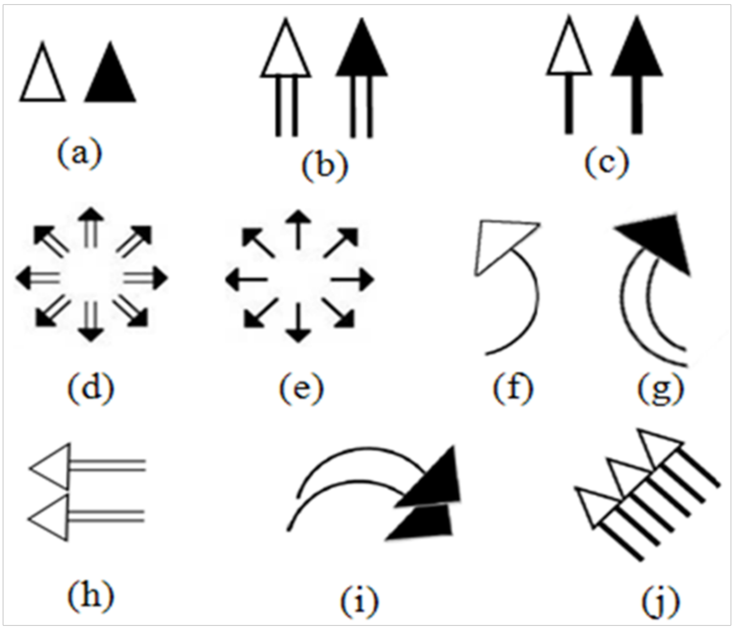


Figure 5-13. SW hand movement symbols showing (a) hand movements by left/right hand; (b) upward hand movement in wall plane; (c) forward hand movement in floor plane; (d) wall plane hand movement directions; (e) floor plane hand movement directions; (f) forward anti-clockwise movement by left hand in floor plane; (g) upward clockwise movement by right hand in wall plane; (h) duplicated left movement by left hand in wall plane; (i) duplicated right clockwise hand movement by right hand in floor plane; (j) tripled up-left movement by left hand in wall plane.

For hand movement recognition, the sequence starts with identification of repeated movements. This is then followed by identification of movement planes, movement directions, path linearity, and clockwise/anti-clockwise movements, respectively.

5.3.2 Identification of Repeated Movements

Fig. 5-14 illustrates a repetitive hand movement, where each non-repeating movement starts around its initial point and ends around its vanishing point. Identification of repeated hand movements is based on the change of the hand movement speed as well as the relative hand moving radius in a specified time interval. Let the starting hand position, defined by the starting hand gesture, be denoted by $\boldsymbol{p_i} = [x_i, y_i, z_i]$; and let the length of the acquired hand motion data for one sign, defined by the ending hand gesture, be denoted by L. If $\boldsymbol{p_n} = [x_n, y_n, z_n]$ denotes a hand position along the hand motion trajectory made by the signer with $n < L$, then the distance between $\boldsymbol{p_i}$ and $\boldsymbol{p_n}$ is given by

$$dist(p_i, p_n) = \sqrt{(x_i - x_n)^2 + (y_i - y_n)^2 + (z_i - z_n)^2} \tag{5.8}$$

Let $\boldsymbol{p}_{n\text{-}1}$, $\boldsymbol{p}_{n\text{-}2}$, and $\boldsymbol{p}_{n\text{-}3}$ denote the three articulated hand positions before the n^{th} hand position along the hand motion trajectory. By computing $dist(\boldsymbol{p}_{n\text{-}1}, \boldsymbol{p}_{n\text{-}3})$, the average hand motion speed $\boldsymbol{v}_n$ at the n^{th} hand position is given by.

$$v_{n=}\frac{dist(p_{n-1}, p_{n-3})}{2} \tag{5.9}$$

Similarly, the average hand motion speed $\boldsymbol{v}_{n+m}$ for the m^{th} hand motion data after the n^{th} hand position data can also be computed, thereby enabling the determination of change in hand movement speed based on their difference.

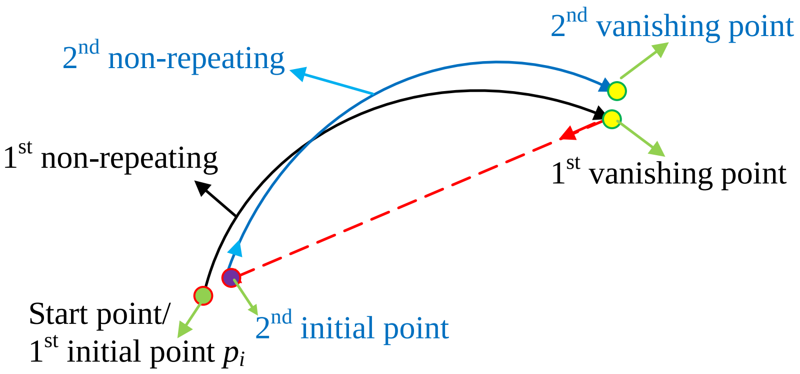


Figure 5-14. Right-up diagonal curving and repeating movement.

Furthermore, the relative hand moving radius over a specified time is obtained as the maximum moving distance within a number of subsequent articulated hand position data with respect to the n^{th} hand position. For example, the relative moving radius of the next m^{th} hand data is given by $max(dist(\boldsymbol{p}_n, \boldsymbol{p}_{n+\text{x}}))$ with $x \in (1,m)$.

In the implementation, $\boldsymbol{p}_n$ will be identified as the first vanishing position reached by the hand (see Fig. 5-14), if $\boldsymbol{v}_{n+m}$ is less than $\boldsymbol{v}_n$ and if the relative hand moving radius given by $max(dist\ (\boldsymbol{p}_n, \boldsymbol{p}_{n+x}))$ is less than a specified threshold. While the first condition corresponds the hand speed that slows down when the hand reaches the end point of each non-repeating hand movement segment, the second condition is to avoid false recognition caused by other movements with a similar speed pattern, such as a hand movement starting with a fast speed and slowing down in the middle of the movement trajectory.

Upon the detection of the first vanish position, a check is made to see if $n = L - 1$. If so, then it indicates that the ending gesture position has been reached and the hand movement is identified as a non-repeating movement. If it is not the case, subsequent hand position data are processed based on their distances to $\boldsymbol{p}_i$ to detect the second starting position for the repeated movement. If $dist(\boldsymbol{p}_i, \boldsymbol{p}_n)$ is less than both $dist(\boldsymbol{p}_i, \boldsymbol{p}_{n\text{-}1})$ and $dist(\boldsymbol{p}_i, \boldsymbol{p}_{n+1})$ by the specified threshold value after encountering the first vanishing hand position, then $\boldsymbol{p}_n$ is identified as the second starting position for the repeated movement. The search of the second vanishing position is then repeated, and the whole process could be repeated for the third time in order to reach the ending hand gesture with $n = L - 1$ (see Fig. 5-14).

5.3.3 Identification of Movement Planes

For two-plane movement classification, hand movements can be interpreted as parallel to either the floor plane or wall plane (see Fig. 5-15). This is achieved based on the angle made by the movement direction vector of each hand position along the hand motion trajectory with respect to the floor plane. As shown in Fig. 5-16, using the Cartesian world coordinate system as defined in Section 3.6.2, with the x-z plane forming the floor plane, x-y plane forming the wall plane, and centered at the starting hand position denoted by p_i, if p_n denotes a hand position along the hand motion trajectory, then the absolute angle made by it with respect to the floor plane is given by

$$\alpha_n = \tan^{-1}\left|\frac{(y_n - y_i)}{\sqrt{(x_n - x_i)^2 + (z_n - z_i)^2}}\right| \tag{5.10}$$

Since the angle of the hand movement direction vector with respect to the floor plane should be less than 45° for the floor plane movement, the hand movement is identified as parallel to the floor plane if all angle values computed for each hand position along the hand motion trajectory are less than 45° with respect to the floor plane, and parallel to the wall plane otherwise.

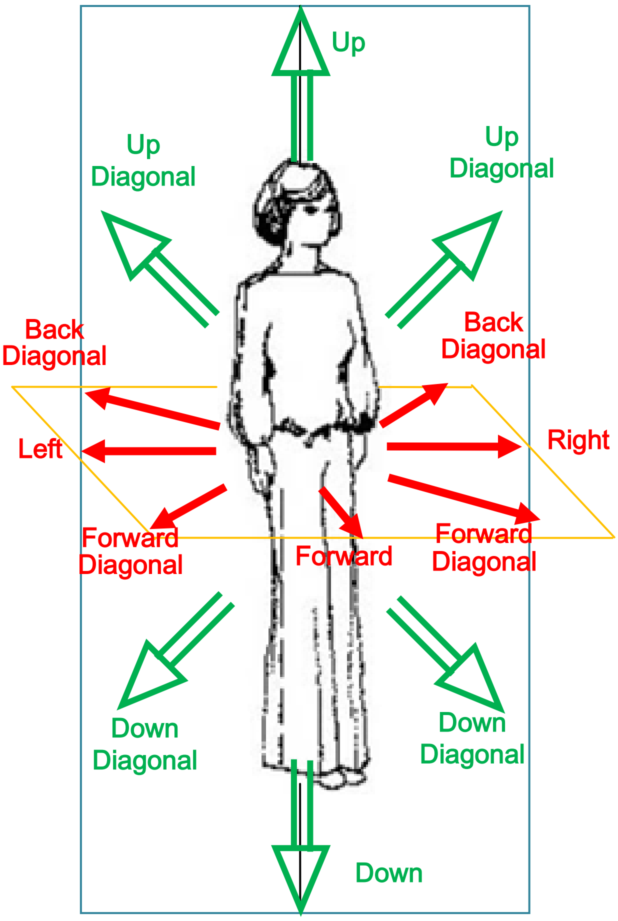


Figure 5-15. Wall/floor plane movements (modified from [250]).

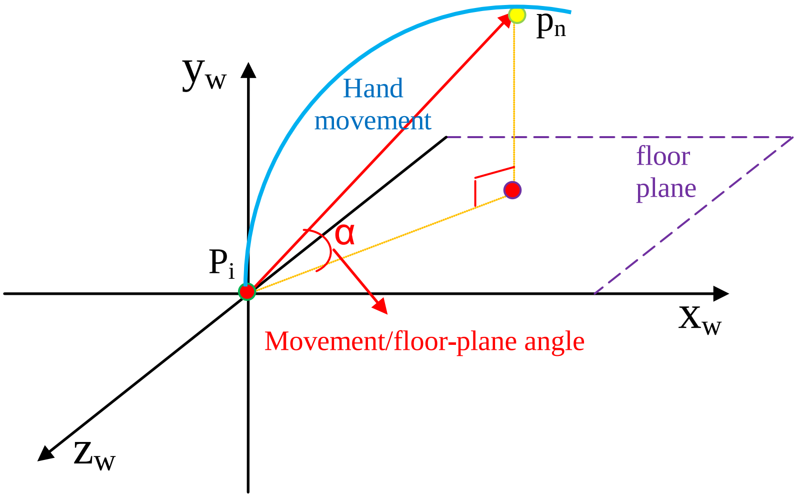


Figure 5-16. Movement direction vector for the determination of movement plane.

5.3.4 Identification of Movement Directions

For eight-direction movement classification, hand movements can be described as left/right horizontal movements, up/down vertical movements, or left/right up/down diagonal movements parallel to either the wall or floor plane as shown in Fig. 5-15. This is achieved based on the angle made by the farthest hand position with respect to the x-axis in the movement plane identified. As shown in Fig. 5-17, if p_v denotes the farthest hand position detected, then the sine and cosine of its angle denoted by β in the wall plane are given by

$$\sin\beta = (y_v - y_i)/\sqrt{(x_v - x_i)^2 + (y_v - y_i)^2} \tag{5.11}$$

$$\cos\beta = (x_v - x_i)/\sqrt{(x_v - x_i)^2 + (y_v - y_i)^2} \tag{5.12}$$

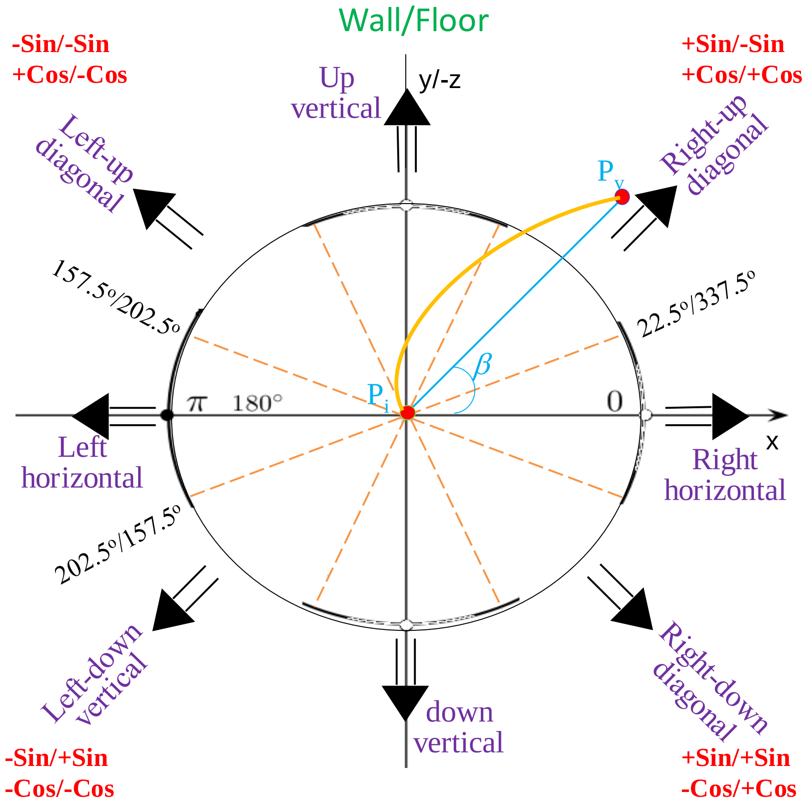


Figure 5-17. Movement directions in wall/floor plane.

To determine the angular value of β, the signs of sinβ and cosβ are used to find the quadrant in which β lies, since, for the wall plane, sinβ is positive for β in the left hand side and cosβ is positive for β in the top half, and for the floor plane sinβ is positive for β in the bottom half and cosβ is positive for β in the right hand side (see Fig. 5-17). Moreover, with each movement plane partitioned using 45° angular sectors centered at eight principal directions as shown in Fig. 5-12, the final movement direction is determined by finding the angular sector in which β lies based on the corresponding limits of sinβ values pre-computed for each angular sector.

5.3.5 Identification of Path Linearity

Hand motion paths in SW can be straight (linear) or curved (non-linear). The identification of path linearity is based on the maximum deviation distance and the corresponding deflection angles with respect to the straight line movement from the starting hand position to the farthest hand position. To explain the algorithm implemented, the hand motion path illustrated in Fig. 5-18 is used as an example, where the movement can be interpreted as a clockwise curved one in parallel to the wall plane. Let (x_i, y_i) and (x_v, y_v) be the coordinates of the starting and farthest hand positions denoted by p_i and p_v in the wall plane. The straight movement from (x_i, y_i) to (x_v, y_v) is described by a line equation with its slope and the y-intercept given by

$$k = \frac{y_v - y_i}{x_v - x_i} \tag{5.13}$$

$$c = \frac{x_v y_i - x_i y_v}{x_v - x_i} \tag{5.14}$$

For a hand position denoted by (x_n, y_n) on the curved path from (x_i, y_i) to (x_v, y_v), the shortest distance with respect to the straight line is given by

$$d_n = \left| \frac{kx_n - y_n + c}{\sqrt{k^2 + 1}} \right| \tag{5.15}$$

Using equation (5.15) to compute the distance from each point on the curved path to the straight line between (x_i, y_i) and (x_v, y_v), the maximum deviation point can be identified and used to compute the maximum deviation distance, denoted by d_s, and its deflection angles, denoted by α and γ, in Fig. 5-18. If the maximum deviation distance, d_s, is less than a threshold which is set as 10 cm by the rule of thumb, as well as its two deflection angles, α and γ, are both less than a threshold which is set to 22.5°, the motion path is identified as a straight one (linear). Otherwise, it is identified as a curved one (non-linear).

In the implementation, with the three vertices, the starting and ending position of the movement as well as the maximum deviation point, forming a triangle shape, the computation of its interior angle α at the movement starting position is done by using

$$\partial = \emptyset_d - \beta \tag{5.16}$$

where angle β and the maximum deflection angle θ_d with respect to the horizontal axis is given by equations (5.11) and equation (5.12).

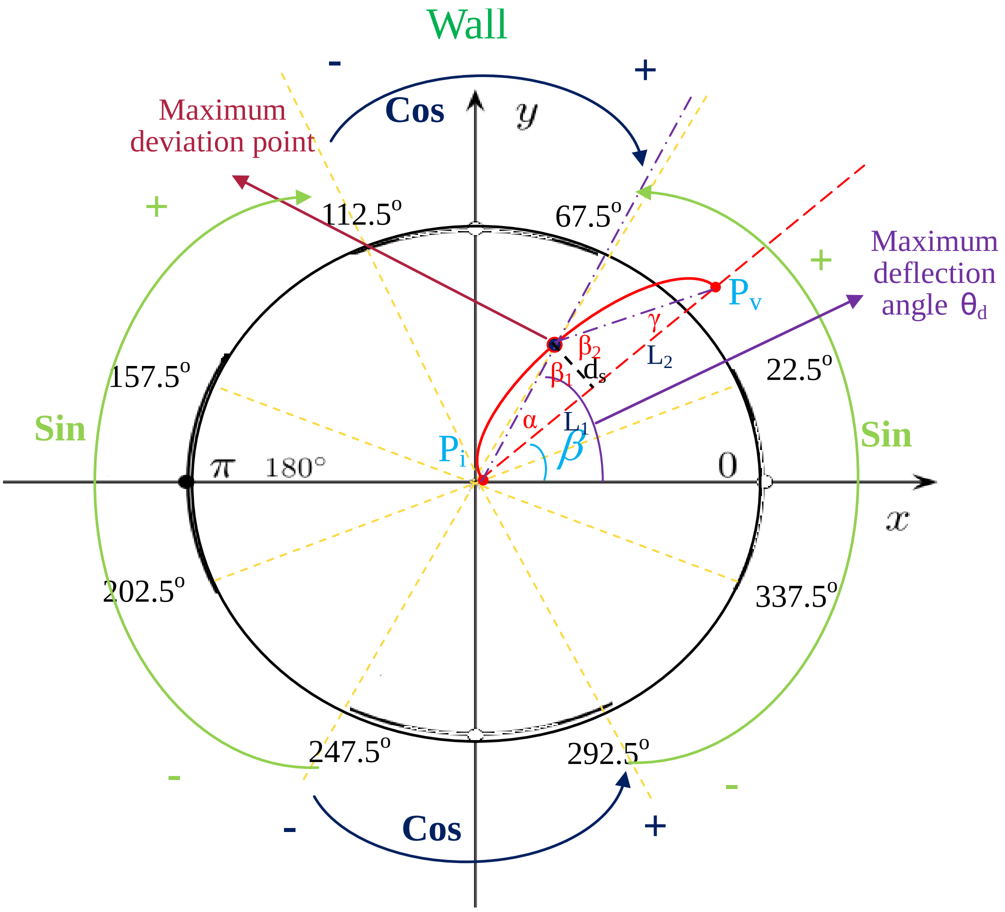


Figure 5-18. Path linearity identification example.

Let *L* denotes the straight line between the starting and ending points of the hand movement, let L_1 denote the length from the starting point to the point with the maximum deflection along *L*, and let L_2 denote the length from the point with the maximum deflection along *L* to the ending point (see Fig. 5-18). L_1 can be computed through equation (5.17), and the length of L_2 can be obtained by L-L_1, where *L* can be computed through equation (5.18). These will enable the other deflection angle denoted by γ at the movement ending position to be obtained from equation (5.19).

$$L_1 = \frac{d_s}{\tan \partial} \tag{5.17}$$

$$L = L_1 + L_2 = \frac{\sin \beta}{p_v.y} \tag{5.18}$$

$$\gamma = \tan^{-1}\left|\frac{d_s}{L_2}\right| \tag{5.19}$$

In addition, the deflection angle α, which is the difference between θ_d and β is also used to determine clockwise/anti-clockwise movement, since a clockwise movement in the left half or an anti-clockwise movement in the right half of the movement plane will result in sinθ_d greater than sinβ, whereas a clockwise movement in the top half or an anti-clockwise movement in the bottom half of the movement plane will result in cosθ_d greater than cosβ. Furthermore, in order to avoid the ambiguity problem caused by those curved movements near the horizontal axis to result in the same cosine value being produced and near the vertical axis to result in the same sine values being produced, the difference of sine values are used for β lying in the angular sector from 292.5° to 67.5° and from 112.5° to 247.5° and the difference of cosine values are used for β lying in the angular sectors from 67.5° to 112.5° and from 247.5° to 292.5° to determine clockwise/anti-clockwise movements, as shown in Fig. 5-18.

5.3.6 Recognition Performance Evaluation

To evaluate the hand movement recognition method developed for DSW, the hand movement data were captured by using the InterSense wrist trackers at a rate of 50 Hz, and various hand movements were performed with different combinations of movement planes and directions as well as repeated movement, trajectory curved and clockwise/anti-clockwise movement. To provide a visual feedback to the user making hand movements, a stereoscopic screen is used to display not only the 3D hand model with the movement but also the corresponding SW movement symbols along the direction of movement in the eight direction boxes. Some representative examples to demonstrate the hand movement recognition capabilities are shown in Figs. 5-19 to 5-25, where the first and second columns show the 3D movement trajectories made by the left and right hands, respectively, and the third column shows the visual feedback, where the centre box displays the SW symbol of the starting gesture and the surrounding eight direction boxes displays the corresponding SW symbols generated based on the hand movement direction. These examples include separate left and right diagonal forward movements by the left and right hands in the floor plane (Fig. 5-19); the left hand performing a diagonal backward movement in the left side of the floor plane with the right hand performing a diagonal up movement in the right side of the wall plane (Fig.

5-20); the left and right hands performing separate clockwise parabolic movements in the left and right sides of the floor plane along the horizontal direction (Fig. 5-21); the left hand performing a diagonal backward clockwise parabolic movement in the left side of the floor plane with the right hand performing a horizontal clockwise movement in the right side of the wall plane (Fig. 5-22); duplicated horizontal outward movements performed by both the left and right hands in the left and right sides of the wall plane (Fig. 5-23); the left and right hands performing the same movement which is an anticlockwise parabolic movement repeated twice along the horizontal direction in the left side of the wall plane (Fig. 5-24); and triplicated movements with the left hand performing repeated horizontal outward movements in the left side of the wall plane and the right hand performing repeated diagonal forward movements in the right side of the floor plane (Fig. 5-25).

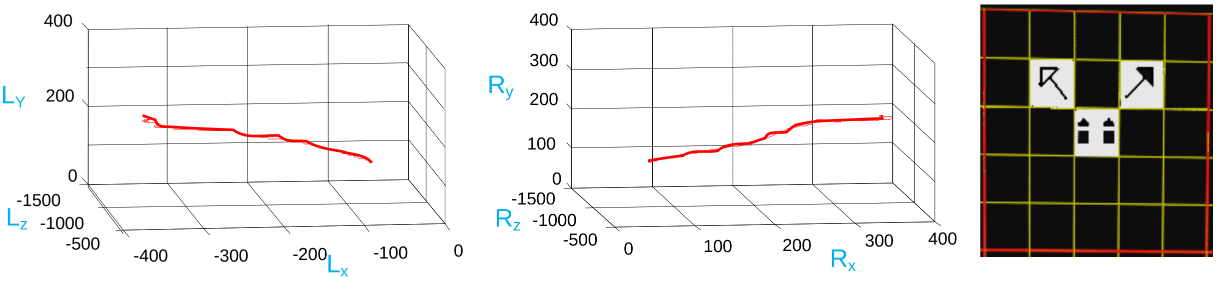


Figure 5-19. Left and right diagonal forward movements in floor plane.

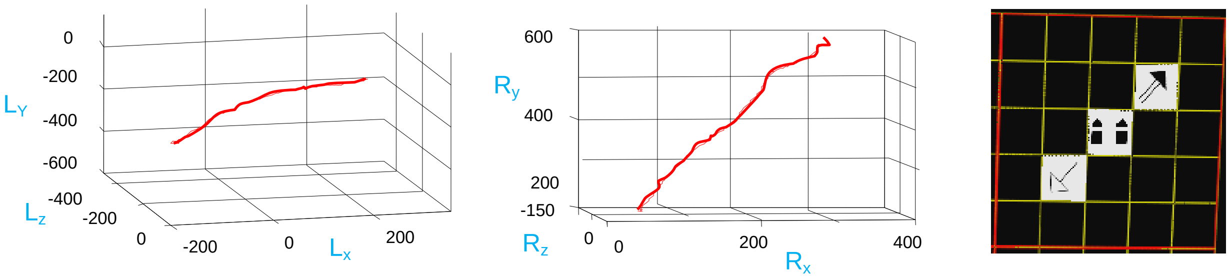


Figure 5-20. Left-back and right- up diagonal movements in floor and wall planes.

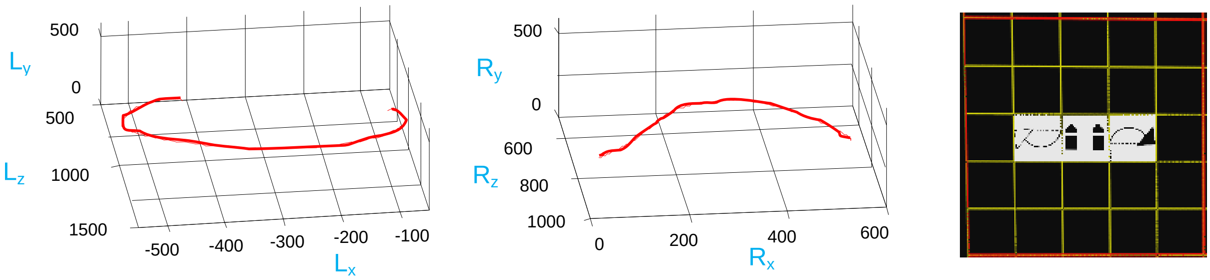


Figure 5-21. Left and right horizontal clockwise parabolic movements in floor plane.

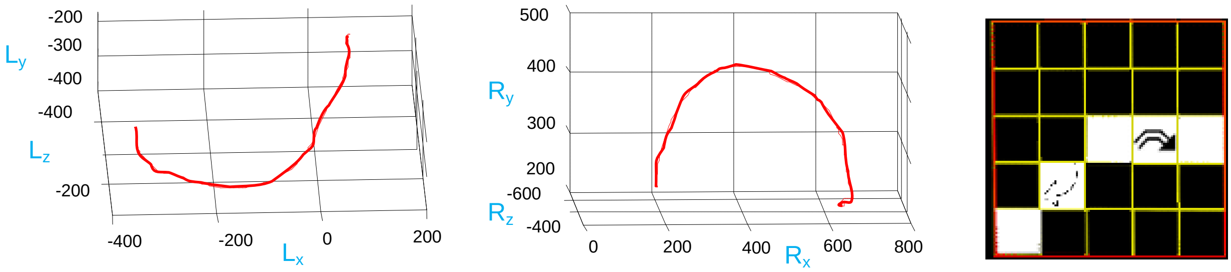


Figure 5-22. Left-back diagonal and right horizontal clockwise parabolic movements in floor and wall planes.

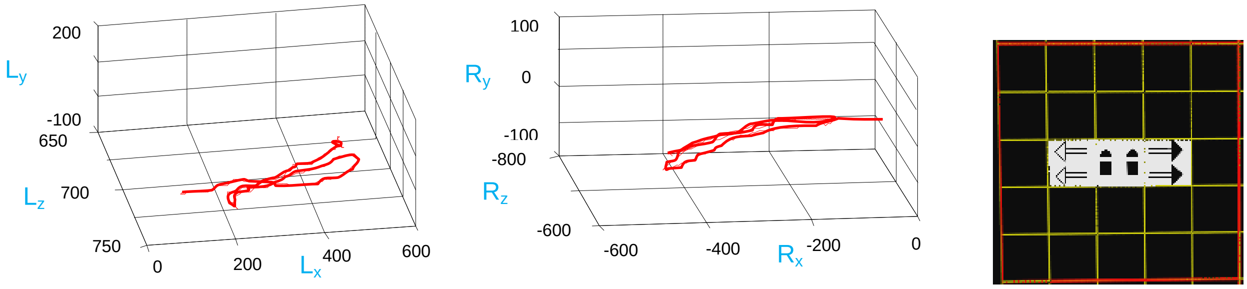


Figure 5-23. Left and right duplicated movements in wall plane.

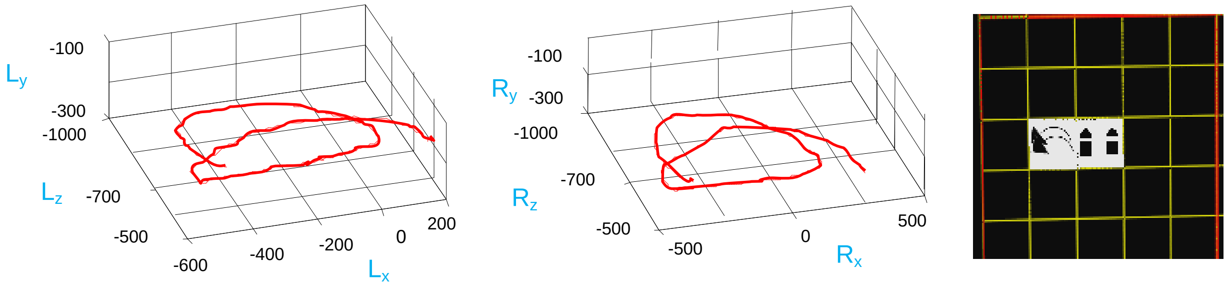


Figure 5-24. Synchronous parabolic anti-clockwise movements in floor plane.

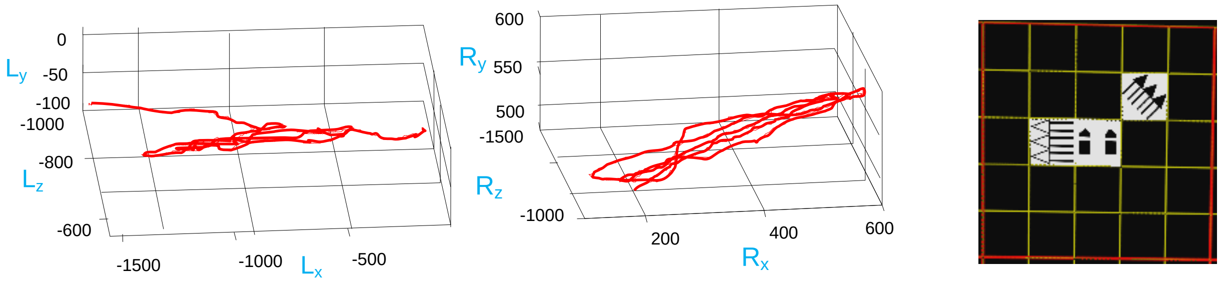


Figure 5-25. Triplicated left and right-up movements in wall and floor planes.

5.4 VISUAL INTERFACE FOR DIRECT SIGN WRITING

5.4.1 Visual Interface Design

To provide an effective visual feedback of the motion and recognition results of the hand gestures, a stereoscopic DSW interface has been developed to produce a stereoscopic view of the hand motions made in 3D and to generate the corresponding SW notations. There are three possible signing sequences for an ordinary sign language, namely:

1) One-stage gesture sequence with a starting hand gesture followed and finished by holding the starting gesture still for a brief moment; or
2) Two-stage gesture sequence with a starting hand gesture followed by the hand movement and finished by holding the hand still for a brief moment at the ending position; or
3) Three-stage gesture sequence with a starting hand gesture followed by the hand movement and finished with a new ending gesture at the ending position.

Since the maximum number of stages that a signing sequence may contain is three, a hierarchical sign-box consisting of a 5-by-5 lattice is constructed to enable three sets of symbols, corresponding to the hand starting gesture, hand movements and ending gesture made for each sign, to be displayed in a sequential manner from the innermost square to the outermost squares. This is illustrated in Fig. 5-26. Recognition of the starting hand gesture results in the corresponding symbols being displayed in the innermost yellow square; recognition of the subsequent movements by two hands results in their symbols being displayed in the sandwiched red squares along the movement directions; and recognition of the ending hand gesture results in the corresponding symbols being displayed in the outermost blue squares along the movement directions. For the implementation, the processing flowchart and the display sequence is shown in Fig. 5-27.

As the processing flow chart shown in Fig. 5-27, the state of the system ready to accept a new sign input is indicated to the signer by the green colour shown in the innermost square of the current sign-box. By continuously checking whether there is a hand gesture input, a valid hand gesture input will result in the green colour in the innermost square of the sign-box to be replaced by the corresponding SW hand gesture symbol, and a traffic light style indicator is activated at its neighbouring square on the right starting from red to indicate that it is ready to accept the following hand movement data input. If there is no hand movement detected within 1 second, the red colour will be replaced by a yellow colour. If the hands continue to be still with no further movement, the yellow colour will disappear after 0.5 seconds and a green colour will show up in the innermost square of the next sign box indicating the end of the current signing sequence and the start of the next signing sequence. However, if it is not the case with the hand movements detected within 1.5 seconds after the hand gesture input, checks will be made according to the other two possible signing sequences. For the case corresponding to the two-stage signing sequence with no ending gesture input, the traffic light style indicator is launched to give a count-down of the time the hands held still after their movements. It will go through the colour sequence of red for 1 second, yellow for 0.5 seconds and green in the next sign box with the recognised SW movement symbol displayed in the middle squares of the current sign box. For the case corresponding to the three-stage signing sequence, a valid new hand gesture input will be treated as the ending hand gesture, and a further check will be made to validate the obtained movement between the starting and ending gestures. If it is a valid movement, then the recognised SW movement symbol will be displayed in the middle squares in the current sign box, together with the ending SW hand gesture symbol displayed in the outermost squares along the movement directions as well as a green colour in the next sign box; if it is an invalid movement, no symbol will be displayed in the middle and outermost squares.

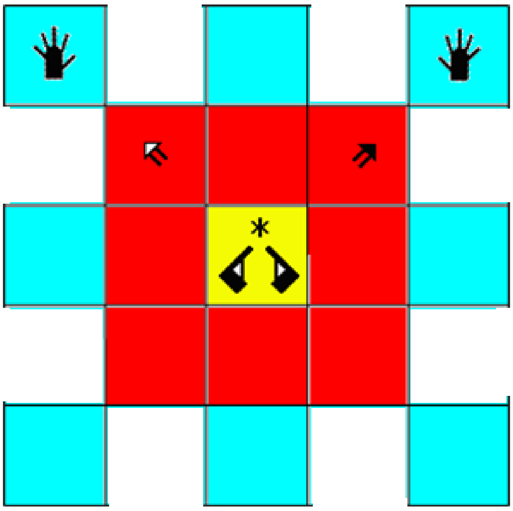


Figure 5-26. Sign-box showing diagonal up movements by two hands in wall plane.

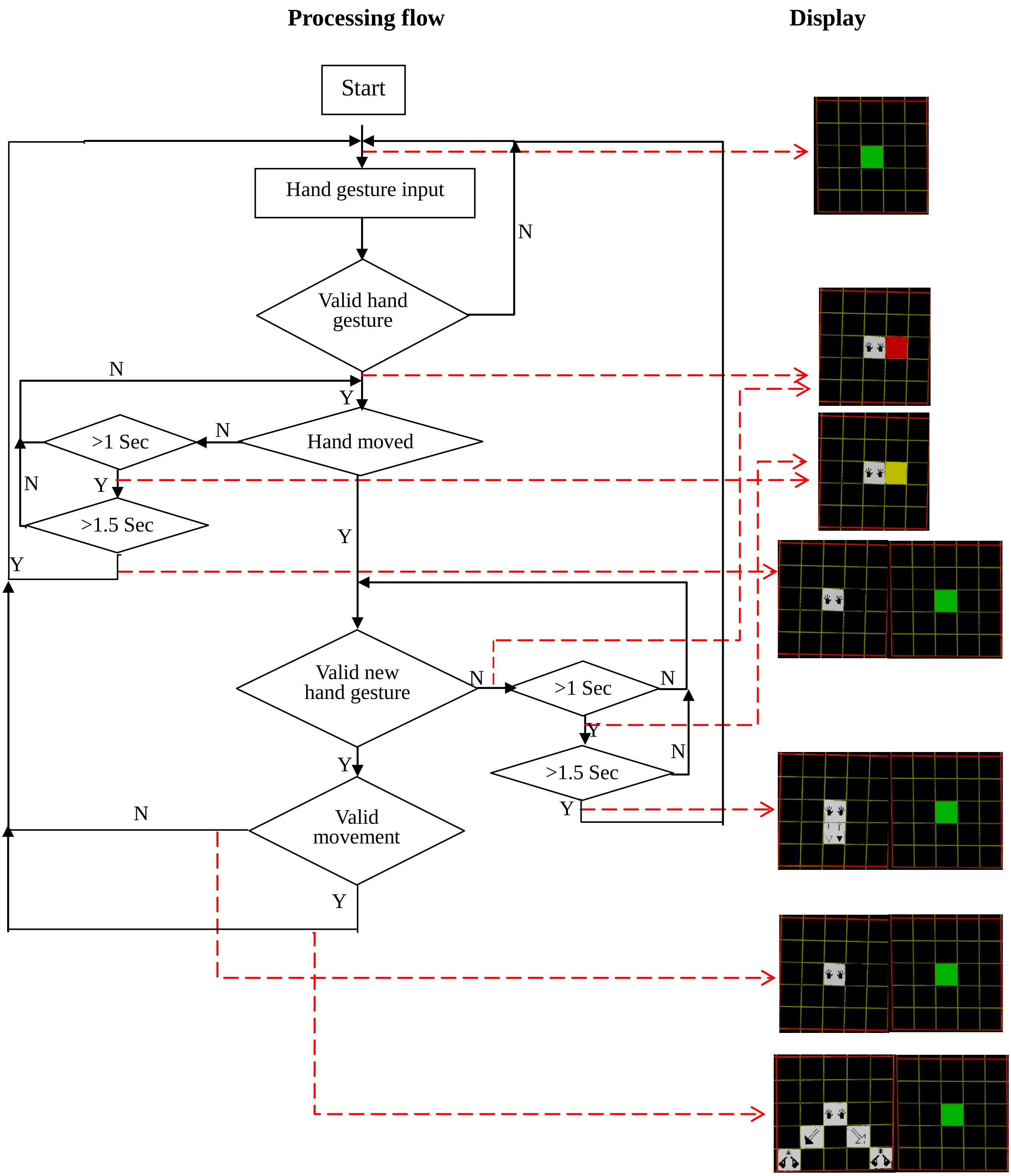


Figure 5-27. DSW interface processing flow and its display sequence.

5.4.2 Visual Interface Evaluation

To evaluate the visual interface for DSW, three representative signing inputs have been made and evaluated according to the three possible signing sequences.

From the visual perspective, by wearing a pair of light-weight polarised glasses, a signer is able to see the graphic models of his/her two hands floating in space, their motions in real-time and in stereoscopic mode with depth impression from his/her view point, as well as the corresponding SW symbols displayed in the 5-by-5 sign-boxes.

For the one-stage gesture input, the ASL gesture of ‘two’ was chosen and is shown in Fig. 5-28a. When the system was ready for input, a green was displayed in the innermost square of current sign-box (see Fig. 5-28b). Soon after the signer made the gesture of ‘two’, the green was replaced by the SW symbol of ‘two’ and a red colour was displayed on the right side of the SW symbol of ‘two’, as shown in Fig.5-28c. With the signer holding the hand gesture still, the red colour turned to yellow after 1 second (see Fig. 5-28d), and a green colour appeared in the innermost square in the next sign-box after 1.5 second (see Fig. 5-28e).

For the two-stage gesture input, the ASL gesture of ‘snow’ was chosen and is shown in Fig. 5-29a. Similarly, once the signer posed the ‘snow’ hand gesture, the hand symbol of ‘snow’ was displayed in the innermost square, and the square on its right turned red (see Fig. 5-29b). Subsequently, with both hands of the signer making falling-down movements followed by holding still at the ending position, the red colour turns to yellow in 1 second (see Fig. 5-29c), and after another 0.5 second, a green colour is displayed in the innermost square of next sign-box, with current sign-box displaying the starting hand gesture in the innermost square as well as the movement around its middle squares (see Fig. 5-29d).

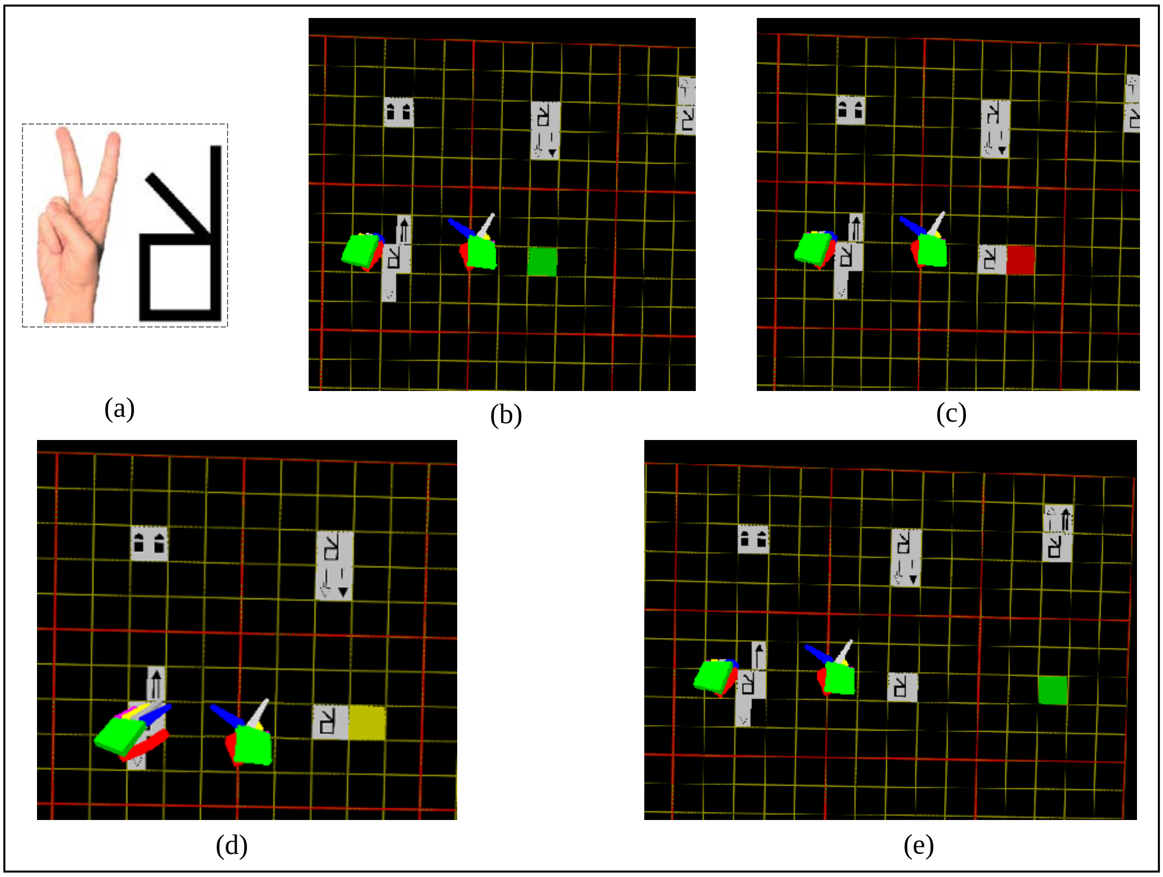


Figure 5-28. Interface output for ASL gesture of ‘two’.

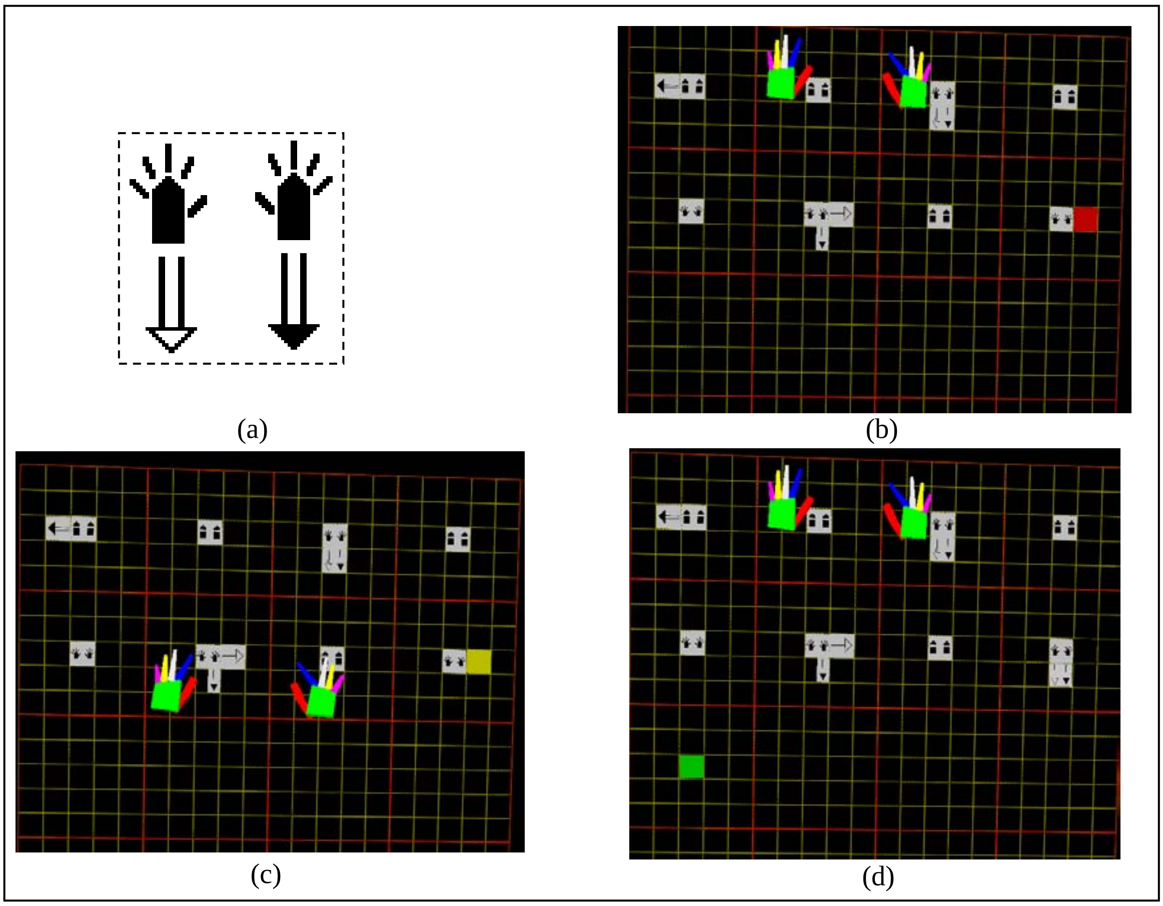
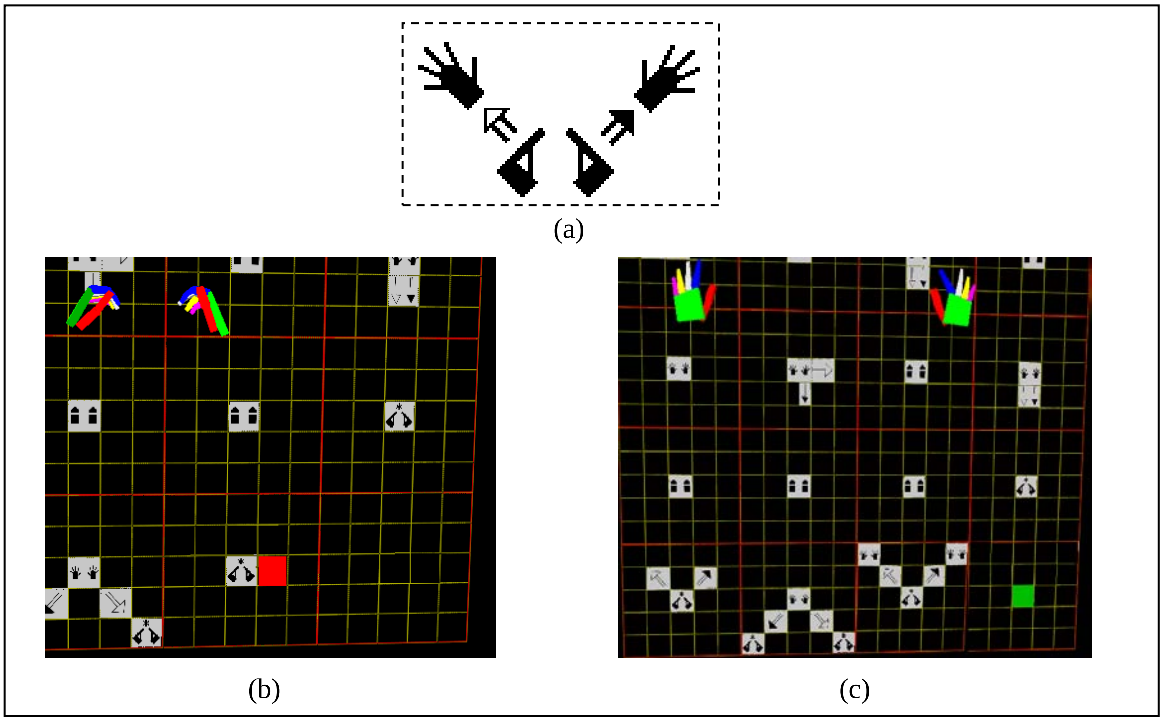


Figure 5-29. Interface output for ASL gesture of ‘snow’.

Finally, for the three-stage gesture input, the ASL gesture of ‘bright’ was chosen and is shown in Fig. 5-30a. With the signer making the starting gesture, the green colour in the innermost square was replaced by the SW hand symbol and a red colour was displayed in the right square next to it, as shown in Fig. 5-30b. After the following hand movement, the signer makes the ending hand gesture, and resulted in the red colour disappeared, the SW hand movement symbols being displayed in the two diagonal direction boxes above the innermost square, the SW ending hand gesture symbols being displayed in the two outer squares along the two diagonal direction, as well as a green colour displayed in the innermost square of the next sign-box (see Fig. 5-30c).


(a)

(b)

(c)

Figure 5-30. Interface output for ASL gesture of ‘bright’.

5.5 SYSTEM EVALUATION AND RESULTS

Since SW is more commonly used for ASL and still under development for BSL, evaluation of the developed system in terms of its usability and performance was conducted based on some selected ASL SW texts. A small section of ASL SW texts was selected for system demonstration from the first page of a children’s book, “Frosty the

Snowman", available from the SW website [256]. Its SW notations and corresponding texts are shown in Fig. 5-31, which contains various hand configurations with fingers open, half-open, close, abduction and adduction, as well as various hand movements with repeated movements, movements on wall/floor planes, movements on different directions, and linear/non-linear movements. Since the 'children' gesture is a special one containing multiple peaks in its movement, the implementation made for its recognition is shown in Appendix C.

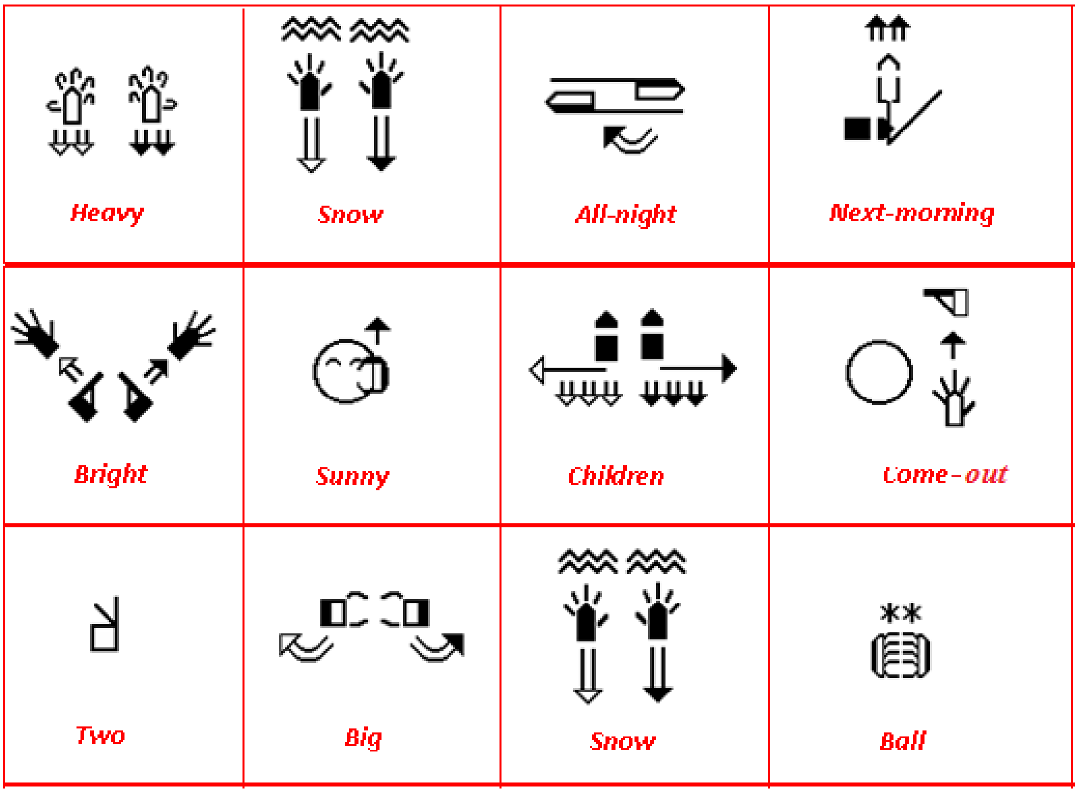


Figure 5-31. First page of the SW book 'Frosty the Snowman'- text: 'All night there was a lot of snow. The next morning was bright and sunny. The children came out and made 2 large snowballs.' (from [256]).

To implement the system interface in accordance with the page layout shown in Fig. 5-31, a visual input page with 4-by-4 blocks of sign-boxes was created. Furthermore, to improve the system's usability in terms of sign-box editing, the visual interface include a manipulation button, 'Manip', to enable symbol selection and movement, a new page button, 'Next', and a trash box for deleting SW symbols entered (see Fig. 5-34).

From the user's perspective, it is required to wear a pair of wireless ShapeHand data gloves to provide hand gesture information, a pair of wireless InterSense tracking devices on the wrists to provide hand orientation and movement information, and a wireless InterSense head tracker to provide the head position and orientation information. A large screen with two back projectors operating in passive circular polarisation mode is used to provide a stereoscopic display. By wearing a pair of light-weight polarised glasses, the user is able to see the graphic models of his/her two hands floating in space, their movements in real-time and in stereoscopic mode with depth impression from his/her viewpoint, as well as SW symbols displayed in the 4-by-4 sign-boxes according to the dynamic hand gestures made. Furthermore, the virtual trash box and two buttons are floating on the right hand side of the 4-by-4 sign-boxes (see Fig. 5-35).

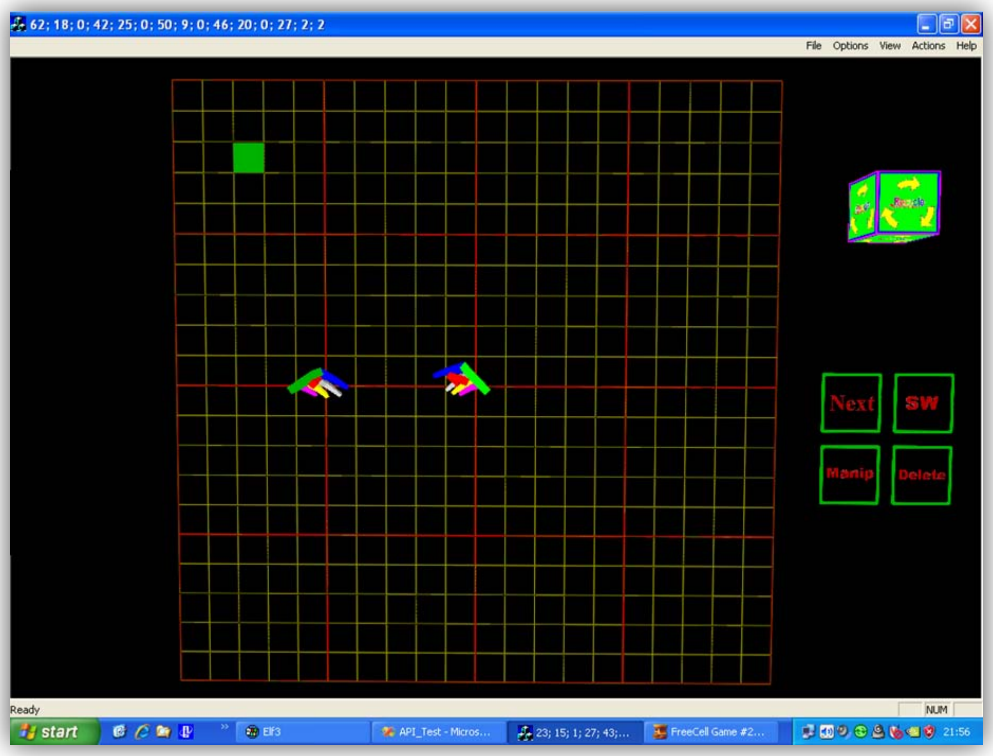


Figure 5-34. DSW visual interface.

In the input mode, the corresponding SW symbols will appear in the active 5-by-5 sign-box from the innermost square to the outermost squares according to the dynamic hand gesture made, and follow the input order from left to right and from top to bottom. After signing one page, a virtual touch of the virtual ‘next’ button enables the system to load a new page for continuing signing. Also, a virtual touch of the virtual ‘manip’ button enables the user to delete any SW symbols entered by selecting and dragging them into the trash box.

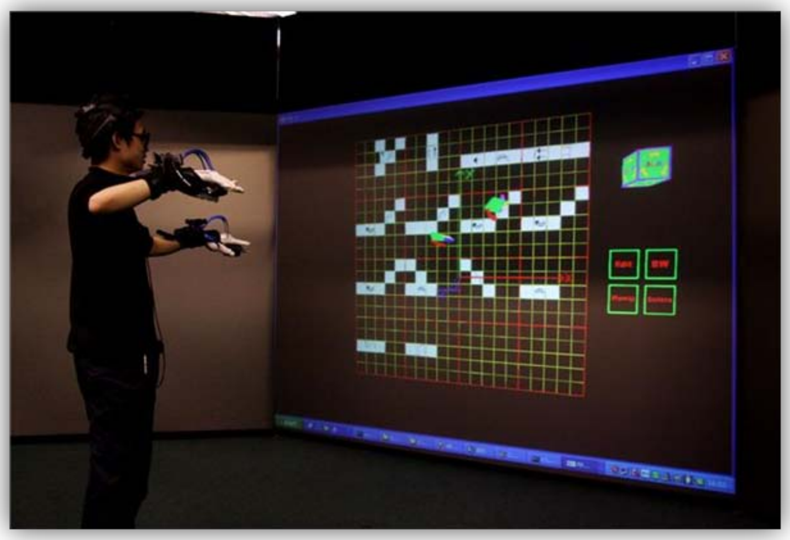


Figure 5-35. User interacting with DSW system.

To access the system's performance, accuracy tests were conducted. Three representative hand gestures were repeated ten times by a user according to the three signing sequences discussed in Section 5.4.1, namely, one-stage gesture 'two', two-stage gesture 'snow' and three-stage gesture 'bright'. The screen-shots of the results are shown in Figs. 5-35 to 5-37, and the recognition rates are listed in Table 5-2. With the signing stages denoted by $\boldsymbol{S_n}$, where n = 1, 2 or 3, and the number of hands involved in each stage denoted by $\boldsymbol{L_n}$ that can be 0, 1 or 2, the recognition rate, ***R***, is defined as:

$$R = S_1 + S_2 + S_3; \qquad (5.23)$$

where

$$S_1 = \frac{1}{n}; S_2 = \sum_1^{L_2} \frac{1}{n \times L_2}; S_2 = \sum_1^{L_3} \frac{1}{n \times L_3}; \qquad (5.22)$$

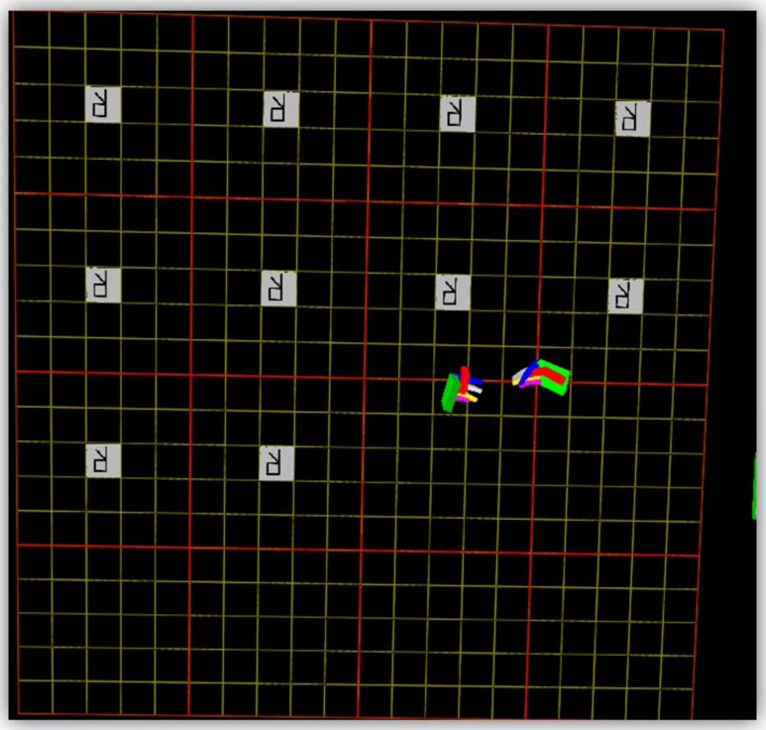


Figure 5-36. Result for repeated gestures of ‘two’.

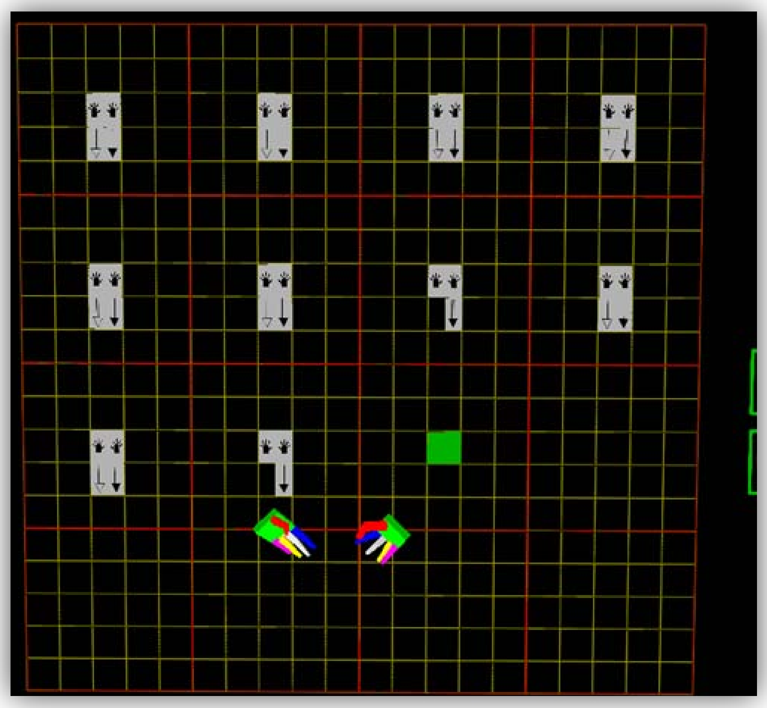


Figure 5-37. Result for repeated gestures of ‘snow’.

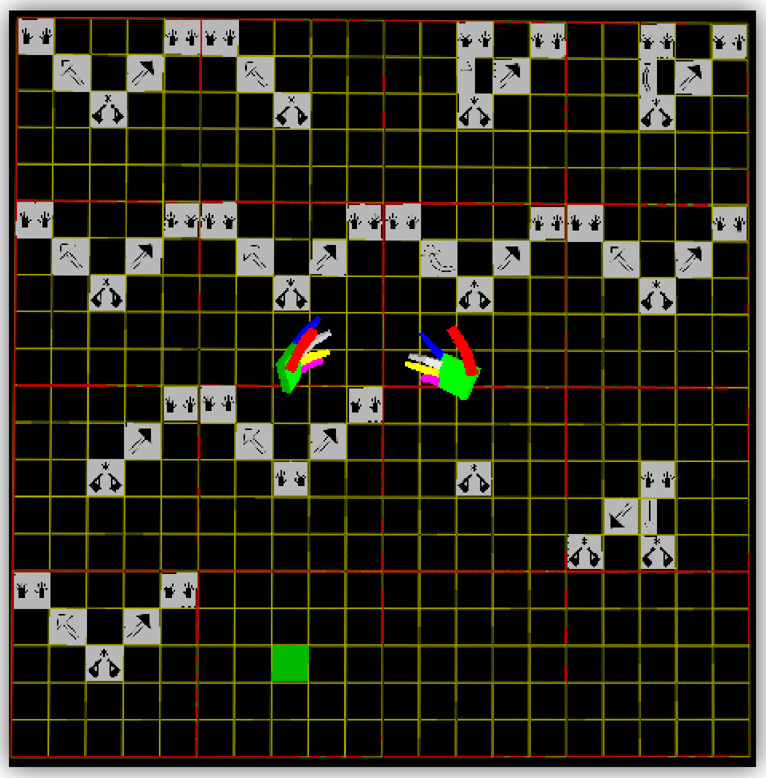


Figure 5-38. Result for repeated gestures of ‘bright’.

Table 5-2: Accuracy test results for three representative hand gestures

Gestures	two	snow	bright
Recognition Rate	100%	95%	88.3%

Tests have also been conducted by a user to make all the dynamic hand gestures to input the first page of the SW book, ‘Frosty the Snowman’, as well as repeating each gesture ten times to measure the error rates. Apart from the gestures of ‘two’, ‘snow’ and ‘bright’, which have been shown previously, the screen-shots of the results are shown in Figs. 5-39 to 5-48, and the error rates are listed in Table 5-3. While the accuracy is 80% (Fig. 5-39) for continuous one page input from ‘Frosty the Snowman’, it varies between the lowest of 80 % for repeated input of ‘come-out’ (Fig. 5-38) to the four highest of 100% for repeated inputs of ‘heavy’ (Fig. 5-40), ‘children’ (Fig. 5-44), ‘two’ (Fig. 5-46), and ‘ball’ (Fig. 5-48).

Table 5-3: Accuracy test results for DSW

Gestures	**all gestures**	**heavy**	**snow**	**all-night**	**next-Morning**	**bright**
Recognition Rate	80%	100%	95%	90%	90%	88.3%
Gestures	**sunny**	**children**	**come-out**	**two**	**big**	**ball**
Recognition Rate	98.3%	100%	80%	100%	92.5%	100%

From the results shown in Tables 5-2 and 5-3, the recognition accuracy is seen to decrease with the increase in the number of signing stages. The high recognition rate of 100% for single stage gestures reflects the advantage of the system which does not requiring highly precise figure posture due to the three level representation of finger posture (open, half-bent and closed). The lower recognition rate of 80% for three-stage gestures is mainly caused by movement recognition errors due to relatively large hand movement variability associated with higher degrees of freedom. The relative low recognition accuracy related to the gesture of ‘all-night’ may be due to occasional occlusion of the InterSense wrist tracker. As the trackers are attached on the back of the wrists and the movement of ‘all-night’ performed by the right hand with the back of the wrist facing down, result in the signal transmission between the ultrasonic transmitter mounted on the ceiling and the tracker receiver is blocked by the wrist. Moreover, the fixed threshold setting used to segment repeating movements based on the vanish points may lead to the failures to recognise the repeating movements of the ‘next-morning’ gesture, since a threshold setting that is too high may result in a slow moving hand with its moving radius lower than the threshold (wrong vanish point being recognised), and a threshold setting that is too low may result in a movement jitter of a still hand with its moving radius higher than the threshold (vanish point not being recognised).

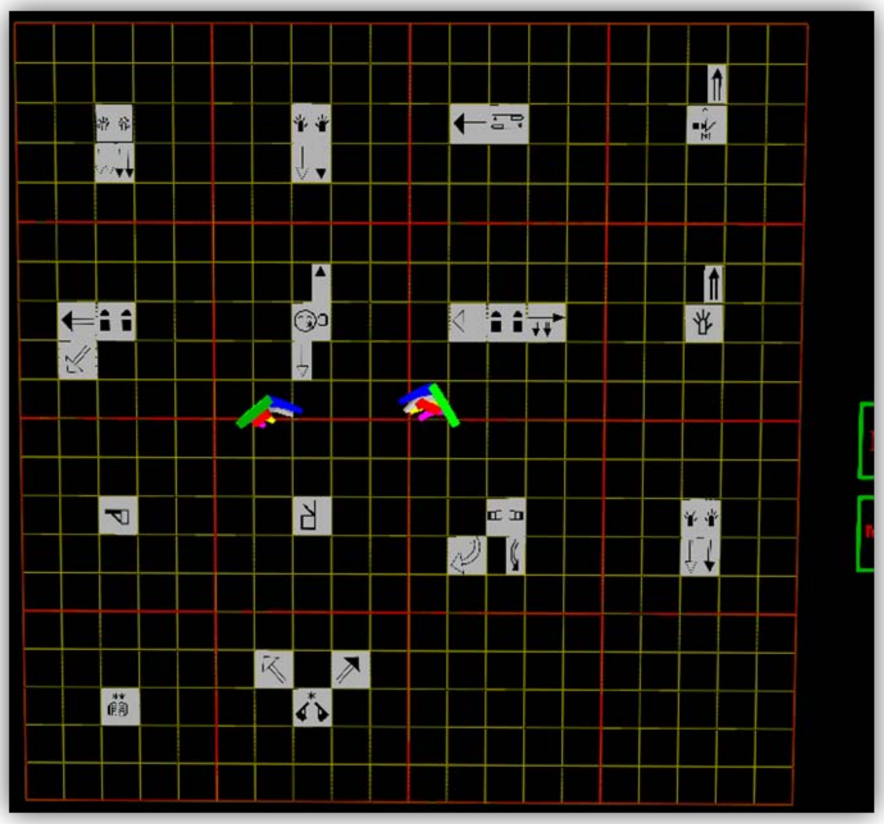


Figure 5-38. Results of continuous input of first page in ‘Frosty the Snowman’.

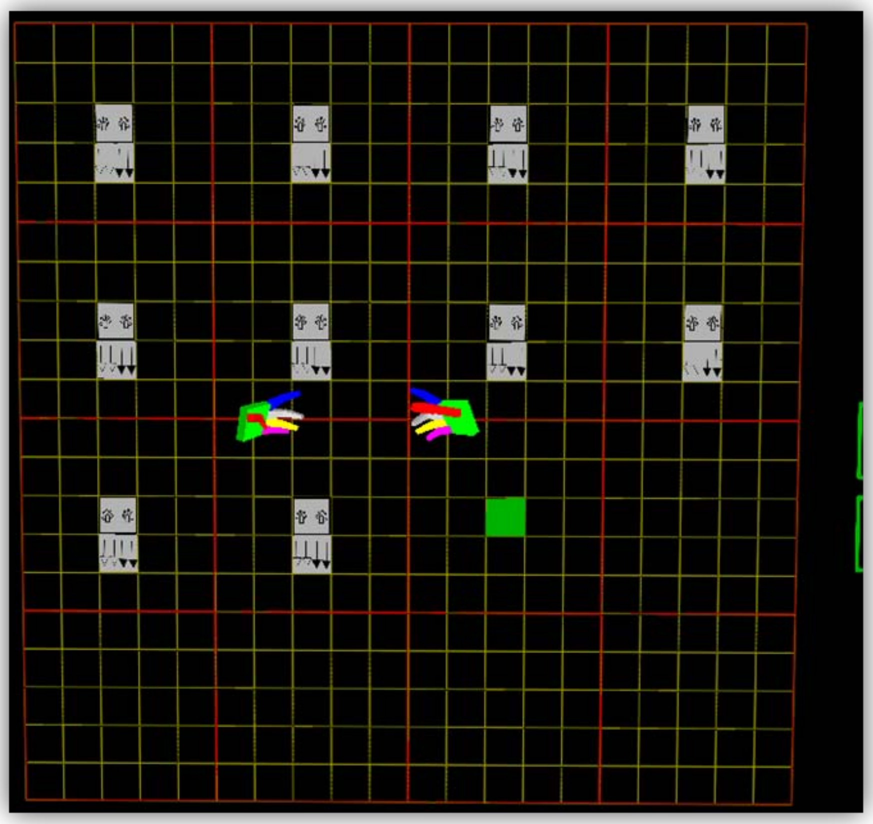


Figure 5-39. Result for repeated gestures of ‘heavy’.

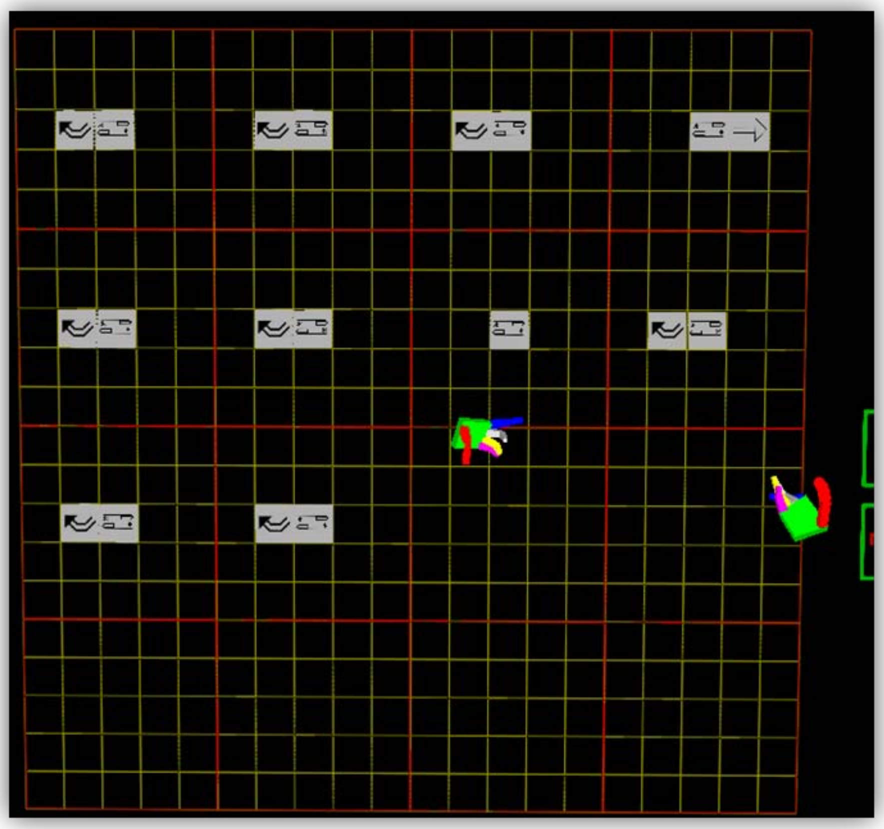


Figure 5-40. Result for repeated gestures of ‘all-night’.

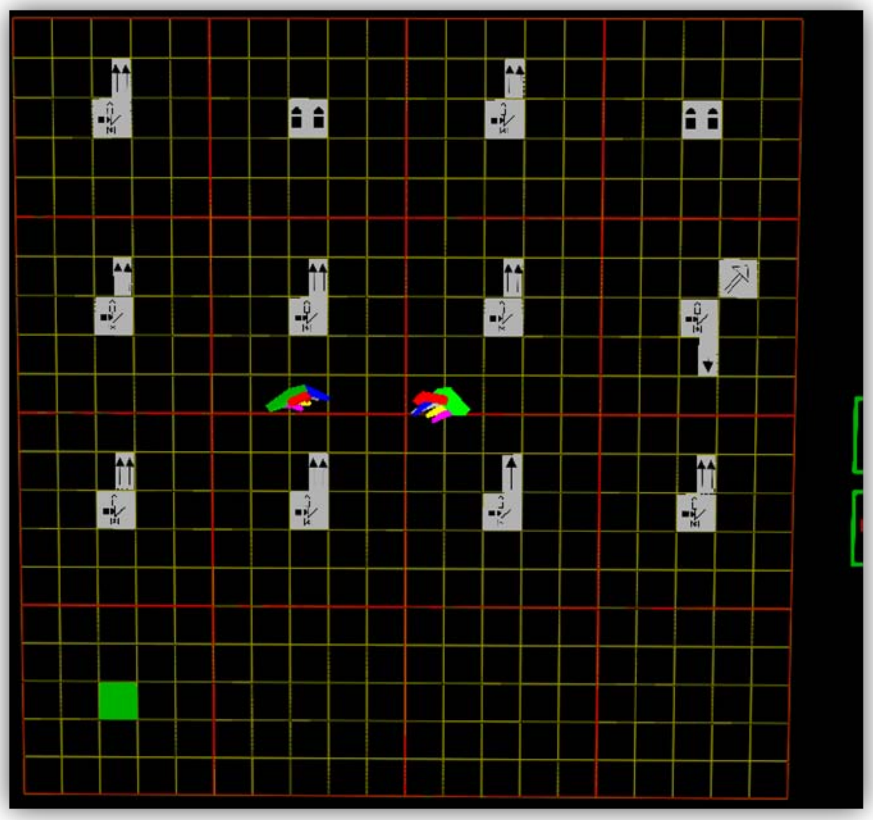


Figure 5-41. Result for repeated gestures of ‘next-morning’.

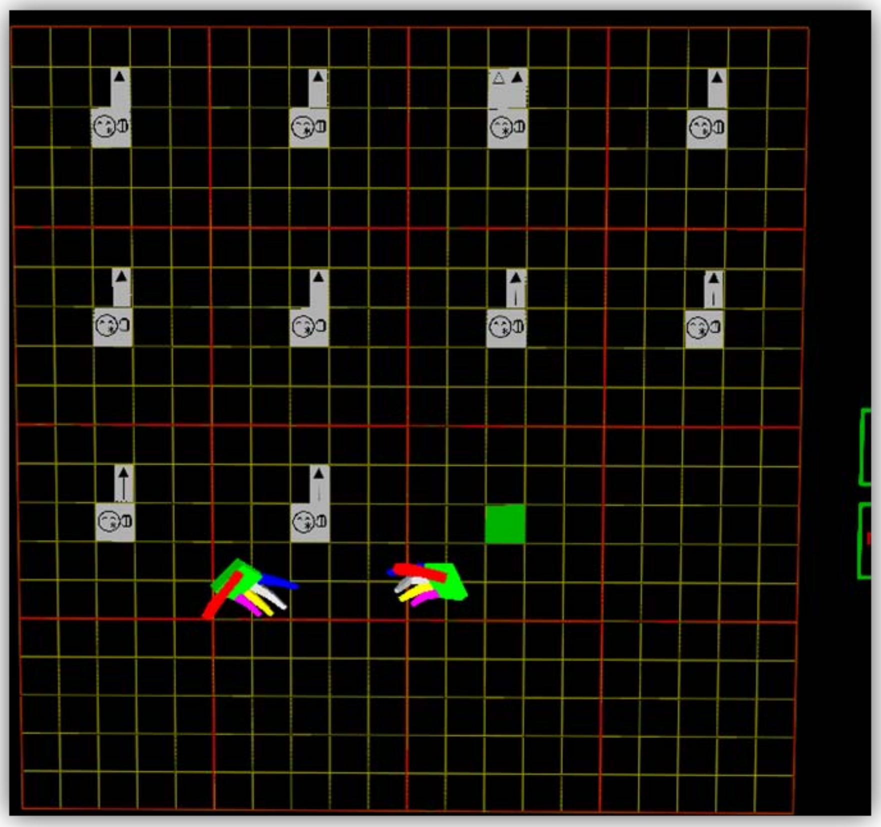


Figure 5-42. Result for repeated gestures of ‘sunny’.

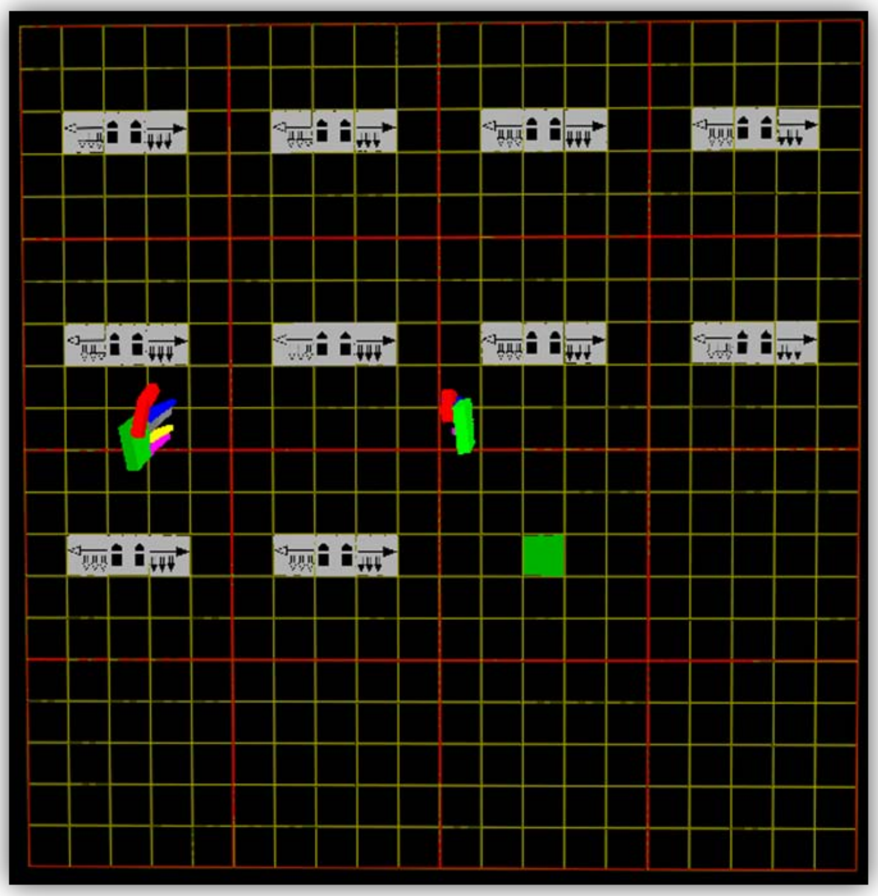


Figure 5-43. Result for repeated gestures of ‘children’.

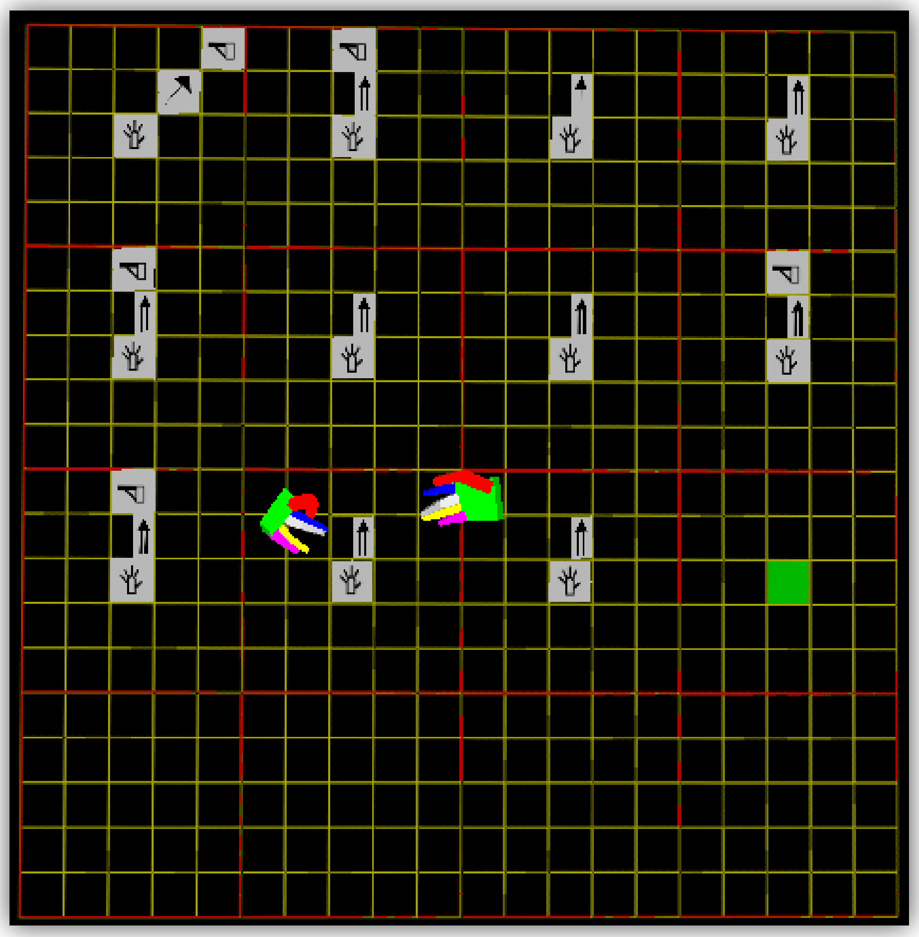


Figure 5-44. Result for repeated gestures of ‘come-out’.

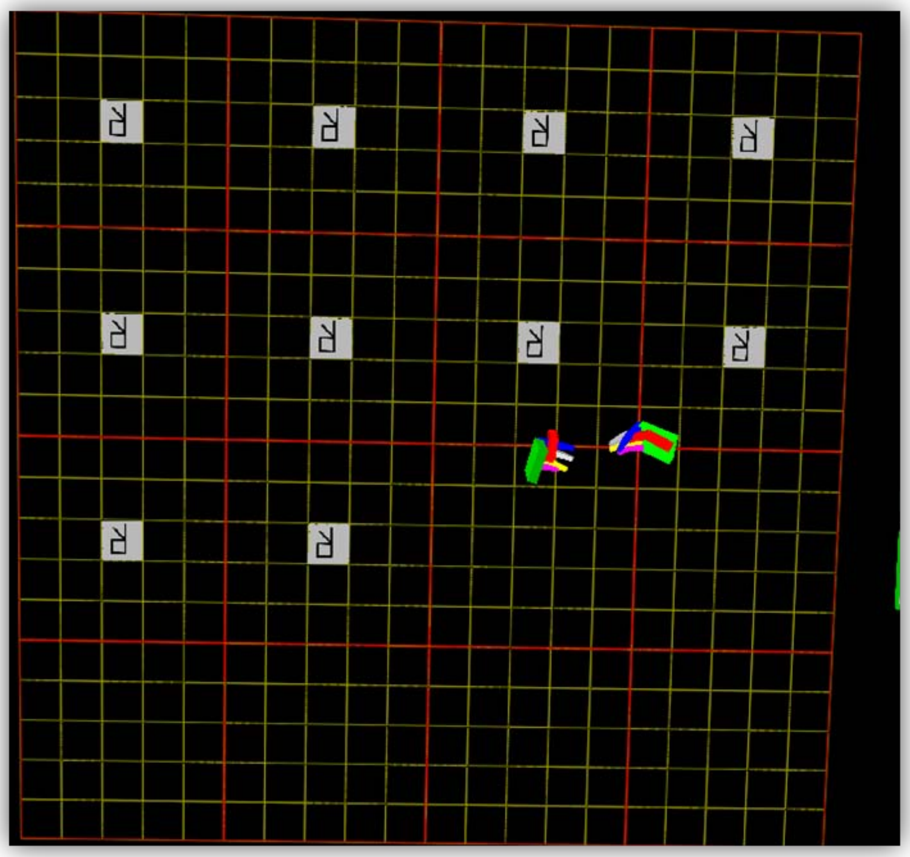


Figure 5-45. Result for repeated gestures of ‘two’.

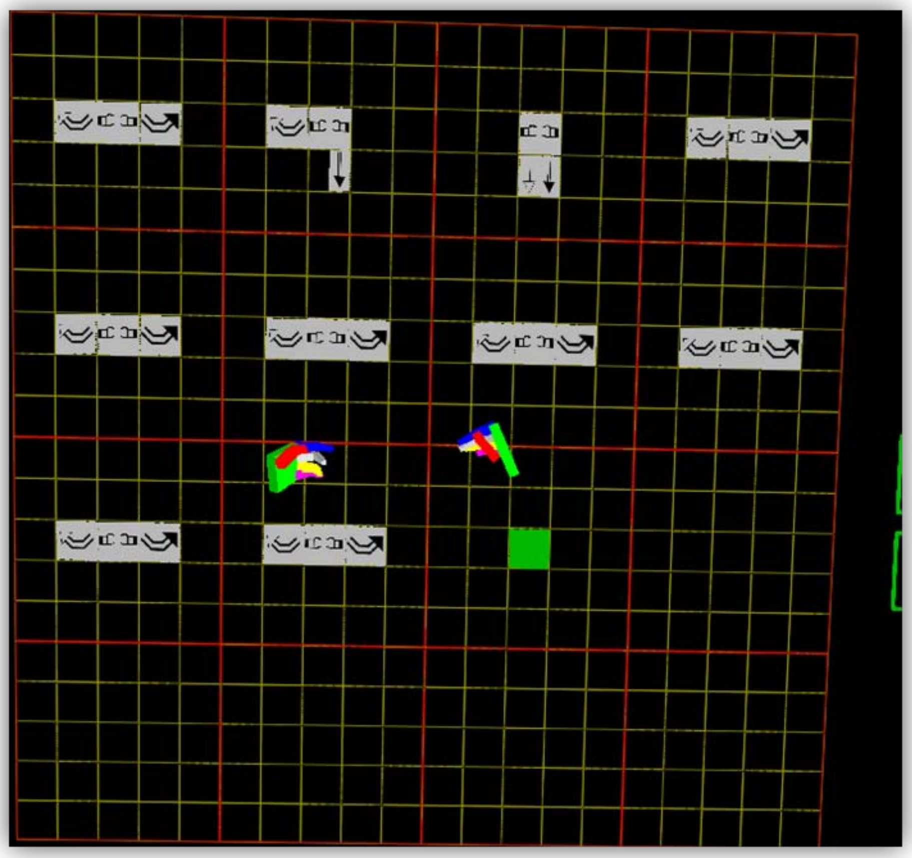


Figure 5-46. Result for repeated gestures of ‘big’.

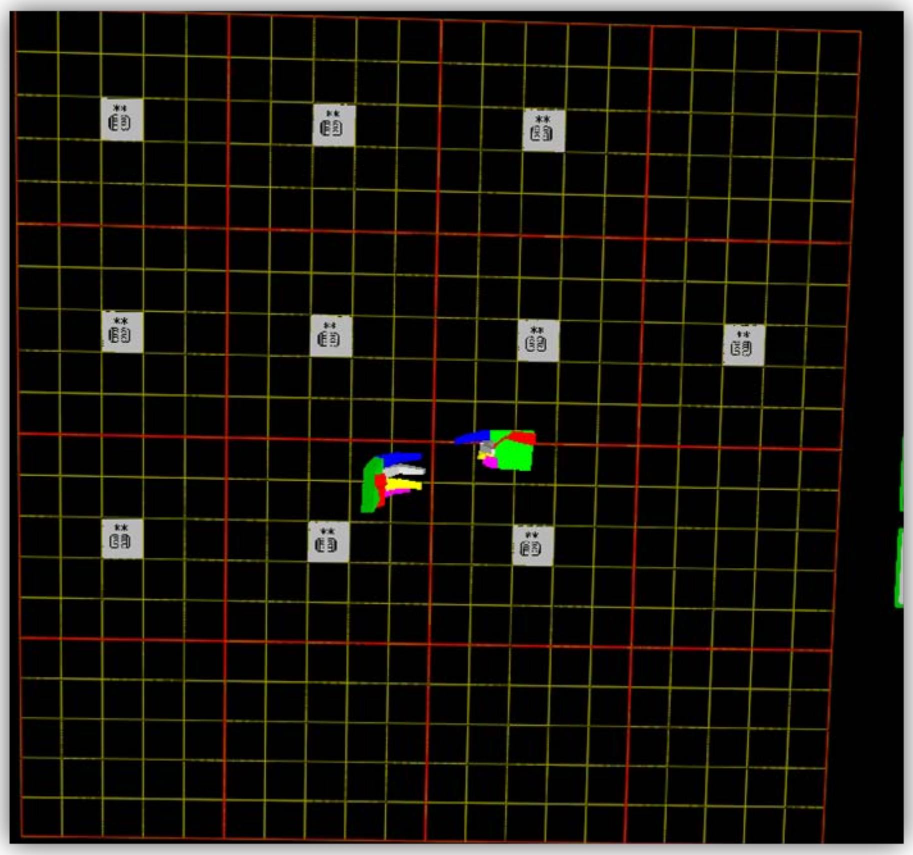


Figure 5-47. Result for repeated gestures of ‘ball’.

To compare the efficiency of the DSW system with the SignWriter keyboard and the hand writing methods, a test has been done by a user using these three different methods to input the first page of ‘Frosty the Snowman’ shown in Fig. 5-31. By recording the starting and ending time to complete the input, 4 minutes were recorded by using DSW system, 45 minutes by using the SignWriter keyboard, and 10 minutes by hand writing. It can be seen that the DSW is significantly quicker than the other two sign writing methods. Moreover, the user experience suggested that for the SignWriter and hand writing sign writing methods, the user needs to transcribe the sign gestures into SW symbols in his/her mind before typing or writing the symbol, which is not as direct as the DSW system.

Furthermore, to assess the system’s usability, two hearing-impaired sign language users, one has been using sign language more than 30 years and the other has been using sign language since he was born, have been invited to test and evaluate the system. One knew SW but does not use it often, and the other never heard about SW. The testing exercises include learning and testing of both the DSW system and the SignWriter keyboard. According to their feedback, the SignWriter keyboard is too complicated to learn and difficult to use, the DSW system barely requires learning and could be used directly. Also, compared to the SignWriter keyboard, the DSW system enables a high signing speed and is easier to use. For future system applications, they suggested that, by integrating the DSW system with the facial expression recognition techniques, it could be put to good use for the SW publishing industry and pre-school linguistic education. Moreover, it could also be applied for message based communication such as e-mail.

5.6 CONCLUDING REMARKS

This chapter reports the development and assessment of a basic but unique and effective DSW system based on the developed system platform discussed in the previous chapters. By using the ShapeHand data gloves to capture the hand posture data and the InterSense wrist tracking devices to capture the hand position and orientation data, together with the developed hand gesture and movement recognition algorithms, the system is able to recognise the user’s dynamic hand gestures and transcribe them into SW symbols directly. In addition, by using a head tracking device and a large stereoscopic display as well as the developed DSW visual interface, the user is able to visualise the SW transcription results immersively and interact with them effortlessly. To demonstrate the system’s performance, evaluation tests have been conducted both systematically and holistically. For systematic evaluation, each system functional component has been tested and evaluated individually. For hand gesture recognition, the results showed that the system was able to recognise the finger configurations of open, half-bent, close, abduction and adduction, as well as eight hand orientations in both the wall and floor planes and three palm facing directions. For hand movement recognition, the results showed that the system was able to recognise the hand movements on different planes, directions as well as repeated movements, curved trajectory and clockwise/anti-clockwise movements. For the DSW visual interface, the results showed that the system was able to provide correct visual feedback according to the user’s input with different signing sequences. For holistic evaluation, the system is demonstrated to be able to recognise the user’s dynamic hand gestures based on a SW ASL demo page selected from a SW children’s book, as well as to display the corresponding SW symbols onto the screen with good accuracy.

The usability comparison study with other sign writing methods has also been conducted including two hearing-impaired users. From the users’ feedback, the DSW system was recognised as a quicker and easier method for sign writing.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORKS

6.1 INTRODUCTION

Under the project title “Real-time human computer interaction based on tracking and recognition of dynamic hand gestures”, the research was carried out in three specific areas, namely, the design of a real-time dynamic hand gestures tracking and recognition HCI system, the construction of an immersive and interactive VR environment for object /data visualisation and manipulation, and the development of an application-oriented system based on DSW. The salient features of the work and the original contributions of the research in these three areas are highlighted in the following three sections of this final chapter, respectively. Further works in these three areas are also recommended. The research has produced two conference papers and one journal paper accepted (see Appendix A).

6.2 REAL-TIME HCI SYSTEM BASED ON DYNAMIC HAND GESTURES TRACKING AND RECOGNITION

One original contribution of this research is the integration and development of a unique system for automatic tracking and recognition of dynamic hand gestures in real-time. The uniqueness lies in the combination of the ShapeHand data gloves to capture hand finger joint position and orientation, the InterSense tracking devices to provide hand wrist position and orientation, and a stereoscopic display to provide immersive visual feedback. Each sub-system has been investigated and been evaluated through performance tests (See Section 3.2.3, 3.3.3 and 3.4.3). Together with the developed algorithms, the integrated system is demonstrated to be able to track and recognise complex hand gestures and movements correctly in real-time through significant system performance testing (See Section 3.7). Particularly, the system is able to capture the hand gesture change at a speed of approximately 14 times per second and to capture the hand movement change with a latency of 94 ms.

Further research in this area includes accuracy improvement of the hand posture data from the ShapeHand data gloves and hand position data from the InterSense tracking devices. While the former suffers from a gradual sensor tape slippage from the calibrated position at the start of the use, the latter suffers from position drifting away due to poor signal reception. Although these problems could be reduced by better hardware designs, such as a better-fitting glove to avoid sensor tape sliding and better signal coverage to reduce drifting, they could also be reduced by adopting more sophisticated tracking algorithms. For gradual slippage of ShapeHand sensor tape, the trend of the changes in finger joint angle data for the same gesture and the finger joint angle data responsible for incorrect symbols identified by the user during input could be used to adjust joint angle threshold settings, thereby achieving continuous calibration with self adaptation. For hand position drifting, the relatively accurate head position could be used as a reference to define the tracking volume of hand movements through a pre-calibration process based on the user arm lengths, thereby preventing the hand position from occurring outside the tracking volume. Since the system was developed with a focus on achieving good real-time system performance, further system development could be carried out to investigate the use of more sophisticated tracking methods presented in Chapter 2, such as using the Kalman filter for hand posture data tracking to avoid the false hand finger shape through using noise-contaminated sensory data and HMM network for dynamic hand gesture recognition, without significantly affecting speed performance.

6.3 VIRTUAL OBJECT MANIPULATION

As the first and a simple application of the real-time HCI system developed, a unique immersive and interactive environment was designed and implemented by the author for virtual object/data manipulation and visualisation based on hand gestures. Using stereoscopic display of virtual objects based on graphics and volume data, the environment is demonstrated to allow a user to select, translate, rotate, scale, release and visualise these virtual objects in 3D space by using natural hand gestures in real-time. Particularly, the system performance testing showed that the system is able to operate at a minimum of 54 frames per second when rendering the graphic cube and at a minimum of 31 frames per second when rendering a real CT medical volume data, which demonstrated its usability in real applications compared to other to other related works (see Section 4.4).

Many further works in this area are possible which include improvement of the realism and immersiveness of the scene by applying more sophisticated OpenGL rendering effects (such as lighting effects); improvement of the system rendering speed by adopting the advanced GPU (Graphic Processing Unit) with embedded rendering algorithms; and implementation of more manipulation methods (such as sculpting) and visualisation methods (such as volume slicing). All of these will enable the system to provide a highly realistic and application specific environment for various uses in a wide range of sectors (such as education, manufacturing, medical and entertainment).

6.4 DIRECT SIGN WRITING

As the second and more sophisticated application of the real-time HCI system developed, a novel prototype of DSW was designed and implemented by the author for hand gesture based sign input. To the best knowledge of the author, there is no any other direct sign writing system is available currently.

With the hand posture data gathered by the ShapeHand data gloves as well as the hand orientation and movement trajectories acquired by the InterSense wrist trackers, the DSW system is shown to be able to produce the corresponding SW symbols by recognising dynamic and complex hand gestures automatically. The DSW system includes a special visual interface to enable the user to see his/her hand movements as well as the SW symbols produced in each stage of the signing sequence. Various tests have been conducted to assess the accuracy and effectiveness of the system. Compared to other sign writing methods, the system has advantages in terms of learning and signing speed. Particularly, the system is seen by hearing-impaired users as forming a good basis for applications in the SW publishing industry, linguistic education and message based commutation (see Section 5.5).

Although it is not the focus of this research, a novelty is also lying in the implementation of BSL fingerspelling recognition, which may propose another research direction. By computing the distance between the index fingertip on the right hand and the fingertips on the left hand in 3D space, the system is demonstrated to be able to recognise five vowel-letters and gives the user a immediate visual feedback once any of the letters has been recognized (See Appendix B).

The successful demonstration of the DSW system prototype and the complexity of sign writing lead to a range of possible future works to extend the recognition to more hand postures, hand orientations and hand movements. For the recognition of more hand postures, it needs to include thumbs which have more degrees of freedom compared to other four fingers and tackles the problem of lower robustness of its data due to poor sensor tape attachment. For the recognition of more hand orientations, it needs to include identification of the hands in the diagonal plane [250]. For the recognition of more hand movements, it needs to include extraction and classification of more trajectory features. Although all of these could be done, investigation needs to be carried out to minimise the increase in the computational costs for these developments without a significant loss of accuracy.

With the prototype system providing an excellent platform towards DSW, the ultimate goal is to combine it with facial expression recognition and body posture recognition methods to enable full human-computer-interaction based on all human gestures.

REFERENCES

1. B. Shackel, *Human-computer interaction-Whence and whither?* Interacting with computers, 2009. **21**(5-6): p. 353-366.

2. B. Hewett, Card, Carey, Gasen, Mantei, Perlman, Strong and Verplank *ACM SIGCHI Curricula for Human-Computer Interaction*. 1996 [cited 2010 07,01]; Available from: http://old.sigchi.org/cdg/cdg2.html.

3. K. A. Butler, R. J. K. Jacob, and B. E. John. *Human-computer interaction: introduction and overview*. in ACM SIGCHI Conference on Human Factors in Computing Systems. May 15-20, 1999. Pittsburgh, Pennsylvania, USA: ACM.

4. M. Fetaji, S. Loskoska, B. Fetaji, and M. Ebibi, *Investigating human computer interaction issues in designing efficient virtual learning environments*, in *Balkan Conference in Informatics*. 2007. Bulgarie.

5. I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. *Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques*. 1998: Wiley Online Library.

6. Mine.Mark, *Virtual environment interaction techniques*, in *Technical report*. 1995, University of North Carolina at Chapel Hill: Chapel Hill, NC.

7. P. Werkhoven, *How Real are Virtual Environments: A Validation of Localization, Manipulation and Design Performance*, in *The Capability of Virtual Reality to Meet Military Requirements*. 2000. p. 3.

8. K. Tollmar, D. Demirdjian, and T. Darrell. *Gesture+ Play Exploring Full-Body Navigation for Virtual Environments*. in 2003 conference on computer vision and pattern recognition workshop 2003.

9. D. A. Bowman, D. B. Johnson, and L. F. Hodges, *Testbed evaluation of virtual environment interaction techniques.* Presence: Teleoperators & Virtual Environments, 2001. **10**(1): p. 75-95.

10. D. A. Bowman and L. F. Hodges. *An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments*. in proceedings of the 1997 symposium on interactive 3D graphics. 1997: ACM.

11. J. Lee and T. Kunii, *Constraint-based hand animation.* Models and techniques in computer animation. Springer-Verlag, Berlin. 1993. p. 110-127.

12. T. Hewett, *ACM SIGCHI curricula for human-computer- interaction*. ACM. 1992

13. A.Dix, J.Finlay, G.Abowd, and R.Beale, *Human-Computer Interaction*. 2 ed. 1997: Prentice Hall.

14. R. Baecker, J.Grudin, W. Buxton, and S. Greenberg, *Readings in human-computer interaction. Toward the Year 2000. 2. ed*. 1995, San Francisco: Morgan Kaufmann.

15. M. A. F. Karray, J. Saleh and M. Arab, *Human-Computer Interaction: Overview on State of the Art.* International journal on smart sensing and intelligent systems, March 2008. **1**.

16. R. J. K. Jacob. *What you look at is what you get: eye movement-based interaction techniques*. in *SIGCHI conference on Human factors in computing systems*. 1990. New York, NY, USA: ACM.

17. I.Starker and R.Bolt, *A gaze-responsive self-disclosing display*, in *SIGCHI conference on Human factors in computer systems*. 1990: New York, NY, USA. p. 3-10.

18. P. Majaranta and K. Räihä. *Twenty years of eye typing: systems and design issues*. in *Proceedings of the 2002 symposium on Eye tracking research & applications*. 2002. New York, NY, USA: ACM.

19. V. Tanriverdi and R. J. K. Jacob. *Interacting with eye movements in virtual environments*. in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2000. New York, NY, USA: ACM.

20. C. H. Morimoto and M. R. M. Mimica, *Eye gaze tracking techniques for interactive applications.* Computer Vision and Image Understanding, 2005. **98**(1): p. 4-24.

21. A. Jaimes and N. Sebe, *Multimodal human-computer interaction: A survey.* Computer Vision and Image Understanding, 2007. **108**(1-2): p. 116-134.

22. A. T. Duchowski, *A breadth-first survey of eye-tracking applications.* Behaviour Research Methods, Instruments, & Computers, 2002. **34**(4): p. 455-70.

23. A. T. Duchowski, *Eye tracking methodology: Theory and practice.* 2007: Springer-Verlag. New York.

24. L. Rabiner and B. H. Juang, *Fundamentals of speech recognition*. Vol. 103. 1993: Prentice hall Englewood Cliffs, New Jersey.

25. J. P. Campbell Jr, *Speaker recognition: A tutorial.* Proceedings of the IEEE, 2002. **85**(9): p. 1437-1462.

26. C. H. Lee, F. K. Soong, and K. K. Paliwal, *Automatic speech and speaker recognition: advanced topics*. 1996: Springer.

27. R. Bolle, S. Pankanti, and A. K. Jain, *Biometrics, Personal Identification in Networked Society: Personal Identification in Networked Society*. 1998: Kluwer Academic Publishers Norwell, MA, USA.

28. B. S. Atal, *Automatic recognition of speakers from their voices.* Proceedings of the IEEE, 2005, April 1976. **64**(4): p. 460-475.

29. G. R. Doddington, *Speaker recognition Identifying people by their voices.* Proceedings of the IEEE, Nov. 1985. **73**(11): p. 1651-1664.

30. D. O'shaughnessy, *Speech communications: human and machine*. 1999: Universities Press.

31. S.Furui, *Recent advances in speaker recognition* Pattern Recognition Letters, May 1998. **18**(9): p. 859-872.

32. *A.Sutherland* and *M.Jack*, *Speaker Verification.* In Aspects of Speech Technology, Edinburgh: Edinburgh University Press, 1988: p. 185-215.

33. B.Zi´Ołko, *Speech Recognition of Highly Inflective Languages. PHD thesis* . 2009: University of York.

34. K. Georgila, K. Sgarbas, A. Tsopanoglou, N. Fakotakis, and G. Kokkinakis, *A speech-based human-computer interaction system for automating directory assistance services.* International Journal of Speech Technology, 2003. **6**(2): p. 145-159.

35. C. Y. Yam, M. S. Nixon, and J. N. Carter, *On the relationship of human walking and running: automatic person identification by gait,* in *International Conference on Pattern Recognition*. 2002: Quebec. p. 287-290.

36. A. Corradini and P. Cohen, *On the relationships among speech, gestures, and object manipulation in virtual environments: Initial evidence.* Advances in Natural Multimodal Dialogue Systems, 2005: p. 97-112.

37. J. Payette. *Advanced human-computer interface and voice processing applications in space*. 1994: Association for Computational Linguistics.

38. M. Blomberg and K. Elenius, *Automatisk igenkänning av tal (‘Automatic recognition of speech’, in Swedish)*, Institutionen for tal, musik och horsel, KTH.2000.

39. D. R. Reddy, *Speech recognition by machine: A review*, in *Proceedings of the IEEE*. April 1976. p. 501 - 531.

40. M. A. Anusuya and S. K. Katti, *Speech Recognition by Machine, A Review.* International Journal of Computer Science and Information Security, Dec. 2009. **6**(3): p. 181-205.

41. R. O'hagan, A. Zelinsky, and S. Rougeaux, *Visual gesture interfaces for virtual environments.* Interacting with computers, 2002. **14**(3): p. 231-250.

42. M. Karam, *A taxonomy of gestures in human computer interactions.* ACM Transactions on. Computer-Human Interactions, 2005.

43. L. Dipietro, A. M. Sabatini, and P. Dario, *A survey of glove-based systems and their applications.* IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2008. **38**(4): p. 461-482.

44. L. W. Howe, F. Wong, and A. Chekima. *Comparison of hand segmentation methodologies for Hand gesture recognition*. 2008: IEEE.

45. B. Fasel and J. Luettin, *Automatic facial expression analysis: a survey.* Pattern Recognition, 2003. **36**(1): p. 259-275.

46. M. R. Ahsan, M. I. Ibrahimy, and O. O. Khalifa, *EMG Signal Classification for Human Computer Interaction: A Review.* European Journal of Scientific Research, 2009. **33**(3): p. 480-501.

47. D.S.Tan and A.Nijholt, *Brain-Computer Interfaces: Applying our Minds to Human-Computer Interaction (Human-Computer Interaction Series),* . 1 ed. 2010: Springer.

48. B. Schuller, M. Lang, and G. Rigoll. *Multimodal emotion recognition in audiovisual communication*. in *Proceedings of IEEE International Conference on ICME '02*. 2002.

49. Argonne National Laboratory, *Nature Bulletin No.611*. Division of Educational Programs, 1960-10-01. Retrieved 2007-12-24.

50. Oxford English Dictionary, *hand.* Oxford University Press.2nd ed., 1989.

51. J. Rehg and T. Kanade. *Visual tracking of high DOF articulated structures: an application to human hand tracking*. in Proceedings of the third European conference on Computer Vision. 1994.

52. Y. Wu, J. Lin, and T. S. Huang, *Analyzing and capturing articulated hand motion in image sequences.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005: p. 1910-1922.

53. Wikipedia. *Hand*. 2011 [cited 2011 01/03]; Available from: http://en.wikipedia.org/wiki/Hand.

54. P. Garg, N. Aggarwal, and S. Sofat. *Vision Based Hand Gesture Recognition*. World Academy of Science Engineering and Technology. 2009. **49:** p. 972-977.

55. C. Wah Ng and S. Ranganath, *Real-time gesture recognition system and application.* Image and Vision Computing, 2002. **20**(13-14): p. 993-1007.

56. E. Sánchez-Nielsen, *Hand gesture recognition for human-machine interaction.* Journal of WSCG, 2003. **12**(1-3).

57. L. Bretzner, I. Laptev, and T. Lindeberg. *Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering*. 2002: Published by the IEEE Computer Society.

58. P. V. Kumar, N.R.V.Praneeth, and Sudheer.V. *Hand And Finger Gesture Recognition System for Robotic Application*. in Proceedings of the International Joint Journal Conference on Engineering and Technology (IJJCET 2010). 2010.

59. T. Coogan, G. Awad, J. Han, and A. Sutherland, *Real time hand gesture recognition including hand*

segmentation and tracking. Advances in Visual Computing, 2006: p. 495-504.

60. J. Canny, *A computational approach to edge detection*. Readings in computer vision: issues, problems, principles, and paradigms, 1987. **184**.

61. Q. Munib, M. Habeeb, B. Takruri, and H. A. Al-Malik, *American Sign Language (ASL) recognition based on Hough transform and neural networks.* Expert Systems with Applications, 2007. **32**(1): p. 24-37.

62. J. Yang and Y. Li, *A New Descriptor for 3D Trajectory Recognition*, in *The Ninth International Symposium on Operations Research and Its Applications*. August 19–23, 2010: Chengdu-Jiuzhaigou, China. p. 362–369.

63. T. Petkovic and J. Krapac, *Technical Report, shape description with Fourier descriptors*. February, 2002.

64. K. Derpanis., *A review of vision-based hand gestures. Internal Report*. 2004, Department of Computer Science, York University.

65. J. Cui and Z. Sun, *Model-based visual hand posture tracking for guiding a dexterous robotic hand.* Optics communications, 2004. **235**(4-6): p. 311-318.

66. E. Dente, A. A. Bharath, J. Ng, A. Vrij, S. Mann, and A. Bull, *Tracking hand and finger movements for behaviour analysis.* Pattern Recognition Letters, 2006. **27**(15): p. 1797-1808.

67. L. Wang, W. Hu, and T. Tan, *Recent developments in human motion analysis.* Pattern Recognition, 2003. **36**(3): p. 585-601.

68. R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff. *3D hand pose reconstruction using specialized mappings*. in Eighth IEEE International Conference on Computer Vision. 2001. Vancouver, BC , Canada

69. T. E. De Campos and D. W. Murray, *Regression-based hand pose estimation from multiple cameras*, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2006: New York, NY

70. M. Donoser and H. Bischof. *Real time appearance based hand tracking*. in 19th International Conference on Pattern Recognition. 2009. Tampa, FL

71. T. Heap and D. Hogg. *Towards 3D hand tracking using a deformable model*. 2002: IEEE.

72. B. Stenger, P. R. S. Mendonca, and R. Cipolla, *Model-based 3D tracking of an articulated hand.* 2001.

73. B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, *Model-based hand tracking using a hierarchical bayesian filter.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006. **28**(9): p. 1372-1384.

74. 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, 2002. **19**(7): p. 677-695.

75. N. Adamo-Villani, J. Heisler, and L. Arns. *Two gesture recognition systems for immersive math education of the deaf*. 2007: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

76. D. J. Sturman and D. Zeltzer, *A Survey of Glove-based Input.* IEEE Computer Graphics and Applications, 1994. **14**(1).

77. T. Defanti and D. Sandin, *Final report to the National Endowment of the arts.* US NEA R. 1977. **60**: p. 34-163.

78. C. M. Ginsberg and D. Maxwell. *Graphical marionette*. 1986: Elsevier North-Holland, Inc.

79. G. J. Grimes, *Digital data entry glove interface device*, in *US Patent 4,414,537*, M.H. Bell Telephone Laboratories, N.J., Editor. 1983.

80. T. G. Zimmerman, *Optical flex sensor*. 1985, Google Patents.

81. T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. *A hand gesture interface device*. 1987: ACM.

82. S. Wise, W. Gardner, E. Sabelman, E. Valainis, Y. Wong, J. Drace, and J. M. Rosen, *Evaluation of a fiber optic glove for semi-automated goniometric measurements.* Journal of Rehabilitation Research and Development, 1990. **27**(4).

83. J. Kramer and L. Leifer, *The Talking Glove:An Expressive and Receptive 'Verbal' Communication Aid for the Deaf, Deaf-Blind, and Nonvocal*. 1989, *tech. report, Stanford University, Dept. of Electrical Engineering, Stanford*: *Calif.* p. 335-340.

84. Intersense Corporation. *3D interaction products*. 2011 [cited 2010 10]; Available from: *http://www.immersion.com.*

85. Virtual Realities. *P5 glove: virtual reality glove*. [cited 2010 11.02]; Available from: http://www.vrealities.com/P5.html. >.

86. D. Bowman, C. Wingrave, J. Campbell, and V. Ly. *Using pinch gloves for both natural and abstract interaction techniques in virtual environments*. in HCI International 2001. 2001.

87. M. Veit, A. Capobianco, and D. Bechmann, *Consequence of Two-handed Manipulation on Speed, Precision and Perception on Spatial Input Task in 3D Modelling Applications.* Journal of Universal Computer Science, 2008. **14**(19): p. 3174-3187.

88. S. Sayeed, S. Andrews, R. Besar, and L. C. Kiong, *Forgery Detection in Dynamic Signature Verification by Entailing Principal Component Analysis.* Discrete Dynamics in Nature and Society, 2007. **2007**: p. 1-8.

89. *Measurand Inc. ShapeHand data glove*. 2009 [cited 2010 07.01]; Available from: http://www.measurand.com.

90. L. Danisch, K. Englehart, and A. Trivett, *Spatially continuous six degree of freedom position and orientation sensor.* Sensor Review, 1999. **19**(2): p. 106-112.

91. V. F. Pamplona, L. A. F. Fernandes, J. L. Prauchner, L. P. Nedel, and M. M. Oliveira. *The image-based data glove*. 2008.

92. V. I. Pavlovic, R. Sharma, and T. S. Huang, *Visual interpretation of hand gestures for human-computer interaction: A review.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. **19**(7): p. 677-695.

93. R. Hassanpour, S. Wong, and A. Shahbahrami, *Vision-Based Hand Gesture Recognition for Human Computer Interaction: A Review*, in *IADIS International Conference Interfaces and Human Computer Interaction 2008*. 2008. p. 125-134.

94. H. Dutağacı, B. Sankur, and E. Yörük, *Comparative analysis of global hand appearance-based person recognition.* Journal of electronic imaging, 2008. **17**: p. 011018.

95. E. Stergiopoulou and N. Papamarkos, *Hand gesture recognition using a neural network shape fitting technique.* Engineering Applications of Artificial Intelligence, 2009. **22**(8): p. 1141-1158.

96. Z. Pan, X. Zhang, A. E. Rhalibi, W. Woo, and Y. Li. *Technologies for E-Learning and Digital Entertainment*. in Third International Conference, Edutainment 2008. June 25-27, 2008. Nanjing, China.

97. A. Jmaa, W. Mahdi, Y. Jemaa, and A. Hamadou, *Hand localization and fingers features extraction:*

application to digit recognition in sign language. Intelligent Data Engineering and Automated Learning-IDEAL 2009, 2009: p. 151-159.

98. R. P. Miller, *Finger dimension comparison identification system*. 1971, US Patent No. 3576538

99. D. P. Sidlauskas, *3D hand profile identification apparatus*. 1988, Recognition systems, Inc. San Jose, Calif.: USA.

100. K. J. Hsiao, T. W. Chen, and S. Y. Chien, *Fast fingertip positioning by combining particle filtering with particle random diffusion*, in *IEEE International Conference on Multimedia and Expo*. 2008: Hannover p. 977-980.

101. K. Oka, Y. Sato, and H. Koike, *Real-time fingertip tracking and gesture recognition.* IEEE Computer Graphics and Applications, 2002: p. 64-71.

102. H. Kim and D. W. Fellner. *Interaction with hand gesture for a back-projection wall*. in Proceedings in Computer Graphics International. June 2004

103. J. Crowley, F. Berard, and J. Coutaz. *Finger tracking as an input device for augmented reality*. in Proceedings in Int'l Workshop on Automatic Face and Gesture Recognition. 1995. Zurich.

104. A. Malima, E. Ozgur, M. Cetin, F. O. E. N. Sci, and I. Sabanci Univ. *A fast algorithm for vision-based hand gesture recognition for robot control*. in IEEE 14th Signal Processing and Communications Applications. 2006. Antalya

105. G. Pradhan, B. Prabhakaran, and C. Li, *Hand-gesture computing for the hearing and speech impaired.* IEEE MultiMedia, 2008. **15**(2): p. 20-27.

106. D. Xu, *A Neural Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG*. Pattern Recognition, 2006. **3**: p. 519-522.

107. C. C. Lien and C. L. Huang, *Model-based articulated hand motion tracking for gesture recognition.* Image and Vision Computing, 1998. **16**(2): p. 121-134.

108. Q. Chen, A. El-Sawah, C. Joslin, and N. D. Georganas. *A dynamic gesture interface for virtual environments based on Hidden Markov Models*. in IEEE International Workshop on haptic Audio Visual Environments and their Applications 2005.

109. J. M. Rehg and T. Kanade. *Digiteyes: Vision-based hand tracking for human-computer interaction*. in In Proceedings of the workshop on Motion of Non-Rigid and Articulated Bodies. 1994. Austin, TX , USA

110. A. A. Argyros and M. I. A. Lourakis, *Binocular hand tracking and reconstruction based on 2D shape matching.* Pattern Recognition, 2006. **1**: p. 207-210.

111. L. Y. Deng, D. L. Lee, H. C. Keh, and Y. J. Liu. *Shape context based matching for hand gesture recognition*. in IET International Conference on Frontier Computing. Theory, Technologies and Applications. 2010 Taichung

112. L. A. Zadeh, *Fuzzy sets.* Information and control, 1965. **8**(3): p. 338-353.

113. C. Von Altrock, *Fuzzy logic and neuro fuzzy applications explained*. 1995: Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

114. E. J. Holden, R. Owens, and G. G. Roy, *Hand movement classification using an adaptive fuzzy expert system.* International Journal of Expert Systems Research and Applications, 1996. **9**(4): p. 465-480.115. E. J. Holden and R. Owens, *Visual sign language recognition.* Multi-Image Analysis, 2001: p. 270-287.

116. M. C. Su, *A fuzzy rule-based approach to spatio-temporal hand gesture recognition.* IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2002. **30**(2): p. 276-281.

117. B. Bedregal, A. Costa, and G. Dimuro, *Fuzzy rule-based hand gesture recognition.* Artificial

Intelligence in Theory And Practice, 2006. **217**: p. 285-294.

118. M. Elmezain, A. Al-Hamadi, and B. Michaelis, *Hand Gesture Recognition Based on Combined Features Extraction.* World Academy of Science, Engineering and Technology, 2009.

119. S. Ge, Y. Yang, and T. Lee, *Hand gesture recognition and tracking based on distributed locally linear embedding.* Image and Vision Computing, 2008. **26**(12): p. 1607-1620.

120. J.K.Aggarwal and N. Nandhakumar, *On the Computations of Motion from Sequence of Images: A Review*, in *Proceedings of the IEEE*. 1988. p. 917 - 935.

121. T. E. De Campos, *3D Visual Tracking of Articulated Objects and Hands*, Thesis (D.Phil.) Department of Engineering Science University of Oxford Trinity. 2006.

122. A. Bobick and J. Davis, *An appearance-based representation of action*, in *Proceedings of the 13th International Conference on Pattern Recognition*. 1996: Vienna , Austria. p. 307-312.

123. A. F. Bobick and J. W. Davis, *The recognition of human movement using temporal templates.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. **23**(3): p. 257-267.

124. A. F. Bobick, *Movement, activity and action: the role of knowledge in the perception of motion.* Philosophical Transactions of the Royal Society B: Biological Sciences, 1997. **352**(1358): p. 1257.

125. J. Shi and C. Tomasi. *Good features to track*. in 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1994. Seattle, WA , USA

126. D. Comaniciu, V. Ramesh, and P. Meer, *Kernel-based object tracking.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003. **25**: p. 564-575.

127. W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma, *Computer vision for computer games*, in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*. 1996: Killington, VT , USA. p. 100-105.

128. R. Polana and R. Nelson. *Low level recognition of human motion*. in *Proceedings of the IEEE Workshop on Non-rigid Motion*. 1994. Austin.

129. B. Heisele, U. Kressel, and W. Ritter. *Tracking non-rigid, moving objects based on colour cluster flow*. in *IEEE Conference on Computer Vision and Pattern Recognition*. 1997. San Juan , Puerto Rico: IEEE Computer Society.

130. J.L.Barron, D.J.Fleet, S.S.Beauchemin, and T.A.Burkitt, *Performance of optical flow techniques*, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1992: Champaign, USA. p. 236 - 242.

131. A. Verri, S. Uras, and E. Demicheli. *Motion segmentation from optical flow*. in *Proc the 5th Alvey Vision Conference*. 1989.

132. H. A. Rowley and J. Rehg. *Analyzing articulated motion using expectation-maximization*. in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1997. San Juan, Puerto Rico: IEEE.

133. A. Shio and J. Sklansky. *Segmentation of people in motion*. in *Proceedings of the IEEE Workshop on Visual Motion*. 1991. Princeton, NJ, USA: IEEE.

134. R. Cipolla, Y. Okamoto, and Y. Kuno. *Robust structure from motion using motion parallax*. in *Proceedings of Fourth International Conference on Computer Vision*. 1993. Berlin , Germany: IEEE.

135. S. X. Ju, M. J. Black, and Y. Yacoob. *Cardboard people: A parameterized model of articulated image motion*. in *Second IEEE International Conference on Automatic Face and Gesture Recognition (FG '96)*. 1996: IEEE.

136. C. Bregler, S. M. Omohundro, M. Covell, M. Slaney, S. Ahmad, D. A. Forsyth, and J. A. Feldman, *Probabilistic models of verbal and body gestures.* Computer Vision for Human-Machine Interaction, 1995: p. 267-290.

137. H. Lu, K. Plataniotis, and A. Venetsanopoulos. *A layered deformable model for gait analysis*. in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*. 2006. Washington, DC, USA: IEEE.

138. R. Ronfard, C. Schmid, and B. Triggs, *Learning to parse pictures of people*, in *Proceedings of the 7th European Conference on Computer Vision*. June 2002: Copenhagen, Denmark. p. 700-714.

139. T. Zhao, R. Nevatia, and F. Lv, *Segmentation and Tracking of Multiple Humans in Complex Situations*, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2001. p. 194-201.

140. T. Zhao and R. Nevatia, *Tracking multiple humans in complex situations*, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2001. p. 194-201.

141. A. Y. Ng and M. I. Jordan. *On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes*. in proceedings of 14th Advances in neural information processing systems. 2002: MIT Press.

142. T. Jaeggli, E. Koller-Meier, and L. Van Gool, *Learning generative models for multi-activity body pose estimation.* Special Issue: Best of the Eighth Asian Conference on Computer Vision, International Journal of Computer Vision, 2008. **83**(2): p. 121-134.

143. C. Lee and A. Elgammal, *Tracking people on a torus.* IEEE Transactions on Pattern Analysis and Machine Intelligence, March 2009. **31**: p. 520-538.

144. Y. Wu, G. Hua, and T. Yu. *Tracking articulated body by dynamic Markov network*. in *Proceedings of Ninth IEEE International Conference on Computer Visio*. 2003. Nice, France.

145. Y. Yacob and L. Davis. *Learned temporal models of image motion*. in Sixth International Conference on Computer Vision. 1998: IEEE.

146. V. Lepetit and P. Fua, *Monocular model-based 3D tracking of rigid objects.* Foundations and Trends in Computer Graphics and Vision, 2005.

147. J. M. Rehg and T. Kanade. *Model-based tracking of self-occluding articulated objects*. in *Proceedings of Fifth International Conference on Computer Vision*. 1995. Cambridge, MA, USA: IEEE.

148. I. Karaulova, P. Hall, and A. D. Marshall. *A hierarchical model of dynamics for tracking people with a single video camera*. in *British Machine Vision Conference*. 2000.

149. M.Shaheen, J.Gall, R.Strzodka, L.Gool, and H.Seidel. *a comparison of 3d model-based tracking approaches for human motion capture in uncontrolled environments*. in IEEE workshop on application of computer vision 2009.

150. J. Deutscher, B. North, B. Bascle, and A. Blake. *Tracking through singularities and discontinuities by random sampling*. in The Proceedings of the Seventh IEEE International Conference on Computer Vision. 1999. Kerkyra: IEEE.

151. M. J. Black and A. D. Jepson, *A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions*, in *Proceedings of the 5th European Conference on Computer Vision*. 1998: London, UK. p. 909 - 924

152. A. J. Davison, J. Deutscher, and I. D. Reid. *Markerless motion capture of complex full-body movement for character animation*. in *Proceedings of the Eurographic workshop on Computer animation and simulation.* 2001. Manchester, UK.

153. B. Thomas, A. Hilton, and V.Krüger, *A survey of advances in vision-based human motion capture and analysis.* Special issue on modelling people: Vision-based understanding of a person's shape, appearance, movement, and behaviour, 2006. **104**(2-3): p. 90-126.

154. B. Li and H. Holstein, *Recognition of Human Periodic Motion - A Frequency Domain Approach*, in *Pattern Recognition*. 2002: *16th International Conference on Pattern Recognition*. p. 10311.

155. R. Polana and R. Nelson. *detecting actives*. in *Proceedings of Computer Vision and Pattern Recognition*. 1993. New York, NY.

156. R. Polana and R. Nelson, *detecting actives.* Journal of Visual Communication and Image Representation, June 1994. **5**: p. 172-180.

157. P. Tsai, M. Shah, K. Keiter, and T. Kasparis, *Cyclic motion detection.* Pattern Recognition, 1993.

158. Y. Makihara, R. Sagawa, Y. Mukaigawa, T. Echigo, and Y. Yagi, *Gait Identification Considering Body Tilt by Walking Direction Changes.* Electronic Letters on Computer Vision and Image Analysis, 2009. **8**(1).

159. R. Cutler and L. S. Davis, *Robust real-time periodic motion detection, analysis, and applications.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. **22**(8): p. 781-796.

160. G. Tolstov and R. Silverman, *Fourier series*. 1976: Courier Dover Publications.

161. J. Walker, *Fourier Series*, in *Academic Press* 2004, *Encyclopedia of Physical Science and Technology*.

162. D. Cunado, J. M. Nash, M. S. Nixon, and J. N. Carter. *Gait extraction and description by evidence-gathering*. in *Proceedings of the Second International Conference on Audio- and Video-Based Biometric Person Authentication*. 1999.

163. C. Y. Yam, M. S. Nixon, and J. N. Carter, *On the relationship of human walking and running: automatic person identification by gait*, in *International Conference on Pattern Recognition*. 2002. p. 10287.

164. R. L. Cosgriff, *Identification of Shape*. 1960, *Ohio State University Research,* Columbus.

165. C. T. Zahn and R. Z. Roskies, *Fourier descriptors for plane closed curves.* IEEE Transactions on Computers, Mar. 1972. **21**: p. 269-281.

166. T.Petkovic and J. Krapac, *Shape description with Fourier descriptors*, in *Technical Report*. 2002.

167. L. Oukhellou and P. Aknin. *Modified Fourier Descriptors: A new parameterisation of eddy current signatures applied to the rail defect classification*. in *III international workshop on advances in signal processing for non destructive evaluation of materials*. 1997. Quebec.

168. Y. Lu, S. Schlosser, and M. Janeczko, *Fourier descriptors and handwritten digit recognition.* Machine Vision and Applications, 1993. **6**(1): p. 25-34.

169. S. Yu, L. Wang, W. Hu, and T. Tan. *Gait analysis for human identification in frequency domain*. in Proceedings of Third International Conference on Image and Graphics. 2004.

170. H. Fujiyoshi and A. J. Lipton. *Real-time human motion analysis by image skeletonization*. in *Fourth IEEE Workshop on Applications of Computer Vision*. 1998. Princeton, NJ, USA. : IEEE.

171. R. Cutler and L. Davis. *Robust periodic motion and motion symmetry detection*. 2000. *IEEE Computer Society Conference on Computer Vision and Pattern* Recognition.

172. J. Davis, W. Richards, and A. Bobick. *Categorical representation and recognition of oscillatory motion patterns*. 2000: Published by the IEEE Computer Society.

173. M. Bhuyan, P. Bora, and D. Ghosh, *Trajectory Guided Recognition of Hand Gestures having only*

Global Motions. International Journal of Computer Science, 2008. **3**: p. 4.

174. D. R. Faria and J. Dias. *3D hand trajectory segmentation by curvatures and hand orientation for classification through a probabilistic approach*. in IEEE/RSJ International Conference on Intelligent Robots and Systems. 2009. St. Louis, MO.

175. W. Chen and S. F. Chang. *Motion trajectory matching of video objects*. in *Proc. SPIE*. 2000.

176. V. Shiv, N. Prasad, V. Kellokumpu, and L. S. Davis, *Ballistic Hand Movements*, in *AMDO 2006, Springer-Verlag*. 2006: Berlin, Heidelberg.

177. J. Bandera, R. Marfil, A. Bandera, J. Rodrígueza, L. Molina-Tanco, and F. Sandoval, *Fast gesture recognition based on a two-level representation.* Pattern Recognition Letters, 2009. **30**(13): p. 1181-1189.

178. S. Wu and Y. Li, *Flexible signature descriptions for adaptive motion trajectory representation, perception and recognition.* Pattern Recognition, 2009. **42**(1): p. 194-214.

179. F. I. Bashir, A. A. Khokhar, and D. Schonfeld, *Object trajectory-based activity classification and recognition using hidden Markov models.* IEEE Transactions on Image Processing, 2007. **16**(7): p. 1912-1919.

180. A. Dyana and S. Das. *Spatio-temporal descriptor using 3D curvature scale space*. in *Proceedings of the 2nd international conference on Pattern recognition and machine intelligence*. 2007. Kolkata, India.

181. C. Rao, A. Yilmaz, and M. Shah, *View-invariant representation and recognition of actions.* International journal of computer vision, 2002. **50**(2): p. 203-226.

182. L. R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, 1989. **77**(2): p. 257-286.

183. C. Vogler and D. Metaxas. *ASL recognition based on a coupling between HMMs and 3D motion analysis*. in *Sixth International Conference on Computer Vision*. 1998 IEEE.

184. T. Starner, J. Weaver, and A. Pentland, *Real-time American sign language recognition using desk and wearable computer based video.* IEEE Transactions on Pattern Analysis and Machine Intelligence. Media Lab., MIT, Cambridge, MA, 1998. **20**(12): p. 1371-1375.

185. M. Ahmad and S. W. Lee, *Human action recognition using multi-view image sequences features*, in *7th International Conference on Automatic Face and Gesture Recognition*. 2006: Southampton. p. 523-528.

186. A. Elgammal, V. Shet, Y. Yacoob, and L. S. Davis. *Learning dynamics for exemplar-based gesture recognition*. in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2003: IEEE.

187. M. Brand, N. Oliver, and A. Pentland. *Coupled hidden Markov models for complex action recognition*. in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1997. San Juan , Puerto Rico.

188. A. Galata, N. Johnson, and D. Hogg, *Learning variable-length Markov models of behaviour.* Computer Vision and Image Understanding, 2001. **81**(3): p. 398-413.

189. S. T. Shivappa, M. M. Trivedi, and B. D. Rao, *Audiovisual Information Fusion in Human computer Interfaces and Intelligent Environments: A Survey.* Proceedings of the IEEE. **98**(10): p. 1692-1715.

190. H. Yu, G. Sun, W. Song, and X. Li. *Human motion recognition based on neural network*. in Proceedings of 2005 International Conference on Communications, Circuits and Systems 2005.

191. Y. Guo, G. Xu, and S. Tsuji. *Understanding human motion patterns*. in *Proceedings of the 12th IAPR International. Conference on Computer Vision & Image Processing*. 1994. Jerusalem, Israel.

192. M. Rosenblum, Y. Yacoob, and L. Davis, *Human expression recognition from motion using a radial basis function network architecture*, in *Proceedings of the 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*. 1994: Austin, TX , USA. p. 43-49.

193. D. R. Faria, H. Aliakbarpour, and J. Dias. *Grasping movements recognition in 3D space using a Bayesian approach*. in *International Conference on Advanced Robotics*. 2009: IEEE.

194. D.Faria and J. Dias, *Hand Trajectory Segmentation and. Classification Using Bayesian Techniques*, in *IROS-2008 workshop on Grasp and Task Learning by Imitation 2008, IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008: *Nice, France*. p. 44-49.

195. J. Rett and J. Dias. *Human-robot interface with anticipatory characteristics based on Laban Movement Analysis and Bayesian models*. in *Proceedings of the IEEE 10th International Conference on Rehabilitation Robotics*. 2008. Noordwijk.

196. D. Faria and J. Dias. *Bayesian Techniques for Hand Trajectory Classification*. in *14th Portuguese Conference on Pattern Recognition*. 31st October, 2008. Coimbra, Portgal.

197. E. J. Keogh and M. J. Pazzani. *Derivative dynamic time warping*. in *First SIAM International Conference on Data Mining*. 2001.

198. C. Myers, L. Rabiner, and A. Rosenberg, *Performance tradeoffs in dynamic time warping algorithms for isolated word recognition.* IEEE Transactions on Acoustics, Speech and Signal Processing, 1980. **28**(6): p. 623-635.

199. C. F. Borges and T. Pastva, *Total least squares fitting of Bézier and B-spline curves to ordered data.* Computer Aided Geometric Design, 2002. **19**(4): p. 275-289.

200. Q. Dong, Y. Wu, and Z. Hu, *Gesture recognition using quadratic curves*, in *Proceedings of the Seventh Asian Conference on Computer Vision*. 2006. p. 817-825.

201. F. Cohen, Z. Huang, and Z. Yang, *Invariant matching and identification of curves using B-splines curve representation.* IEEE Transactions on Image Processing, 1995. **4**(1): p. 1-10.

202. M. C. Shin, L. V. Tsap, and D. M. B. Goldgof, *Gesture recognition using Bezier curves for visualisation navigation from registered 3D data.* Pattern Recognition, 2004. **37**(5): p. 1011-1024.

203. W. Niu, J. Long, D. Han, and Y. F. Wang. *Human activity detection and recognition for video surveillance*. in IEEE International Conference on Multimedia and Expo. 2004. Taipei.

204. InterSense, Inc. *IS-900 precision inertial-ultrasonic motion tracking system*. 2011 [cited 2010 10.01]; Available from: http://www.isense.com.

205. D. Wormell and E. Foxlin. *Advancements in 3D interactive devices for virtual environments*. 2003: Citeseer.

206. Measurand Inc. *ShapeHand data glove*. 2009 [cited 2010 07.01]; Available from: http://www.measurand.com.

207. T. Grossman, R. Balakrishnan, and K. Singh. *An interface for creating and manipulating curves using a high degree-of-freedom curve input device*. 2003: ACM.

208. R. Balakrishnan, G. Fitzmaurice, G. Kurtenbach, and K. Singh. *Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip*. 1999: ACM.

209. R. Bachnak and S. King. *Non-destructive evaluation and position tracking of flaws in conductive materials*. 2008: World Scientific and Engineering Academy and Society (WSEAS).

210. Y. Baillot, J. J. Eliason, G. S. Schmidt, J. Swan, D. Brown, S. Julier, M. A. Livingston, and L.

Rosenblum. *Evaluation of the ShapeTape tracker for wearable, mobile interaction*. in Proceedings of IEEE Virtual Reality 2003: IEEE.

211. D. H. Rubine, *The automatic recognition of gestures*. 1991, University of Carnegie-Mellon, Pittsburgh, PA.

212. E. Foxlin, M. Harrington, and Y. Altshuler. *Miniature 6-DOF inertial system for tracking HMDs*. in In Procceedings of SPIE Helmet and Head-Mounted Displays III. 1998. Orlando.

213. L. Fang, P. J. Antsaklis, L. A. Montestruque, M. B. Mcmickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, and M. Xie, *Design of a wireless assisted pedestrian dead reckoning system-the NavMote experience.* IEEE Transactions on Instrumentation and Measurement, 2005. **54**(6): p. 2342-2358.

214. Y. Xiaoping, E. R. Bachmann, and R. B. Mcghee, *A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements.* IEEE Transactions on Instrumentation and Measurement, 2008. **57**(3): p. 638-650.

215. H. Zhou and H. Hu, *Reducing drifts in the inertial measurements of wrist and elbow positions. IEEE Transactions on Instrumentation and Measurement*, 2009. **59**(3): p. 575-585.

216. M. Zaoui, D. Wormell, Y. Altshuler, E. Foxlin, and J. Mcintyre, *A 6 DOF opto-inertial tracker for virtual reality experiments in microgravity.* Acta Astronautica, 2001. **49**(3-10): p. 451-462.

217. Intersense, Inc. *Product Manual for use with InertiaCube3 and the InertiaCube3 Processor* 2005: Bedford, MA.

218. E. Foxlin, M. Harrington, and G. Pfeifer. *Constellation: a wide-range wireless motion-tracking system for augmented reality and virtual set applications*. in Proceedings of the 25th annual conference on Computer graphics and interactive techniques 1998 New York, NY, USA.

219. V. Kindratenko, *A survey of electromagnetic position tracker calibration techniques.* Virtual Reality, 2000. **5**(3): p. 169-182.

220. V. Kindratenko, *A comparison of the accuracy of an electromagnetic and a hybrid ultrasound-inertia position tracking system.* Presence: Teleoperators & Virtual Environments, 2001. **10**(6): p. 657-663.

221. K. Meyer, H. L. Applewhite, and F. A. Biocca. *A survey of position trackers*. 1992.

222. H. R. Jones, *Magnetic Position and Orientation Tracking System.* IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, 1979. **15**(5): p. 709.

223. R. G. Belleman, *Interactive exploration in virtual environments, Thesis*. 2003, Universiteit van Amsterdam.

224. G. Jones, D. Lee, N. Holliman, and D. Ezra. *Controlling Perceived Depth in Stereoscopic Images*. in *Stereoscopic displays and virtual reality systems viii*. 2001.

225. R. Piroddi. *White Paper, Stereoscopic 3D Technologies.* Innovation in the Multi-Screen World, April 2010.

226. J. Malik, B. L. Anderson, and C. E. Charowhas, ***Stereoscopic occlusion junctions*. Nature America**, **1999**. **2**(9): p. **840**.

227. N. Holliman, *Mapping perceived depth to regions of interest in stereoscopic images.* Stereoscopic Displays and Virtual Reality Systems XI, AJ Woods, JO Merritt, SA Benton, and MT Bolas, eds. **5291**(1): p. 117?28.

228. A. Woods, *Compatibility of display products with stereoscopic display methods.* 2005.

229. H. Yamanoue. *the differences between toed-in camera configurations and parallel camera*

configurations in shooting stereoscopic images. in 2006 IEEE International Conference on Multimedia and Expo. 2006. Toronto.

230. W. Kang and S. Lee. *Horizontal parallax distortion correction method in toed-in camera with wide-angle lens*. in 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Vide 2009: IEEE.

231. MSDN. *Document/View Architecture*. 2011 [cited 2011 06.20]; Available from: http://msdn.microsoft.com/en-us/library/4x1xy43a(v=vs.80).aspx

232. C. Bonacic, C. Garcia, M. Marin, M. Prieto, F. Tirado, and C. Vicente, *Improving Search Engines Performance on Multithreading Processors*. High Performance Computing for Computational Science-VECPAR 2008, 2008: p. 201-213.

233. K. Lutz, S. Koeneke, T. Wustenberg, and L. Jancke, *Asymmetry of cortical activation during maximum and convenient tapping speed*. Neuroscience letters, 2004. **373**(1): p. 61-66.

234. B. A. Myers, *A brief history of human-computer interaction technology.* interactions, 1998. **5**(2): p. 44-54.

235. D. Aliakseyeu, S. Subramanian, J. B. Martens, and M. Rauterberg. *Interaction Techniques for Navigation through and Manipulation of 2D and 3D Data*. 2002: Eurographics Association.

236. J. Rönkkö, J. Markkanen, R. Launonen, M. Ferrino, E. Gaia, V. Basso, H. Patel, M. D'cruz, and S. Laukkanen, *Multimodal astronaut virtual training prototype*. International Journal of Human-Computer Studies, 2006. **64**(3): p. 182-191.

237. M. R. Mine, F. P. Brooks Jr, and C. H. Sequin. *Moving objects in space: exploiting proprioception in virtual-environment interaction*. 1997: ACM Press/Addison-Wesley Publishing Co.

238. M. Ikits, J. Kniss, A. Lefohn, and C. Hansen, *Volume Rendering Techniques*, in *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. April 1, 2004, Addison-Wesley Professional.

239. M. Levoy, H. Fuchs, S. M. Pizer, J. Rosenman, E. L. Chaney, G. W. Sherouse, V. Interrante, and J. Kiel. *Volume Rendering in Radiation Treatment Planning*. 1990: IEEE Computer Society Press.

240. W. Li, A. Kaufman, and K. Kreeger. *Real-time volume rendering for virtual colonoscopy*. 2001: Springer Verlag Wien.

241. S. Roettger, S. Guthe, D. Weiskopf, T. Ertl, and W. Strasser. *Smart hardware-accelerated volume rendering*. 2003: Eurographics Association.

242. X. Tong, W. Wang, W. Tsang, and Z. Tang. *Efficiently rendering large volume data using texture mapping hardware*. in *Joint EUROGRAPHICS - IEEE TCCG Symposium on Visualization Proceedings*. 26-28 May 1999. Vienna, Austria.

243. S. Lombeyda, L. Moll, M. Shand, D. Breen, and A. Heirich. *Scalable interactive volume rendering using off-the-shelf components*. in *IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics*. Oct. 2001. San Diego, CA, USA.

244. D. Phan, T. Nguyen, and T. Bui, *A 3D Conversational Agent for Presenting Digital Information for Deaf People*. Agent Computing and Multi-Agent Systems, 2009: p. 319-328.

245. W. C. Stokoe, D. C. Casterline, and C. G. Croneberg, *A dictionary of American Sign Language on linguistic principles*. 1976: Linstok Press.

246. S. Valerie and A. Frost. *SignWriting: Sign Languages Are Written Languages*. in *2nd Annual Visual and Iconic Language Conference*. 2008. San Diego, California.

247. Sign Language Notation System. *home page (English) for HamNoSys*. [cited 2010 11.01]; Available from: http://www.sign-lang.uni-hamburg.de/projects/hamnosys.html.

248. Dancewritingsite. *home of Sutton DanceWriting*. 2011 [cited 2011 01.12]; Available from: http://www.dancewriting.org/.

249. S. Valerie., *Dance Writing Shorthand for Classical Ballet. Sutton Movement Writing Center*, September1997.

250. *SignWriting® Site*. 2011 [cited 2011 1.1]; Available from: http://www.signwriting.org.

251. Elghoul and M.J.O., *An Avatar Based Approach for Automatic Interpretation of Text to Sign Language.* Challenges for assistive technology: AAATE 07, 2007: p. 266.

252. Z. Mo and U. Neumann. *Lexical gesture interface*. in *IEEE International Conference on Computer Vision System*. 2006: IEEE.

253. K. Clark and D. Gunsauls., *The SignWriter Newsletter. written in signed and spoken languages*, Spring Issue, 1997.

254. Signwriting. *ISWA 2008* 2008 [cited 2010 02.10]; Available from: http://www.movementwriting.org/symbolbank/ISWA2008/.

255. V. Sutton. *Lessons in SignWriting*. 2011 [cited 2011 02.01]; Available from: http://www.signwriting.org/lessons/lessonsw/lessonsweb.html.

256. A. Frost. *Frosty the Snowman pg. 1, 1999 Version*. February 07, 2010 [cited 2011 02.13]; Available from: http://www.signbank.org/SignPuddle1.5/searchword.php?ui=1&sgn=5&sid=459.

257. S. Liwicki and M. Everingham. *Automatic recognition of fingerspelled words in British sign language*. 2009: IEEE.

258. Resources. *British Sign Language, What is the Fingerspelling Alphabet?* 2011 [cited 2010 11.1]; Available from: <*http://www.british-sign.co.uk/what_fingerspelling.php*>.

259. Hubpages. *An Introduction to Finger Spelling*. 2011 [cited 2010 11.1]; Available from: http://hubpages.com/hub/An-Introduction-to-Finger-Spelling.

260. Wikipedia. *British Sign Language*. 31 January 2011 [cited 2010 11.01]; Available from: <*http://en.wikipedia.org/wiki/British_Sign_Language*>.

261. Wikipedia. *Fingerspelling*. 17 January 2011 [cited 2010 11.02]; Available from: http://en.wikipedia.org/wiki/Fingerspelling.

262. A. Design. *BSL / British Sign Language tutorial*. [cited 2010 11.01]; Available from: <*http://www.aspexdesign.co.uk/bsl.htm*>.

263. D. P. Crossroads. *Did the Celts Know Sign Language?* January 16 2008 [cited 2010 11.01]; Available from: http://deafpagancrossroads.com/2008/01/16/did-the-celts-know-sign-language/.

APPENDIX A

PUBLICATIONS

1. Dynamic Hand Gesture Tracking and Recognition for Real-Time Immersive Virtual Object Manipulation
   2009 International Conference on CyberWorlds, pp.29-35, 2009.

2. Hand motion recognition and visualisation for direct sign writing
   2010 International Conference on Information Visualisation, pp. 467 - 472, 2010.

3. Immersive manipulation of virtual objects through glove based hand gesture interaction.
   Journal of Virtual Reality. Augest, 2011.

Dynamic Hand Gesture Tracking and Recognition for Real-time Immersive Virtual Object Manipulation

Gan Lu, Lik-kwan Shark, and Geoff Hall
ADSIP Research Centre,
University of Central Lancashire,
Preston UK PR1 2HE
glu@uclan.ac.uk

Ulrike Zeshan
International Centre for Sign Languages and Deaf Studies,
University of Central Lancashire
Preston UK PR1 2HE

Abstract

Immersive visualisation is increasingly being used for comprehensive and rapid analysis of objects in 3D and object dynamic behaviour in 4D. Challenges are therefore presented to provide natural user interaction to enable effortless virtual object manipulation. Presented in this paper is the development and evaluation of a human-computer interaction system based on natural hand gestures. By employing a hybrid inertial and ultrasonic tracking system to provide the absolute positions and orientations of the user's head and hands as well as a pair of high degrees-of-freedoms data glove to provide the relative positions and orientations of finger joints and tips in both hands, the proposed system is shown to be able to automatically track and recognise a number of simple hand gestures. The effectiveness and potential of the proposed system is demonstrated through the five basic object manipulation tasks involving selection, release, translation, rotation and scaling of a 3D virtual cube.

1. Introduction

Stereoscopic display [1] enables viewing of an object as an image in 3D with depth information and the object behaviour as an image sequence in 4D with depth and time information. This is increasingly being used in comprehensive and rapid visualisation of complex data sets, such as 3D CT (Computed Tomography) data and 4D dynamic MR (Magnetic Resonance) data in medical diagnosis and treatment planning [2-5]. A more immersive visualisation can be achieved by the use of a large wall display, as well as head tracking to allow the user to move around and to view from different perspectives based on the user's head position and orientation. However, there is a challenge to provide an immersive interaction with the virtual object projected in stereoscopic mode without using indirect manipulation methods, such as keyboard based control, mouse based 3D widgets or hand-held input devices like wireless 3D wands.

Direct manipulation based on tracking and recognition of hand gestures provides a more immersive interaction with virtual objects. A possible approach is based on the use of video camera [6], whereby the hand movements and gestures are recognised based on the dynamic hand shapes extracted from the video sequence. The difficulties associated with this approach include

- self occlusion resulting in capture of partial hand gestures due to the restricted camera view angle;
- incorrect image segmentation of hands due to different lighting and background conditions; and
- high computation cost due to the requirements to track rapid hand motion and to handle high complexity of a hand with at least one degree of freedom (DOF) for each finger joint and 27 DOF for just one hand.

These difficulties can be overcome by wearing a pair of data gloves with sensors to provide finger movement information at the expense of introducing a small inconvenience to the user [7]. The simplest type of data glove is based on contact, using conductive patches in the glove [8]. With the requirement of fingers touching each other to make electrical contact, the recognisable gestures are not necessary natural, and the number of identifiable gestures is limited. A more sophisticated data glove providing more finger movement information is based on flexure. It uses fibre-optic, mechanical, or piezoresistive sensors to measure the bending of each finger. For the work described in this paper, the hand gesture recognition is based on a high DOF data glove, called ShapeHand from Measurand [9-10]. It can be considered as one of the most sophisticated data gloves, which uses fibre-optic sensors to provide all finger joint movement information, 27 DOF for each hand.

As the first step to achieve dynamic hand gesture tracking and recognition, presented in this paper is a small scale system implemented for demonstration and evaluation of real-time immersive virtual object manipulation.

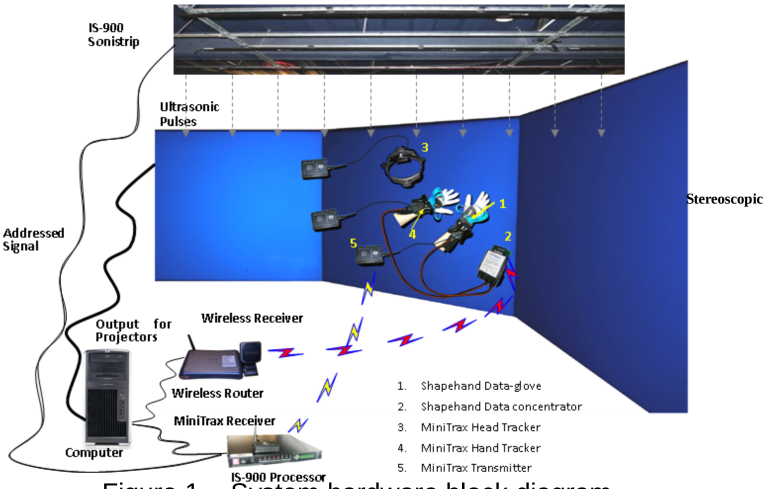


Figure 1. System hardware block diagram

The system integrates three elements to achieve immersion:

- a wireless hybrid inertial and ultrasonic tracking system [11-12] to provide the 3D positions and 3D orientations of the user head and two hands;
- a pair of wireless high DOF data glove [9] to provide all the finger joint movement information; and
- a large screen with two stereoscopic back projections to show the virtual object being manipulated and to provide a visual feedback of two hands with respect to the virtual object in 3D.

2. System development

This section presents hardware and software aspects associated with system development in terms of integration.

2.1. Hardware integration

The demonstration system developed for real-time immersive object manipulation is illustrated in Figure 1 in a block diagram form. Driven by a desktop computer, the system is an integration of three subsystems for gesture data acquisition, position data acquisition, and stereoscopic display.

For gesture data acquisition from hands, a pair of wireless ShapeHand data glove from Measurand is used. This data glove is based on flexible tapes embedded with multiple fibre optic curvature sensors arranged to sense bend and twist along the length of each tape. By attaching the tapes to run along each finger with one end at the finger tip and the other end fed to a small data acquisition box at wrist, the gesture movements of fingers introduce deformation of the tapes. Also, the bend and twist measured at each sensor location with respect to the wrist end of the tape enables relative positions and orientations of each finger joint to be determined. As shown in Figure 1, via the ShapeHand Data Concentrator, the collected hand gesture data are transmitted to a wireless receiver/router connected to the Ethernet port of the computer.

Since absolute hand position and orientation data in 3D space are not provided by ShapeHand data glove, an IS-900 wireless tracking system from InterSense [11] is used to provide the required hand position data. The system is also used to provide the head position and orientation data in order to generate correct view. The operation of the system is based on a combination of inertial tracking and ultrasonic tracking. Whilst the outputs from the inertial sensors, consisting of accelerometers and gyros, are used to determine the position and orientation of each sensor in 3D space, the range measurements based on time-of-flight between ultrasonic emitters and receivers are used to correct the drifting effect inherent with the inertial sensors. As shown in Figure 1, IS-900 SoniStrips containing ultrasonic emitters are mounted on ceiling, which transmit ultrasonic pulses upon receiving addressed signals from the IS-900 processor connected to the serial port of the computer. Three MiniTrax tracking devices containing inertial sensors and ultrasonic receivers are used with two attached to the user wrists and one attached to the user head. Each MiniTrax tracking device performs time-of-flight range measurement based on the ultrasonic pulses received, and transmit its position and orientation data to the corresponding MiniTrax receiver connected to the IS-900.

For stereoscopic display, the demonstration system uses one large screen with size of 2.74m x 2.06m (shown as the middle screen in Figure 1), and two back projectors operating in passive circular polarisation mode are connected to the computer through two dual DVI graphics card output ports. 3D objects with depth effect are seen by the user wearing a pair of light-weight polarised glasses.

The computer used in the demonstration

system runs on Microsoft Windows XP, and is based on an Intel Xeon 3.06GHz CPU with 2 GB RAM and NVIDIA Quadro FX 3000 Graphics Card with 256MB memory.

2.2. Software implementation

The system software is implemented as a Windows XP based application using C++. To minimise development time, the software utilises the Microsoft Foundation Classes (MFC) to build the user interface and control units. A modified version of the standard Document View Model has been implemented to allow the input data to update the document object (containing the current user head and hand position as well as gesture data), and the output display to be treated as an individual "view" of the document object. Furthermore, the software is implemented following a multi-thread approach to minimise response time for interactive object manipulation. There are five parallel program threads with two of them performing hand gesture data acquisition and extraction from ShapeHand, and the other three performing position data acquisition from InterSense, gesture recognition, and stereoscopic display, respectively.

The two program threads for hand gesture data acquisition and extraction are implemented based on the ShapeHand API (Application Programming Interface). With steps including initiation of data collection, receiving data and checking received data, the program thread for data acquisition obtains raw data from a pair of wireless ShapeHand data glove via the Ethernet port of the computer. Using the raw data obtained, the required positions and orientations of a finger joint are determined in the program thread for data extraction.

The program thread of position data acquisition is implemented based on the InterSense API. It performs the data acquisition from the three MiniTrax tracking devices attached to the head and two wrists of the user via the serial port of the computer to provide their position and orientation data. Steps in this thread include data collection and data updating if the incoming data are found to be different from the previously received data.

The program thread of gesture recognition is based on the algorithm presented in the next section. Essentially, it involves data merging through coordinate transformations, as well as tracking and recognising a number of pre-specified hand gestures for manipulation of the displayed virtual object.

The program thread of stereoscopic display is implemented based on OpenGL to provide 3D visual feedback to the user. This is done by generating two views of the virtual object and two hands. As viewing through polarisation glasses results in each eye seeing only the view generated for it, it creates visual immersion with depth impression. Steps in this program thread include the use of the head position data acquired to specify the viewing position and direction of the left and right eyes, configuration of the viewing frustum for each eye, and stereo rendering to draw the left and right images of the 3D object and hand models by perspective projection.

These five program threads are executed simultaneously by the computer with each thread assigned a slice of its CPU (Central Processing Unit) time. The scheduling of the threads is done in a round-robin manner with all threads having the same priority. In execution of the program thread of position data acquisition during its allocated time slice, no change in the received data will result in early switching to the next program thread.

3. Hand tracking and gesture recognition

This section focuses on two data processing operations, namely, coordinate transformation to fuse multiple position and orientation data sets acquired using different referencing systems, and the dynamic gesture tracking and recognition method for immersive object manipulation.

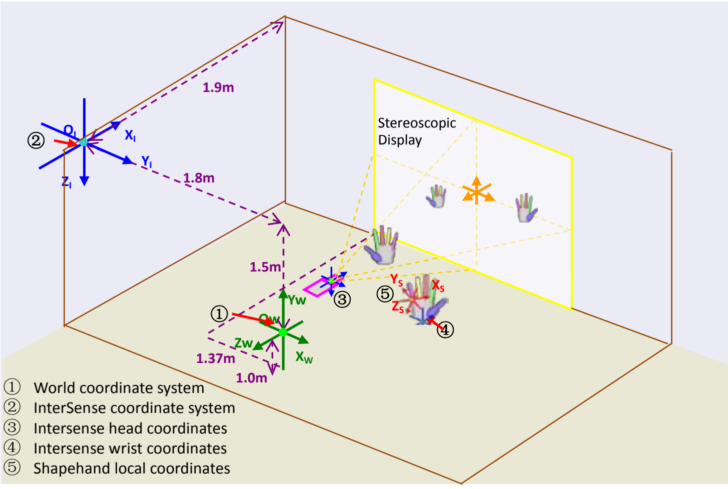
3.1. Data integration via coordinate transformation

With different equipment using different coordinate systems for data acquisition, processing and display, coordinate transformations are required to bring different data sets into a common coordinate system.

Figure 2 illustrates the spatial relationships between different coordinate systems. The world coordinate system is defined to have the same orientation as the stereoscopic display. With the x-axis (denoted by X_w) pointing towards right, the y-axis (denoted by Y_w) pointing upwards, and the z-axis (denoted by Z_w) pointing towards the viewer, this forms a right-handed coordinate system with a positive rotation about the axis in the anticlockwise direction. Furthermore, the origin of the world coordinate system (denoted by O_w) is located at the middle of the screen along the x-axis (1.37m away from the screen right edge), 1m above the floor along the y-axis, and 1.9m in front of the screen along the z-axis.

For the InterSense system with its coordinate axes denoted by (X_I, Y_I, Z_I), the position data acquired for head and wrists are calibrated with respect to its origin denoted by O_I at (–1.8m, 1.5m, 0) in the world coordinate system as shown in Figure 2. Furthermore, two rotation operations are required to align the orientations of the InterSense coordinate system with the orientations of the world coordinate system, namely, rotation of –90° about the InterSense y-axis to make the new InterSense x-axis parallel to the world coordinate x-axis, and rotation of 90° about the new InterSense x-axis to make the new InterSense y and z axes parallel to the world coordinate systems. If $\boldsymbol{i} = [x_i, y_i, z_i, 1]'$ denotes the homogeneous coordinates of a position in the InterSense coordinate system, then its corresponding homogeneous coordinates in the specified world coordinate system denoted by $\boldsymbol{i}_w = [x_{iw}, y_{iw}, z_{iw}, 1]'$ are given by:

$$i_w = \boldsymbol{T}^{I\rightarrow W} i \tag{1}$$



1.9m
②
O_I
X_I
Y_I
Z_I
1.8m
Stereoscopic Display
1.5m
Y_W
③
⑤
Y_S
X_S
Z_S
④
①
O_W
Z_W
X_W
1.37m
1.0m
① World coordinate system
② InterSense coordinate system
③ Intersense head coordinates
④ Intersense wrist coordinates
⑤ Shapehand local coordinates

Figure 2. Coordinate systems

where $\boldsymbol{T}^{I\rightarrow W}$ is the matrix for geometric transformation from the InterSense coordinate system to the world coordinate system. Based on the geometric relationship between the two coordinate systems described above, $\boldsymbol{T}^{I\rightarrow W}$ is given by

$$\boldsymbol{T}^{I\rightarrow W} = \begin{bmatrix} 0 & 1 & 0 & -1.8 \\ 0 & 0 & -1 & 1.5 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

For the ShapeHand system, the position data of each finger joint are acquired using the local tape coordinate system. When the hand is fully open as shown in Figure 2, the x-axis (denoted by X_S) runs along the length of the narrow flat tape (along each finger towards the finger tip), the y-axis (denoted by Y_S) is perpendicular to the face of the narrow flat tape (perpendicular to the palm), and the z-axis (denoted by Z_S) is towards the side of the narrow flat tape (in the direction across the palm).

Furthermore, the origin is fixed at the bottom of the palm in the middle of the wrist. With the wrist position and orientation data provided by the InterSense system, the finger joint position data need to be transformed from its local coordinate system to the InterSense coordinate system first and to the world coordinate system subsequently. If $\boldsymbol{s} = [x_s, y_s, z_s, 1]$ denotes the homogeneous coordinates of a position in the local ShapeHand coordinate system, then its corresponding homogeneous coordinates in the specified world coordinate system denoted by $\boldsymbol{s}_w = [x_{sw}, y_{sw}, z_{sw}, 1]'$ are given by:

$$\boldsymbol{s}_w = \boldsymbol{T}^{I\rightarrow W}\boldsymbol{T}^{S\rightarrow I}\boldsymbol{s} \tag{3}$$

where $\boldsymbol{T}^{S\rightarrow I}$ is the matrix for geometric transformation from the ShapeHand coordinate system to the InterSense coordinate system. If the position and orientation data provided by the InterSense system are denoted by (x_i, y_i, z_i) and $(\alpha_i, \beta_i, \gamma_i)$, then $\boldsymbol{T}^{S\rightarrow I}$ is given by

$$\boldsymbol{T}^{S\rightarrow I} = \begin{bmatrix} \boldsymbol{c}_\alpha \boldsymbol{s}_\beta - \boldsymbol{s}_\alpha \boldsymbol{c}_\beta \boldsymbol{s}_\gamma & \boldsymbol{c}_\beta \boldsymbol{c}_{\boldsymbol{i}} & \boldsymbol{c}_\alpha \boldsymbol{c}_\beta \boldsymbol{s}_\gamma + \boldsymbol{s}_\alpha \boldsymbol{s}_\beta & \boldsymbol{x}_{\boldsymbol{i}} \\ \boldsymbol{s}_\alpha \boldsymbol{c}_\gamma & -\boldsymbol{s}_\gamma & \boldsymbol{c}_\alpha \boldsymbol{c}_\gamma & \boldsymbol{y}_{\boldsymbol{i}} \\ -\boldsymbol{s}_\alpha \boldsymbol{s}_\beta \boldsymbol{s}_\gamma - \boldsymbol{c}_\alpha \boldsymbol{c}_\beta & \boldsymbol{s}_\alpha \boldsymbol{c}_\gamma & \boldsymbol{c}_\alpha \boldsymbol{s}_\beta \boldsymbol{s}_\gamma - \boldsymbol{s}_\alpha \boldsymbol{c}_\gamma & \boldsymbol{z}_{\boldsymbol{i}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

In equation (4), where *c* and *s* denoting *cos* and *sin* functions with subscripts denoting the orientation angles from the InterSense system.

3.2. Gesture recognition and virtual object manipulation

A basic sequence in immersive virtual object manipulation can be considered as consisting of object selection at the start, followed by object manipulation which can be a combination of translation, rotation, and scaling, and object release at the end. The implementation requires selection of a set of meaningful hand gestures as well as computation of the distances between hands and objects. For natural interaction and user comfort, Figure 3 shows three hand gestures selected for implementation of the five basic object manipulation operations, where the index finger pointing gesture shown in Figure 3(a) is used for not only selection of a virtual object but also for object translation and rotation based on the position of the left or right index fingertip and the hand orientation; the hand open gesture shown in Figure 3(b) is used for release of a selected object; and the gesture of two moving hands with the ring and small fingers closed shown in Figure 3(c) is for object scaling.

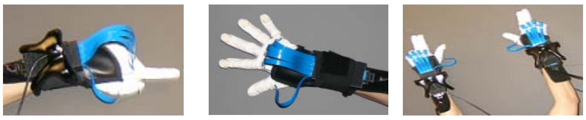
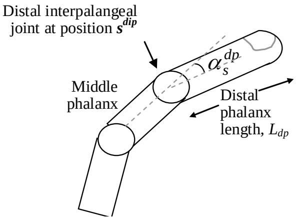

(a) (b) (c)

Figure 3. Hand gestures: (a) object selection, translation and rotation; (b) object release; and (c)object scaling



Distal interpalangeal joint at position $\boldsymbol{s}^{dip}$
α_s^{dp}
Middle phalanx
Distal phalanx length, L_{dp}

Figure 4. Index finger model

In order to execute the object selection and manipulation operation using the selected gestures, the 3D position of the left and right index fingertips need to be determined and tracked. As a hinged joint, there is only one DOF for the distal interphalangeal joint in the index finger. With the distal phalanx length known through the measurement of the user index finger, the index fingertip position can be determined based on the distal interphalangeal joint position and the distal phalanx bending angle with respect to the middle phalanx provided by the ShapeHand data glove as illustrated in Figure 4.

As shown in Figure 4, if $\boldsymbol{s}^{dip}$ = $[x_s^{dip}, y_s^{dip}, z_s^{dip}, 1]'$ denotes the homogeneous coordinates of the distal interphalangeal joint and α_s^{dp} denotes the distal phalanx bending angle in the local ShapeHand coordinate system, then the homogeneous coordinates of the index fingertip position in the world coordinate system denoted by $\boldsymbol{s}_w^{tip} = [x_{sw}^{tip}, y_{sw}^{tip}, z_{sw}^{tip}, 1]'$ are given by

$$\mathbf{s}_w^{tip} = \boldsymbol{T}^{I\rightarrow W}\boldsymbol{T}^{S\rightarrow I}\begin{bmatrix} 1 & 0 & 0 & \cos\alpha_s^{dp}\boldsymbol{L}_{dp} \\ 0 & 1 & 0 & \sin\alpha_s^{dp}\boldsymbol{L}_{dp} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\mathbf{s}^{dip} \tag{5}$$

where L_{dp} denotes the distal phalanx length.

Furthermore, the three selected gestures shown in Figure 3 are seen to consist of a combination of bending down and extending thumb and fingers in each hand, which can be determined based on the flexion angle of the proximal phalanx of the thumb or finger with respect to the back of the hand. With the finger flexion angle calibrated to 0° to correspond to a fully extended position (by hand opening) and 90° to correspond to a fully bending down position (by hand closing), the selected hand gestures can be recognised by expressing them using the corresponding binary state based on a threshold of 45°. Let two hands be denoted by H with its binary state set to logic 0 for the left and logic 1 for the right, and let the thumb and four fingers in each hand be denoted by T, I, M, R, S with the binary state of each one set to logic 0 if its flexion angle, α_s^{pp} , measured by the ShapeHand data glove is less than 45° and logic 1 otherwise. The index finger pointing gesture is given by

$$(\overline{H}\,\overline{T}I\overline{M}\,\overline{R}\,\overline{S})\cup(H\overline{T}I\overline{M}\,\overline{R}\,\overline{S}) \tag{6}$$

the hand open gesture is given by

$$(\overline{H}TIMRS)\cup(HTIMRS) \tag{7}$$

and the two hand moving gesture for object scaling is give by

$$(\overline{H}\,\overline{T}\,\overline{I}\,\overline{M}RS)\cap(H\overline{T}\,\overline{I}\,\overline{M}RS) \tag{8}$$

Apart from the object release operation which requires only recognition of the corresponding hand gesture, other object manipulation operations requires additional information of the object with respect to the user hands in terms of its location, orientation and size. Let the virtual object to be manipulated be denoted by $\boldsymbol{o}$ centred at (o_x, o_y, o_z) in the world coordinate system, with orientation of (o_α, o_β, o_γ), and with its bounding box defined by lengths of (L_x, L_y, L_z). Object selection requires not only recognition of the index finger pointing gesture by using equation (6) but also the position of the index fingertip within the object volume. Although different distance measures can be used, this was implemented by computing the distance between the index fingertip and object centre, $dist(\mathbf{s}_{w,i}^{tip}$, $\boldsymbol{o})$ where subscript $i = l$ for the left hand and $i = r$ for the right hand, since the object can be selected by using either the left or right hand. If the coordinates of the left or right index fingertip in the world coordinate system are given by ($x_{sw,i}^{tip}$, $y_{sw,i}^{tip}$, $z_{sw,i}^{tip}$), then

$$dist(\boldsymbol{s}_{w,i}^{tip}, o) = \sqrt{(x_{sw,i}^{tip} - o_x)^2 + (y_{sw,i}^{tip} - o_y)^2 + (z_{sw,i}^{tip} - o_z)^2} \tag{9}$$

In the implementation, a cube centred at (o_x, o_y, o_z) with all of its sides equal to the shortest length of the object bounding box is defined as the object selection volumes, whereby the object is selected when

$$\{dist(\mathbf{s}_{w,l}^{tip}, \mathbf{o}) \le \min(L_x, L_y, L_z)/2\} \cup$$
$$\{dist(\mathbf{s}_{w,r}^{tip}, \mathbf{o}) \le \min(L_x, L_y, L_z)/2\} \tag{10}$$

When the object is selected, the object bounding box is highlighted to provide a visual feedback to the user.

For object translation and rotation following object selection, it was implemented by making the object centre to follow the current 3D position of the index fingertip computed using equation (5) and the object 3D orientation to follow the current wrist orientation provided by Intersense. If both hands are making the index finger pointing gestures, the position of the selected object will follow the index finger with minimum distance to the object centre. Since a user may use one pointing index finger to do object selection, translation and rotation with the other hand open, the object release operation is disabled if equation (10) is satisfied.

Object scaling requires not only recognition of the two hand gesture using equation (8) but also the positions of the left and right index fingertips within the object selection volume. This was implemented by computing *dist*($\boldsymbol{s}_{w,l}^{tip}$, $\boldsymbol{o}$) and *dist*($\boldsymbol{S}_{w,r}^{tip}$, $\boldsymbol{o}$), with the scaling operation activated only when

$$\{dist(\mathbf{s}_{w,l}^{tip}, \mathbf{o}) \le \min(L_x, L_y, L_z)/2\} \quad \cap$$

$$\{dist(\mathbf{s}_{w,r}^{tip}, \mathbf{o}\} \le \min(L_x, L_y, L_z)/2\} \qquad (11)$$

Furthermore, with the scaling operation activated, the 3D object size is enlarged or reduced uniformly in 3D by the setting the length of the object bounding box equal to the distance between two index fingertips.

$$L = \max[abs(x_{w,l}^{tip} - x_{w,r}^{tip}), abs(y_{w,l}^{tip} - y_{w,r}^{tip}), abs(z_{w,l}^{tip} - z_{w,r}^{tip})] \qquad (12)$$

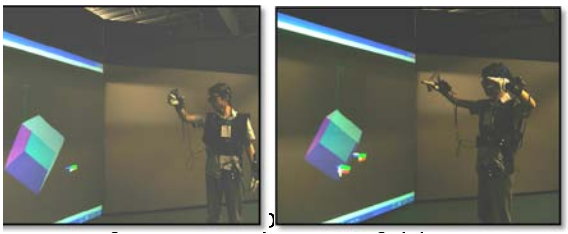
4. System performance

Presented in this section are the evaluation performed on the system developed, which includes manipulation of a virtual cube as an example to demonstrate its usability as well as speed and latency evaluation involving the use of a high speed camera.

4.1. Manipulation of virtual cube

To demonstrate the usability and evaluate the performance of the system developed, a simple six-colour cube was created as a virtual object for immersive manipulation by the user wearing a pair of wireless ShapeHand data gloves, a pair of wrist tracking devices, a head tracking device, and a pair of polarised glasses.

From the visual perspective, the user is able to see the stereoscopic images of the cube and his/her hands displayed through two projectors positioned at the back of a large screen operating in passive circular polarisation mode. With the head tracking device providing the position and orientation of the user, the user can physically moves around in front of the display screen with an impression of a 3D virtual cube floating in space, whereby a forward movement causes the 3D cube to appear nearer with a bigger size, a backward movement causes the 3D cube to appear further away with a smaller size, and a side movement with a side look via head rotation causes a different side the 3D cube to appear. Furthermore, to provide a visual feedback to the user, the side of the cube nearest to the pointing index finger is highlighted if the cube is selected.



translation and rotation; and (b) Cube scaling

From the interaction perspective, the simplicity and intuitiveness of the hand gestures were seen to enable a new user to quickly handle and manipulate the 3D cube quickly, namely, pointing the index finger to touch (select), drag and rotate the 3D cube in 3D space as shown in Figure 5(a), passing the 3D cube from one hand to another hand (from one pointing index finger to another pointing index finger), sliding two hands (with the ring and small finger closed) with respect to each other to enlarge and reduce the cube size as shown in Figure 5(b), and opening the hand(s) to detach from the cube.

Furthermore, the system is able to perform the required operations with certain deviation in the gestures made such as fingers not fully open and closed, and highly robust recognition can be achieved by performing calibration of hand close and open gestures at the start.

As an example, Figure 6 shows some of the data acquired from performing a short sequence of selection, translation and release of the 3D cube, namely, (o_x, o_y, o_z) to show the 3D position variation of the cube centre using red, green and blue dotted lines, ($x_{sw,r}^{tip}$, $y_{sw,r}^{tip}$, $z_{sw,r}^{tip}$) to show the 3D position variation of the right hand index finger tip using red, green and blue solid lines, and α_s^{pp} to show the flexion angle of the proximal phalanx of the right hand middle finger in a black solid line.

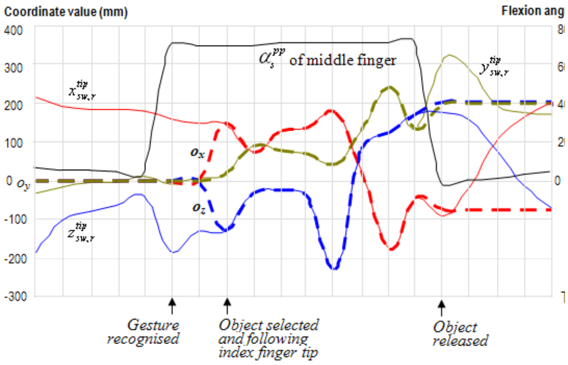


Figure 6. A sequence of dynamic gesture data

With an open hand gesture at the start of the sequence, it is seen from Figure 6 that α_s^{pp} is around 0º, the user right hand moves in the horizontal plane as indicated by the changing coordinate values of $x_{sw,\,r}^{tip}$ and $z_{sw,\,r}^{tip}$ with $y_{sw,\,r}^{tip}$ roughly constant, and the virtual cube is stationary at the origin of the world coordinate system as indicated by (o_x, o_y, o_z) = (0, 0, 0). Soon after the hand gesture changed into an index finger pointing gesture as indicated by the sharp rise of α_s^{pp} from 0º to around 70º due to the middle finger closed, the index finger tip is seen to approach the cube with the coordinate values of ($x_{sw,\,r}^{tip}$, $y_{sw,\,r}^{tip}$, $z_{sw,\,r}^{tip}$) moving towards (o_x, o_y, o_z). With the cube size of 250x250x250 mm, the cube is selected when the distance between the index finger tip and the cube centre, dist ($\boldsymbol{s}_{w,\,r}^{tip}$, $\boldsymbol{o}$) computed using equation (9), is less than 125 mm as specified by equation (10). Once selected, the cube is seen to follow the index finger tip with the coordinate values of o_x following $x_{sw,\,r}^{tip}$, o_y following $y_{sw,\,r}^{tip}$, and o_z following $z_{sw,\,r}^{tip}$. Finally, the user hand opens to release the object as indicated by α_s^{pp} falling back to around 0º due to the opening of the middle finger, the cube is seen to stay at its final position with (o_x, o_y, o_z) fixed as the user hand moves away.

4.2. System performance evaluation and analysis

A number of tests were also conducted to assess the key measures of real-time performance in terms of speed and latency.

For performance evaluation of speed based on manipulation of a virtual cube, a counter is inserted at the start of each program thread to record the number of times to run the thread per second. With continuous hand movement in 3D space, a typical example showing the frequency of executing each program thread over a period of one minute is shown in Figure 7.

From Figure 7, the program threads of hand gesture recognition and position data acquisition based on InterSense are seen to be relatively fast with relatively large fluctuations. Whilst the former is seen to be the fastest one with an average execution frequency of 136 times per second and the largest variation between the maximum of 171 times per second and the minimum of 102 times per second, the latter is the second fastest with an average execution frequency of 118 times per second and a variation between the maximum of 159 times per second and the minimum of 99 times per second. Very similar behaviour of the execution frequencies for the ShapeHand gesture data acquisition program thread and the stereoscopic display program thread are also seen from Figure 7, with the former slightly faster at 62 times per second on average between the maximum of 65 times per second and the minimum of 59 times per second, and the latter at 60 times per second between the maximum of 64 times per second and the minimum at 54 times per second. Since the stereoscopic display program thread is the slowest, the speed of the system depends on the complexity and the number of the objects to be displayed. Based on the lowest execution frequency shown in Figure 7, the system can operate at 54 frames per second for simple objects.

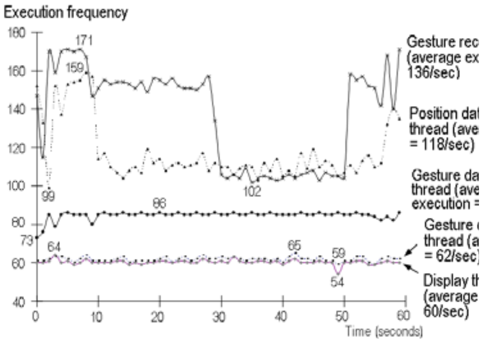


Figure 7. Thread execution frequency versus time

To confirm the adequacy of the system speed performance for dynamic gesture recognition, a test was also carried out to check the tracking speed against the maximum speed of finger movement. By wearing the ShapeHand data glove with the index finger open and closed repeatedly at the highest possible speed, the virtual hands on the stereoscopic display was found to follow the angular movement of the index finger at a maximum speed around 14 times per second.

For performance evaluation of latency, a high speed video camera was used to record the hand movement made by user wearing the ShapeHand data glove as well as the movement of the virtual hands appeared on the stereoscopic screen. With the hand opening and closing repeatedly, the video was captured at 64 frames per second, and video analysis of the corresponding hand gestures showed a delay around 6 frames of the virtual hand movement with respect to the real hand movement, which is equivalent to a latency of approximately 94ms.

5. Conclusions

The paper demonstrates an approach to achieve immersive virtual object manipulation using natural hand gestures. In particular, the paper describes (a) integration of a wireless high DOF hand gesture data glove, a wireless position tracking system, and a stereoscopic display; (b) algorithms developed for recognition of dynamic hand gestures; and (c) system performance evaluation conducted to assess its usability. Overall, the system is shown to provide an immersive and interactive environment, whereby a user can visualise in stereoscopic mode and interact in 3D with virtual objects using natural hand gestures. Furthermore, the simplicity and intuitiveness of the selected hand gestures as well as robustness in recognition of imprecise hand gestures enable users to quickly master the object manipulation operations with little effort. Although the virtual object used for system demonstration is a simple one based on the object bounding box, the work is seen as an important first step toward manipulations of complex objects. Particularly, it has been shown that the system can operate at 54 frames per second in the worst case with a latency time of approximately 94 ms using a PC with 3 GHz CPU.

Acknowledgment: The authors would like to thank Dr. X. Chen for his technical support.

References

[1] Woods, A.J., "Compatibility of display products with stereoscopic display methods", *in Proceedings of the International Display Manufacturing Conference*, Taipei, Taiwan, 2005.

[2] Wang, X.H., Good, W.F., Fuhrman, C.R., Sumkin, J.H., Britton, C.A., and Golla, S.K., “Stereo CT Image Compositing Methods for Lung Nodule Detection and Characterization”, *Academic Radiology*, 2005, Volume 12, Issue 12, pp. 1512 – 1520.

[3] Patel, D., Muren, L.P., Mehus, A., Kvinnsland, Y., Ulvang, D.M., and Villanger, K.P., “A virtual reality solution for evaluation of radiotherapy plans”, *Radiotherapy and Oncology*, 2007, Volume 82, Issue 2, pp. 218-221.

[4] Zhang, S., Demiralp, C., Keefe, D., DaSilva, M., Laidlaw, D.H., Greenberg, B.D., Basser, P.J., Pierpaoli, C., Chiocca, E.A., and Deisboeck, T.S., “An immersive virtual environment for DT-MRI volume visualization applications: a case study”, *in Proceedings of IEEE Visualization Conference*, San Diego, USA, 2001, pp. 437-440.

[5] Kober, C., Boerner B.I., Mori, S., Tellez, C.B., Klarhöfer, M., Scheffler, K., Sader, R., and Zeilhofer, H.F., “Stereoscopic 4D-Visualization of Craniofacial Soft Tissue based on Dynamic MRI and 256 Row 4D-CT”, *Advances in Medical Engineering*, Springer Proceedings in Physics, 2007, Part II, pp. 175-180.

[6] Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., and Twombly, X., “Vision-based hand pose estimation: A review”, *Computer Vision and Image Understanding*, 2007, Volume 108, pp. 52–73.

[7] Sturman, D.J., and Zeltzer, D., “A survey of glove-based input”, *IEEE Computer Graphics and Applications*, 1994, Volume 14, Issue 1, pp. 30-39.

[8] Bowman, D., Wingrave, C., Campbell, J., and Ly, V., “Using pinch gloves for both natural and abstract interaction techniques in virtual environments", *in Proceedings of HCI International*, New Orleans, USA, 2001, pp. 629-633.

[9] ShapeHand data glove, Measurand Inc., (www.measurand.com).

[10] Danisch, L., Englehart, K., and A. Trivett, “Spatially continuous six degree of freedom position and orientation sensor,” *Sensor Review*, 1999, Volume 19, Issue 2, pp. 106-112.

[11] IS-900 precision inertial-ultrasonic motion tracking system, InterSense Inc., (www.isense.com)

[12] Wormell, D., and Foxlin, E., “Advancements in 3D interactive devices for virtual environments”, *in Proceedings of the Workshop on Virtual Environments*, Zurich, Switzerland, 2003, pp. 47-56.

Hand motion recognition and visualisation for direct sign writing

Gan Lu*, Lik-Kwan Shark*, Geoff Hall* and Ulrike Zeshan+
*Applied Digital Signal & Image Processing Research Centre (ADSIP)
+International Centre for Sign Languages and Deaf Studies
University of Central Lancashire, PR1 2HE
{Glu@uclan.ac.uk}

Abstract

Although SignWriting provides an intuitive notation system based on pictorial symbols to enable any sign based language in the world to be transcribed into a written form, it is a time consuming process for keyboard based input. To address the challenge of direct sign writing, the paper presents a human-computer-interaction system developed for recognition and visualisation of hand movements. The system is shown to be able to display the corresponding SignWriting symbols for various hand movements performed by two hands based on motion characteristics such as movement planes, movement directions, straight/curve movement paths, clockwise/anti-clockwise movements, and single/repeated movements.

1. Introduction

There are several notation systems to enable a sign language communicated in a visual-gestural form to be transcribed into a written form, and SignWriting (SW) developed by Valerie Sutton in 1974 is a popular one among the deaf communities [1]. Each sign in SW is represented by a sign-box containing a composition of basic pictorial symbols to depict postures and movements of body, hand and head as well as facial expressions. On one hand, it is simple to use SW to write any sign language in the world; on the other hand, it is a time consuming process for keyboard based input due to a large number of pictorial symbols for selection and a number of spatial manipulations (rotation and translation) of selected symbols required to compose a sign [2]. A challenge is therefore presented to develop a Human-Computer-Interaction (HCI) system to enable automatic transcription of articulated signs into corresponding sign-boxes in an electronic form without using keyboards. With hand movements forming a core part of any sign language, the paper focuses on the development of automatic recognition and visualisation of hand motion in 3D.

One possible approach for hand motion recognition is based on the use of single or multiple video cameras [3], whereby hand movements in 3D are tracked across the video sequence based on hand positions detected from each frame using a computer vision algorithm. Possible problems associated with this approach include occlusion resulting in loss of hand positions due to the restricted camera view angle, and high computation cost due to the requirement to track rapid hand motion and to process a large data set. Since these problems can be overcome by wearing a sensor at the expense of introducing a small inconvenience to the signer, it is adopted as the approach presented in this paper, whereby an IS-900 wireless tracking system from InterSense with two tracking devices attached to the signer wrists [4] is used to provide the required hand position data in a more robust and rapid manner.

With the paper focusing on recognition of the dynamic hand movements in 3D between a starting gesture and an ending gesture, the proposed method assumes that the starting and ending hand gestures can be recognised using a data glove. ShapeHand from Measurand which uses fibre-optic sensors to provide all finger joint movement information has been used by the authors for this purpose [5][6].

The paper is organised as follows. While Section 2 describes hand motions and their corresponding notations in SignWriting, Section 3 presents algorithms developed for hand motion recognition. These are followed by Section 4 showing a 3D stereoscopic user interface developed to generate corresponding SignWriting notations and to produce a stereoscopic view of hand movements made. Finally, some concluding remarks are given.

2. SignWriting Hand Motions and Notations

According to International SignWriting Alphabet (ISWA) 2008 [7], there are 30 pictorial symbol groups containing 639 basic pictorial symbols. Since each basic symbol can have up to a maximum of 96 variations (up to 6 different fills and 16 rotations), it results in a large dictionary that currently contains a total of 35,023 valid symbols. A much longer time is therefore needed to select and combine symbols to input a single sign than to type a word.

By using the ISWA basic symbols, some hand movement examples in 3D can be transcribed as shown in Fig.1, where unfilled/filled arrow heads are used to indicate left/right hands, double-stem/single-stem arrows are used to indicate movements parallel to the wall/floor planes, and arrow orientations are used to indicate movement directions in either plane (with hand movements in each plane divided into eight principal directions). Furthermore, arrows can be straight/curved to indicate movement paths, and duplicated/triplicated for repeated movements.

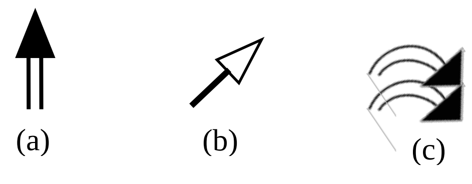


Figure 1 SW symbols showing (a) vertical movement made by right hand in the wall plane; (b) diagonal forward movement made by left hand in the floor plane; (c) two repeated curved movements made by left hand along the horizontal direction in the wall plane.

3. Hand Motion Recognition

For simplicity of implementation, the input of each sign is based on a fixed three-phase signing sequence that consists of making a hand gesture at the starting position, performing a follow-up hand movement, and finishing with a hand gesture at the ending position. While the starting and ending hand gestures are recognised by processing finger joint positions acquired from a pair of ShapeHand data glove from Measurand, hand movements are recognised by processing hand positions acquired from the InterSense tracking system. The hand motion data is acquired at a rate of 50 Hz, the starting and ending hand gestures are used to start and end the storage of the acquired hand motion data in a memory buffer for processing. The sequence of processing starts with identification of repeated movements, and it is followed by identification of movement planes, movement directions, path linearity, and clockwise/anti-clockwise movements, respectively.

3.1. Identification of repeated movements

Identification of repeated hand movements is based on the number of times the hand has moved from the space around the starting position to the space around the farthest position between the starting and ending gestures. Let the starting hand position, defined by the starting hand gesture, be denoted by $\boldsymbol{p_i} = [x_i, y_i, z_i]$; and let the length of the acquired hand motion data for one sign, defined by the ending hand gesture, be denoted by L. If $\boldsymbol{p_n} = [x_n, y_n, z_n]$ denotes a hand position along the hand motion trajectory made by the signer with $n < L$, then the distance between $\boldsymbol{p_i}$ and $\boldsymbol{p_n}$ is given by

$$dist(p_i, p_n) = \sqrt{(x_i - x_n)^2 + (y_i - y_n)^2 + (z_i - z_n)^2} \tag{1}$$

Let $\boldsymbol{p}_{n\text{-}1}$ and $\boldsymbol{p}_{n+1}$ denote two neighbouring hand positions along the hand motion trajectory before and after p_n. By computing $dist(\boldsymbol{p}_i, \boldsymbol{p}_{n\text{-}1})$, $dist(\boldsymbol{p}_i, \boldsymbol{p}_n)$ and $dist(\boldsymbol{p}_i, \boldsymbol{p}_{n+1})$ starting from $\boldsymbol{p_i}$ for each hand motion position, if $dist(\boldsymbol{p}_i, \boldsymbol{p}_n)$ is greater than both $dist(\boldsymbol{p}_i, \boldsymbol{p}_{n\text{-}1})$ and $dist(\boldsymbol{p}_i, \boldsymbol{p}_{n+1})$ by a specified threshold value, then $\boldsymbol{p}_n$ is identified as the first farthest position reached by the hand. The use of a threshold is to avoid the small movement jitter problem in the acquired position data.

Upon detecting the first farthest position, a check is made to see if $n = L - 1$. If it is, then it indicates that the ending gesture position has been reached and the hand movement is identified as a non-repeating movement. If it is not the case, subsequent hand position data are processed based on their distances to $\boldsymbol{p_i}$ to detect the second starting position for the repeated movement. If $dist(\boldsymbol{p}_i, \boldsymbol{p}_n)$ is less than both $dist(\boldsymbol{p}_i, \boldsymbol{p}_{n\text{-}1})$ and $dist(\boldsymbol{p}_i, \boldsymbol{p}_{n+1})$ by the specified threshold value after encountering the first farthest hand position, then $\boldsymbol{p}_n$ is identified as the second starting position for the repeated movement. The search of the farthest position is then repeated to detect the second farthest position reached by the hand, and the whole process could be repeated for the third time in order to reach the ending hand gesture with $n = L - 1$.

3.2. Identification of movement planes

For two-plane movement classification, hand movements can be interpreted as parallel to either the floor plane or the wall plane. This is achieved based on the angle made by the movement direction vector of each hand position along the hand motion trajectory with respect to the floor plane. Using the Cartesian co-ordinate system with the x-y plane forming the wall plane and the x-z plane forming the floor plane centred at the starting hand position denoted by p_i, if p_n denotes a hand position along the hand motion trajectory, then the absolute angle made by it with respect to the floor plane is given by

$$\alpha_n = \tan^{-1}\left|\frac{(y_n - y_i)}{\sqrt{(x_n - x_i)^2 + (z_n - z_i)^2}}\right| \tag{2}$$

Since the angle of the hand movement direction vector with respect to the floor plane should be less than 45^o for the floor plane movement, the hand movement is identified as parallel to the floor plane if all angle values computed for each hand position along the hand motion trajectory are less than 45^o with respect to the floor plane, otherwise it is considered as parallel to the wall plane.

3.3. Identification of movement directions

For eight-direction movement classification, hand movements can be interpreted as left/right horizontal movements, up/down vertical movements, or left/right up/down diagonal movements parallel to either the wall or floor plane as shown in Fig. 2. This is achieved based on the angle made by the farthest hand position with respect to the x axis in the movement plane identified. As shown in Fig. 2, if p_v denotes the farthest hand position detected, then the sine and cosine of its angle denoted by β in the wall plane are given by

$$\sin\beta = (y_v - y_i)/\sqrt{(x_v - x_i)^2 + (y_v - y_i)^2} \quad (3)$$

$$\cos\beta = (x_v - x_i)/\sqrt{(x_v - x_i)^2 + (y_v - y_i)^2} \quad (4)$$

To determine the angular value of β, the signs of sinβ and cosβ are used to find the quadrant in which β lies, since, for the wall plane, sinβ is positive for β in the left hand side and cosβ is positive for β in the top half, and for the floor plane sinβ is positive for β in the bottom half and cosβ is positive for β in the right hand side (see Fig. 2). With each movement plane partitioned using 45° angular sectors centred at eight principal directions as shown in Fig. 2, the final movement direction is determined by finding the angular sector in which β lies based on the corresponding limits of sinβ values precomputed for each angular sector.

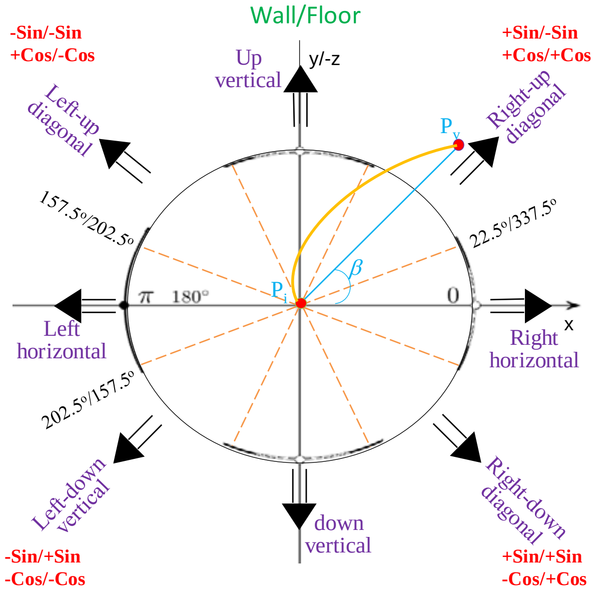


Figure 2 Movement Directions in Wall/Floor Plane

3.4. Identification of path linearity

Hand motion paths in SW can be straight (linear) or curved (non-linear). The identification of path linearity is based on the maximum deflection angle with respect to the straight line movement from the starting hand position to the farthest hand position.

To explain the algorithm implemented, the hand motion path illustrated in Fig. 3 is used as an example, where the movement can be interpreted as a curved clockwise one in parallel to the wall plane. Let (x_i, y_i) and (x_v, y_v) be the coordinates of the starting and farthest hand position denoted by p_{i} and p_v in the wall plane. The straight movement from (x_i, y_i) to (x_v, y_v) is described by a line equation with its slope and the y-intercept point given by

$$k = \frac{y_v - y_i}{x_v - x_i} \quad (5)$$

$$c = \frac{x_v y_i - x_i y_v}{x_v - x_i} \quad (6)$$

For a hand position denoted by (x_n, y_n) on the curved path from (x_i, y_i) to (x_v, y_v), the shortest distance with respect to the straight line is given by

$$d_n = \left| \frac{kx_n - y_n + c}{\sqrt{k^2 + 1}} \right| \quad (7)$$

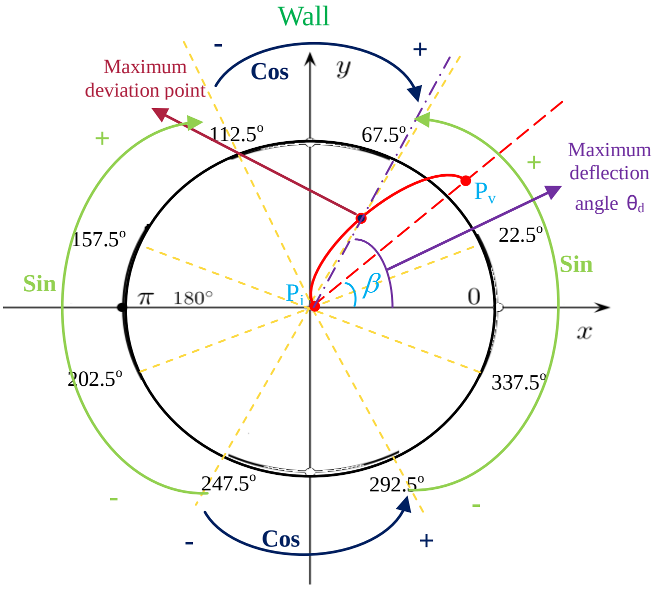


Figure 3 Path Linearity Identification Example

Using (7) to compute the distance from each point on the curved path to the straight line between (x_i, y_i) and (x_v, y_v), the maximum deviation point can be identified and used to compute the maximum deflection angle with respect to the horizontal axis denoted by θ_d in Fig. 3. If the difference between θ_d and β is less than the deflection angle threshold that is set to 11.25°, then the motion path is identified as a straight one (linear). Otherwise, it is identified as a curved one (non-linear).

In the implementation, the difference between θ_d and β is determined based on the values of sinθ_d, and cos θ_d as well as sinβ and cosβ obtained from the previous stage of identification of the movement direction (see Section 3.3). The differences between these values are also used to determine clockwise/anti-clockwise movements, since a clockwise movement in the left half or an anti-clockwise movement in the right half of the movement plane will result in sinθ_d greater than sinβ, whereas a clockwise movement in the top half or an anti-clockwise movement in the bottom half of the movement plane will result in cosθ_d greater than sinβ. Furthermore, in order to avoid the ambiguity problem caused by those curved movements near the horizontal axis to result in the same cosine value being produced and near the vertical axis to result in the same sine values being produced, the difference of sine values are used for β lying in the angular sector from 292.5° to 67.5° and from 112.5° to 247.5° and the difference of cosine values are used for β lying in the angular sectors from 67.5° to 112.5° and from 247.5° to 292.5° to determine clockwise/anti-clockwise movements, as shown in Fig. 3.

4. Direct Sign Writing Interface

To provide an effective visual feedback of hand motion and its recognition, a stereoscopic direct sign writing interface has been developed to produce a stereoscopic view of the hand movements made in 3D and to generate corresponding SignWriting notations. Since each sign is articulated in a sequence with three gestural parts, a hierarchical sign-box consisting of a 5-by-5 lattice is constructed to enable three sets of symbols, corresponding to the hand and motion gestures made for each sign, to be displayed in a sequential manner from the most inner square to the outer squares. This is illustrated in Fig. 4. Recognition of the starting hand gesture results in the corresponding symbols being displayed in the most inner yellow square; recognition of the subsequent movements by two hands results in their symbols being displayed in the sandwiched red squares along the movement directions; and recognition of the ending hand gesture results in the corresponding symbols being displayed in the outer blue squares along the movement directions.

Figure 5 shows the interface developed for direct sign writing with a user making signs. From the user perspective, it is required to wear a pair of wireless ShapeHand data glove to provide hand gesture information, a wireless InterSense tracking device on each wrist to provide hand motion information, and a wireless InterSense head tracker to provide the position and orientation of the user head. A large screen with two back projectors operating in passive circular polarisation mode is used to provide a stereoscopic display. By wearing a pair of light-weight polarised glasses, the user is able to see the graphic models of his/her two hands floating in space, their movements in real-time and in stereoscopic mode with depth impression from his/her viewpoint, as well as SW symbols displayed in 5-by-5 lattice blocks according to the hand movements made.

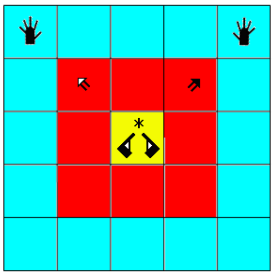


Figure 4 Sign-box showing diagonal up movements by two hands in wall plane

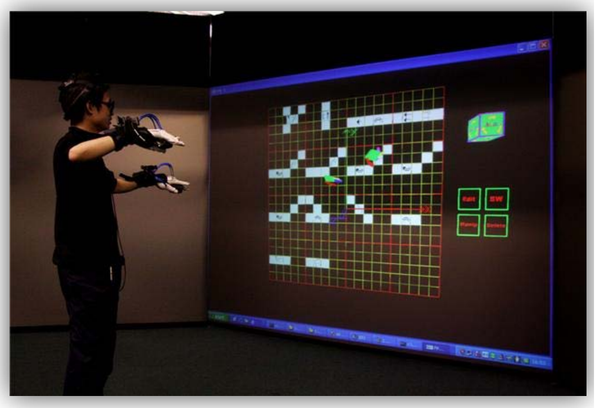


Figure 5 Direct sign writing interface

Various hand movements with different combinations of movement planes and directions as well as repeated movements, trajectory curved and clockwise/anti-clockwise movements were performed to evaluate the recognition performance. Some representative examples to demonstrate the capability of hand movement recognition are shown in Figs. 6-12, where the first and second columns show the 3D movement trajectories made by the left and right hands, respectively, and the third column shows the corresponding SW symbols generated. These examples include separate left and right diagonal forward movements by the left and right hands in the floor plane (Fig. 6); the left hand performing a diagonal backward movement in the left side of the floor plane with the right hand performing a diagonal up movement in the right side of the wall plane (Fig. 7); the left and right hands performing separate clockwise parabolic movements in the left and right sides of the floor plane along the horizontal direction (Fig. 8); the left hand performing a diagonal backward clockwise parabolic movement in the left side of the floor plane with the right hand performing a horizontal clockwise movement in the right side of the wall plane (Fig. 9); duplicated horizontal outward movements performed by both the left and right hands in the left and right sides of the floor plane (Fig. 10); the left and right hands performing the same movement which is an anti-clockwise parabolic movement repeated twice along the horizontal direction in the left side of the wall plane (Fig. 11); and triplicated movements with the left hand performing repeated horizontal outward movements in the left side of the wall plane and the right hand performing repeated diagonal forward movements in the right side of the floor plane (Fig. 12).

Conclusions

This paper presents the work done to develop a unique human-computer-interaction system for direct sign writing with a particular focus on recognition and visualisation of hand movements. Based on the SW hand motions and notations, the paper describes the algorithms developed for recognition of hand motions based on various motion characteristics, and the direct sign writing interface implemented for 3D hand motion visualisation and SW notation display. The system is shown to involving various combinations of motion characteristics provide a good visual feedback and an intuitive sign-box display format. A good mixture of hand movements can be recognised which include different movement planes and directions as well as repeated movements, trajectory curved, and clockwise/anti-clockwise movements. Although the development could be viewed as at its early stage with a significant amount of further work in order to cover the full spectrum of hand motions, an excellent basis is offered by the system to achieve the goal of rapid and direct sign writing without using keyboards.

Acknowledgements

The authors would like to thank Dr. Chen Xin for technical help.

References

SignWriting® Site url: http://www.signwriting.org/.

K. Clark and D. Gunsauls. Written in signed and spoken languages. *The SignWriter Newsletter*. Spring Issue, 1997.

J. Rett, S. Luis and J. Dias. Laban Movement Analysis for Multi-Ocular Systems. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept, 22-26, 2008, France

IS-900 precision inertial-ultrasonic motion tracking system. InterSense Inc. url:www.isense.com.

ShapeHand. Measurand Inc. url: www.measurand.com.

G. Lu, L. Shark, G. Hall and U. Zeshan, Dynamic Hand Gesture Tracking and Recognition for Real-Time Immersive Virtual Object Manipulation, *2009 International Conference on CyberWorlds*, pp.29-35.

SignWriting International SignWriting Alphabet (ISWA 2008). url: http://www.signwriting.org/lessons/iswa/

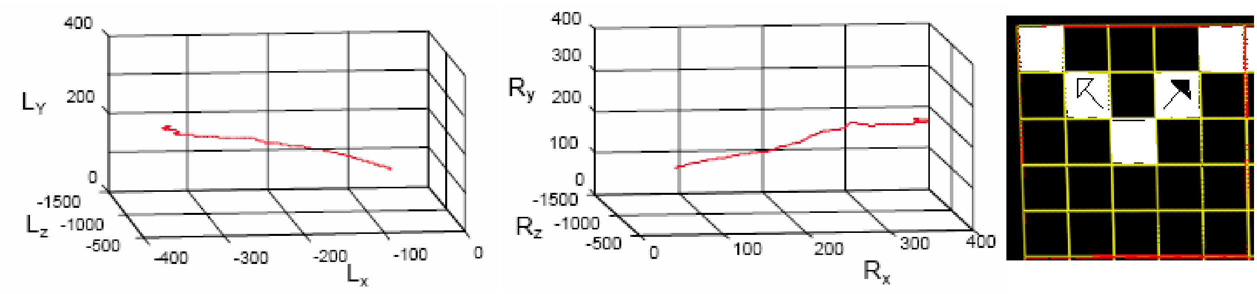


Figure 6 Left and right diagonal forward movements in floor plane

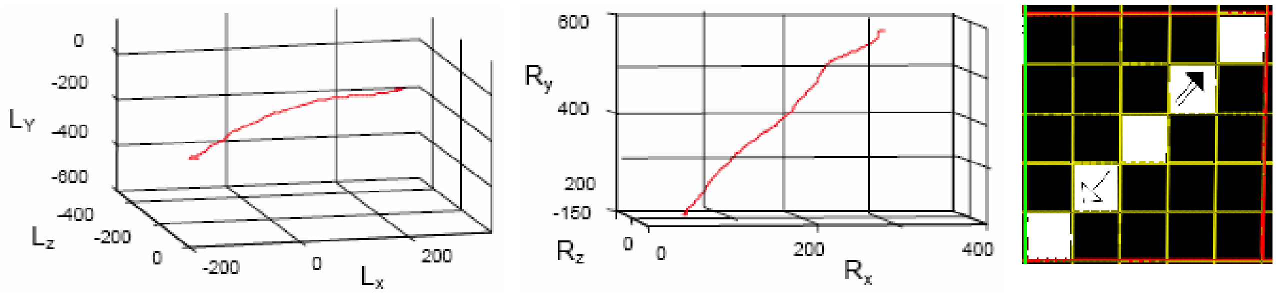


Figure 7 Left-back and right- up diagonal movements in floor and wall planes

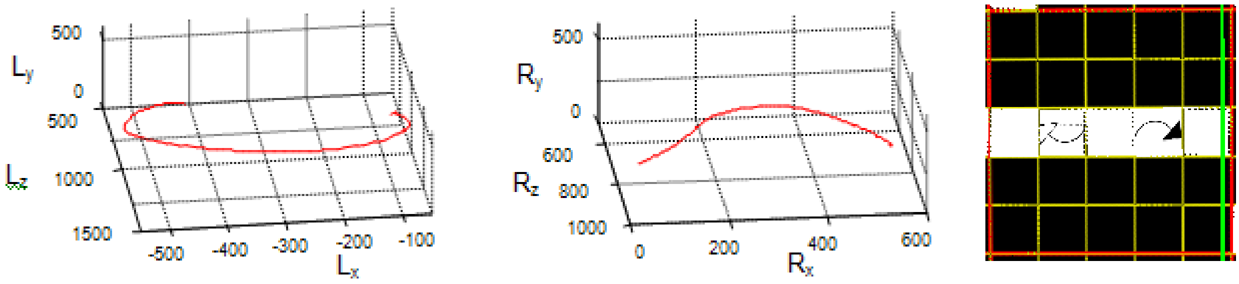


Figure 8 Left and right horizontal clockwise parabolic movements in floor plane

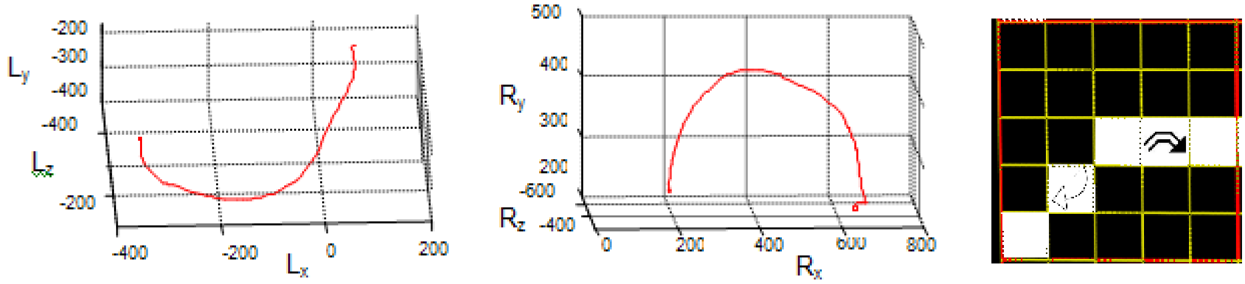


Figure 9 Left-back diagonal and right horizontal clockwise parabolic movements in floor and wall planes

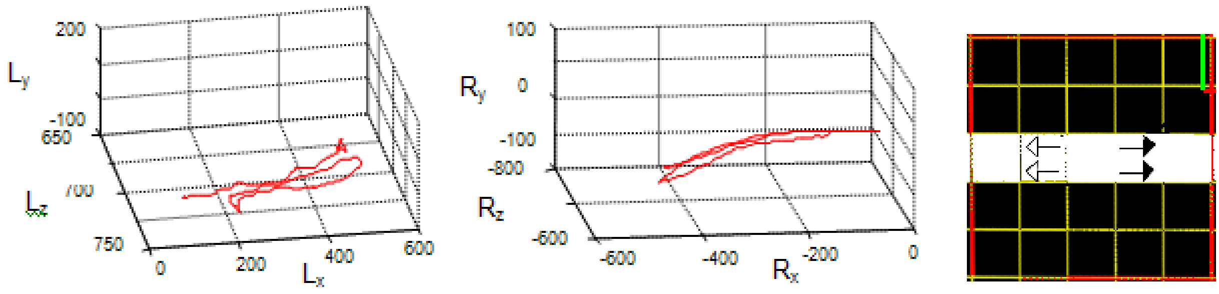


Figure 10 Left and right duplicated movements in floor plane

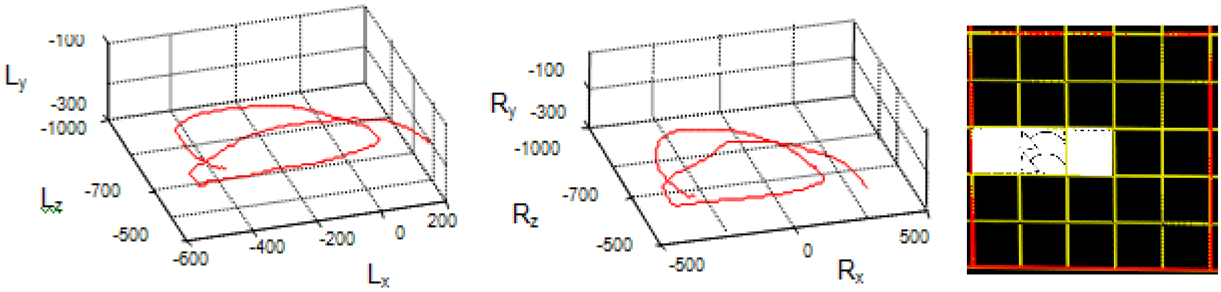


Figure 11 Synchronous parabolic anti-clockwise movements in floor plane

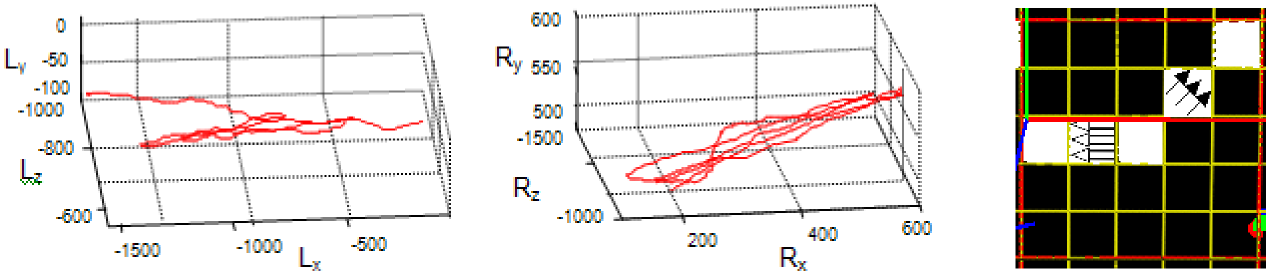


Figure 12 Triplicated left and right-up movements in wall and floor planes

Immersive manipulation of virtual objects through glove-based hand gesture interaction

Gan Lu, Lik-Kwan Shark, and Geoff Hall
Applied Digital Signal
and Image Processing Research Centre,
University of Central Lancashire,
Preston, UK. PR1 2HE
e-mail:glu@uclan.ac.uk

Ulrike Zeshan
International Centre for Sign Languages
and Deaf Studies,
University of Central Lancashire
Preston, UK. PR1 2HE

***Abstract*—Immersive visualisation is increasingly being used for comprehensive and rapid analysis of objects in 3D and object dynamic behaviour in 4D. Challenges are therefore presented to provide natural user interaction to enable effortless virtual object manipulation. Presented in this paper is the development and evaluation of an immersive human–computer interaction system based on stereoscopic viewing and natural hand gestures. For the development, it is based on the integration of a back-projection stereoscopic system for object and hand display, a hybrid inertial and ultrasonic tracking system to provide the absolute positions and orientations of the user's head and hands, as well as a pair of high degrees-of-freedom data gloves to provide the relative positions and orientations of digit joints and tips on both hands. For the evaluation, it is based on a two-object scene with a virtual cube and a CT (computed tomography) volume created for demonstration of real-time immersive object manipulation. The system is shown to provide a correct user view of objects and hands in 3D with depth, as well as to enable a user to use a number of simple hand gestures to perform basic object manipulation tasks involving selection, release, translation, rotation and scaling. Also included in the evaluation are some quantitative tests of the system performance in terms of speed and latency.**

Keywords - Hand gesture tracking and recogntion, Immersive stereoscopic visualisation, virtual object manipulation.

1 Introduction

Stereoscopic display (Woods 2005) enables viewing of an object as an image in 3D with depth information and the object behaviour as an image sequence in 4D with depth and time information. This is increasingly being used in comprehensive and rapid visualisation of complex data sets, such as 3D CT (computed tomography) data and 4D dynamic MR (magnetic resonance) data in medical diagnosis and treatment planning (Wang et al. 2005; Patel et al. 2007; Greenberg 2001; Kober et al. 2007). A more immersive visualisation can be achieved by the use of a large wall display, as well as head tracking to allow the user to move around and to view from different perspectives based on the user's head position and orientation. However, there is a challenge to provide an immersive interaction with the virtual object projected in stereoscopic mode without using indirect manipulation methods such as keyboard-based control, mouse-based 3D widgets, or hand-held input devices like wireless 3D wands.

Although more direct manipulation of virtual objects could be achieved by sensing and processing various natural human communication signals such as audible speech and hidden electroencephalography (EEG) (Demirdjian et al. 2005; Corradini and Cohen 2002; Le´cuyer et al. 2008; Toyama 2006), none of these human–computer interface strategies can be compared with the hand gesture–based user interface that provides a far more intuitive, natural and immersive interaction for users to manipulate 3D virtualobjects based on normal human actions (Dipietro and Sabatini 2008; O'Hagan et al. 2002; Garg et al. 2009). Among various approaches to recognise hand gestures, one is based on the use of a video camera (Erol et al. 2007), whereby the hand movements and gestures are recognised based on the dynamic hand shapes extracted from the video sequence. The difficulties associated with this approach include:

- self-occlusion resulting in capture of partial hand gestures due to the restricted camera view angle;
- incorrect image segmentation of hands due to different lighting and background conditions; and
- high computation cost due to the requirements to track rapid hand motion and to handle high complexity of hand with at least one degree of freedom (DOF) for each hand digit joint and 27 DOF for just one hand (Adamo-Villani et al. 2007).

These difficulties can be overcome by

wearing a pair of data gloves with sensors to provide finger and thumb movement information at the expense of introducing a small inconvenience to the user (Sturman and Zeltzer 1994). The simplest type of data glove is based on contact, using conductive patches in the glove (Bowman et al. 2001). With the requirement of hand digits touching each other to make electrical contact, the recognisable gestures are not necessarily natural, and the number of identifiable gestures is limited. A more sophisticated data glove providing more digit movement information is based on flexure. It uses fibre-optic, mechanical or piezoresistive sensors to measure the deformation of each hand digit. For the work described in this paper, the hand gesture recognition is based on a high DOF data glove, called Shape-Hand from Measurand (ShapeHand Data Glove, Measurand Inc., http://www.measurand.com, Danisch et al. 1999). It can be considered as one of the most precise and sophisticated data gloves, which uses 40 fibre-optic sensors on each glove to provide all finger joint movement information, 27 DOF for each hand. To achieve immersive manipulation of virtual objects through glove-based hand gesture interaction, presented in this paper is a small-scale system implemented for demonstration and evaluation. The systemintegrates three elements:

• a pair of wireless high DOF data gloves (Measurand ShapeHand Data Glove, Measurand Inc., http://www.measurand.com) to provide all the hand digit joint movement information;

• a wireless hybrid inertial and ultrasonic tracking system (IS-900 Precision Inertial-ultrasonic Motion Tracking System, InterSense Inc., http://www.isense.com; Wormell and Foxlin 2003) to provide the 3D positions and 3D orientations of the user's head and two hands; and

• a large screen with two stereoscopic back projections to show the virtual object being manipulated and to provide a visual feedback of two hands with respect to the virtual object in 3D with depth.

The paper is organised as follows. While Sect. 2 presents system development, which includes a description of the hardware integration in Sect. 2.1 and software implementation in Sect. 2.2. Section 3 presents algorithms developed for data integration in Sect. 3.1, for gesture recognition in Sect. 3.2 and for object distance computation in Sect. 3.3. These are followed by some system demonstration and evaluation results in Sect. 4 to show system performance. Finally, some concluding remarks are given in Sect. 5.

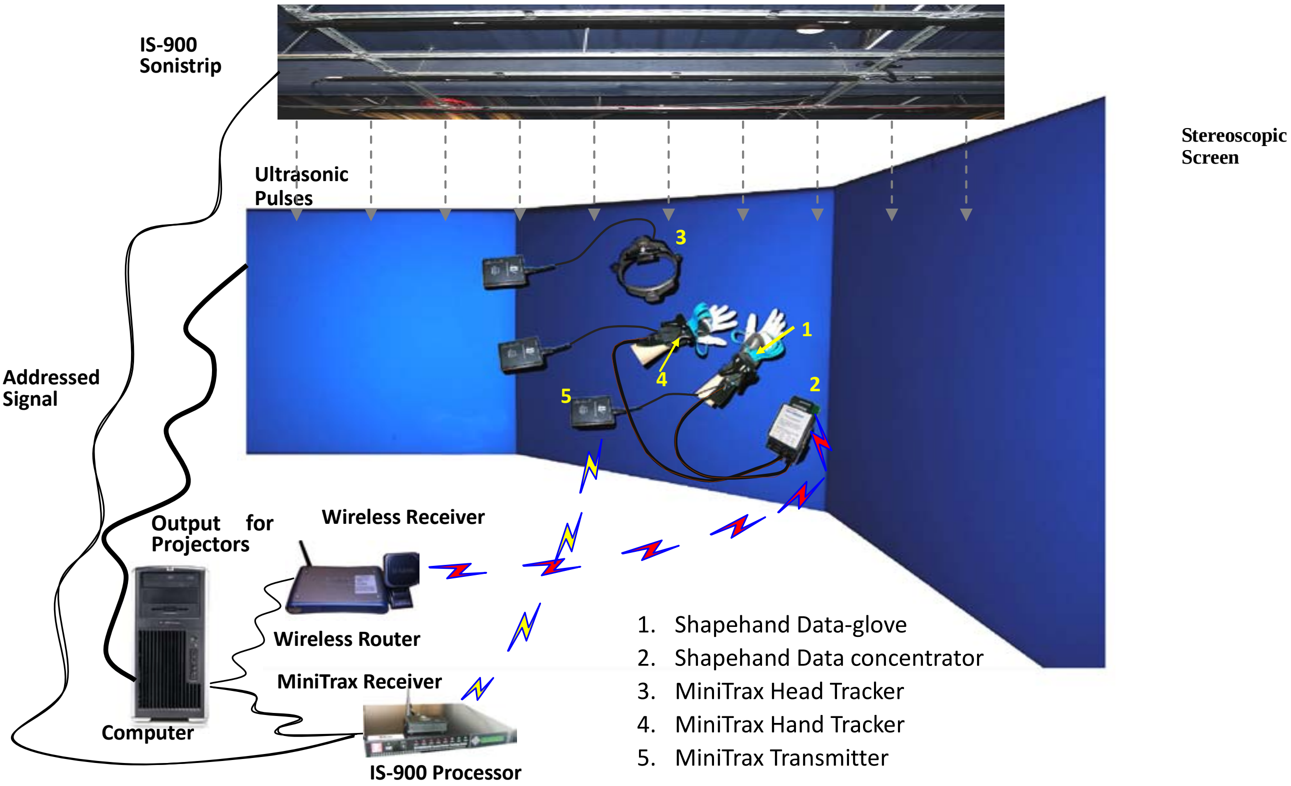


Fig. 1. System hardware block diagram

2 System development

This section presents hardware and software aspects associated with system development in terms of integration.

2.1 Hardware integration

The demonstration system developed for real-time immersive object manipulation is illustrated in Fig. 1. Driven by a desktop computer, the system is an integration of three subsystems for gesture data acquisition, position data acquisition and stereoscopic display.

For gesture data acquisition from hands, a pair of wireless ShapeHand data gloves from Measurand is used. These data gloves are based on flexible tapes embedded with multiple fibre-optic curvature sensors arranged to sense bend and twist along the length of each tape. By attaching the tapes to run along each digit with one end at the finger tip and the other end fed to a small data acquisition box at wrist, the gesture movements of digits introduce deformation to the tapes. Also, the bend and twist measured at each sensor location with respect to the wrist end of the tape enables relative positions and orientations of each digit joint to be determined. As shown in Fig. 1, via the ShapeHand Data Concentrator, the collected hand gesture data are transmitted to a wireless receiver/router connected to the Ethernet port of the computer. Since absolute hand position and orientation data in 3D space are not provided by ShapeHand data gloves, an IS- 900 wireless tracking system from InterSense (IS-900 Precision Inertial-ultrasonic Motion Tracking System, InterSense Inc., http://www.isense.com) is used to provide the required hand position data. The system is also used to provide the head position and orientation data in order to generate a correct view. The operation of the system is based on a combination of inertial tracking and ultrasonictracking. Whilst the outputs from the inertial sensors, consisting of accelerometers and gyros, are used to determine the position and orientation of each sensor in 3D space, the range measurements based on time-of-flight between ultrasonic emitters and receivers are used to correct the drifting effect inherent within the inertial sensors.

As shown in Fig. 1, IS-900 SoniStrips containing ultrasonic emitters are mounted on the ceiling, which transmit ultrasonic pulses upon receiving addressed signals from the IS-900 processor that is connected to the serial port of the computer. Three MiniTrax tracking devices containing inertial sensors and ultrasonic receivers are used with two attached to the user's wrists and one attached to the user's head. Each MiniTrax tracking device performs time-of-flight range measurement based on the ultrasonic pulses received, and transmit its position and orientation data to the corresponding MiniTrax receiver connected to the IS-900.

For stereoscopic display, the demonstration system uses one large screen with size of 2.74 m 9 2.06 m (shown as the middle screen in Fig. 1), and two back projectors operating in passive circular polarisation mode are connected to the computer through two dual DVI graphics card output ports. 3D objects with depth effect are seen by the user wearing a pair of light-weight polarised glasses. The computer used in the demonstration system runs on Microsoft Windows XP and is based on an Intel Xeon 3.06 GHz CPU with 2 GB RAM and NVIDIA Quadro FX 3,000 Graphics Card with 256 MB memory.

2.2 Software implementation

The system software is implemented as a Windows XPbased application using C??. To minimise development time, the software utilises the Microsoft Foundation Classes (MFC) to build the user interface and control units. A modified version of the standard Document View Model has been implemented to allow the input data to update the document object (containing the current user head and hand position, as well as gesture data) and the output display to be treated as an individual 'view' of the document object. Furthermore, the software is implemented following a multithread approach to minimise response time for interactive object manipulation. There are five parallel program threads with two of them performing hand gesture data acquisition and extraction from ShapeHand, and the other three performing position data acquisition from InterSense, gesture recognition and stereoscopic display, respectively.

The two program threads for hand gesture data acquisition and extraction are implemented based on the ShapeHand API (Application Programming Interface). With steps including initiation of data collection, receiving data and checking received data, the program thread for data acquisition obtains raw data from a pair of wireless ShapeHand data glove via the Ethernet port of the computer.

Using the raw data obtained, the required positions and orientations of a digit joint are determined in the program thread for data extraction.

The program thread of position data acquisition is implemented based on the InterSense API. It performs the data acquisition from the three MiniTrax tracking devices attached to the head and two wrists of the user via the serial port of the computer to provide their position and orientation data. Steps in this thread include data collection and data updating if the incoming data are found to be different from the previously received data.

The program thread of gesture recognition is based on the algorithm presented in the next Section. Essentially, it involves data merging through coordinate transformations, as well as tracking and recognising a number of prespecified hand gestures for manipulation of the displayed virtual objects.

The program thread of stereoscopic display is implemented based on OpenGL programming to provide 3D visual feedback to the user. This is done by generating two views of the virtual objects and two hands. As viewing through polarisation glasses results in each eye seeing only the view generated for it, it creates a visual immersion with depth impression. Steps in this program thread include the use of the head position data acquired to specify the viewing position and direction of the left and right eyes, configuration of the viewing frustum for each eye and stereo rendering to draw the left and right images of the 3D objects and hand models by perspective projection.

These five program threads are executed simultaneously by the computer with each thread assigned a slice of its CPU (central processing unit) time. The scheduling of the threads is done in a round-robin manner, with all threads having the same priority. In execution of the program thread of position data acquisition during its allocated time slice, no change in the received data will result in early switching to the next program thread.

3 Hand tracking and gesture recognition

This section focuses on two data processing operations, namely, coordinate transformation to fuse multiple position and orientation data sets, acquired using different referencing systems, and the dynamic hand gesture tracking and recognition methods for immersive objects manipulation.

3.1 Data integration through coordinate transformation

With different equipments using different coordinate systems for data acquisition, processing and display, coordinate transformations are required to bring different data sets into a unified coordinate system.

Figure 2 illustrates the spatial relationship between different coordinate systems. The world coordinate system is defined to have the same orientation as the stereoscopic display. With the x-axis (denoted by Xw) pointing towards the right, the y-axis (denoted by Yw) pointing upwards and the z-axis (denoted by Zw) pointing towards the viewer, this forms a right-handed coordinate system with a positive rotation about the axis in the anticlockwise direction. Furthermore, the origin of the world coordinate system (denoted by Ow) is located at the middle of the screen along the x-axis (1.37 m away from the screen right edge), 1 m above the floor along they-axis and 1.9 m in front of the screen along the z-axis.

For the InterSense system with its coordinate axes denoted by (XI, YI, ZI), the position data acquired for head and wrists are calibrated with respect to its origin denoted by OI at (-1.8, 1.5, 0m) in the world coordinate system as shown in Fig. 2. Furthermore, two rotation operations are required to align the orientations of the InterSense coordinate system with the orientations of the world coordinate system, namely, rotation of -90_ about the InterSense yaxis, to make the new InterSense x-axis parallel to the world coordinate x-axis, and rotation of 90_ about the new InterSense x-axis to make the new InterSense y- and z- axes parallel to the world coordinate systems. If i = [xi, yi, zi, 1]’ denotes the homogeneous coordinates of a position in the InterSense coordinate system, then its corresponding

homogeneous coordinates in the specified world coordinate system denoted by iw = [xiw, yiw, ziw, 1]’ are given by

$$i_w = \boldsymbol{T}^{I \to W} i \qquad (1)$$

where $\boldsymbol{T}^{I \to W}$ is the matrix for geometric transformation from the InterSense coordinate system to the world coordinate system.

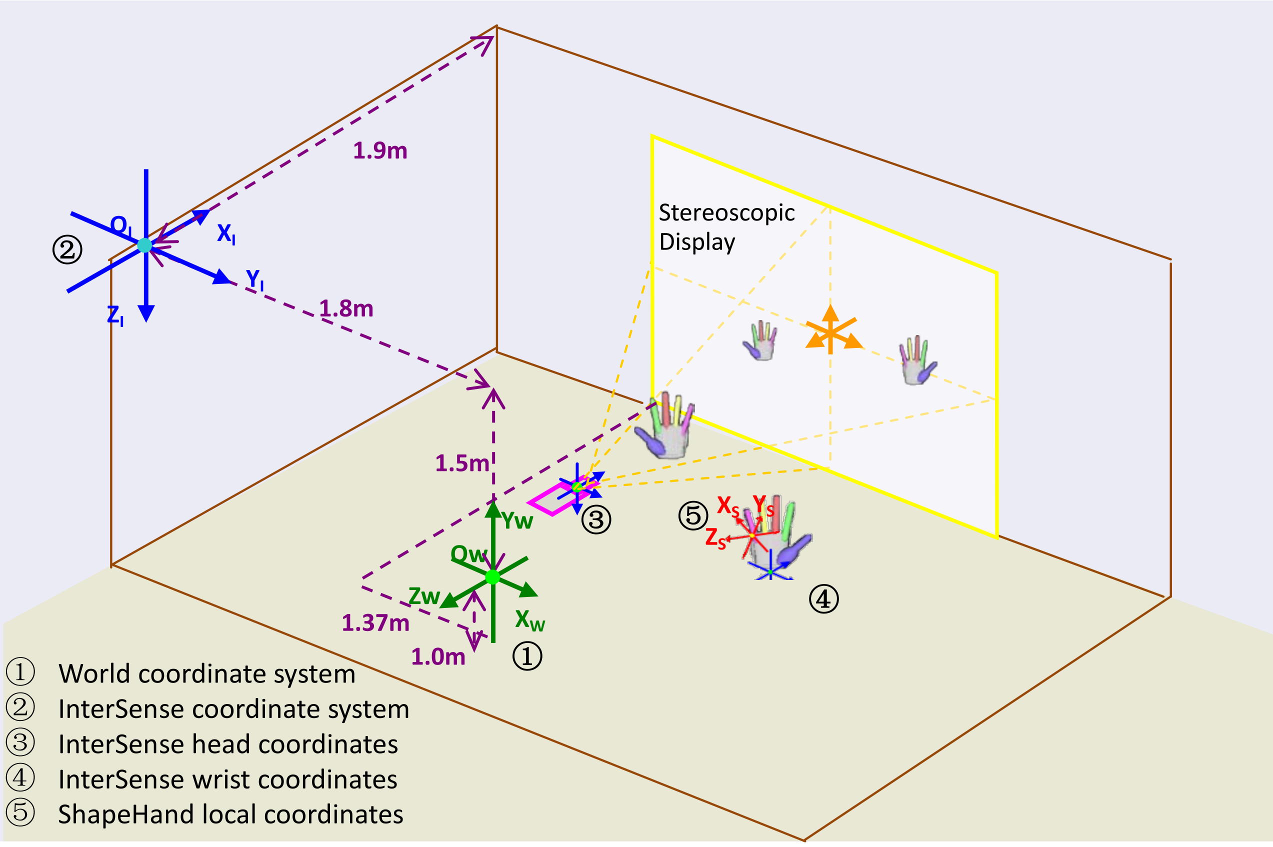

1.9m
②
O_I
X_I
Y_I
Z_I
1.8m
Stereoscopic
Display
1.5m
Yw
③
Qw
Zw
Xw
1.37m
1.0m
①
⑤
X_S
Y_S
Z_S
④
① World coordinate system
② InterSense coordinate system
③ InterSense head coordinates
④ InterSense wrist coordinates
⑤ ShapeHand local coordinates

Fig. 2. Coordinate systems

Based on the geometric relationship between the two coordinate systems described above, $\boldsymbol{T}^{I\to W}$ is given by (2)

$$\boldsymbol{T}^{I\to W} = \begin{bmatrix} 0 & 1 & 0 & -1.8 \\ 0 & 0 & -1 & 1.5 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

For the ShapeHand system, the position data of each digit joint are acquired using the local tape coordinate system. When the hand is fully open as shown in Fig. 2, the x-axis (denoted by X_S) runs along the length of the narrow flat tape (along each finger towards the finger tip), the y-axis (denoted by Y_S) is perpendicular to the back of the narrow flat tape (perpendicular to the palm back), and the z-axis (denoted by Z_S) is towards the side of the narrow flat tape (in the direction across the palm). Furthermore, the origin is fixed at the bottom of the palm in the middle of the wrist. With the wrist position and orientation data provided by the InterSense system, the digit joint position data need to be transformed from their local coordinate system to the InterSense coordinate system first and to the world coordinate system subsequently.

If $\boldsymbol{s}$ = [x_s, y_s, z_s, 1] denotes the homogeneous coordinates of a position in the local ShapeHand coordinate system, then its corresponding homogeneous coordinates in the specified world coordinate system denoted by $\boldsymbol{s}_w = [x_{sw}, y_{sw}, z_{sw}, 1]'$ are given by

$$\boldsymbol{s}_w = \boldsymbol{T}^{I\to W}\boldsymbol{T}^{S\to I}\boldsymbol{s} \qquad (3)$$

where $\boldsymbol{T}^{S\to I}$ is the matrix for geometric transformation from the ShapeHand coordinate system to the InterSense coordinate system. If the position and orientation data provided by the InterSense system are denoted by (x_i, y_i, z_i) and (α_ι, β_ι, γ_ι), then $\boldsymbol{T}^{S\to I}$ is given by

$$\boldsymbol{T}^{S\to I} = \begin{bmatrix} \boldsymbol{c}_\alpha \boldsymbol{s}_\beta - \boldsymbol{s}_\alpha \boldsymbol{c}_\beta \boldsymbol{s}_\gamma & \boldsymbol{c}_\beta \boldsymbol{c}_i & \boldsymbol{c}_\alpha \boldsymbol{c}_\beta \boldsymbol{s}_\gamma + \boldsymbol{s}_\alpha \boldsymbol{s}_\beta & \boldsymbol{x}_i \\ \boldsymbol{s}_\alpha \boldsymbol{c}_\gamma & -\boldsymbol{s}_\gamma & \boldsymbol{c}_\alpha \boldsymbol{c}_\gamma & \boldsymbol{y}_i \\ -\boldsymbol{s}_\alpha \boldsymbol{s}_\beta \boldsymbol{s}_\gamma - \boldsymbol{c}_\alpha \boldsymbol{c}_\beta & \boldsymbol{s}_\alpha \boldsymbol{c}_\gamma & \boldsymbol{c}_\alpha \boldsymbol{s}_\beta \boldsymbol{s}_\gamma - \boldsymbol{s}_\alpha \boldsymbol{c}_\gamma & \boldsymbol{z}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4)$$

where *c* and *s* denoting *cos* and *sin* functions with subscripts denoting the orientation angles from the InterSense system.

3.2 Gesture recognition

A basic sequence in immersive virtual object manipulation can be considered as consisting of object selection at the start, followed by object manipulation, which can be a combination of translation, rotation and scaling, and object release at the end. The implementation requires selection of a set of meaningful hand gestures, as well as computation of the distances between hands and objects. For natural interaction and user's comfort, Fig. 3 shows three hand gestures selected for implementation of the five basic object manipulation operations, where the index finger pointing gesture shown in Fig. 3a is used not only for selection of a virtual object but also for object translation and rotation based on the position of the left or right index fingertip and the hand orientation; the hand-open gesture shown in Fig. 3b is used for the release of a selected object; and the gesture of two moving hands with the ring and small fingers closed shown in Fig. 3c is for object scaling.In order to execute the object selection and manipulation operation using the selected gestures, the 3D position of the left and right index fingertips need to be determined and tracked. As a hinged joint, there is only one DOF for the distal interphalangeal joint in the index finger.

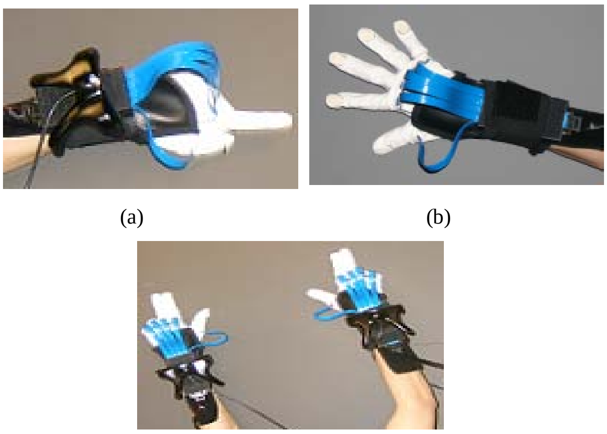


Fig. 3. Hand gestures: (a) object selection, translation and rotation; (b) object release; and (c) object scaling

In order to execute the object selection and manipulation operation using the selected gestures, the 3D position of the left and right index fingertips need to be determined and tracked. As a hinged joint, there is only one DOF for the distal interphalangeal joint in the index finger.

With the distal phalanx length known through the measurement of the user's index finger, the index fingertip position can be determined based on the distal interphalangeal joint position and the distal phalanx bending angle with respect to the middle phalanx provided by the ShapeHand data glove as illustrated in Fig. 4.

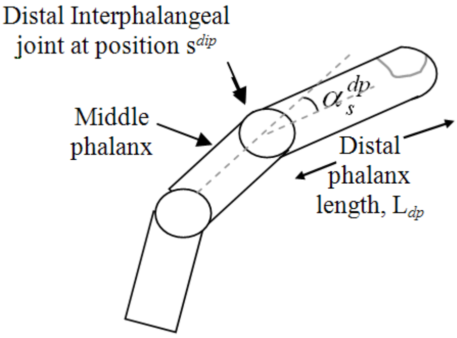

Distal Interphalangeal joint at position s^{dip}
Middle phalanx
α_s^{dp}
Distal phalanx length, L_{dp}

Fig. 4. Index finger model

As shown in Fig. 4, if $\boldsymbol{s}^{dip} = [x_s^{dip}, y_s^{dip}, z_s^{dip}, 1]'$ denotes the homogeneous coordinates of the distal interphalangeal joint and α_s^{dp} denotes the distal phalanx bending angle in the local ShapeHand coordinate system, then the homogeneous coordinates of the index finger tip position in the world coordinate system denoted by $\mathbf{s}_w^{tip} = [\boldsymbol{x}_{sw}^{tip}, \boldsymbol{y}_{sw}^{tip}, \boldsymbol{z}_{sw}^{tip}, 1]'$ are given by

$$\mathbf{s}_w^{tip} = \boldsymbol{T}^{I\rightarrow W}\boldsymbol{T}^{S\rightarrow I}\begin{bmatrix} 1 & 0 & 0 & \cos\alpha_s^{dp}\boldsymbol{L}_{dp} \\ 0 & 1 & 0 & \sin\alpha_s^{dp}\boldsymbol{L}_{dp} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\mathbf{s}^{dip} \quad (5)$$

where L_{dp} denotes the distal phalanx length.

Moreover, the three selected gestures shown in Fig. 3 are seen to consist of a combination of bending down and extending thumb and fingers in each hand, which can be determined based on the flexion angle of the proximal phalanx of the thumb or finger with respect to the back of the hand. With the digit flexion angle calibrated to 0° to correspond to a fully extended position (by hand opening) and 90° to correspond to a fully bending down position (by hand closing), the selected hand gestures can be recognised by expressing them using the corresponding binary state based on a threshold of 45°. Let two hands be denoted by *H* with its binary state set to logic 0 for the left and logic 1 for the right, and let the thumb and four fingers in each hand be denoted by *T*, *I*, *M*, *R*, *S* with the binary state of each one set to logic 0 if its flexion angle, α_s^{pp}, measured by the ShapeHand data glove is less than 45°, and logic 1 otherwise. The index finger pointing gesture is given by

$$(\overline{H}\,\overline{T}I\overline{M}\,\overline{R}\,\overline{S}) \cup (H\overline{T}I\overline{M}\,\overline{R}\,\overline{S}) \quad (6)$$

the hand open gesture is given by

$$(\overline{H}TIMRS) \cup (HTIMRS) \quad (7)$$

and the two hand moving gesture for object scaling is given by

$$(\overline{H}TIM\,\overline{RS}) \cap (HTIM\,\overline{RS}) \quad (8)$$

3.3 *Object distance computation*

Apart from the object release operation that requires only recognition of the corresponding hand gesture, other object manipulation operations require additional information of the object with respect to the user's hands in terms of its location, orientation and size.

Let the virtual object to be manipulated be denoted by $\boldsymbol{o}$ centred at (o_x, o_y, o_z) in the world coordinate system, with orientation of (o_α, o_β, o_γ), and with its bounding box defined by lengths of (L_x, L_y, L_z). Object selection requires not only recognition of the index finger pointing gesture by using (6), but also the position of the index finger tip with respect to the object bounding box in 3D space. A virtual object is selected, if the index finger tip (left or right) touches anywhere in one of the side faces of the bounding box. Hence, upon recognition of the index finger pointing gesture, two more conditions need to be satisfied for a virtual object to be selected. One is based on the distances between the index finger tip and the side face centres of the bounding box, and the other is based on the distances between the index finger tip and the side face planes of the bounding box.

For the first condition, if S_i with i = 1, 2, …, 6, denote the six side planes of the virtual object bounding box, and $\boldsymbol{P}_{i,1}$ = [$x_{Pi,1}$, $y_{Pi,1}$, $z_{\ Pi,1}$] the coordinates of each side plane centre, then the distance between the pointing index finger tip and each side plane centre is given by

$$\boldsymbol{dist}(\mathrm{s}_w^{\boldsymbol{tip}}, \boldsymbol{P}_{i,1}) = \sqrt{(\boldsymbol{x}_{\boldsymbol{sw}}^{\boldsymbol{tip}} - \boldsymbol{x}_{\boldsymbol{Pi},1})^2 + (\boldsymbol{y}_{\boldsymbol{sw}}^{\boldsymbol{tip}} - \boldsymbol{y}_{\boldsymbol{Pi},1})^2 + (\boldsymbol{z}_{\boldsymbol{sw}}^{\boldsymbol{tip}} - \boldsymbol{z}_{\boldsymbol{Pi},1})^2} \quad (9)$$

For the second condition, three non-collinear points lying on each side of the bounding box are selected to represent each side plane. If these three points are denoted by $\boldsymbol{P}_{i,j}$ with i =1, 2, …, 6 and j = 1, 2, 3, then the distance between the pointing index finger tip and each side plane of the bounding box is given by

$$\boldsymbol{dist}(\boldsymbol{s}_{\boldsymbol{sw}}^{\boldsymbol{tip}}, \boldsymbol{S_i}) = \frac{\left|(\boldsymbol{a_i}\boldsymbol{x}_{\boldsymbol{sw}}^{\boldsymbol{tip}} + \boldsymbol{b_i}\boldsymbol{y}_{\boldsymbol{sw}}^{\boldsymbol{tip}} + \boldsymbol{c_i}\boldsymbol{z}_{\boldsymbol{sw}}^{\boldsymbol{tip}} + \boldsymbol{d_i})\right|}{\sqrt{\boldsymbol{a_i}^2 + \boldsymbol{b_i}^2 + \boldsymbol{c_i}^2}} \quad (10)$$

where a_i, b_i, c_i, and d_i are the coefficients of the plane equation for S_i, and are obtained by solving the following equation

$$\begin{bmatrix} \boldsymbol{x}_{\boldsymbol{Pi},1} & \boldsymbol{y}_{\boldsymbol{Pi},1} & \boldsymbol{z}_{\boldsymbol{Pi},1} & 1 \\ \boldsymbol{x}_{\boldsymbol{Pi},2} & \boldsymbol{y}_{\boldsymbol{Pi},2} & \boldsymbol{z}_{\boldsymbol{Pi},2} & 1 \\ \boldsymbol{x}_{\boldsymbol{Pi},3} & \boldsymbol{y}_{\boldsymbol{Pi},3} & \boldsymbol{z}_{\boldsymbol{Pi},3} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{a_i} \\ \boldsymbol{b_i} \\ \boldsymbol{c_i} \\ \boldsymbol{d_i} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (11)$$

In the implementation, the three non-collinear points selected for each side of the bounding box include the corresponding side plane centre, and a virtual object is selected when the following condition is true

$$\begin{aligned} &\left\{\left(\overline{\boldsymbol{H}}\,\overline{\boldsymbol{T}}\boldsymbol{I}\overline{\boldsymbol{M}}\,\overline{\boldsymbol{R}}\,\overline{\boldsymbol{S}}\right) \cup \left(\boldsymbol{H}\overline{\boldsymbol{T}}\boldsymbol{I}\overline{\boldsymbol{M}}\,\overline{\boldsymbol{R}}\,\overline{\boldsymbol{S}}\right)\right\} \\ &\cap \left\{\boldsymbol{dist}\left(\boldsymbol{s}_{\boldsymbol{sw}}^{\boldsymbol{tip}}, \boldsymbol{P}_{\boldsymbol{i},1}\right) \le \min(\boldsymbol{L_x}, \boldsymbol{L_y}, \boldsymbol{L_z})/2\right\} \cap \left\{\boldsymbol{dist}\left(\boldsymbol{s}_{\boldsymbol{sw}}^{\boldsymbol{tip}}, \boldsymbol{S_i}\right) = 0\right\} \end{aligned} \quad (12)$$

In (12), the first two terms come from (6) and are used to confirm the index finger pointing gesture that can be made by the left and/or right hand, the third and fourth terms indicate the index finger tip touching a side face of the bounding box. When (12) is satisfied, the bounding box of the virtual object is highlighted to provide a visual feedback to the user.

For object translation and rotation following object selection, it was implemented by making the object centre follow the current 3D position of the index fingertip computed using (5) and the object 3D orientation to follow the current wrist orientation provided by InterSense. If both hands are making the index finger pointing gestures, the position of the selected object will follow the index finger with minimum distance to the object centre. Since a user may use one pointing index finger to do object selection, translation and rotation with the other hand open, the object release operation is disabled if (12) is satisfied.

Object scaling requires not only recognition of the two hand gesture using (8) but also the positions of the left and right index fingertips with respect to the object bounding box in 3D space. Hence, the function to activate the scaling operation can be expressed by (13)

$$\begin{aligned} &\left\{\left(\overline{H}TIM\,\overline{RS}\right) \cap \left(HTIM\,\overline{RS}\right)\right\} \\ &\cap \left\{dist\left(s_{sw,l}^{tip}, P_{i,1}\right) \le \min(L_x, L_y, L_z)/2\right\} \cap \left\{dist\left(s_{sw,l}^{tip}, S_i\right) = 0\right\} \\ &\cap \left\{dist\left(s_{sw,r}^{tip}, P_{i,1}\right) \le \min(L_x, L_y, L_z)/2\right\} \cap \left\{dist\left(s_{sw,r}^{tip}, S_i\right) = 0\right\} \end{aligned} \quad (13)$$

where the first two terms come from (8) and are used to recognise the object scaling gesture, the middle two terms indicate the left index finger tip touching a side of the bounding box, and the last two terms indicated the right index finger tip touching a side of the bounding box. When (13) is satisfied, the scaling operation is activated, and the virtual object size is enlarged or reduced uniformly in 3D by setting the length of the object bounding box equal to the distance between two index fingertips.

$$L = \max[abs(x^{tip}_{w,l} - x^{tip}_{w,r}), abs(y^{tip}_{w,l} - y^{tip}_{w,r}), abs(z^{tip}_{w,l} - z^{tip}_{w,r})] \quad (14)$$

4 SYSTEM PERFORMANCE

Presented in this section are the evaluations performed on the developed system, which include manipulation of two virtual objects that created to demonstrate the system usability, as well as speed and latency assessment involving the use of a high speed camera.

4.1 Immersive virtual object manipulation

To demonstrate the usability and evaluate the performance of the developed system, a scene with two virtual objects was created for immersive manipulation by the user wearing a pair of wireless ShapeHand data gloves, a pair of wrist tracking devices, a head tracking device, and a pair of polarised glasses. One object is a simple six-colour cube with an initial size of 80x80x80 m^3, and the other is a medical CT volume with 256x256x256 voxels.

From the visual perspective, the user is able to see the stereoscopic images of the cube and CT volume as well as his/her hands displayed through two projectors placed at the back of a large screen operating in passive circular polarisation mode. With the head tracking device providing the position and orientation of the user's head, the user can physically moves around in front of the display screen with an impression of a 3D virtual cube and a 3D CT volume floating in space, whereby a forward movement causes each object to appear nearer and larger, a backward movement causes each object to appear further and smaller, and a side movement with a side look via head rotation causes a different side of each object to appear.

From the interaction perspective, the use is able to see his/her hands in 3D with respect to the virtual objects, as well as the gestures made. Furthermore, when the pointing index finger tip of the user reaches a side plane of the virtual object, the object bounding box is highlighted to provide a visual feedback of object selected.

The simplicity and intuitiveness of the hand gestures were seen to enable a new user to quickly handle and manipulate individual object or both objects simultaneously, namely, pointing the index finger(s) to touch (select), drag, rotate the 3D cube or the CT volume, or both in 3D space as shown in Fig. 5, passing the selected object from one hand to another hand (from one pointing index finger to another pointing index finger), sliding two hands (with the ring and small finger closed) with respect to each other to enlarge and reduce the size of the selected as shown in Fig. 6, and opening the hand(s) to detach from the selected object(s).

Furthermore, the system is able to perform the required operations with certain deviation in the gestures made such as hand digits not fully extended and closed, and highly robust recognition can be achieved by performing calibration of hand close and open gestures at the start.

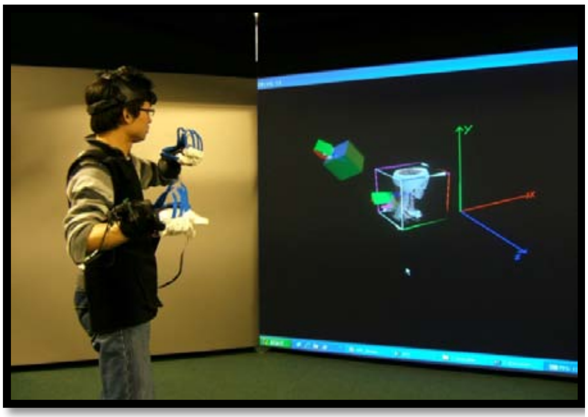


Fig. 5. User performing simultaneous translation and rotation of two objects

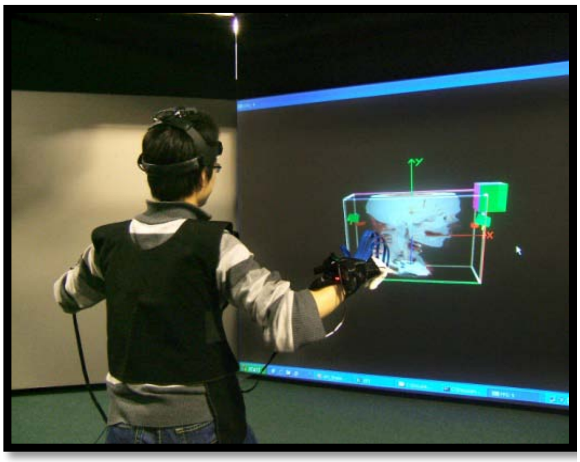


Fig. 6. User performing scaling of CT volume

As an example, Fig. 7 shows some of the data acquired from performing a short sequence of selection, translation and release of one of the virtual object in the scene, namely, (o_x, o_y, o_z) to show the 3D position variation of the virtual object centre using red, green and blue dotted lines, ($x_{sw,r}^{tip}$, $y_{sw,r}^{tip}$, $z_{sw,r}^{tip}$) to show the 3D position variation of the right hand index finger tip using red, green and blue solid lines, and α_s^{pp} to show the flexion angle of the proximal phalanx of the right hand middle finger in a black solid line.

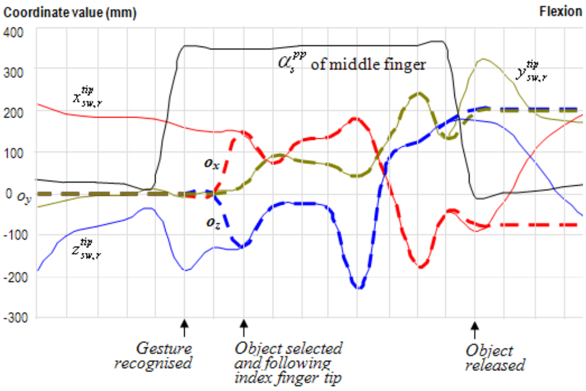


Fig.7. A sequence of dynamic gesture data

With an open hand gesture at the start of the sequence, it is seen from Fig. 7 that α_s^{pp} is around 0º, the user’s right hand moves in the horizontal plane as indicated by the changing coordinate values of $x_{sw,r}^{tip}$ and $z_{sw,r}^{tip}$ with $y_{sw,r}^{tip}$ roughly constant, and the virtual object is stationary at the origin of the world coordinate system as indicated by (o_x, o_y, o_z) = (0, 0, 0). Soon after the hand gesture changed into an index finger pointing gesture as indicated by the sharp rise of α_s^{pp} from 0º to around 70º due to the middle finger closed, the index finger tip is seen to approach the virtual object with the coordinate values of ($x_{sw,r}^{tip}$, $y_{sw,r}^{tip}$, $z_{sw,r}^{tip}$) moving towards (o_x, o_y, o_z). When the distance is sufficiently small, the object is seen to be selected. Once selected, the virtual object is seen to follow the index finger tip with the coordinate values of o_x following $x_{sw,r}^{tip}$, o_y following $y_{sw,r}^{tip}$, and o_z following $z_{sw,r}^{tip}$. Finally, the user’s hand opens to release the object as indicated by α_s^{pp} falling back to around 0º due to the opening of the middle finger, the virtual object is seen to stay at its final position with (o_x, o_y, o_z) fixed as the user’s hand moves away.

4.2 System performance evaluation and analysis

A number of tests were also conducted to assess the key measures of system real-time performance in terms of speed and latency.

For manipulation of a graphics-based object, speed performance evaluation was based on the virtual cube, and for manipulation of a data-based object, it was based on the CT volume. The evaluation was implemented by inserting a counter at the start of each program thread to record the number of times to run the thread per second.

For manipulation of the virtual cube, Figure 8 shows a typical example showing the frequency of executing each program thread over a period of one minute with continuous hand movement in 3D space. From Fig. 8, the program threads of hand gesture recognition and position data acquisition based on InterSense are seen to be relatively fast with relatively large fluctuations. Whilst the former is seen to be the fastest one with an average execution frequency of 136 times per second and the largest variation between the maximum of 171 times per second and the minimum of 102 times per second, the latter is the second fastest with an average execution frequency of 118 times per second and a variation between the maximum of 159 times per second and the minimum of 99 times per second. Very similar behaviour of the execution frequencies for the ShapeHand gesture data acquisition program thread and the stereoscopic display program thread are also seen from Fig. 8, with the former slightly faster at 62 times per second on average between the maximum of 65 times per second and the minimum of 59 times per second, and the latter at 60 times per second between the maximum of 64 times per second and the minimum at 54 times per second.

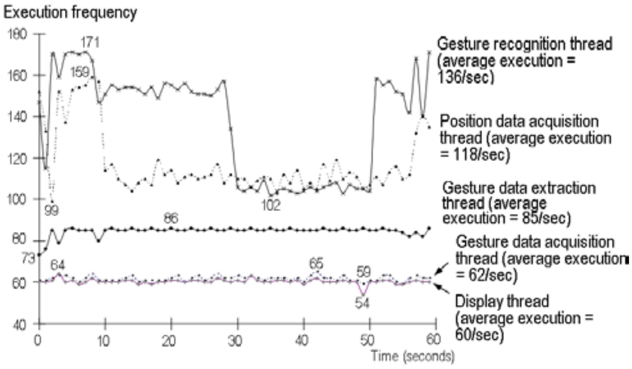


Figure 8. Thread execution frequency versus time

For manipulation of the CT volume, the stereoscopic display program thread was also found to be the slowest with the worst case execution frequency of 31 times per second. Hence, the speed of the system depends on the complexity and the number of the objects to be displayed.

To confirm the adequacy of the system speed performance for dynamic gesture recognition, a test was also carried out to check the tracking speed against the maximum speed of finger movement. By wearing the ShapeHand data glove with the index finger opening and closing repeatedly at the highest possible speed, the virtual hands on the stereoscopic display was found to follow the angular movement of the index finger at a maximum speed around 14 times per second.

For performance evaluation of latency, a high speed video camera was used to record the hand movement made by a user wearing the ShapeHand data glove as well as the movement of the virtual hands appeared on the stereoscopic screen. With the hand opening and closing repeatedly, the video was captured at 64 frames per second, and video analysis of the corresponding hand gestures showed a delay around 6 frames of the virtual hand movement with respect to the real hand movement, which is equivalent to a latency of approximately 94ms.

5 CONCLUSIONS

The paper demonstrates an approach to achieve immersive manipulation of virtual objects using natural hand gestures. In particular, the paper describes (a) the integration of a wireless high DOF hand gesture data glove, a wireless position tracking system, and a stereoscopic display; (b) algorithms developed for recognition of dynamic hand gestures and object distances; and (c) system performance evaluation conducted to assess its usability. Overall, the system is shown to provide an immersive and interactive environment, whereby a user can visualise in stereoscopic mode and interact in 3D with virtual objects using natural hand gestures. Furthermore, the simplicity and intuitiveness of the selected hand gestures, together with robustness in recognition of imprecise hand gestures, enables users to quickly master the object manipulation operations with little effort. Particularly, it has been shown that the system can operate at 54 frames per second for graphics-based objects, and 31 frames per second for data-based objects, as well as a latency time of approximately 94 ms using a PC with 3GHz CPU.

Although the virtual objects used for system demonstration are simple, with one based on a cube and the other based on a 3D-CT volume, the work is seen as an important first step towards simultaneous manipulations of multiple objects with complex shapes, deformable surface and mutual physical interaction. Further work will extend the system to recognise not only a wide range of more intricate hand gestures but also complex hand movement trajectories in 3D space, thereby enabling the use of natural hand movements and gestures to perform complex virtual object manipulation, such as assembling and disassembling tasks in equipment maintenance and repair (Corvaglia 2004; Badler et al. 1993; Johnson and Rickel 1997), and complex virtual tool operation, such as control of robots in medical surgery (Liverneaux et al. 2009; Scharver et al. 2004).

ACKNOWLEDGMENT

The authors thank Dr. X. Chen for his technical support.

References

Adamo-Villani N., Heisler J, Arns L (2007) Two gesture recognition systems for immersive math education for the deaf. In: ACM proceedings of IMMERSCOM 2007, Verona, Italy, pp 10–12

Badler NI, Philips CB, Webber BL (1993) Simulating humans, computer graphics animation and control. Oxford University Press, New York

Bowman D, Wingrave C, Campbell J, Ly V (2001) Using pinch gloves for both natural and abstract interaction techniques in virtual environments. In: Proceedings of HCI international, New Orleans, USA, pp 629–633

Corradini A, Cohen P (2002) On the relationships among speech, gestures, and object manipulation in virtual environments: initial Evidence. In: Proceedings of the international CLASS workshop on natural, intelligent and effective interaction in multimodal dialogue systems, Copenhagen, Denmark

Corvaglia D (2004) Virtual training for manufacturing and maintenance based on Web3d technologies. In: Proceeding of LETWeb3D 2004: 1st international workshop on Web3D technologies in learning, education and training, pp 28–33

Danisch L, Englehart K, Trivett A (1999) Spatially continuous six degree of freedom position, orientation sensor. Sens Rev 19(2):106–112

Demirdjian D, Ko T, Darrell T (2005) Untethered gesture acquisition and recognition for virtual world manipulation. Virutal Real 8:222–230

Dipietro L, Sabatini AM (2008) A survey of glove-based systems and their applications. IEEE Trans Syst Man Cybern Part C: Appl Rev 38:461–482

Erol A, Bebis G, Nicolescu M, Boyle RD, Twombly X (2007) Vision-based hand pose estimation: a review. Comput Vis Image Underst 108:52–73

Garg P, Aggarwal N, Sofat S (2009) Vision based hand gesture recognition. In: Proceedings of world academy of science, engineering and technology, pp 972–977

IS-900 Precision Inertial-ultrasonic Motion Tracking System, InterSense Inc. http://www.isense.com

Johnson WL, Rickel J (1997) Steve: an animated pedagogical agent for procedural training in virtual environments. Sigart Bulletin, ACM Press 8(1–4): 16–21

Kober C, Boerner BI, Mori S, Tellez CB, Klarho¨fer M, Scheffler K, Sader R, Zeilhofer HF (2007) Stereoscopic 4D-visualization of craniofacial soft tissue based on dynamic MRI and 256 row 4DCT. Adv Med Eng Springer Proc Phys Part II, pp 175–180

Le´cuyer A, Lotte F, Reilly R, Leeb R, Hirose M, Slater M (2008) Brain-computer interfaces, virtual reality, and videogames. Computer 41:66–72

Liverneaux P, Nectoux E, Taleb C (2009) The future of robotics in hand surgery. Chirurgie de la Main 28:278–285

O'Hagan RG, Zelinsky A, Rougeaux S (2002) Visual gesture interfaces for virtual environments, interacting with computers, vol 14, pp 231–250, April, 2002

Patel D, Muren LP, Mehus A, Kvinnsland Y, Ulvang DM, Villanger KP (2007) A virtual reality solution for evaluation of radiotherapy plans. Radiother Oncol 82(2):218–221

Scharver C, Evenhouse R, Johnson A, Leigh J (2004) Designing cranial implants in a haptic augmented reality environment. Commun ACM 47:32–38

ShapeHand Data Glove, Measurand Inc. http://www.measurand.com

Sturman DJ, Zeltzer D (1994) A survey of glove-based input. IEEE Comput Graph Appl 14(1):30–39

Toyama H (2006) Trials on grasping of a 3D virtual object in CAVE using EEG signals. IEIC Tech Rep 91:53–56

Wang XH, Good WF, Fuhrman CR, Sumkin JH, Britton CA, Golla SK (2005) Stereo CT image compositing methods for lung nodule detection, characterization. Acad Radiol 12(12):1512–1520

Woods AJ (2005) Compatibility of display products with stereoscopic display methods. In: Proceedings of the international display manufacturing conference, Taipei, Taiwan

Wormell D, Foxlin E (2003) Advancements in 3D interactive devices for virtual environments. In: Proceedings of the workshop on virtual environments, Zurich, Switzerland, pp 47–56

Zhang S, Demiralp C, Keefe D, DaSilva M, Laidlaw DH, Greenberg, BD, Basser PJ, Pierpaoli C, Chiocca EA, Deisboeck TS (2001) An immersive virtual environment for DT-MRI volume visualization applications: a case study. In: Proceedings of IEEE visualization conference, San Diego, USA, pp 437–440

APPENDIX B

FINGERSPELLING RECOGNITION

In order to illustrate the system's versatility, recognition of the BSL fingerspelling was investigated by developing an algorithm to enable the system to automatically display the text on screen according to the BSL fingerspelling gestures made by a hearing impaired signer, thereby enabling audience to understand immediately the meaning of the hand signs through reading of the displayed text.

The BSL is a visual sign language used in the UK, and is the first or preferred language of the deaf people in the UK. BSL uses word level signs (gestures), non-manual features, e.g. facial expression and body posture, as well as fingerspelling (letter-by-letter signing) to convey meaning [257]. Fingerspelling (see Fig. B.1) is an important part of BSL and is mainly used to spell words for which no sign to express, e.g., names or technical terms, acronyms, unknown words to signers, or to clarify a sign that is unfamiliar to the 'observer' who is reading the signer. Moreover, a conversation with a BSL signer could be simply carried out by using the 26 BSL fingerspelling alphabets, albeit it is time consuming [258-261]. Compared to the fingerspelling methods in some other Sign Languages, such as the ASL, BSL fingerspelling has an inherent difficulty for recognition, since it involves both hands for all letter signs (except 'c') instead of using one hand like ASL fingerspelling [257]. On the other hand, most BSL fingerspelling signs are based on the shape of the letter, which are 'drawn' by the fingers of the right hand on the 'page' of the left for the right-handed user. The exceptions are the five vowels, i.e., *A*, *E*, *I*, *O*, and *U*. They are indicated in a systematic way, rather than graphic signs, by using the right index finger to touch the tips of the left thumb and fingers [262, 263]. As the first step to achieve the BSL fingerspelling recognition by using the proposed system, the work presented in this section is a small scale system design for the BSL fingerspelling gesture recognition based on these five vowels letters.

As shown in Fig. B.1, the BSL fingerspelling gestures of the five vowels consist of a combination of the left hand opening and right hand index finger pointing postures, as well as the right hand index finger tip touching each tip of the thumb/fingers on the left hand. The left hand opening hand gesture can be identified by using equation (4.2) in

Section 4.2, and the right hand index finger pointing gesture by using equation (4.1). Moreover, the touch between the right hand index finger tip and other tips of the thumb/fingers on the left hand is measured by computing the distance between them using equation (4.4). Therefore, for instance, the recognition condition of letter ‘A’ spelled out using BSL is given by

$$\begin{gathered}(\overline{H}TIMRS)\cap(H\overline{T}I\overline{M}\,\overline{R}\overline{S})\\ \cap\left\{dist\left(s_{sw,r}^{tip,i},s_{sw,l}^{tip,t}\right)\leq\min\left[dist\left(s_{sw,r}^{tip,i},s_{sw,l}^{tip,m}\right)\right]\right\}\\ \cap\left\{dist\left(s_{sw,r}^{tip,i},s_{sw,l}^{tip,t}\right)\leq T_1\right\}\end{gathered} \tag{B.1}$$

where $m = T, I, M, R, S$, denotes the five fingers of a hand, respectively.

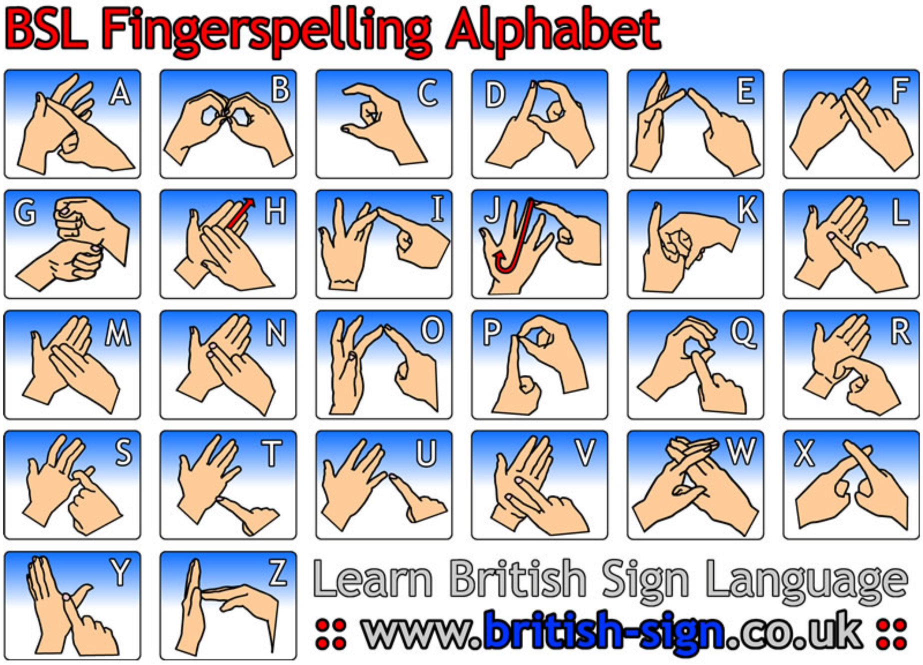

BSL Fingerspelling Alphabet
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
Learn British Sign Language
:: www.british-sign.co.uk ::

Figure B.1. BSL Fingerspelling Alphabe (from [258]).

In equation (B.1), the first two terms come from equations (4.2) and (4.1) respectively, indicating the left hand open and the right hand making the index finger pointing gesture; the middle two terms indicate the distance between the right hand index finger tip and left hand thumb tip being the minimum compared to the distances between the right hand index finger tip to the other four left hand finger tips; and the last term indicates the distance between the right hand index finger tip and left hand thumb tip is less than the specified threshold. In the implementation, the threshold for the touching distance between two finger tips of two hands is set to be 20mm based on empirical data, since the finger width is approximate 15mm and the distance between two fingertips is approximate 50mm with the hand open. When the distance between the index finger tip on the right hand and a finger tip on the left hand is detected to be the minimum compared to other fingertips, and is less than 15mm, a vowel letter will be displayed on the screen immediately to correspond the fingerspelling gesture.

Tests have been done to evaluate the system’s performance in terms of BSL fingerspelling recognition. As illustrated from Figs. B(2-6), the system is shown to be able to recognise the five vowels fingerspelled by the user by outputting the corresponding vowel letters on the screen. Furthermore, to provide a clear visual feedback, the fingers colour shall change according to different fingerspelling gesture inputs, with red for ‘*A*’, blue for ‘*E*’, grey for ‘*I*’, yellow for ‘*O*’, and pink for ‘*U*’.

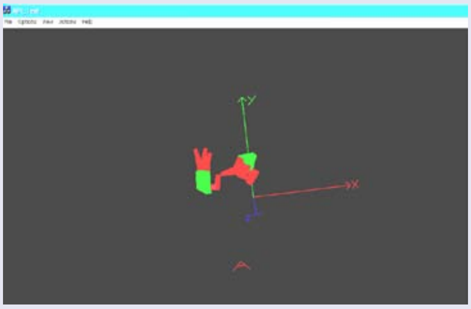


Figure B.2. Fingerspelling of ‘*A*’.

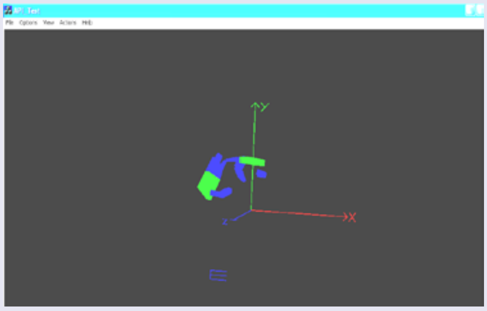


Figure B.3. Fingerspelling of ‘E’.

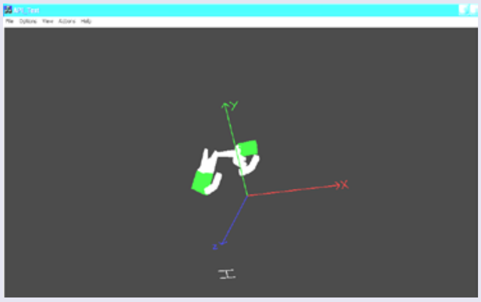


Figure B.4. Fingerspelling of ‘*I*’.

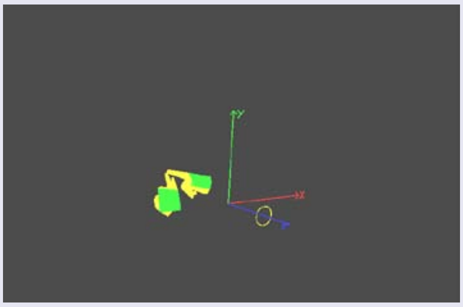


Figure B.5. Fingerspelling of ‘*O*’.

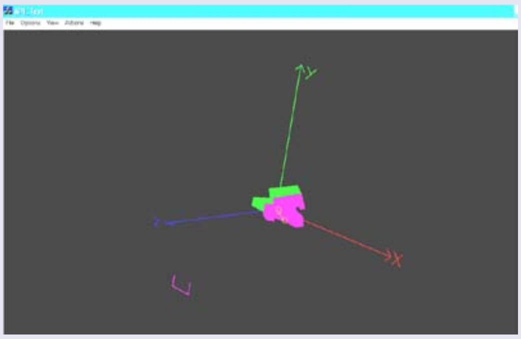


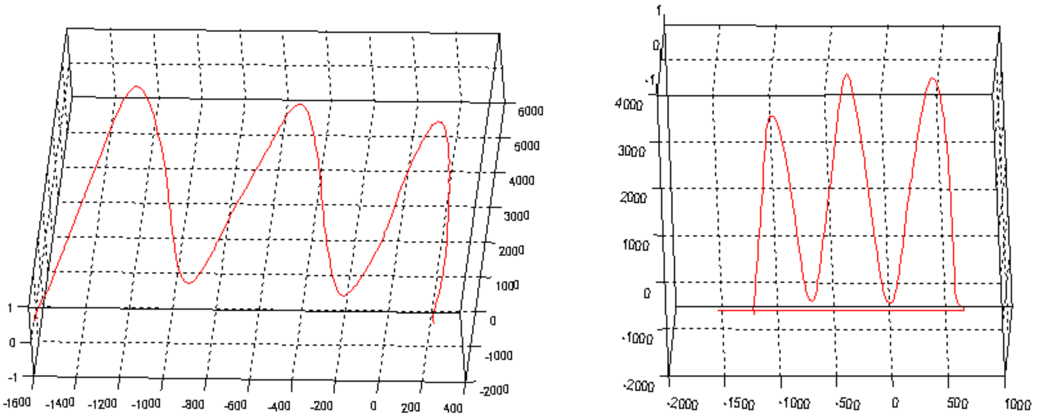
Figure B.6. Fingerspelling of ‘*U*’.

APPENDIX C

RECOGNITION OF ‘CHILDREN’ GESTURE

Fig. C.1 shows the movement trajectories of the left and right hands making the ‘children’ gesture, where two hands are seen to move outwards in the opposite directions in the floor plane. Furthermore, the two hand movement trajectories are similar, with each one containing three local peaks and two local valleys. Therefore, the recognition of the ‘children’ movement gesture is implemented based on the movement directions as well as the number of local peaks and valleys along the movement curve. As shown in Fig. C.2, using the right hand movement trajectory as an example, after the hand movement being recognised as a right horizontal direction movement by using the method presented in Section 5.3.4 and a curved movement by using the method presented in Section 5.3.5, a further check is made to see if this movement contains multiple peaks and valleys. Let the starting hand position and a hand position along the hand motion trajectory be denoted by $\boldsymbol{p_i} = [x_i, y_i, z_i]$ and $\boldsymbol{p_n} = [x_n, y_n, z_n]$, respectively. The height of a hand position is defined as the relative difference between $\boldsymbol{p_i}$ and $\boldsymbol{p_n}$ in the vertical plane (parallel to the y-axis). For example, h_n, the height of $\boldsymbol{p_n}$ with respect to $\boldsymbol{p_i}$ is computed by:

$$h_n = |y_n - y_i| \tag{C.1}$$



(a) (b)

Figure C.1. Recorded ‘Children’ gesture hand movement trajectories of (a) left hand and (b) right hand.

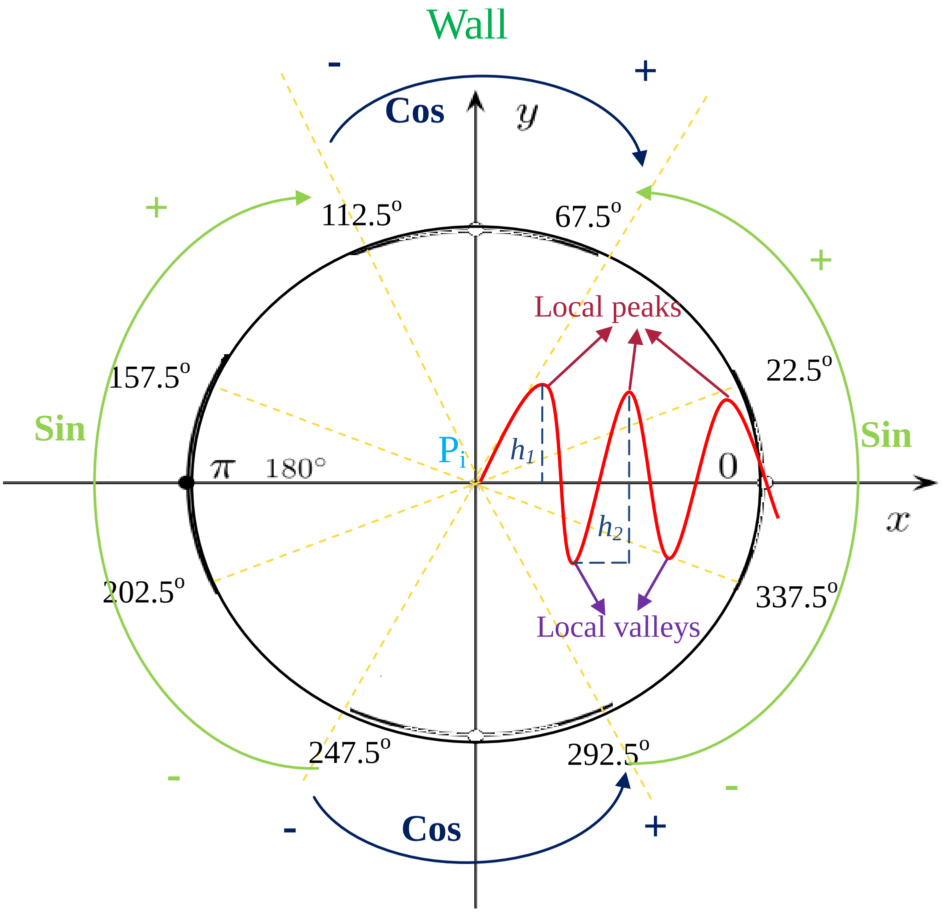


Figure C.2. ‘Children’ right hand gesture movement trajectory.

Thus, the first local peak point is defined as the first point which has the maximum height, h_1, with respect to the starting position compared to its four neighbouring points. Similarly, the first local valley point is defined as the point which has the lowest height, h_2, with respect to the first obtained local peak position compared with its four neighbouring points. This process is then continued to find other local peak and valley points, until it reaches its ending position. Base d on the number of local peaks and valleys, the curve can be identified as a right hand ‘children’ movement if the number is greater than or equal to five; and a normal curved movement otherwise.