APPLICATION OF AUGMENTED REALITY IN MANUAL ASSEMBLY DESIGN AND PLANNING

WANG ZHENBIAO

(B.Eng)

**A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE**

2012

Declaration

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Wang Zhenbiao

7 May 2013

Acknowledgements

This research is a result of the support and assistance from many different people, whom I wish to acknowledge. First and foremost, I would like to express my sincere gratitude to my supervisors, Associate Professor Ong Soh Khim and Professor Andrew Nee Yeh Ching for their constant encouragement. Without their invaluable guidance, assistance and suggestions, this research would not have been possible. I am deeply grateful to Prof. Ong for her continuous support of my Ph.D. study and research, for her patience, enthusiasm and immense knowledge. I benefited much from her constructive critiques and comments. I appreciate all her contributions of time, ideas, and efforts to make my Ph.D. experience productive and stimulating. I am equally grateful to Prof. Nee. He has supported me throughout my Ph.D. study with his great patience and erudite knowledge. He has been my advisor, mentor and guardian to over last four years. With his professional advice, research life became smooth and rewarding for me.

I am grateful to NUS for providing this opportunity and rewarding me with a research scholarship for my Ph.D. research and study.

I thank all my friends in the ARAT Lab for their kindly support and valuable assistance. Special thanks go to Dr. Yuan Miaolong, Dr. Louis Fong, Dr. Zhang

Jie, Dr. Shen Yan, Dr. Chen Chengjun, Dr. Li Zhenghao, Dr. Fang Hongchao, Jiang Shuai, Ng Lai Xing, Zhu Jiang, Andrew Yew, Du Yue, Yu Lu and Wang Xin.

Lastly, I would like to thank my family for all their love and encouragement. I am proud of my parents. Without their love, trust and support in all my pursuits, I cannot reach this point in my life. Most important of all, I would like to thank my loving, supportive, encouraging and patient wife Meiling for her faithful support. I have a better life in Singapore with her accompanying me.

Table of Contents

Summary

Assembly is the capstone process of connecting components to make a complete product in manufacturing. It brings together the processes of design, engineering and manufacturing to create a functional product. Assembly design and planning during the initial product design stage are highly significant to industrial product development. Augmented reality (AR) is a technology in which intuitive interaction experience is provided to the users by combining the real environment with various computer-generated texts, images, 3D CAD models, etc. AR technologies have opened up a powerful set of tools for computer-aided assembly design and planning.

In this research, an AR interactive manual assembly design and planning (ARIMADP) system has been developed to support manual assembly design with a 3D natural bare-hand interaction (3DNBHI) interface. The ARIMADP system can provide the users with a high level of awareness of the surrounding real environment.

The use of human hands as an attractive alternative to traditional devices has been an important part of current human-computer interaction (HCI). The 3DNBHI interface developed in this thesis uses bare hands as a robust HCI approach for

AR applications. The hand region segmentation has been achieved by using an efficient colour segmentation algorithm. The segmentation from input video stream is adaptable to different lighting conditions and complex background. The fingertip features can be detected and tracked in real time. Hands and fingertips differentiation algorithms have been formulated and implemented for developing a dual-hand interaction interface. Using stereo vision technologies, 3D information of the fingertips can be retrieved. The thumb tip and index fingertip have been used for the pinch operations of the virtual objects in this research. The experimental results showed that the virtual objects can be manipulated and orientated simultaneously by the user. The dual-hand interface is robust and provides direct and intuitive interactions between the user and the virtual objects in an AR environment.

The ARIMADP system is a constraint-based assembly simulation, design, and planning system that can be used by the users to simulate manual assembly process without the need for any auxiliary CAD information. The ARIMADP system can enhance the users' experience and reduce preparation time. To support the simulation and design of manual assemblies in an AR-based environment, a Tri-layer Assembly Data Structure (TADS) forming the basis of the ARIMADP system has been used to help users manage the assembly data. Several algorithms including a robust feature-based automatic constraint recognition algorithm and constraint-based assembly constraints recognition and refinement algorithms have been formulated for the ARIMADP system. Assembly tools and tool-assisted operations have been considered to make the assemblies in an AR environment more realistic. By integrating the 3DNBHI interface, ARIMADP can achieve an intuitive cognition on the mating relations between the components and the functions of the assembly.

List of Tables

List of Figures

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
APIs	Application Programming Interfaces
AR	Augmented Reality
ATS	Assembly Tree Structure
BCCS	Bumblebee2 Camera Coordinate System
CAD	Computer-Aided Design
CamShift	Continuously Adaptive Mean-shift
CAPP	Computer-Aided Process Planning
CCI	Connection Change Index
CI	Continuity Index
CV	Computer-Vision
DCI	Direction Change Index
DHM	Digital Human Modeling
DOF	Degrees of Freedom
DMU	Digital Mock-up
FCCS	Firefly Camera Coordinate System
FOV	Field of View
HCI	Human-Computer Interaction
HCS	Hand Coordinate System
HMD	Head-Mounted Display
HIS	Hand Strain Index
MI	Manipulability Index
OAI	Overall Assembly Index
OBS	Object Coordinate System

OPI	Operation Preference Index
PEs	Precedence Expressions
PM	Precedence Matrix
PTS	Part-Tree Structure
RCE	Restricted Coulomb Energy
RMS	Root Mean Square
ROI	Region of Interest
STR	Sequences Task Representations
TADS	Tri-layer Assembly Data Structure
TCI	Tool Change Index
VR	Virtual Reality
VP	Virtual Prototyping
WCS	World Coordinate System

List of Symbols

P_{LC}	Centre of the line segment between two fingertips
P_m	Mean position of all the five fingertips
P_{HC}	Centre of the hand
P_{th}	Tip of thumb finger
t_d	Position constraint parameter
t_θ	Orientation constraint parameter
M_{LS}	Location-solving transformation matrix
${}^{B}_{W}T$	Transformation matrix between the WCS and the BCCS
${}^{F}_{W}T$	Transformation matrix between the WCS and the FCCS
${}^{W}_{O}T$	Transformation matrix of the object in the current frame
${}^{W}_{O'}T$	Transformation matrix of the object in the previous frame
${}^{H}_{O}T$	Transformation matrix between the HCS and the OBS
${}^{H'}_{H}T$	Transformation matrix between the HCS in two frames
X_B, Y_B, Z_B	Coordinates of a point in the BCCS
X_F, Y_F, Z_F	Coordinates of a point in the FCCS
X_H, Y_H, Z_H	Coordinates of a point in the HCS
X_W, Y_W, Z_W	Coordinates of a point in the WCS

Chapter 1 Introduction

1.1 Problem Statement and Research Motivation

In the absence of high fidelity simulation tools, traditionally, physical prototyping in the design loop to verify proper functioning and ease of assembly is the main method for assembly design and evaluation. Designers can obtain valuable feedback easily and identify unexpected defects to improve assembly design using physical prototypes. However, physical prototyping is time-consuming and expensive. Furthermore, once built and tested, physical prototypes are either difficult or impossible to modify.

Currently, Virtual Reality (VR) techniques together with the concept of Virtual Prototyping (VP) [Zorriassatine et al, 2003] have been widely used to simulate and evaluate assembly in the early design stage. VR attempts to replace the designer's perception of the surrounding environment with an immersive, interactive and 3D computer-synthesized environment. By adopting VR, a designer can see, touch and operate a future product before its physical implementation with lower cost and effort. The elimination of physical prototyping and on-site verification makes virtual assembly a powerful tool to reduce the time-to-market of a product and adapt changes or introduce new products. However, there is an obvious shortcoming of current virtual environments as a medium for assembly evaluation which is that the “realism” experience is limited while users operate virtual objects due to a lack of suitable sensory feedback. In addition, a great deal of computation resources is needed to simulate a complex assembly process in a pure virtual environment. Therefore, it is often difficult to satisfy the requirement of real-time simulation although computers available currently are very powerful.

An alternative that is growing in popularity in simulating and evaluating assembly in the early design stage is the use of the Augmented Reality (AR) techniques, which mixes real objects (e.g., physical prototypes, tools, robots, etc.) with virtual objects (e.g., virtual prototypes, information, tools, etc.) to create a mixed reality interface. Unlike the VR technologies which replace the surrounding world with a totally artificial one, this hybrid prototyping technique combines physical representations of parts and physical feedback with computer generated information to analyse the behaviours or properties of future products. With an AR environment, engineers can manipulate and evaluate the virtual prototypes of new product designs in the real assembly environment. There is no need to model the complex assembly ambience using computer graphics techniques and engineers are able to obtain a more realistic feeling for making better decisions for the assembly operations while working in the real assembly environment.

Furthermore, using the virtual assembly parts, the evaluation and revision process can be performed in a fast, economic and efficient manner.

In summary, AR has good potential for reducing costs and lead time, and enhancing efficiency in product design and assembly planning. This research is motivated by the objective to take advantage of the AR technologies for manual assembly design in the early design stage.

1.2 Objectives and Research Scope

The purpose of this research is to explore the application of AR in manual assembly simulation and design using natural bare-hand interaction. This research is focused on the realization and evaluation of an AR interactive manual assembly design system, namely the ARIMADP system, which can be used by users to simulate the manual assembly process and achieve a near-optimal assembly sequence. The objectives of this research are listed as follows.

1) Develop an AR-based assembly system to support manual assembly simulation and design in an AR environment.
2) Develop an efficient methodology for constraint-based assembly simulations without any prepared CAD information (such as assembly constraints, assembly hierarchy, final positions of parts, etc.).

3) Develop a robust bare-hand interaction tool to allow the users to interact with virtual assembly components directly and intuitively.

4) Develop a mechanism to support assembly sequence planning after the assembly is available for the users.

This research focuses on supporting manual assembly simulation, design and planning in an AR environment. An immersive and intuitive HCI interface that allows the users to simulate and design assemblies in the early design stage will be built. In the proposed system, the users can view, touch and operate a product before its physical implementation with lower cost and effort. Bare-hand interaction tool will be explored for natural HCI. A robust hand segmentation algorithm will be investigated and employed. The tips of the thumbs and index fingers will be used to achieve 3D pinch operations. A hands and fingertips differentiation algorithm will be explored to achieve a dual-hand interaction interface. The information extracted from the solid CAD model, such as feature parameters, surface type, etc., and the information that can be recorded during an assembly simulation, are deposited in a data structure. A methodology for on-line constraint recognition and confirmation will be developed to support the proposed ARIMADP system. A methodology for assembly sequence planning and evaluation will be developed.

1.3 Overview of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 presents a literature review of the status of the AR technologies and several research domains relevant to this thesis. A review of the AR technologies and some of AR applications are made, followed by a discussion of the benefits of the application of AR to facilitate collaborative design. This is followed by a detailed literature review of assembly design and planning systems.

Chapter 3 describes the overall architecture of the system that has been studied and developed in this research. The tri-layer data structure used in this system for managing the assembly data is described. In addition, the coordinate systems used in this system are presented.

Chapter 4 presents a natural and robust 3D bare-hand interaction tool for AR systems.

By using an efficient colour segmentation algorithm, the hand region segmentation from input video stream is adaptable to different lighting conditions and complex background. The fingertip features are detected and tracked in real time. Correct occlusion between real fingers and virtual objects is realized. A dual-hand interaction interface is also developed. The interface allows the user to manipulate and orient virtual objects simultaneously.

Chapter 5 discusses a systematic methodology for manual assembly design and planning using AR technologies. A hybrid method based on constraint analysis has been implemented to allow the users to simulate a manual assembly process without the need for auxiliary CAD information. Algorithms for assembly constraint recognition and assembly location refinement have been developed. Algorithms for assembly sequence planning and evaluation have also been proposed.

Chapter 6 describes the system configuration of the AR assembly environment. Two case studies are presented to show the manual assembly simulation and design using the methodologies proposed in Chapter 4 and Chapter 5.

Finally, Chapter 7 summarizes this thesis and gives the conclusions, contributions and recommendations for future research.

Chapter 2 Literature Review

2.1 Introduction

This chapter presents the background knowledge of several domains relevant to the research. Previous works related to this research can be classified into two broad areas: AR technologies and assembly design and planning. A historical overview and a summary of the current research in these areas are presented.

2.2 Augmented Reality

AR is a recently developed technology that aims at mixing real objects (e.g., physical prototypes, tools, robots, etc.) with virtual objects (e.g., virtual prototypes, information, tools, etc.) to create an augmented environment to the users. A number of technologies, such as sensor technologies, computer vision and graphics, etc., have been developed and implemented to facilitate the generation of such augmented environment. VR technologies immerse a user completely inside a computer-generated environment so that the user cannot see the surrounding real world. In contrast, AR allows the user to see the real environment with virtual objects superimposed upon the real world. Therefore, the user's perception can be greatly enhanced by AR. The virtual continuum presented by Milgram and Kishino [Milgram and Kishino, 1994] is often used to describe the relationship between AR, VR and the real world (Figure 2-1).

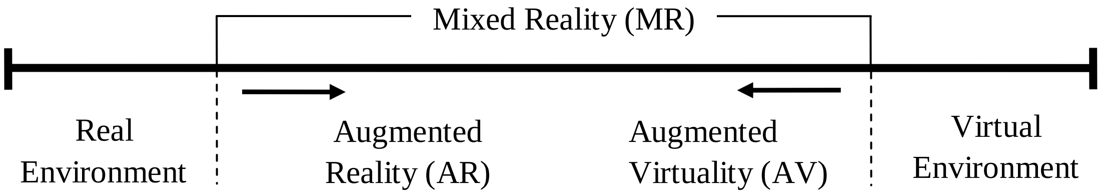


Figure 2-1 Virtual continuum [Milgram and Kishino, 1994]

2.2.1 AR Enabling Technologies

In a survey by Azuma [Azuma, 1997], AR is a variation of VR and AR-based systems have three main characteristics which are combining real and virtual, providing interaction in real-time and registering in 3D. Therefore, the basic enabling technologies for setting up an AR environment include display, tracking and interaction. These enabling technologies are highlighted and discussed in detail in this section.

2.2.1.1 Display

In an AR environment, the user is provided with a synthetic image which is a combination of real and virtual images. Vallino [Vallino, 1998] discussed two types of hardware configurations for the users to observe an augmented world in AR systems, namely, the video see-through and the optical see-through, as shown in Figure 2-2.

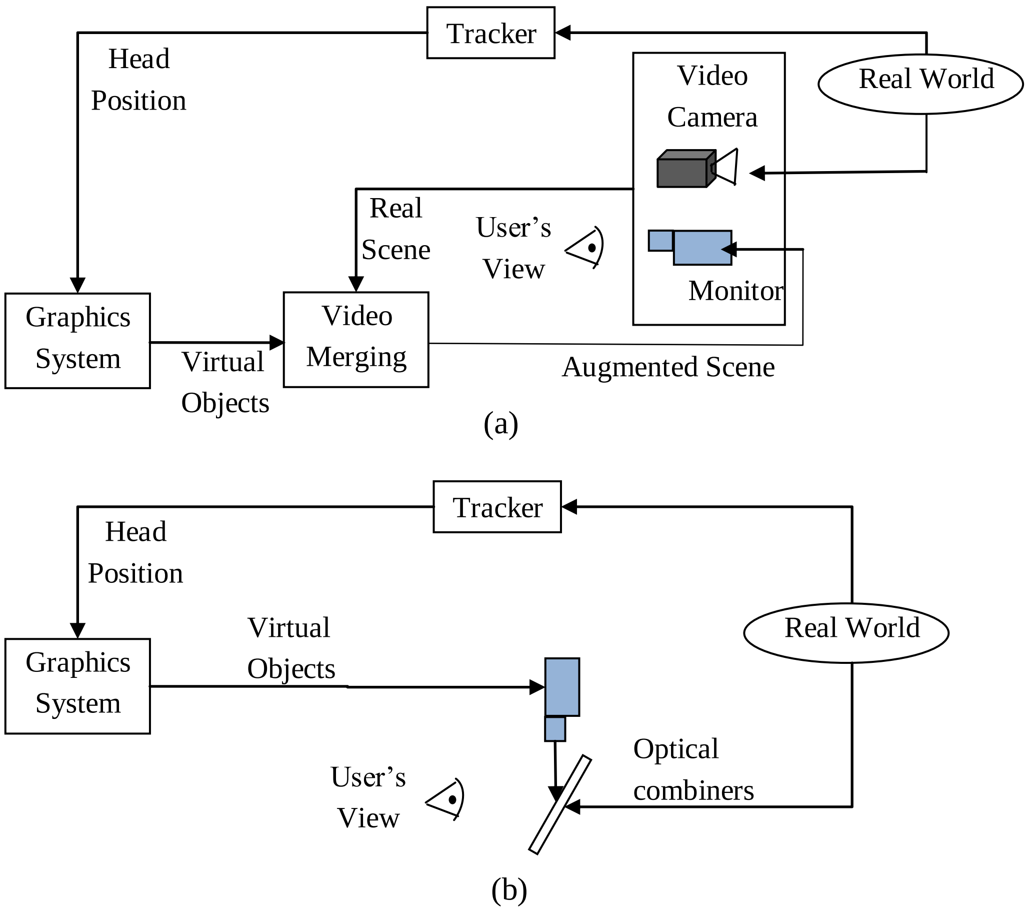
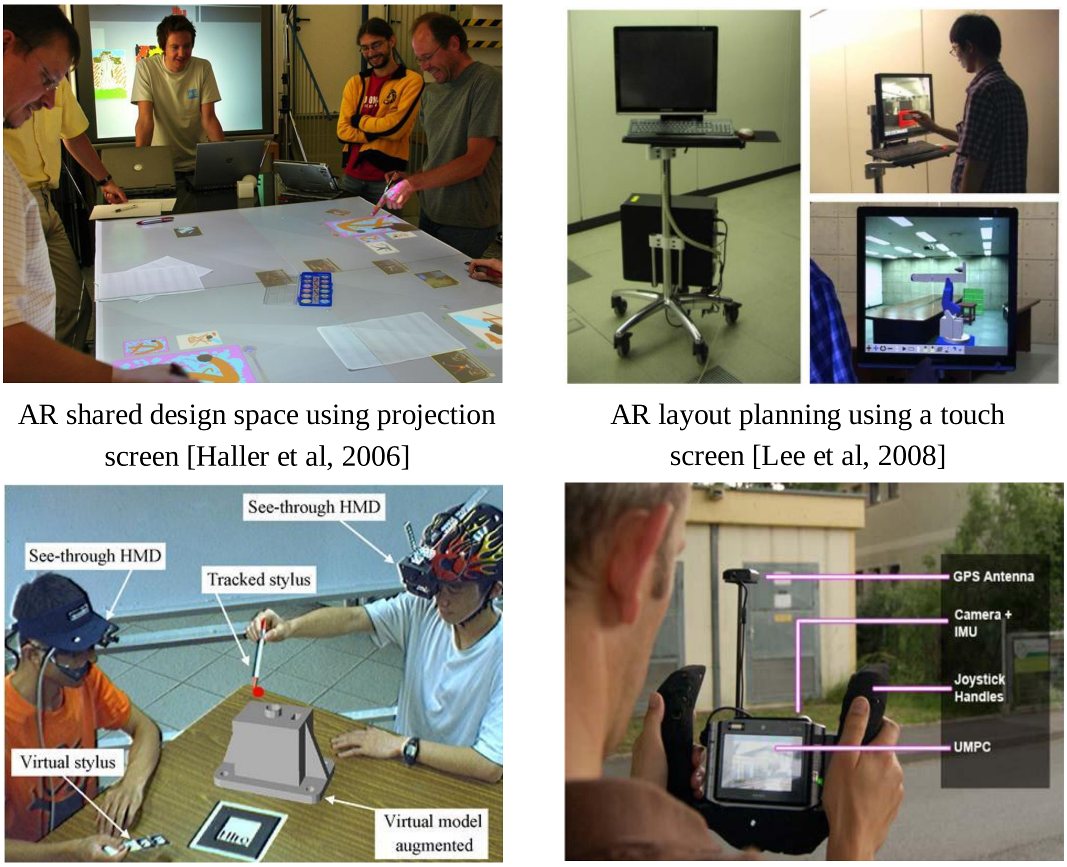


Figure 2-2 Two typical architectures of AR systems: (a) video see-through architecture, (b) optical see-through architecture [Vallino, 1998]

These two configurations have both advantages and disadvantages [Azuma, 1997]. While in the video-see through configuration, real scenes from the camera are combined with the virtual objects created by the graphics system. In contrast, in the optical see-through configuration, the merging of the real world with the virtual objects is performed using the optical combiners. The optical see-through devices have the advantages of low cost, safer and less eye offset. However, the optical combiners usually reduce the amount of light that the user sees from the real world. In addition, the optical see-through systems are more challenging because a high frame rate is required to ensure that the virtual information does not lag behind the real world during augmentation. Therefore, in the video see-through configuration, the user has a heightened sense of immersion in the AR environment.

The most widely used display hardware devices in AR systems are monitors, head-mounted display (HMD), hand-held displays and projection screens. Figure 2-3 shows four set-ups from the AR systems presented in the literature.



AR shared design space using projection screen [Haller et al, 2006]

AR layout planning using a touch screen [Lee et al, 2008]

AR product design using HMDs [Ong and Shen, 2009]

Handheld AR visualization system [Schall et al, 2009]

Figure 2-3 Display hardware

While using a monitor display device or a projection screen, the user has little sensation of being immersed in the environment created by the display and cannot move around in the environment. Due to the fact that mobile devices have become increasingly more powerful and some hardware, such as high resolution camera, touch screen and gyroscope, have already been embedded in these mobile devices, there is a growing trend of using hand-held devices in AR systems [Pasman et al, 2004; Wagner et al, 2006; Xin et al, 2008]. An AR-assisted assembly guidance system using a mobile phone and a computer was presented [Hakkarainen et al, 2008]. However, hand-held devices can restrict the user's hand flexibility.

HMD's can give the user an immersed feeling and have been widely used in AR systems. The choice of video see-through HMD or optical see-through HMD depends on the requirement of specific AR applications. For applications in which realistic visualization is not needed, such as assembly guidance, outdoor AR applications and maintenance, optical see-through HMD will be a good choice. For applications where high registration accuracy and realistic visualization are required, such as surgery, engineering design and robot path planning, video see-through HMDs are more suitable. Pang [Pang, 2006] concluded that the video see-through HMD is currently a suitable choice for the AR-based assembly design and planning systems as it can provide better registration between real and virtual entities and realistic feeling.

Although a HMD can provide immersive feeling to the users, it suffers from a narrow field of view (FOV). The FOV which most HMDs can offer is less than the one which humans have. In order to provide a greater sense of immersion and better situational awareness for the users, a HMD with wider FOV should always be selected for an AR-based system.

2.2.1.2 Tracking

In an AR environment, to ensure that the virtual information is rendered and overlaid properly and precisely on the real scene, the information from the surrounding environment, e.g., places for superimposing the virtual information, the orientation and position of the user's view point (head position), etc., must be obtained. There are five general tracking technologies, namely, time-frequency measurement, spatial scan, inertial sensing, mechanical linkages and direct-field sensing [Rolland et al, 2001]. Trackers often use time-frequency measurement to determine the transmission time of a signal, compare the phase difference of the signal to a reference one, or obtain the time differences indirectly with the help of frequency measurement techniques. Trackers based on the spatial scan principle detect the position and orientation of a target by scanning a work volume. Trackers using inertial sensors measure the acceleration and rotation relative to the reference frame of the earth and have been widely used in navigation for ships, submarines and airplanes. For mechanical trackers, mechanical linkages and encoders are typically equipped to record the rotation or/and displacement of the target with respect to the reference. Trackers based on the direct-field sensing principle carry out tracking through either determining the intensity and direction of the magnetic field or detecting the change of the orientation with respect to the inertial reference frame of the earth. Generally, tracking measurements are subject to signal noise, degradation with distance and interference sources.

Computer vision methods for tracking are becoming more popular as they can avoid the calibration of external sensors and provide the potential for accurate tracking without bulky and costly equipment. These methods have a disadvantage that the tracking process could be affected by poor illumination. Therefore, hybrid tracking methods integrating the advantages of two or more tracking devices to obtain the best quality tracking and minimize the tracking errors have been explored [Fischer et al, 2007; Yang et al, 2007].

Fiducial marker is one of the most commonly used vision-based tracking methods in AR-based systems. The ARToolKit library developed by Kato and Billinghurst [Kato and Billinghurst, 1999] has been widely used in tangible AR applications. ARToolKit is a C and C++ language software library that let the users develop AR applications easily. Thresholding, template matching and line fitting are performed for each frame to obtain reliable projective positions of the four vertices of the markers, which are used to estimate the pose of the camera. As shown in Figure 2-4, ARToolKit uses computer vision techniques to calculate the real camera position and orientation relative to markers and render the virtual information relative to the marker coordinate system in real-time. ARTag utilizes multiple markers with a fixed layout to provide robust and occlusion-free tracking [Hakkarainen et al, 2008; Salonen et al, 2007].

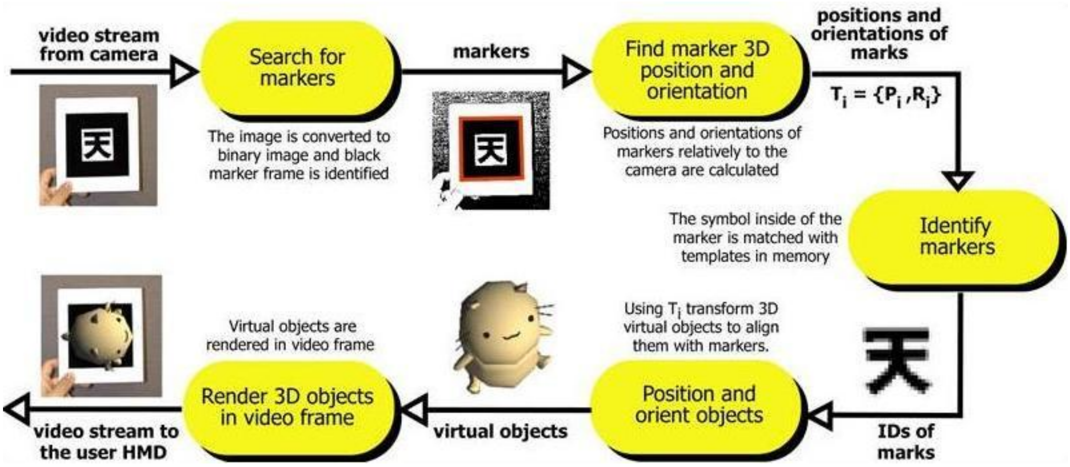

video stream from camera
Search for markers
The image is converted to binary image and black marker frame is identified
markers
Find marker 3D position and orientation
Positions and orientations of markers relatively to the camera are calculated
positions and orientations of marks
$T_i = \{P_i, R_i\}$
Identify markers
The symbol inside of the marker is matched with templates in memory
IDs of marks
Using T_i transform 3D virtual objects to align them with markers.
Position and orient objects
virtual objects
Virtual objects are rendered in video frame
Render 3D objects in video frame
video stream to the user HMD

Figure 2-4 The principle of ARToolKit [ARToolKit, 1998]

As described in the ARToolKit manual [Kato et al, 1999], the working process of ARToolKit is as follows: (1) transform a live video image into a binary image; (2) search the binary image for squares; (3) for every square, match with the pre-trained patterns; and (4) if a match is found with a pre-defined marker, determine the position and orientation of the camera relative to the marker.

This ARToolKit library is easy to use because only a single fiducial marker is required to perform the tracking and rendering process. The disadvantages of ARToolKit are the low accuracy, low occlusion tolerance and the jittering of the virtual information due to the presence of noise in the tracking data. Maximum tracking errors for four different tracking distances are discussed in a study by Malbezin et al [Malbezin et al, 2002], as shown in Table 2-1. ARToolKitPlus is an updated version of ARToolKit which provides new pose estimation algorithms for giving more stable tracking [ARToolKitPlus, 2006].

Table 2-1 Maximum tracking error values for four distances [Malbezin et al, 2002]

Distance (m)	1	1.5	2	2.5
Error (mm)	±14	±18	±22	±27

In order to solve the problem that the virtual objects cannot be augmented when the markers are partially occluded or outside the field of view of the camera while using ARToolKit or ARToolKitPlus, during the last ten years, marker-less methods that use natural features directly for tracking instead of the markers have been developed by many researchers [Ong et al, 2006; Fong et al, 2010]. In these studies, natural features are extracted from two successive frames and correspondences are achieved based on feature similarity. Figure 2-5 shows a marker-less AR application.

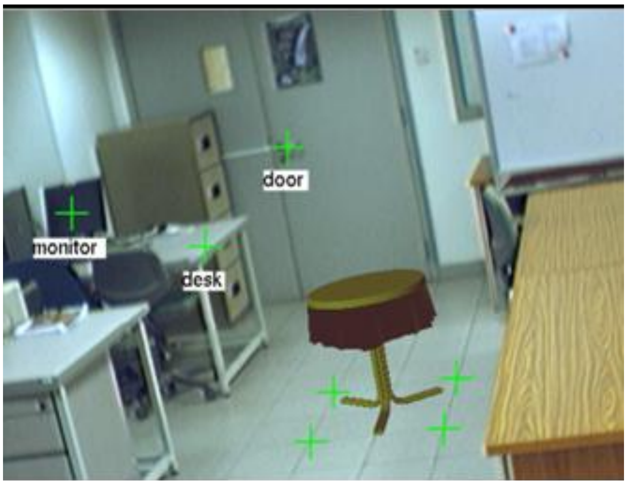


Figure 2-5 Marker-less-based AR application [Ong et al, 2006]

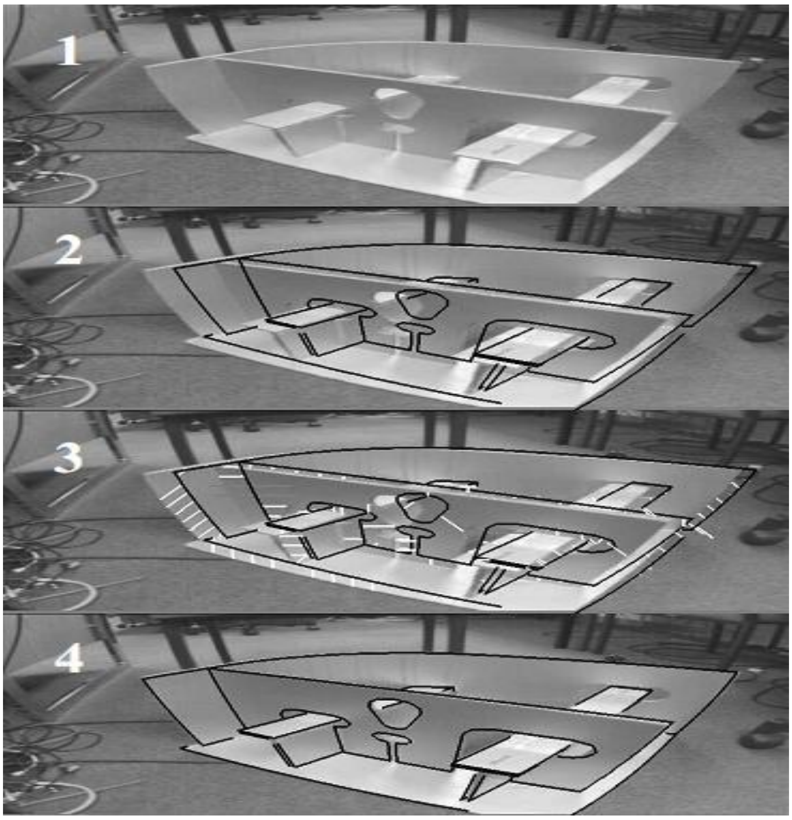


Figure 2-6 Model-based tracking [Klein and Drummond, 2003]

Lepetit et al [Lepetit et al, 2003] presented a 3D tracking method by introducing a rough 3D CAD model of an object inside the view of the camera. Klein and

Drummond [Klein and Drummond, 2003] developed a system in which visual tracking was performed by minimizing the errors between the image frames and the model projections. The tracking loop is shown in Figure 2-6. They further used gyroscopes to track rapid camera motions.

2.2.1.3 Discussion

As high registration accuracy and realistic visualization are required for assembly design and planning in an AR environment, video see-through HMDs are used in this research for display. The computational cost of marker-less tracking is high. In addition, the implementation of marker-less tracking in an uncontrolled environment is more complex and time-consuming. Therefore, stable marker-based tracking using ARToolKitPlus library has been adopted in this research.

2.2.2 Interaction

Basically, the interaction functions in an AR-based system allow users to interact and control the 2D or 3D virtual information in an AR environment. A goal of AR is to come up with interaction that feels as natural as possible to the user. The effectiveness and intuitiveness of visualizing and manipulating activities would be largely determined by the interaction techniques. A review of various interaction methods will be presented in the following two sections.

2.2.2.1 Existing Interaction

The most widely used interaction techniques in AR/VR systems can be classified into three types, namely, command language, distance manipulation and direction manipulation.

The command language method, the earliest form of interaction style, which has been used in desktop-based systems for a long time, is still being used and is very useful in an immersive environment, e.g., to trigger actions and to select virtual objects. The command language method can be implemented in an AR/VR environment using virtual buttons, gesture or speech recognition engines. Speech and gesture are recognized for controlling the virtual models in VECA (Virtual Environment for Collaborative Assembly) to allow collaborative assembly tasks to be performed by users in geographically dispersed locations [Chen et al, 2005]. Besides using gesture or speech recognition engine, some systems used virtual buttons or keyboards for the command language. A control tool called Virtual Interaction Panel (VirIP) for AR systems was developed [Yuan et al, 2004]. The VirIP is composed of virtual buttons, which have meaningful information that can be activated using an interaction pointer during the augmentation process, as shown in Figure 2-7.

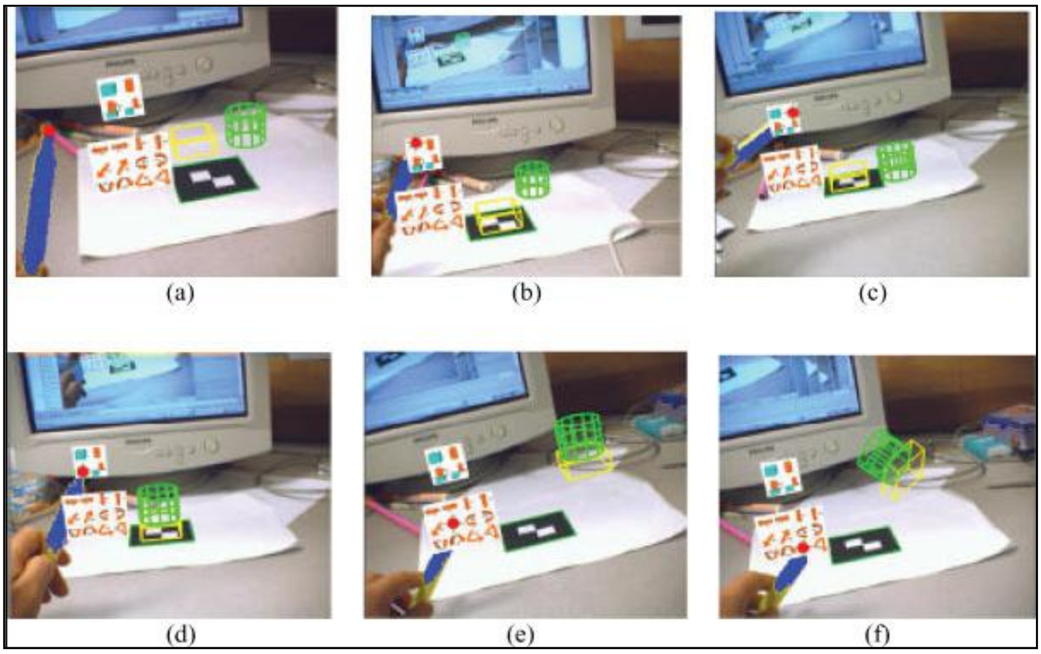


Figure 2-7 The Virtual Panel interaction tool [Yuan et al, 2004]

Unlike systems using command language methods for interaction, some applications allow users to manipulate the virtual objects outside their arm reach regions. In these scenarios, the users can manipulate the virtual object through some interaction tools using distance manipulation methods. A virtual laser was used to achieve the selection of virtual objects in an immersive virtual system [Mine et al, 1997]. A set of interactive tools was designed using the distance manipulation method to give users Virtual X-Ray vision [Bane and Hollerer, 2004], as shown in Figure 2-8.

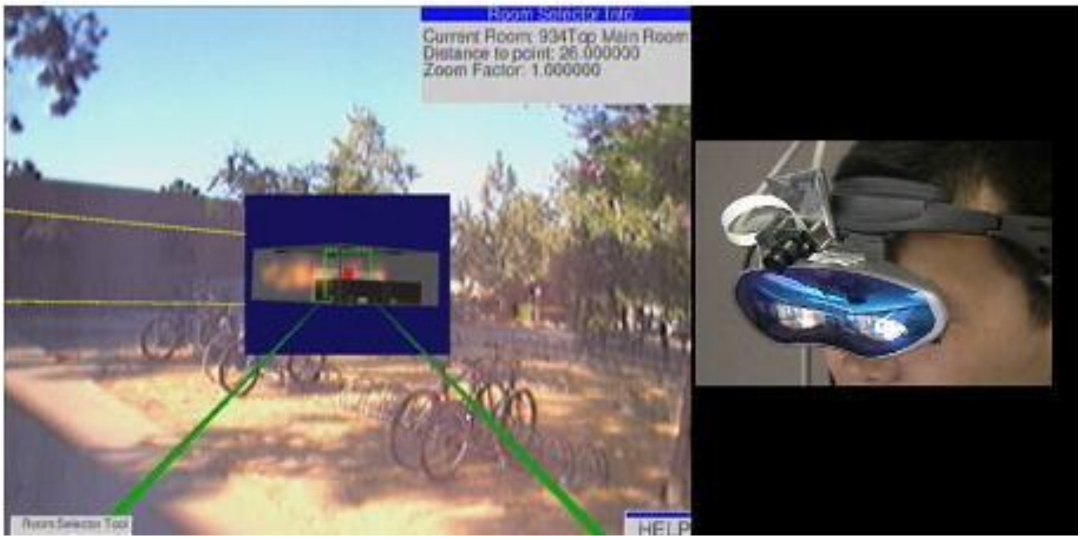
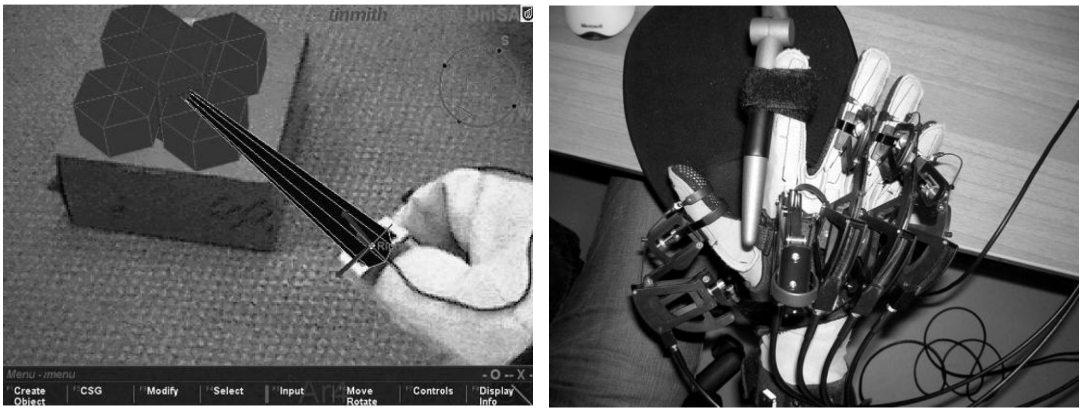


Figure 2-8 Virtual x-ray version using an AR system [Bane and Hollerer, 2004]

As compared with the command language methods and distance manipulation methods, direct manipulation is a more intuitive way to manipulate virtual objects in an immersive environment. A data-glove is an interactive device, resembling a glove worn on the hand, which facilitates tactile sensing and fine-motion control in robotics and VR. Various sensor technologies have been used to capture hand gestures. Sensors mounted on a data-glove send signals to a processing unit which translates hand gestures into data. These data are then interpreted by the software accompanying the data-glove. Data-gloves have been widely used to capture the hand motions in device-assisted hand interaction systems. Although data-gloves are usually used in gesture-based command input systems, data-gloves have been used for direct manipulation of virtual objects in an AR environment in several reported systems. A text entry mechanism based on data-glove was developed to support symbolic manipulation of virtual objects [Thomas and Piekarski, 2002], as shown in Figure 2-9 (a). An augmented system which can provide fingertip-based interaction with virtual objects was developed [Buchmann et al, 2004]. In this system, image processing software and finger-based and hand-based fiducial markers were used to track gestures of the user and fingertip-based haptic feedback devices were used to enable the user to feel virtual objects. Although data-gloves can be used easily to realize HCI, the users may feel uncomfortable when wearing the data-gloves and the cost of these data-gloves can be very high. In comparison with data-gloves, PHANToM, another commonly used device manufactured by Sensable Technologies [PHANToM, 2011], can provide force feedback to users using the concept of direct manipulation. A PHANToM device can provide force feedback caused by the collisions between virtual objects and allow a user to "feel" the surface attributes of virtual prototypes. Using a PHANToM desktop device, the users can simulate manual operations, such as touching, picking and moving virtual prototypes. A haptic VR tool designed by integrating CyberGrasp data-glove [CyberGrasp, 2010] and PHANToM for direct manipulation was developed to enhance the accessibility for the visually impaired [Nikolakis et al, 2004], as shown in Figure 2-9 (b). Although PHANToM devices can provide haptic feedback for direct manipulation, these devices are not portable and are very expensive for the common users.

Many tasks carried out in large-scale AR environments are manipulation tasks with many degrees-of-freedom (DOF), leading to greater difficulties for the users to manipulate the virtual objects using traditional interaction methods as discussed above. To achieve natural and intuitive HCI, human hands can be used as interaction devices.



(a) Data-glove [Thomas and Piekarski, 2002]

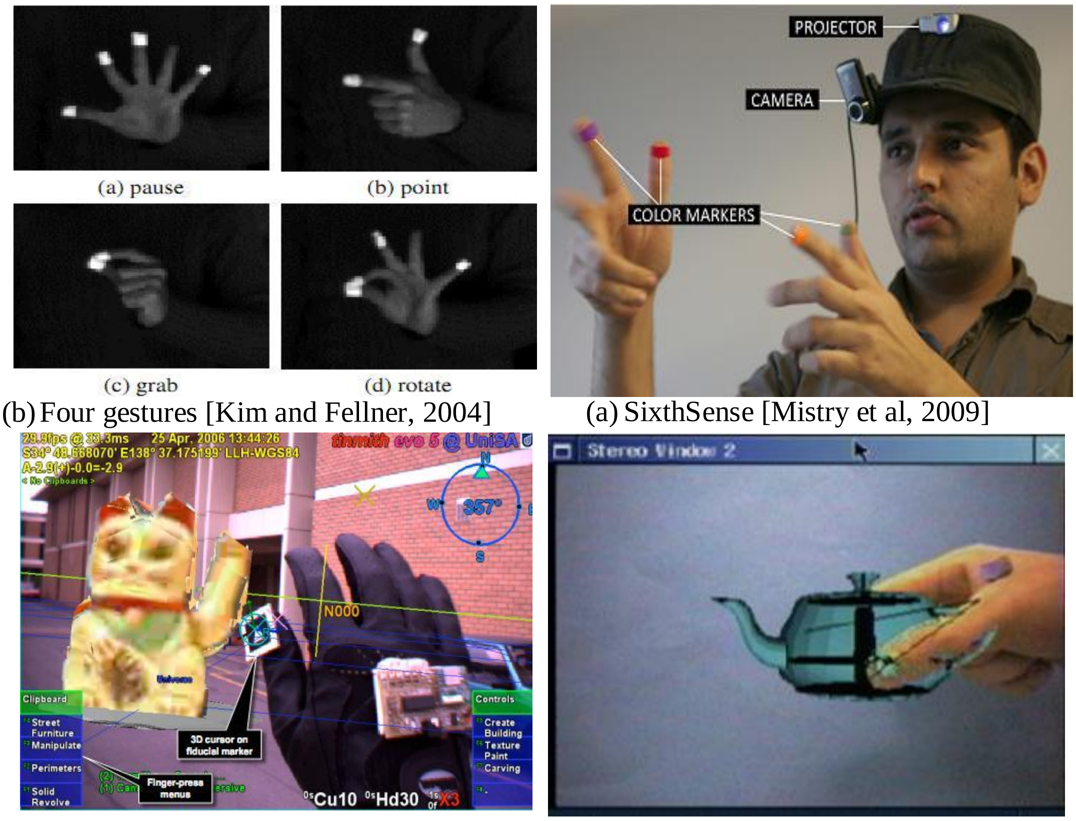
(b) Combination of PHANToM and CyberGrasp [Nikolakis et al, 2004]

Figure 2-9 Direct manipulation user interfaces in AR systems

2.2.2.2 Bare-hand Interaction

As compared with traditional HCI methods, hand interactions are less intrusive and more convenient for users to interact with the virtual contents and explore the 3D augmented environment. Using bare hands interactions will make HCI more intuitive and easier to operate than traditional methods, especially in high DOF manipulation tasks in an AR environment.

Vision-based human hand detection and tracking systems can identify the gestures of bare hands from video streams and use them as commands, which computers can understand and respond to. There are a number of AR interfaces that use computer-vision (CV) based hand tracking. Earlier attempts to use human hands as interaction tools typically track special markers that are attached on the hand features. A hand interaction interface was developed by using thimble-shaped markers and invisible light sources, which radiate in an ultraviolet wavelength region, to detect the position of four fingertips [Kim and Fellner, 2004]. Four gestures (pause, point, grab and rotate) were designed as the selection and manipulation commands to interact with virtual objects in this system, as shown in Figure 2-10 (a). A CV-based wearable and gestural information interface called SixthSense was designed to augment the physical world with digital information and natural hand gestures are used to interact with the augmented information [Mistry et al, 2009], as shown in Figure 2-10 (b). Colour markers at the tip of the users' fingers are used to track the locations of the fingertips using CV techniques. Hand-worn gloves (Figure 2-10 (c)) for interacting with mobile outdoor AR systems in 3D space were designed by tracking the fiducial markers attached on the thumbs of these gloves [Piekarski and Smith, 2006]. An interface which allows the user to pick up, move and rotate virtual objects was developed [Kojima et al, 2001]. The motion of the fingertips was captured by tracking the colour makers attached on the user's hand, as shown in Figure 2-10 (d). Using markers is an effective method to simplify the hand feature detection procedure, and the gesture parameters can be calculated easily and efficiently. However, the markers must be specially designed for calibration and tracking purposes. Moreover, if a hand is attached with too many markers, segmenting each marker will cause problems.



(b) Four gestures [Kim and Fellner, 2004]

(a) SixthSense [Mistry et al, 2009]

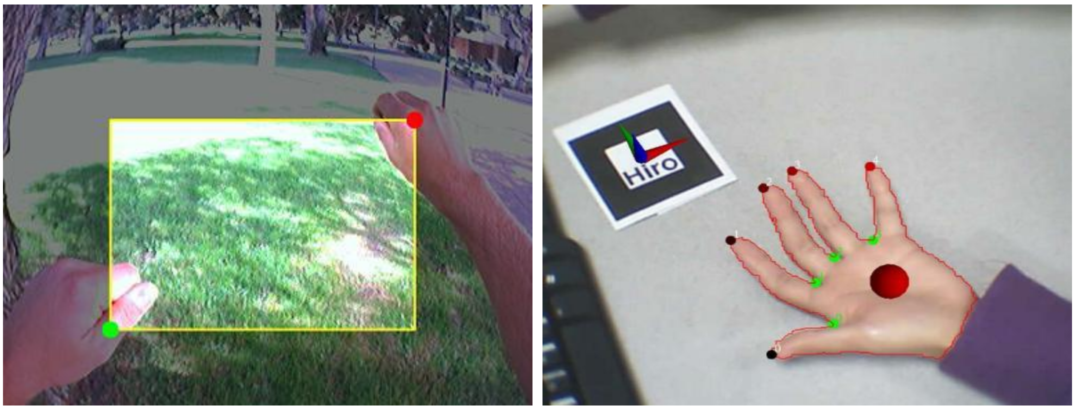
(c) Hand-worn glove [Piekarski and Smith, 2006]

(d) Interaction using Fingertips [Kojima et al, 2001]

Figure 2-10 CV-based hand interaction

Bare-hand interaction methods can be classified roughly into two groups, namely, indirect bare-hand interaction and direct bare-hand interaction. Indirect bare-hand interaction uses non-contact hand gestures for control and manipulation.

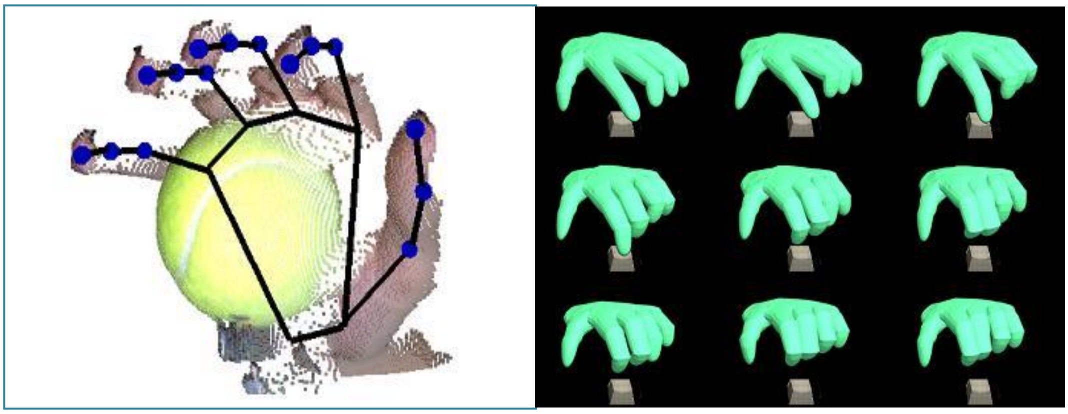
Vision-based human hand detection and tracking systems can identify the gestures of bare hands from video streams and use them as commands, which computers can understand and respond to. A procedure to determine the gestures was presented and a system to recognize simple sign language was tested [Nielsen et al, 2004]. A single camera-based video input system which allows 3D interaction with existing computer applications using bare hands was developed [Dhawale et al, 2006]. A vision-based hand gesture interface for navigating the learning object repository mapped in a 3D virtual environment was designed and implemented [Chen et al, 2007]. A VR-based 3D assembly system which uses two-handed gestures for virtual objects handling was developed [Wang et al, 2011]. A hand shadow-based interaction was developed to enable users to interact with virtual objects using the space above an interactive tabletop [Hilliges et al, 2009]. HandVu [Kölsch, 2004], which can be used in both VR and AR applications, is a vision-based system for on-line gestures recognition and it provides 2D image coordinates of the locations of the user's hand, as shown in Figure 2-11 (a). Hand gestures have also been widely used for game control [Schlattmann et al, 2009; Yoon et al, 2006]. Besides interaction using hand gestures, using bare hands to hold and interact with 3D virtual objects has been explored in recent years. There are several AR interfaces that allow the users to hold and inspect 3D virtual objects by rendering virtual objects on their hands. HandyAR [Lee & Höllerer, 2007] used a human hand as a tracking pattern to display augmented virtual objects on the hand. In the HandyAR system, fingertips are detected and tracked to reconstruct the camera pose frame-by-frame. Stiff fingertips would cause changes in the 3D coordinate system and a calibration process is required. As the convexity defect points of the human hands are relatively static during hand movements, a vision-based hand interaction methodology based on four convexity defect points to obtain the camera pose frame-by-frame in an AR environment was designed [Shen et al, 2011], as shown in Figure 2-11 (b).



(a) HandVu [Kölsch, 2004] (b) Virtual object on hand [Shen et al, 2011]

Figure 2-11 Indirect bare-hand interaction

Direct bare-hand interaction will be triggered when there is a contact between the human bare hand and the virtual objects. The best way for direct manipulation of virtual objects is to use the 3D hand model for tracking. A system that can track the articulated 3D pose of a hand while the hand is interacting with objects was designed [Hamer et al, 2009], as shown in Figure 2-12 (a). This system can track the configuration of the local hand parts using soft constraints and model occlusion explicitly using both hands and the virtual objects. A 3D hand model fitting method was developed to recover accurate finger positions for a virtual keyboard system [Du and Charbon, 2007], as shown in Figure 2-12 (b). This system uses a structured light sensor to measure the motion of the user's hand using a 3D hand model that consists of a polygonal skin driven by an underlying skeleton system. Due to the high dimensionality of a user's hand, the 3D model-based hand tracking methods are computationally expensive. Therefore, these methods are difficult to process in real time and are not well suited for interactive applications.



(a) Tracking of a3D hand pose [Hamer et al, 2009]

(b) 3Dhand model fitting [Du and Charbon, 2007]

Figure 2-12 Direct bare-hand interaction

2.2.2.3 Discussion

Most of the existing systems do not support 3D natural hand interactions in real time. In this research, a 3D natural bare-hand interaction methodology and a dual-hand interface based on this methodology have been developed, which will be discussed in detail in Chapter 3.

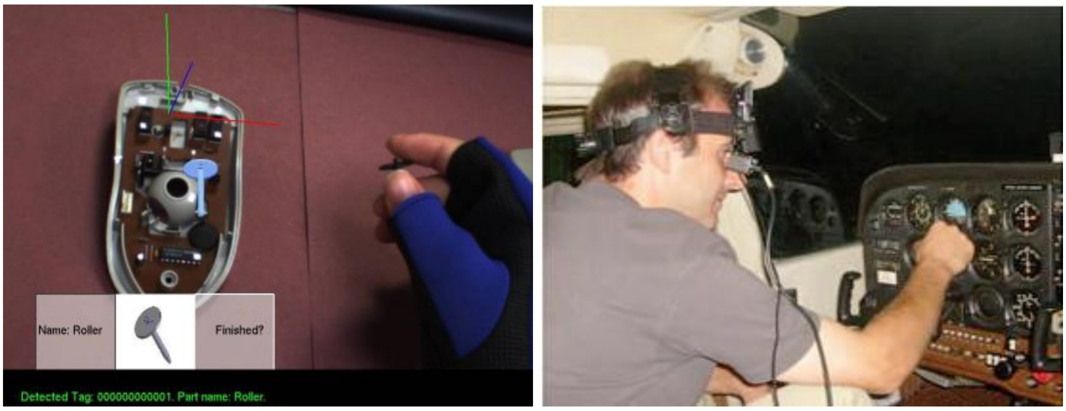
2.3 AR Applications

AR has become increasingly more popular in industry applications. In the last fifteen years, many of AR applications have been reported in various areas, e.g. medical [State et al, 1996; Harders et al, 2007], entertainment [Schlattmann et al, 2009; Yoon et al, 2006], education [Kaufmann and Schmalstieg, 2003; Zagoranski and Divjak, 2003] and engineering design and planning [Doil et al, 2003; Wang and Jen, 2006]. In this section, a brief review of the industry applications of AR is introduced.

2.3.1 Assembly Guidance and Maintenance

Many researchers have focused their effort on on-line AR-assisted assembly training, assembly guidance and maintenance. An AR interface can provide assembly instructions in real time and in the real environment without alternating attentions between the assembly workspace and the instructions available on other computers or paper manuals. An AR-assisted system in the aerospace industry was developed for the purpose of improving workers' performance of manufacturing activities through the use of head-mounted display technologies [Caudell and Mizell, 1991]. An AR-aided assembly work using a Visual Assembly Tree Structure (VATS) was presented [Yuan et al, 2008]. A general procedure for AR-assisted assembly training system was developed [Chimienti et al, 2010]. This system was used for training operators during the assembly of a planetary gearbox with the help of a hand-held device. RFID technologies were adopted in an application for assembly guidance in an AR environment [Zhang et al, 2010], as shown in Figure 2-13 (a). Traditional assembly support media were compared with AR-assisted assembly (ARsembly) and AR support proved to be more suitable for difficult assembly tasks [Wiedenmaier et al, 2003].

The Knowledge-based Augmented Reality (KARMA) system was developed for 3D maintenance tasks [Feiner et al, 1993]. Using automatic knowledge-based generation of output depending on a series of rules and constraints, this system can support the performance of simple tasks through augmenting virtual information onto a user's view. An AR-based system that provides marker-less tracking and 3D animation overlap on several similar apparatuses on different airplanes and in different lighting conditions was developed for aircraft maintenance [De Crescenzio et al, 2011], as shown in Figure 2-13 (b). A prototype AR application to support military mechanics conducting routine maintenance tasks inside an armoured vehicle turret was developed [Henderson and Feiner, 2009]. This prototype used a tracked HMD to augment a mechanic's natural view with text, labels, arrows, and animated sequences designed to facilitate task comprehension, location, and execution.



(a) Assembly guidance [Zhang et al, 2010]

(b) Aircraft maintenance [De Crescenzio et al, 2011]

Figure 2-13 AR-based assembly guidance and maintenance

2.3.2 Product Design

A spaced design system which is an innovative AR system addressing the aesthetic design of free form curves and surfaces was presented [Fiorentino et al, 2002]. The Fata Morgana system [Klinker, 2002] investigated the AR presentation for automobile design evaluation. A framework of an AR system for rapid evaluation of product prototypes through mixed prototypes was presented [Balcisoy et al, 2000]. An augmented prototyping system which aims at generating and utilizing prototypes that can represent not only the primary function but also other functions, such as looking nice in aesthetic shape and keeping good in overall structure was presented [Park et al, 2009], as shown in Figure 2-14. This system facilitates the tangible interaction, realistic visualization, and functional behaviour simulation of a digital handheld product in an AR environment. A video see-through AR interface was integrated into three prototype 3D applications in three different domains: engineering systems, geospace, and multimedia [Kim and Dey, 2010].

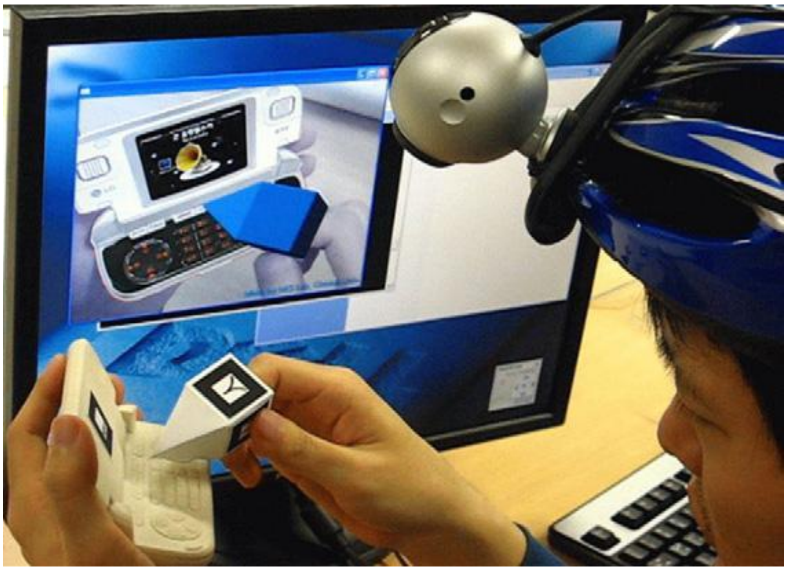


Figure 2-14 AR-based product design [Park et al, 2009]

2.3.3 Factory Layout Planning

The application of digital manufacturing technology to verify a real environment in a virtual reality environment has been extended to the augmented reality environment. A mixed reality-based virtual factory layout planning system was developed and its effectiveness in the simulation of process layout planning was demonstrated [Lee et al, 2011], as shown in Figure 2-15. An AR-based application tailored to the needs of industrial factory planning processes was presented [Doil et al, 2003; Pentenrieder et al, 2008]. A prototype of an AR-based construction planning tool (AR Planner) with virtual elements sets and tangible interface was developed [Wang and Jen, 2006]. Worksite planning rules were integrated into the AR planner with the purpose of preventing potential planning errors and process inefficiency intelligently.

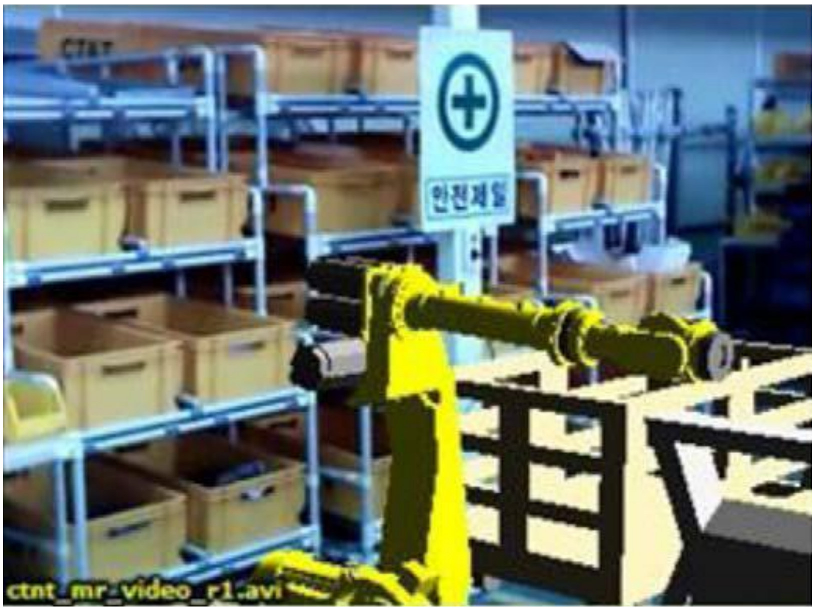


Figure 2-15 AR-based layout planning [Lee et al, 2011]

2.3.4 Robotics

Preliminary investigation of a robot system in a simulation environment offers many advantages over direct implementation. Not only is virtual simulation safe, but it also allows the user to observe the complete state of the virtual world, interact with it and visualize the performance of each system component under various conditions. A method for intuitive and efficient programming of industrial robots based on AR is presented [Zaeh and Vogl, 2006]. An AR cueing method designed to improve teleoperator performance under conditions of display control misalignment is investigated [Chintamani et al, 2010]. An automatic mobile robot system controlled using a marker and remote indication employing the augmented reality technology was developed [Ohmori and Sakamoto, 2010]. An AR-based system was proposed to assist the users in planning and programming robotic tasks complying with robot dynamics constraints [Fang et al, 2011], as shown in Figure 2-16.

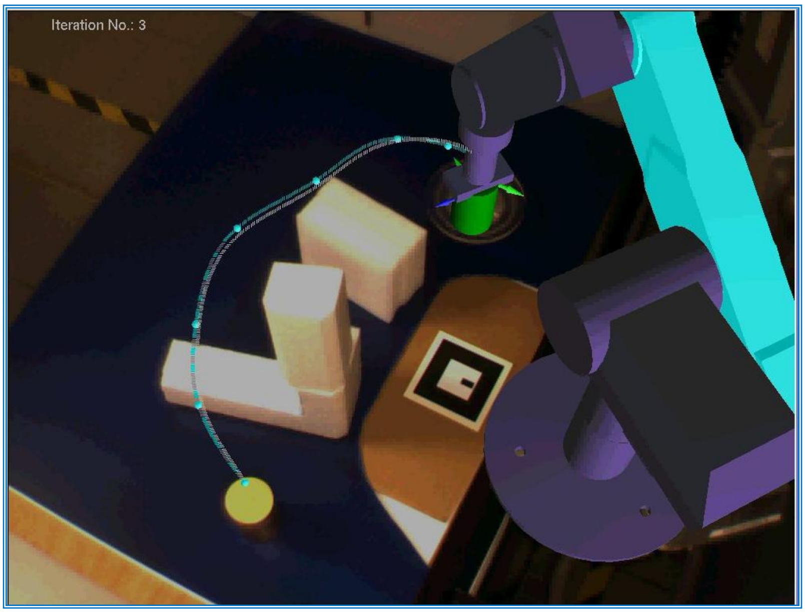


Figure 2-16 AR-based robotics planning [Fang et al, 2011]

2.3.5 Discussion

An AR environment provides the promising potential to generate a highly immersive and intuitive HCI for engineering design and planning. Engineers are able to manipulate and evaluate the virtual prototypes of a product in a real assembly environment, capture good assembly intent and make better decisions in assembly design and planning as compared to working in the conventional desktop-based CAD systems or VR-based systems. In this research, an AR-based system for manual assembly design and simulation has been developed.

A video-based AR system using a video see-through HMD is implemented in this research as it provides a more realistic and accurate registration result as compared to the optical see-through HMD. For the tracking technologies, the computer vision-based technologies are adopted as they are simpler, cheaper and easier solutions to set up a satisfactory AR environment as compared to sensor-based technologies. ARToolKitPlus [ARToolKitPlus, 2006] library is used to build the AR assembly environment. A 3D bare-hand interaction tool has been developed to allow the user to manipulate and orientate virtual parts simultaneously in real time.

Currently, the main use of AR in most of the reported AR assembly systems is to display the assembly information to guide the assembly operations. In these systems, the assembly design process has been completed in advance and predefined assembly information is imported from CAD software and used to guide the assembly operations. This is not significant in reducing the product design lead time and improving the assembly design and planning. In addition, there is limited intuitive interaction between the users and the AR assembly systems in the early design stage. The first area of interest in this thesis is to develop a natural 3D bare-hand interaction tool to enhance the interactions between the users and the virtual components in an AR-based environment. The second area of interest in this thesis is to develop a systematic methodology to realize interactive constraint-based augmented assembly for manual assembly simulation. The manual assembly design system is augmented by adopting AR technologies to allow users to conduct assembly design and planning using virtual components in a real environment. More detailed discussion of the assembly design and planning issues is presented in the next section.

2.4 Assembly Design and Planning

In concurrent engineering, there is the need for integrating design with assembly planning. A concurrent engineering platform should be able to perform preliminary assembly planning during conceptual design stages so that alternative assembly plans can be evaluated and compared for redesigning or refining a design with a promising candidate assembly plan. In this section, existing AR-based assembly design and planning systems are first reviewed and discussed. Activities involved in assembly design and planning, such as assembly modelling and constraint-based assembly, will also be discussed.

2.4.1 AR-based Assembly Design and Planning Systems

Some researchers have focused their effort on on-line AR-based assembly design and planning. Wang et al [Wang et al, 2010] discussed several key aspects in AR-based assembly systems, such as occlusion handling, collision detection and human-computer interaction. A prototype system was implemented to illustrate the application of these techniques. The WebShaman Digiloop system [Halttunen and Tuikka, 2000] augmented virtual prototypes with physical objects to examine the functionality and features of products through assembly operations in an AR environment. Pang et al [Pang, 2006] presented an AR system to support the design of assembly features. This system can help users design the assembly features to provide proper part-part constraints in the early design stage. The virtual assembly features were rendered on the real assembly platform using a marker-less registration techniques. The model-based collision detection technique was implemented for assembly constraint evaluation. Ong et al [Ong et al, 2007] used AR to integrate the assembly product design and planning activities with the workplace design and planning process to improve the assembly procedures and layout. The designers can analyse and improve this product assembly design through different feature operations, e.g., changing the position of a feature, or modifying the feature attributes. Valentini [Valentini, 2009] presented a virtual assembly system using a data-glove as an interaction tool. The proposed methodology was based on the recognition and interpretation of the user's intent about the grasping, moving and constraining of the virtual objects in the scene. All these three tasks can be performed by deducing equations about specific geometrical reference frames attached to the objects.

Assembly sequence planning has always been a key issue in the manufacturing industry [Gottschlich et al, 1994]. Raghavan et al investigated an information presentation scheme [Raghavan et al, 1999] for AR stimuli in assembly sequence planning. The developed AR-based assembly evaluation tool would allow a manufacturing engineer to interact with the assembly planner while manipulating the real and virtual prototype components in an assembly environment. This mixed prototyping can provide a better understanding of the different constraints and factors involved in assembly design and evaluation. Liverani et al [Liverani et al, 2004] described a personal active assistant (PAA) tool to assist and evaluate the assembly operations. PAA exploits a CAD tool connection to improve object recognition, best assembly sequence optimization, and operator instructions generation in an integrated environment based on CAD assembly software and on an AR wearable system.

2.4.2 Assembly Modelling and Planning with Assembly Features

Assembly features have been widely used to improve the efficiency of the assembly design and planning process [Sung et al, 2001; van Holland and Bronsvoort, 2000]. There are significant variations in the definition of assembly features. Assembly features are described as “mating pairs of form features with parameters and compatibility constraints as part of each feature definition” [Shah and Rogers, 1993]. This definition is restrictive as only mating information between two components can be obtained from the features. Assembly features are defined as “regions of geometry that are identifiable in the manufacturing sense” [Anantha et al, 1996]. This definition describes machined features in an assembly rather than spatial or proximity relationships between components. Assembly features are described as “features with significance for assembly processes” and are subdivided into connection features (such as final position, insertion path/point, tolerances) and handling features, (characteristics that give the locations on an assembly component that can be handled safely using a gripper during assembly) [van Holland and Bronsvoort, 2000].

The common theme in all the above definitions is that assembly features are the geometric or non-geometric attributes (mating relation) of a discrete part of which the presence or dimensions are relevant to the manufacture, engineering analysis and usage of the part.

Assembly features are used in both assembly modelling and assembly planning to improve the assembly efficiency [van Holland and Bronsvoort, 2000]. The system is implemented in a prototype assembly modelling program together with several other assembly planning modules, such as grip planning, stability analysis and assembly sequence planning. Assembly features are used in the assembly planning process for the assembling of an aeroplane [Deneux, 1999]. The first step creates a four by four triangular relationship matrix that encodes the set of possible relations between components of the same category. The assembly features are retrieved from the matrix by considering typical complex assemblies and analysing the relationships between the components.

2.4.3 Constraint-based Assembly

To assemble products in an AR/VR environment, traditionally, users would have to transfer assembly information, such as geometric information of parts, assembly tree, assembly constraints, etc., to the assembly environment using a special data converter in advance. In a constraint-based assembly system, the user can position each part accurately by satisfying all the constraints according to the assembly information. The VADE system [Jayaram et al, 1999] used Pro/Toolkit to import assembly data for simulating assembly operations in a VR environment. Pre-defined geometric constraints are activated to simulate constrained motion when parts approach mutual proximity. A constraint-based assembly system in a VR environment was presented [Yang et al, 2007]. In this system, geometric modelling and assembly design of products are carried out in a CAD system in advance and the geometric and assembly information are transferred to the VR environment using a data converter subsequently. Constraint recognition is performed when the bounding boxes of two interrelated parts are intersecting. A VR-based assembly modelling approach through constraint-based manipulations was developed [Zhong et al, 2005]. By integrating manipulation with a feature mating knowledge base, the pairs of mating features between components can be recognized.

2.4.4 Assembly Planning

During the last two decades, assembly planning has been an important research area both in mechanical engineering and in computer science. Early assembly planning systems are mostly based on user interaction. Many systems focus a user's attention either on a connection between a pair of parts or a single assembly operation in the assembly [De Fazio and Whitney, 1987; Wilson, 1995]. These approaches generate all the precedence relationships of an assembly between its liaisons by querying a set of structured questions based on its liaison graph. However, this querying method requires users to answer either a large number of easy questions or a small number of difficult questions. Obviously, this method is far from automation. To support automatic assembly planning, much research effort has been made to develop computer-aided assembly planning systems. A number of geometric reasoning approaches to automated assembly planning have been proposed. One general approach is the cut-set method used by many researchers [Baldwin et al, 1991; Homem and Sanderson, 1991]. The cut-set method follows the compute-and-test scheme, where all the possible ways to partition an assembly into two connected subassemblies are generated, and each partitioning is tested for local and global freedom using geometric reasoning methods.

To generate good assembly plans, non-geometric assembly data should also be considered in assembly planning. There have been several approaches that generate assembly sequences by using high-level expert knowledge or experience [Chakrabarty and Wolter, 1997; Swaminathan and Barber, 1996]. However, a common drawback of these existing interactive or automatic systems is that the role of the human in assembly planning is not considered and included. They fail to address issues related to ergonomics, such as awkwardness to perform assembly operations, etc.

Currently, assembly planning processes are supported by many computer-aided tools. Computer-aided process planning (CAPP) has been developed to simplify and improve process planning. Several systems integrate CAD and CAPP for assembly process planning [Grabowik et al, 2005; Zhou et al, 2007]. Digital mock-up (DMU) is a realistic computer simulation of a product for its entire life cycle in a 3D manner. DMU system has been widely used in product development in the early design stage without physical mock-ups [Song and Chung, 2009]. Digital Human Modelling (DHM) can be used to estimate work load in real-life tasks. DHM has been widely used in ergonomic evaluation in assembly planning processes [Chang and Wang, 2007; Fritzsche, 2010].

VR techniques have been widely researched on to support assembly design and planning. VR assembly systems can visualize the behaviour of the human operators during the assembly process. A VR assembly system was implemented in industrial case studies and it demonstrated that VR is able to model the interactions of the actual assembly of a large press machine and address ergonomics issues in the assembly of a wheel to a truck [Jayaram et al, 2007]. An experiment that investigated the potential benefits of VR in supporting assembly planning was presented [Ye et al, 1999]. It was observed that the participants could on average perform the assembly operations in approximately half the time in the immersive and non-immersive VR environments than in the traditional environment using blueprints. A virtual assembly design environment that allows engineers to plan, evaluate and verify the assembly of mechanical systems was described and the software and hardware technologies that are needed to perform VR assembly were discussed [Jayaram et al, 1997; Jayaram et al, 1999]. In a survey of VR assembly systems, it was concluded that the requirements of a successful VR assembly system are the support of geometric and subjective design and assembly rules, collision detection of the interacting parts, detection and management of inter-part constraints, physics-based modelling and interactions that are close to the real thing for human factor analysis [Seth et al, 2010]. A knowledge-based virtual assembly system was developed for automatic assembly sequences planning [Fan and Dong, 2001]. Most VR systems are able to fulfil these requirements but the ergonomic issue may be difficult to solve in a pure VR environment. This is due to the fact that VR systems usually take a human operator away from the actual assembly workspace.

The intuitive interactions provided by AR technologies allow assembly design and planning to be achieved in a robust way by considering human ergonomics during the design and planning process. Raghavan et al investigated an information presentation scheme [Raghavan et al, 1999] for AR stimuli in assembly sequence planning. The developed AR-based assembly evaluation tool allows a manufacturing engineer to interact with the assembly planner while manipulating the real and virtual prototype components in an assembly environment. This mixed prototyping approach can provide a better understanding of the different constraints and factors involved in assembly design and evaluation. A personal active assistant (PAA) tool which can be used to assist and evaluate the assembly operations was described [Liverani et al, 2004]. A virtual assembly system using data-glove as an interaction tool in an AR environment was presented [Valentini, 2009]. The proposed methodology was based on the recognition and interpretation of the user's intent about the grasping, moving and constraining of the virtual objects in the scene. All these three tasks can be performed by deducing equations about specific geometrical reference frames attached to the objects.

2.4.5 Discussion

Feature-based assembly systems offer a convenient way to simulate and design assembly operations. During an assembly process, a particular assembly operation is always related to a single or compound assembly feature. With the attributes of the related assembly features, it is possible to generate useful information automatically, e.g., the geometric assembly constraint, for evaluating an assembly operation, e.g., estimate the manipulability of an assembly operation, etc. Designing an assembly is to ensure proper assembly features for delivering the proper assembly intent. In this thesis, only the geometric attributes of a part are considered as assembly features. This thesis has been focused on on-line assembly constraints recognition and confirmation using assembly features. Interactive assembly planning and evaluation methods have been developed for the user to obtain a near-optimal sequence.

2.5 Summary

In this chapter, general descriptions of AR technologies and applications, and assembly design and planning processes have been presented. The basic AR enabling technologies including video and optical techniques, tracking, registration, and interaction are first discussed. A brief review of the various industrial AR applications is presented. It is observed that CV-based AR technologies provide a simpler and more economical solution to set up a satisfactory AR environment and will be used in this research. In order to develop a systematic methodology to improve assembly design and planning in the AR assembly environment, the published research work in assembly design and planning has been reviewed and discussed in this chapter.

Chapter 3 Detailed System Description

3.1 Introduction

The objective of this research is to develop an AR-assisted system to perform manual assembly design and planning. Using this system, the user can assemble components together to complete an assembly using his/her bare hands. The 3DNBHI interface enables the user to manipulate the virtual objects robustly. The system can achieve an intuitive recognition of the mating relations between the components and the functions of the assembly. In this chapter, detailed architecture of the ARIMADP system is presented. The TADS forming the basis of the system is introduced. The software and hardware used in this system is presented. This chapter also presents the coordinate systems used in the ARIMADP system.

3.2 System Architecture

The ARIMADP system consists of five units, namely, an assembly platform, a HMD, a Firefly camera, a BumbleBee2 camera and a PC. The PC is used as the computing unit for image processing, virtual information generation and rendering, assembly constraint recognition and confirmation. The system has four functional modules, which are Interaction, Visual Output, Assembly and Data management. The architecture of ARIMADP is depicted in Figure 3-1.

1. The 3DNBHI tool is realized in the Interaction Module. CV technologies have been used to formulate and develop the 3DNBHI method. A dual-hand interface based on 3DNBHI has been implemented for the users to manipulate and orientate parts or sub-assemblies. By closely replicating the real world interactions, the dual-hand interface makes the assembly operations in ARIMADP more realistic. The details of the 3DNBHI tool and the dual-hand interface are discussed in Chapter 4.
2. The augmented view is provided to the user through a video see-through HMD in the Visual Output Module. Wearing a HMD, the user can visualize 3D virtual components from different perspectives. By visualizing the rotation and translation of virtual components that are achieved using the 3DNBHI, the user can have an immersive experience.
3. Constraint-based augmented assembly is realized in the Assembly Module. Without the need for any auxiliary CAD information, this module provides the user an intuitive cognition on the mating relations between the components and the functions of the assembly. Based on an interactive evaluation method, a near-optimal sequence can be selected from a number of feasible sequences.. The details of the constraint-based assembly process and the assembly planning algorithms are discussed in Chapter 5.

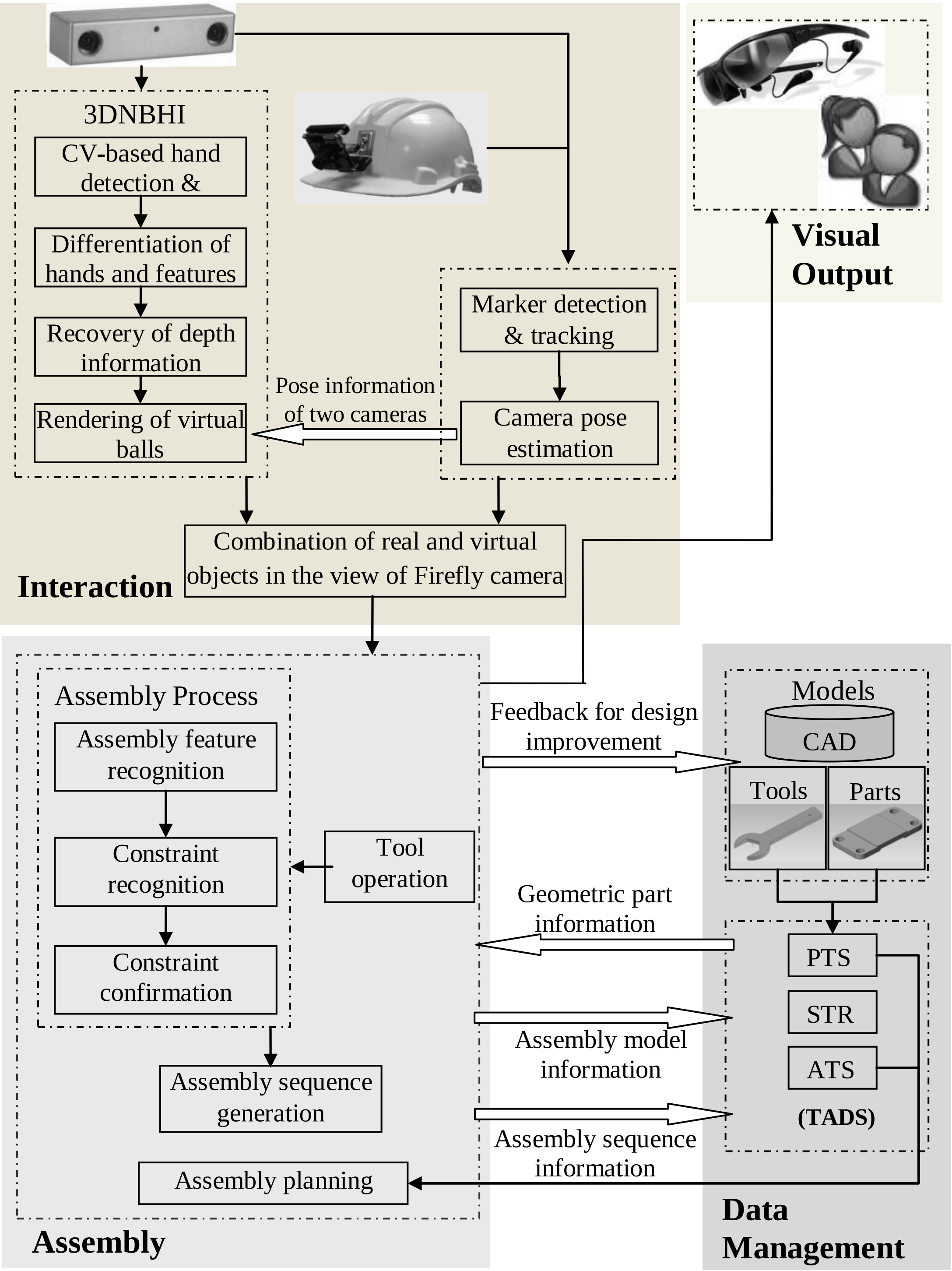

3DNBHI
CV-based hand detection &
Differentiation of hands and features
Recovery of depth information
Rendering of virtual balls
Pose information of two cameras
Marker detection & tracking
Camera pose estimation
Combination of real and virtual objects in the view of Firefly camera
Interaction
Visual Output
Assembly Process
Assembly feature recognition
Constraint recognition
Constraint confirmation
Tool operation
Assembly sequence generation
Assembly planning
Assembly
Feedback for design improvement
Geometric part information
Assembly model information
Assembly sequence information
Models
CAD
Tools
Parts
PTS
STR
ATS
(TADS)
Data Management

Figure 3-1 The architecture of ARIMADP

4. A Tri-layer Assembly Data Structure (TADS) is adopted in the Data Management Module to help users manage the assembly data. The details of the TADS are discussed in the next section.

3.3 Tri-layer Assembly Data Structure

To support assembly modelling and design in an AR-based environment, the TADS forming the basis of the ARIMADP system is developed to help users manage the assembly data.

The first layer stores the geometric information required for assembly modelling. A part-tree structure (PTS) is used to manage the geometric information of the parts. An assembly tool is considered as a part in PTS. In the ARIMADP system, different part IDs are created for different models and different surface IDs are created for different surfaces. The root nodes of PTS consist of the IDs of the parts loaded into the system at the initialization step. Each intermediate node consists of an ID of the surface of its parent (part) and each leaf node contains the geometric information (type and parameter of its parent (surface), as shown in Figure 3-2.

The assembly sequence is stored in the second layer. The method to represent assembly sequences using task representations (STR) [Homem de Mello and Sanderson, 1991] is used in ARIMADP. With the assumptions that no two assembly tasks share the same input sub-assembly (either an assembly component or a sub-assembly of a group of assembly parts) and each assembly task contains two input sub-assemblies, the assembly sequence is stored in an ordered list of task representations. One assembly operation is executed for each task. In this research, three types of assembly operations are considered, namely, Insert, Placement and Fasten. Given an assembly that has N parts, the assembly sequence is an ordered list of assembly tasks: $\{t_1, t_2 \cdots, t_{N-1}\}$. For any task t_i $(1 \le i \le N-1)$, the IDs (ID) of parts (the part) in the input sub-assemblies s_{i1} and s_{i2} are stored, and the name of the assembly tool used in this assembly task is also attached to each task representation, as shown in Figure 3-3. In ARIMADP, when an assembly is completed, the system stores the assembly sequence for further assembly evaluation.

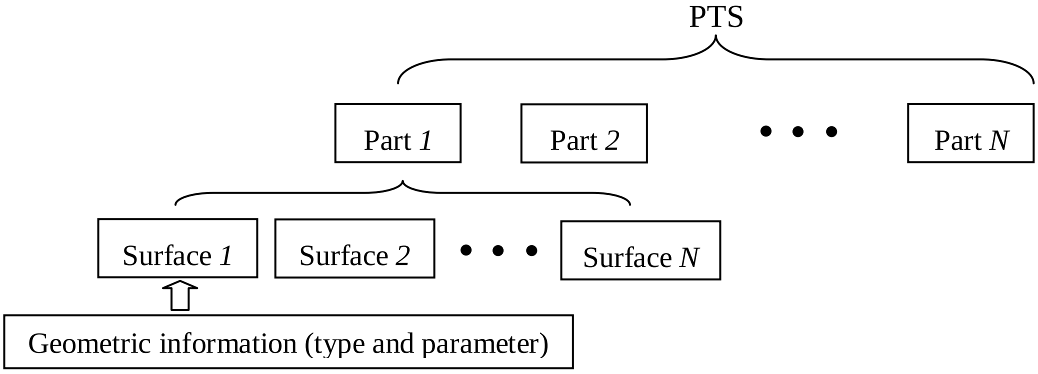


Figure 3-2 The part tree structure (PTS)

The information of the completed assembly model is stored in the third layer using an assembly tree structure (ATS). The root nodes of ATS consist of the IDs of the parts in the part-pairs that are assembled together. Each intermediate node consists of the IDs of surfaces in the surface-pairs of the part-pairs stored in the parent nodes. The leaf nodes contain the mating information of the geometric constraint established between the two surfaces in the surface-pair stored in the parent node. The mating information includes the type of the constraints and the mating direction vectors specified for the mating conditions. Figure 3-4 shows the configuration of ATS. The assembly model can be used for further disassembly and re-assembly planning.

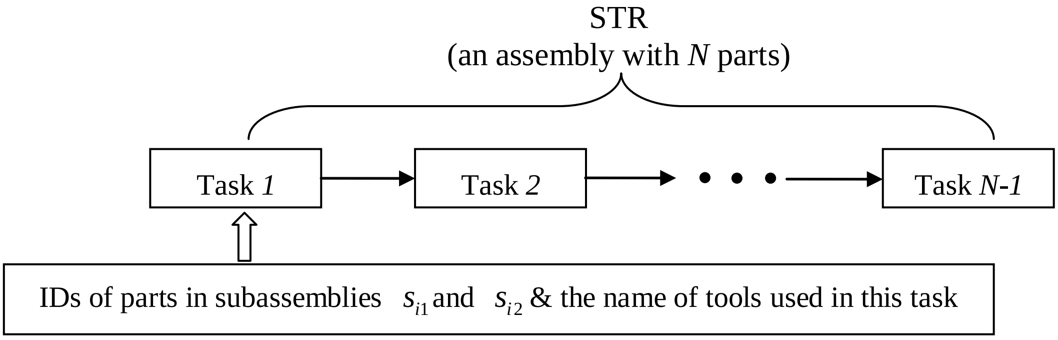

STR
(an assembly with *N* parts)
Task *1*
Task *2*
Task *N-1*
IDs of parts in subassemblies s_{i1} and s_{i2} & the name of tools used in this task

Figure 3-3 The structure of STR

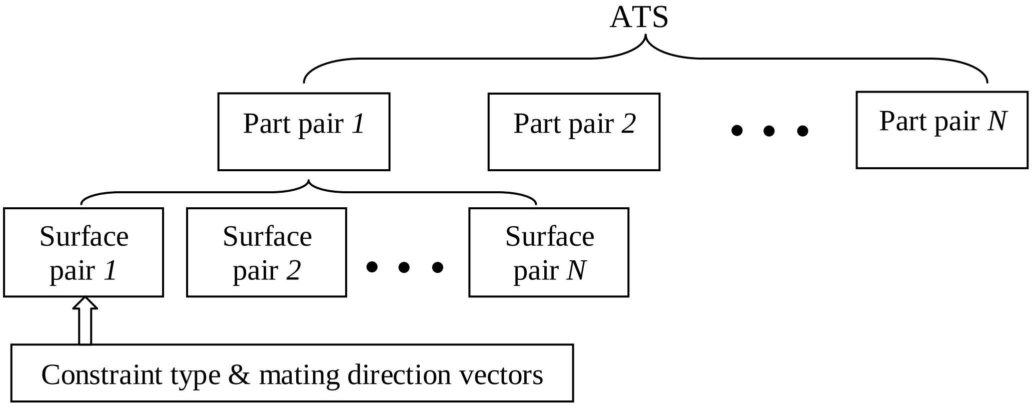


Figure 3-4 The assembly tree structure (ATS)

3.4 Software and Hardware in the System

The ARIMADP system was developed in Visual Studio 2008 as a MFC application. The system applies a few open source application programming interface (API) and libraries, namely OpenGL and OpenCV. OpenGL was used for rendering of virtual objects and information. The camera coordinates can be transformed into a set of screen-coordinates via the OpenGL [OpenGL, 2011] transformation pipeline [Martx, 2006], as shown in Figure 3-5. Some CV functions of OpenCV [OpenCV, 2011] were adopted for image processing in the developing of the 3DNBHI. The details will be discussed in the next chapter.

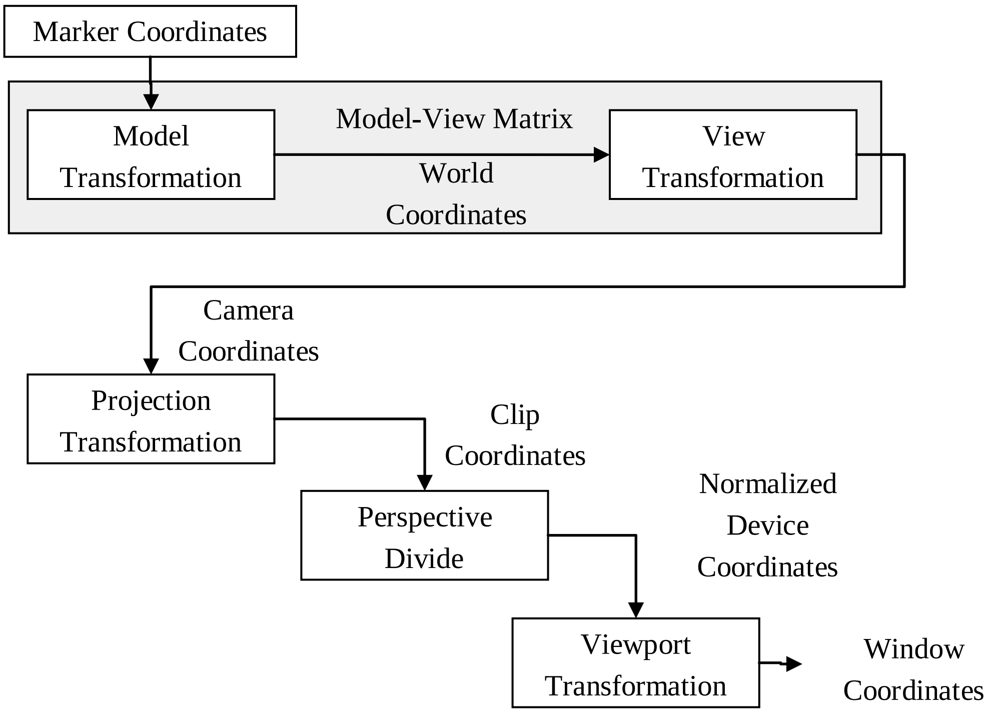


Figure 3-5 The OpenGL transformation pipeline [Martx, 2006]

The ARIMADP system uses a Firefly camera from Point Grey Research® Inc. to stream the augmented scene to the visual output module. The output display used is VUZIX WRAP 920 Video iWear (VUZIX, 2011), which is a HMD with 640×480 display resolution in a 67 inch screen. A BumbleBee2 camera with 640×480 display resolution from Point Grey Research® Inc. is used for depth information retrieval in the interaction module. A DELL desktop (i7-870 2.93 GHZ/ 4GB RAM/ 1GB graphic card) is employed as the computing unit. The operating system is Windows 7 professional.

3.5 Coordinate Systems

In the ARIMADP system, the marker-based registration approach from ARToolKitPlus is used. Using the marker-based method, a known-size square marker is prepared and positioned in the real assembly environment. The configuration of the prototype system is discussed in Chapter 6. Four coordinate systems are needed to set up the ARIMADP environment, *i.e.*, the world coordinate system (WCS) based on the “Hiro” marker, the Firefly camera coordinate system (FCCS), the Bumblebee2 camera coordinate system (BCCS) and the hand coordinate system (HCS). To avoid ambiguities, the definitions of these coordinate systems are presented in this section. The details of the HCS will be discussed in the next chapter. The relationship between the WCS, the FCCS and the BCCS is shown in Figure 3-6.

The WCS is attached to the marker. As shown in Figure 3-6, the origin of the WCS is the centre of the marker, X and Y-axes are the directions of two different parallel sides respectively and Z-axis is the cross product of X and Y-axes. Coordinates in the WCS are denoted as (X_W, Y_W, Z_W).

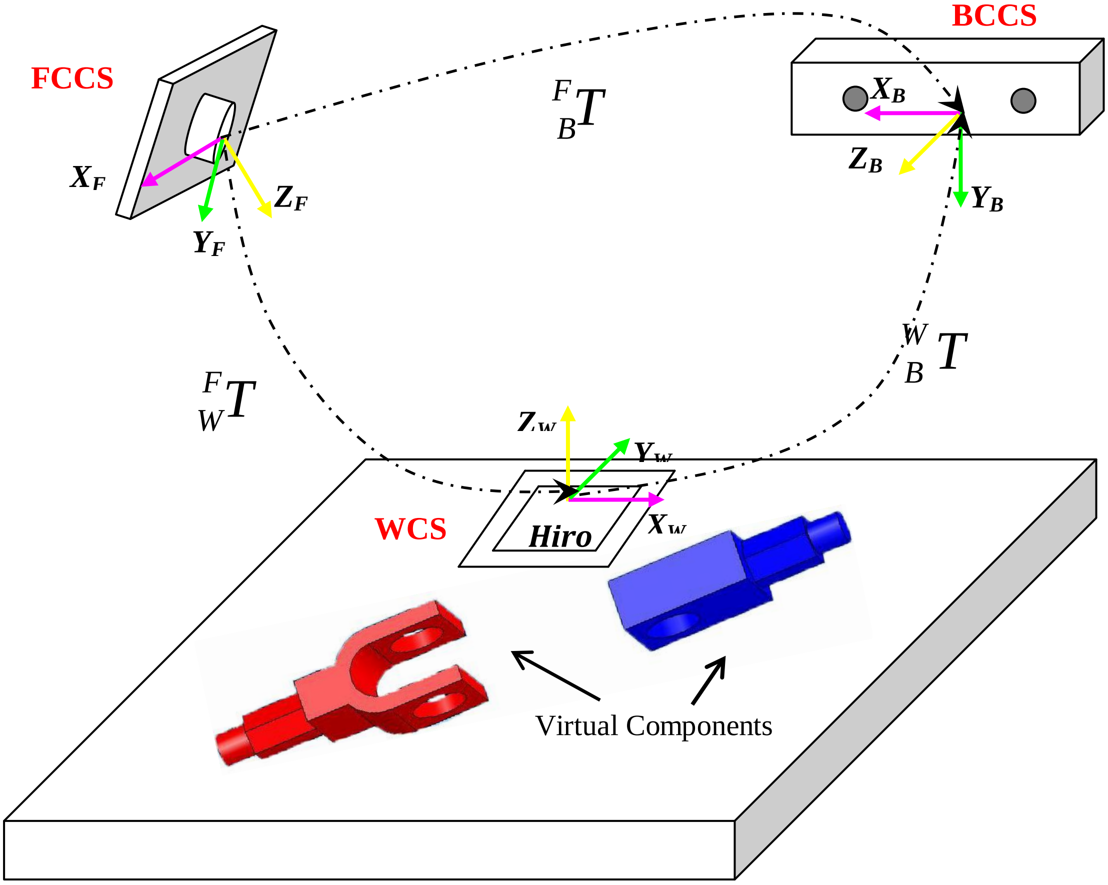


Figure 3-6 Transformation of the coordinate systems in ARIMADP

The FCCS and BCCS are attached to the Firefly camera and the BumbleBee2 camera respectively. The Z-axis of the FCCS is the optical axis of the Firefly camera. Coordinates in the FCCS are denoted as (X_F, Y_F, Z_F). The Z-axis of the BCCS is the optical axis of the BumbleBee2 camera. Coordinates in the BCCS are denoted as (X_B, Y_B, Z_B).

The image coordinates of the four vertices of the square marker in FCCS and BCCS are tracked using the marker-based tracking technique provided by ARToolKitPlus. The camera pose is estimated using the method presented by Scheweighofer and Pinz [Scheweighofer and Pinz, 2006] and the image coordinates and the world coordinates of square vertices. Next, the transformation matrix ${}_W^F T$ between the WCS and the FCCS and ${}_W^B T$ between the WCS and the BCCS can be obtained. The camera coordinates are then transformed into a set of screen-coordinates via the OpenGL transformation pipeline. The OpenGL transformation pipeline transforms vertices in WCS into FCCS and BCCS using linear algebra.

3.6 Summary

Existing AR-based assembly systems have been developed mainly to display the assembly information to guide the assembly operations. In these systems, the assembly design process has been completed in advance and predefined assembly information is imported from CAD software and used to guide the assembly operations. The ARIMADP system is an AR-assisted system to perform manual assembly design and planning using a natural bare-hand interaction tool. In this chapter, the overall system architecture has been described. The data structure forming the basis of the system has been introduced. The software and hardware adopted and used in this system have been discussed. The coordinate systems to set up the AR-based environment have been presented.

Chapter 4 Dual-hand Interface based on Bare-hand Interaction

4.1 Introduction

In this chapter, a 3D bare-hand interaction tool is described. By using an efficient colour segmentation algorithm, the hand region segmentation from input video stream is adaptable to different lighting conditions and complex background. The fingertip features are detected and tracked in real time. A dual-hand interaction interface is developed to allow users to manipulate and orientate virtual objects simultaneously.

4.2 3D Bare-hand Interaction Methodology

As discussed in the literature review and the system description chapters, in order to use human hand as a robust interaction tool in an AR environment, the first step is to obtain the hand region from the input video frame using a robust segmentation algorithm that is adaptable to different lighting conditions and complex background. The hand features, which are fingertips, are then extracted in the next step. The 3D information of these features is retrieved using a stereo camera and the 3D information is used for the 3D interaction which is a 3D pinch operation in this research between the bare hand and the virtual object. The hands differentiation and fingertips differentiation algorithms are implemented to realize the dual-hand interface. The flowchart of the algorithm for 3D bare-hand interaction is shown in Figure 4-1.

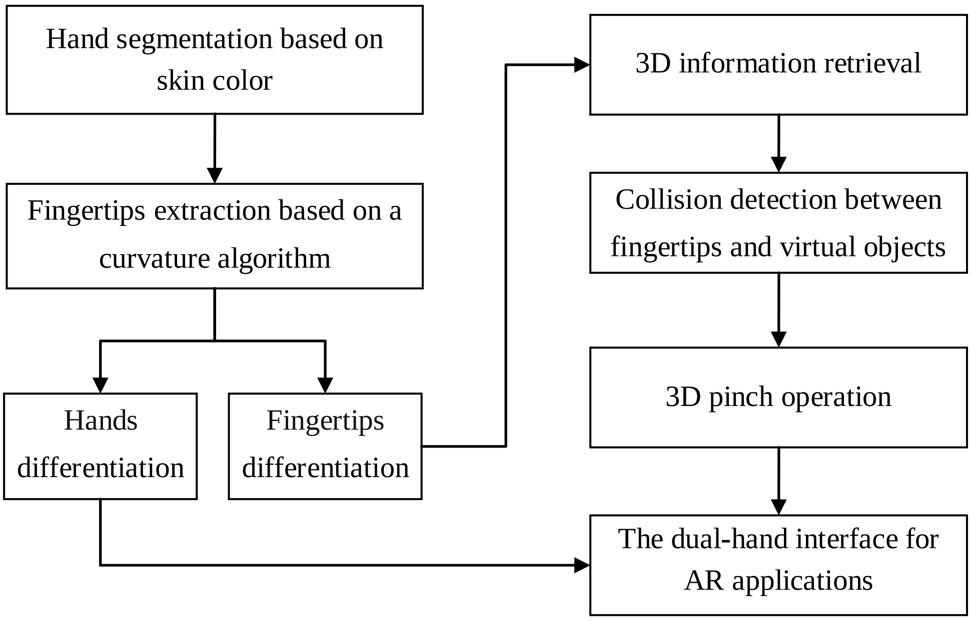

Hand segmentation based on skin color
Fingertips extraction based on a curvature algorithm
Hands differentiation
Fingertips differentiation
3D information retrieval
Collision detection between fingertips and virtual objects
3D pinch operation
The dual-hand interface for AR applications

Figure 4-1 Flowchart of the bare-hand interaction methodology

4.3 Hand Segmentation

Human skin detection has been widely used in hand segmentation. Background image subtraction and skin colour segmentation are typically used to extract the image regions corresponding to the human skin. Background subtraction methods are not suitable for AR applications because the user is moving and the background is changing continuously. Skin colour segmentation approaches are widely used in algorithms for hand and face detections currently. These methods are very effective because they usually involve a small amount of computation, and the skin colour is relatively stable and distinguishable from most of the background colours.

Most existing skin-color segmentation methods can be sorted into two categories, namely, parametric methods [Greenspan et al, 2001; Menser and Wien, 2000] and non-parametric methods [Chari and Bouzerdoum, 2000]. However, it is very difficult to specify the parameters of a skin colour distribution when using parametric methods because they can vary significantly among different people under various lighting conditions, such that a large memory space is needed.

4.3.1 Hand Segmentation using a Bayesian Approach

The YC_bC_r colour space is defined in response to increasing demands for digital approaches in handling video information, and has since become a widely used model in digital video [Chari and Bouzerdoum, 2000]. Y represents luminance, and C_b and C_r are the blue-difference and red-difference chrominance. YC_bC_r separates luminance from chrominance in RGB values using Equation (4.1) [Chari and Bouzerdoum, 2000], which consists of a weighted sum of the three components.

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.1}$$

Previous study [Chai and Ngan, 1999] has found that pixels belonging to a skin region exhibit similar Cb and Cr values. Furthermore, it has been shown that skin colour model based on Cb and Cr values can provide good coverage of all human races. Figure 4-2 shows the result of skin colour classification using a Bayesian method [Chari and Bouzerdoum, 2000] in the YCbCr colour space. It is found that the detection result is not good enough and the computational cost is high when using this non-parametric method.

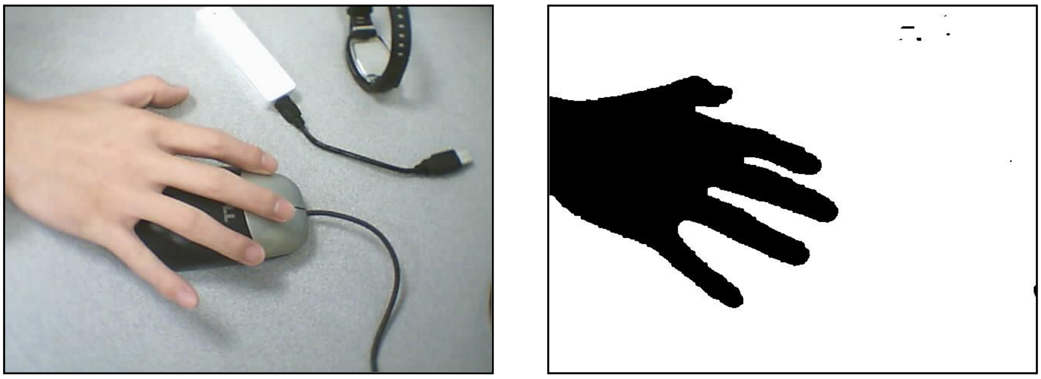


Figure 4-2 Skin colour classification using a Bayesian approach

4.3.2 Hand Segmentation using a Neural Network

The Restricted Coulomb Energy (RCE) neural network is a simple but effective colour classification technique. The RCE neural network is known to be a specific design of the hyper-spherical classifier that can serve as a general adaptive pattern classification engine [Yin et al, 2001; Yuan et al, 2008]. A RCE neural network is used for segmenting the hand from the background image. The RCE network consists of three layers of neuron cells, with a full set of connections between the first and second layers, and a partial set of connections between the second and third layers, as shown in Figure 4-3. The L*a*b* colour space, which is defined by the CIE (Commission International de l'Éclairage) in 1976, is selected to detect the skin colour. L* represents luminance and a* and b* are the channels of chrominance. Three cells in the input layer contain the colour values, namely, L*, a* and b*. The cells in the middle layer are called the prototype cells. Each prototype cell contains colour information, *i.e.*, a learned colour class in the training data, and each cell in the output layer corresponds to a different colour class presented in the training data set.

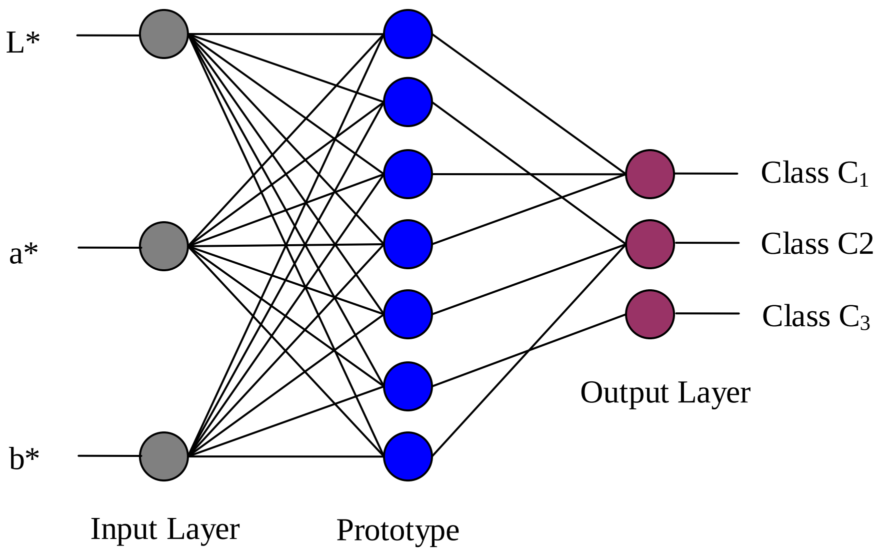


Figure 4-3 Architecture of a RCE neural network

The advantage of using the L*a*b* colour space is that the skin colour distributions in this colour space under different lighting conditions are lowly correlated and translated along the “L*” axis with the change of lighting conditions [Yin et al, 2001]. The values of the L*, a* and b* can be computed using the Equations (4.2) and (4.3) [Kasson and Plouffe, 1992].

$$L^* = \begin{cases} 166\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 & if\ \frac{Y}{Y_n} > 0.008856 \\ 903.3\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 & if\ \frac{Y}{Y_n} \le 0.008856 \end{cases}$$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \tag{4.2}$$

$$b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right]$$

where

$$f(t) = \begin{cases} t^{\frac{1}{3}} & if\ t > 0.008856 \\ 7.787t + \frac{16}{116} & if\ t \le 0.008856 \end{cases} \tag{4.3}$$

X_n, Y_n and Z_n are the tristimulus values of a perfect reflecting diffuser. In the proposed system, they are selected to be 250.410, 255.000 and 301.655, respectively. X, Y and Z are the tristimulus values of the specimen, which are obtained from the R, G, B values of each pixel based on the Equation (4.4).

$$\begin{aligned} X &= 0.607R + 0.174G + 0.201B \\ Y &= 0.299R + 0.587G + 0.114B \\ Z &= 0.066G + 1.117B \end{aligned} \tag{4.4}$$

Hand segmentation using the RCE neural network includes two processes, namely, the RCE network training and the hand image segmentation. The flowchart of the hand image segmentation algorithm is shown in Figure 4-4.

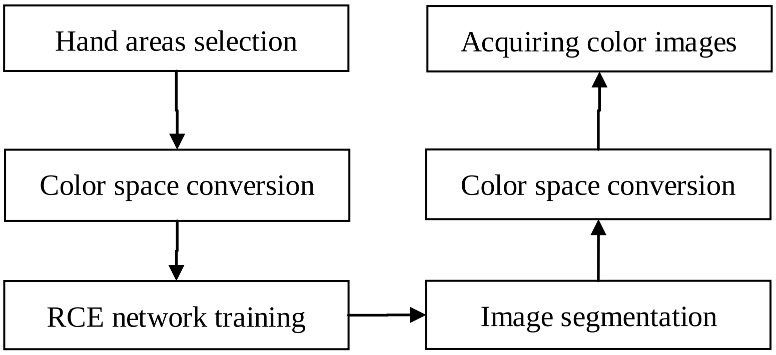


Figure 4-4 Flowchart of hand image segmentation using RCE

For each incoming frame in a live video, every pixel of the frame can be directly categorized to be either a skin-colour pixel or a non-skin-colour pixel. It is assumed that a large portion of the input colour image is the hand region, meaning that no other part of human body will be captured by the camera. The hand contour is extracted using some CV functions of the OpenCV library. The segmentation result, as shown in Figure 4-5, is used for hand tracking.

Using the RCE neural network, the hand segmentation is robust under various lighting conditions. However, the frame rate for hand segmentation using neural network is only about 9 fps. In the next section, a real-time segmentation of the hand regions from the input video stream which is more robust against complex backgrounds has been achieved by using Continuously Adaptive Mean-shift (CamShift) algorithm.

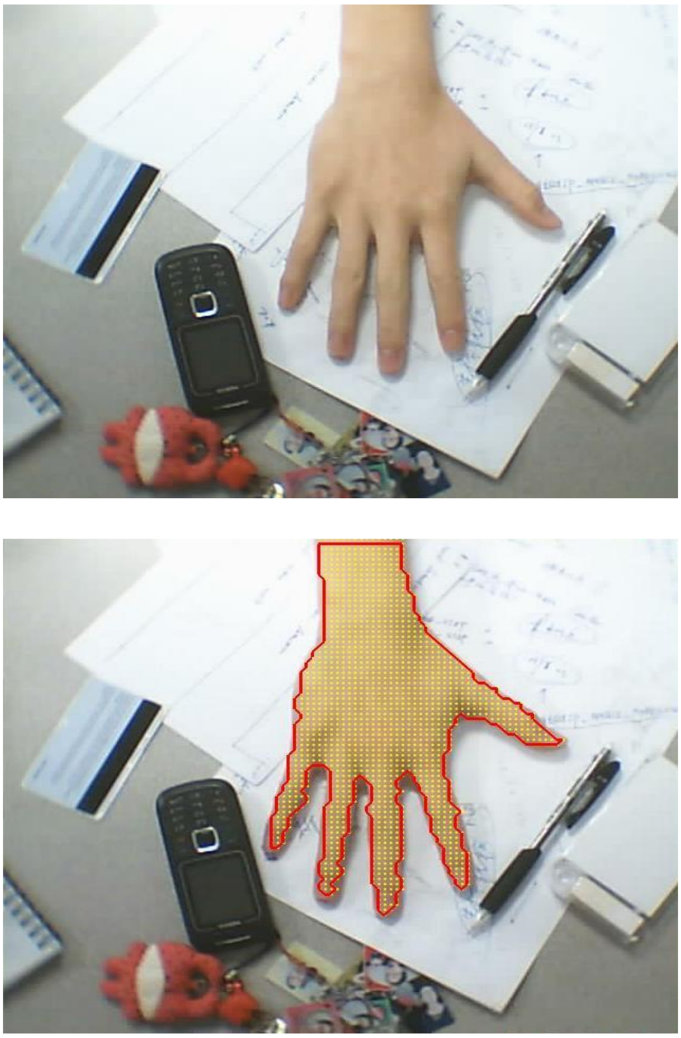


Figure 4-5 Hand segmentation result using RCE

4.3.3 Hand Segmentation using CamShift

The CamShift algorithm is used to track the human hands using a 1D histogram consisting of quantized channels from the Hue Saturation Value (HSV) colour space. HSV, developed in 1970s, is one of the most common cylindrical-coordinate representations of points in an RGB colour model, which rearranges the geometry of RGB in an attempt to be more intuitive and perceptually relevant than the Cartesian representation. The values of H, S and V are calculated according to the pseudo codes shown in Figure 4-6.

```
//r = [0,1], g = [0,1], b = [0,1]
//h = [0, 360], s = [0,1], v = [0,1]

    r = r/255;
    g = g/255;
    b = b/255;

The value of v is equal to the maximum value of r,g and b.
    v = max(r, g, b);
The value of s is calculated using the following equation.
    s = (max(r, g, b) – min(r, g, b))/v;
The value of h is calculated using the following equation.
    If (r is equal to max(r, g, b))
        h = (g - b)/ (max(r, g, b) – min(r, g, b));
    else if (g is equal to max(r, g, b))
        h = 2 + (b - r)/ (max(r, g, b) – min(r, g, b));
    else
        h = 4 + (r - g)/ (max(r, g, b) – min(r, g, b));

    h = h*60;

    if (h is smaller than zero)
        h = h+ 360;
```

Figure 4-6 Pseudo codes of the transformation from RGB to HSV

Given a probability density image, the CamShift algorithm can find the mean of the distribution by iterating in the direction of the maximum increase in the probability density through adapting the Mean Shift algorithm [Bradski, 1998]. The HSV colour space with the hues separated out from the saturation and the intensity is used to create a model of the desired hue using a colour histogram, as all humans (except albinos) have basically the same hue. This discrete hue model is then used to segment the hand region from the input video stream.

In the proposed methodology, a region of interest (ROI) on the hand has to be selected by the user at the beginning of the process. Next, the hues derived from the skin pixels in the ROI are sampled from the H channel and binned into a 1D histogram. The training results, which include information of the skin colour distribution, the histogram and the ROI, are stored in a text file. This text file will be reloaded automatically the next time when the same user is using the system. Otherwise, a new text file will be generated for a new user. In this way, the training process does not have to be repeated for the same user. For each frame of the input video stream, the stored skin colour histogram is used to categorize every pixel of the frame to be either a skin-color pixel or a non-skin-color pixel. It is assumed that a large portion of the input colour image is the hand region, meaning that no other part of the human body will be captured by the camera. Figure 4-7 shows the flowchart of the hand segmentation algorithm.

The hand contours are also detected and extracted using the OpenCV library. To remove small noise pixels, the connected component, which consists of a number of connected pixels, with the largest perimeter is selected as the hand contour. Figure 4-8 shows the hand segmentation result.

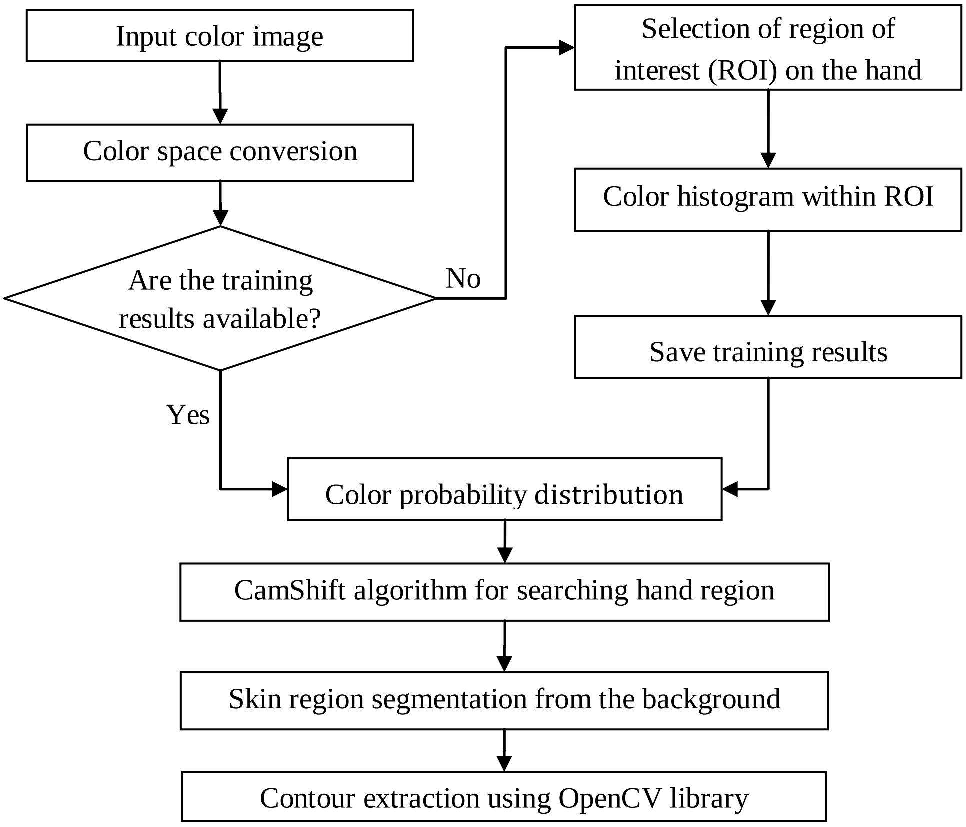

Input color image
Color space conversion
Are the training results available?
No
Yes
Selection of region of interest (ROI) on the hand
Color histogram within ROI
Save training results
Color probability distribution
CamShift algorithm for searching hand region
Skin region segmentation from the background
Contour extraction using OpenCV library

Figure 4-7 Flowchart of the algorithm of hand segmentation using CamShift

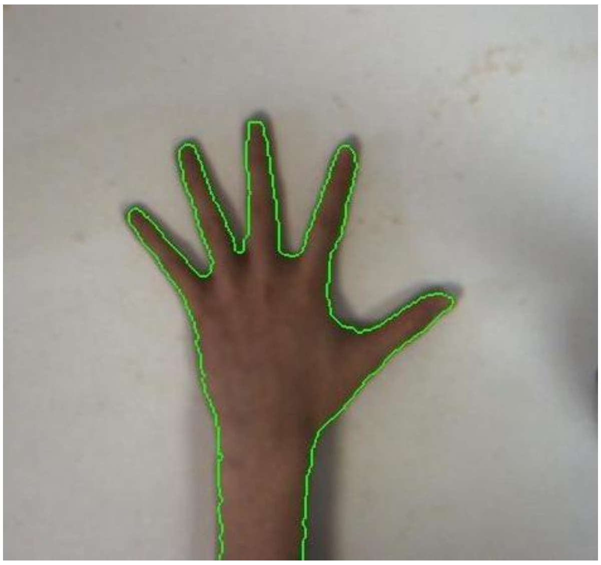


Figure 4-8 Hand segmentation result using CamShift

4.4 Hand Feature Extraction

In the proposed methodology, the fingertips are detected from the hand contour using a curvature-based algorithm [Segen and Kumar, 1998]. The curvature of a contour point is measured through computing the dot product of $\overrightarrow{P_iP_{i-l}}$ and $\overrightarrow{P_iP_{i+l}}$ according to Equation (4.5), where P_i is the i th point in the hand contour, P_{i-l} and P_{i+l} are the preceding and following points respectively, and l is the point index on the hand contour. In the proposed system, l is set to 15. Figure 4-9 shows the vectors used in Equation (4.5).

$$C_l(P_i) = \frac{\overrightarrow{P_iP_{i-l}} \cdot \overrightarrow{P_iP_{i+l}}}{\left\|\overrightarrow{P_iP_{i-l}}\right\| \left\|\overrightarrow{P_iP_{i+l}}\right\|} \qquad (4.5)$$

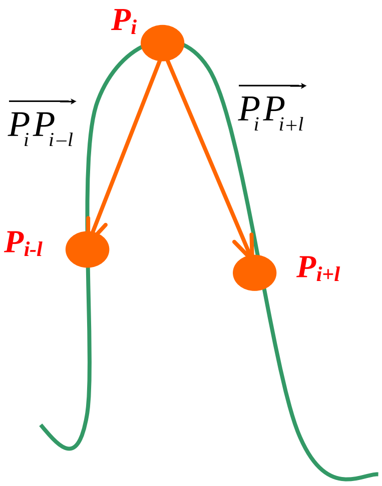


Figure 4-9 Vectors used in Equation (4.5)

The points with curvature values higher than a threshold, which is set to 0.5 in this research, are selected as candidates for the fingertips. Next, all the candidates are separated into different candidate groups. In addition, the directions indicated by the cross product of the two vectors in Figure 4-9 are used to determine whether a point is a fingertip point or a valley point between the fingertips. An ellipse is fitted to the hand contour using the least-square fitting method, and the centre point of the hand is specified as the centre of the ellipse. For each candidate group, the distance between each candidate and the centre point of the hand region is computed and the point with the longest distance from the hand centre will be specified as the fingertip. Figure 4-10 shows the fingertip extraction result.

The hand segmentation and features extraction processes are sufficiently fast. Thus, the tracking of the hands and features can be achieved by executing the detection and extraction processed for each frame of the input video stream.

4.5 Hands Differentiation

At the start of the process, the user needs to put his/her one hand or two hands in the camera's view with the palm facing down to make sure all the five fingertips can be detected. For each hand, the tip of the thumb P_{th} is determined as the farthest fingertip from the mean position P_m of all the five fingertips. The mean position is calculated through averaging the positions of all the five fingertips, as shown in Equation (4.6).

$$\begin{cases} x_m = \frac{1}{5}\sum_{i=0}^{5} x_f(i) \\ y_m = \frac{1}{5}\sum_{i=0}^{5} y_f(i) \end{cases} \tag{4.6}$$

where (x_m, y_m) is the 2D coordinate of the mean position P_m and $(x_f(i), y_f(i))$ is the 2D coordinate of the i th fingertip.

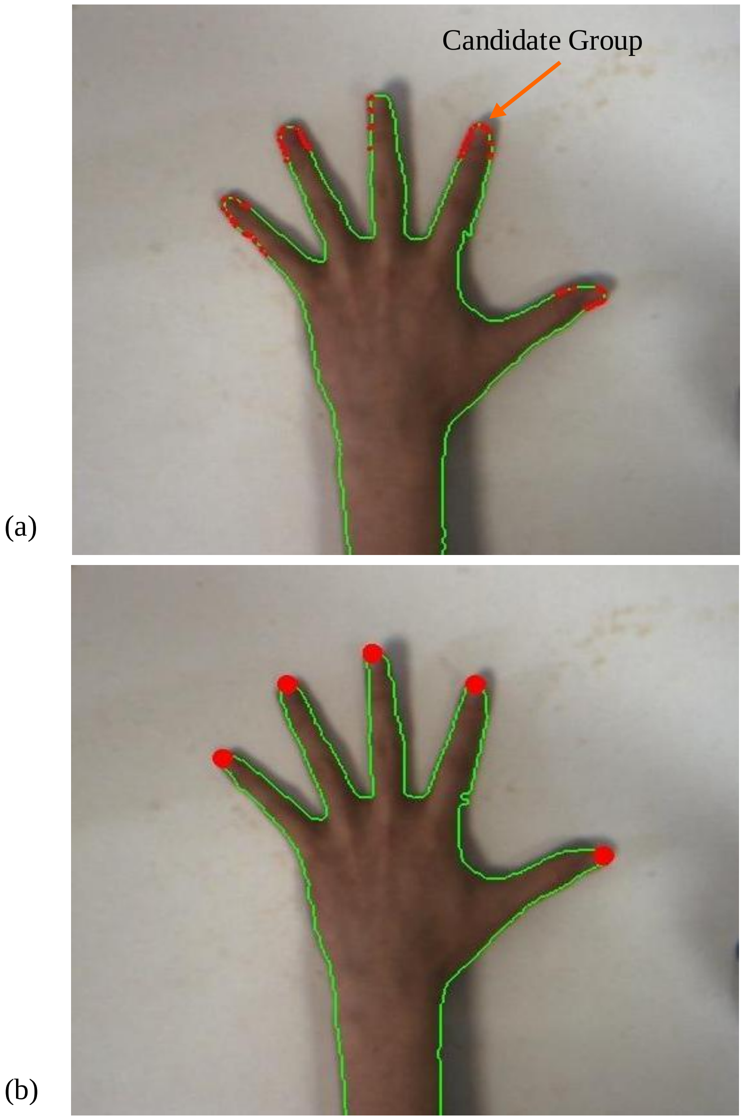


Figure 4-10 Fingertips extraction: (a) candidate group; (b) specified fingertips

A cross product $\overrightarrow{D_H}$ of $\overrightarrow{P_{HC}P_{th}}$ and $\overrightarrow{P_{HC}P_{th}}$ is calculated according to Equation (4.7), where P_{HC} is the centre of the hand and P_{th} is the tip of thumb finger. All the coordinates and vectors are calculated in the image coordinates system, as shown in Figure 4-11.

$$\overrightarrow{D_H} = \frac{\overrightarrow{P_{HC}P_{th}} \times \overrightarrow{P_{HC}P_m}}{\left\|\overrightarrow{P_{HC}P_{th}}\right\| \left\|\overrightarrow{P_{HC}P_m}\right\|} \tag{4.7}$$

Next, the user's hand can be differentiated by checking the direction of $\overrightarrow{D_H}$. The vector $\overrightarrow{D_H}$ can be represented in Equation (4.8).

$$\overrightarrow{D_H} = H_x\vec{i} + H_y\vec{j} + H_z\vec{k} \tag{4.8}$$

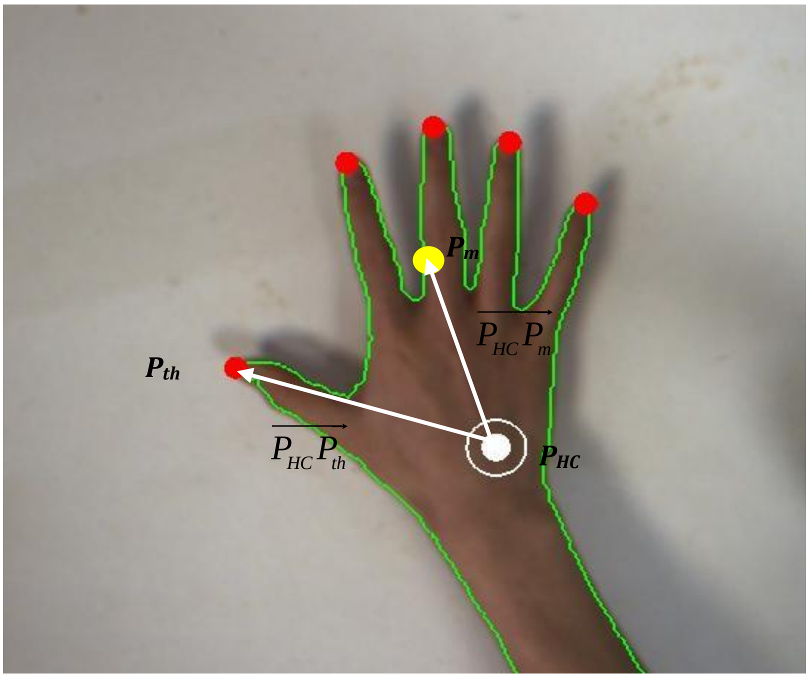


Figure 4-11 Hands differentiation

The system can determine the number of hands in the camera's view. If only one hand is used, this hand would be specified either as the "right hand" or "left hand" and this specification will not change until this hand is moved out of the camera's view. If both of the user's hands are used concurrently, the two hand centres are tracked using a matching algorithm that minimizes the displacement between the centres of the pair of hands over two successive frames after the hand differentiation process. Therefore, these two hands can always be differentiated in each frame. This proposed method cannot handle hands with palms facing up or the crossing of two hands. When the user crosses his/her two hands, only one hand will be detected by the system. Palms facing up may cause system errors. For instance, two "right hands" will be recognized by the system when the user put his left hand (palm up) and right hand (palm down) in the view of the camera.

4.6 Fingertips Differentiation

After the user's hands have been recognized and differentiated, the system differentiates the thumb and index fingertips automatically. Similar to the hands differentiation process, a cross product $\overrightarrow{D(P_i)}$ of $\overrightarrow{P_{HC}P_i}$ and $\overrightarrow{P_{HC}P_{LC}}$ is calculated according to Equation (4.9).

$$\overrightarrow{D(P_i)} = \frac{\overrightarrow{P_{HC}P_i} \times \overrightarrow{P_{HC}P_{LC}}}{\left\|\overrightarrow{P_{HC}P_i}\right\| \left\|\overrightarrow{P_{HC}P_{LC}}\right\|} \tag{4.9}$$

where P_{HC} is the centre of the hand, P_{LC} is the centre of the line segment between two fingertips and P_i is the i th fingertip. All the coordinates and vectors are calculated in the image coordinates system. Figure 4-12 shows the vectors used in Equation (4.9).

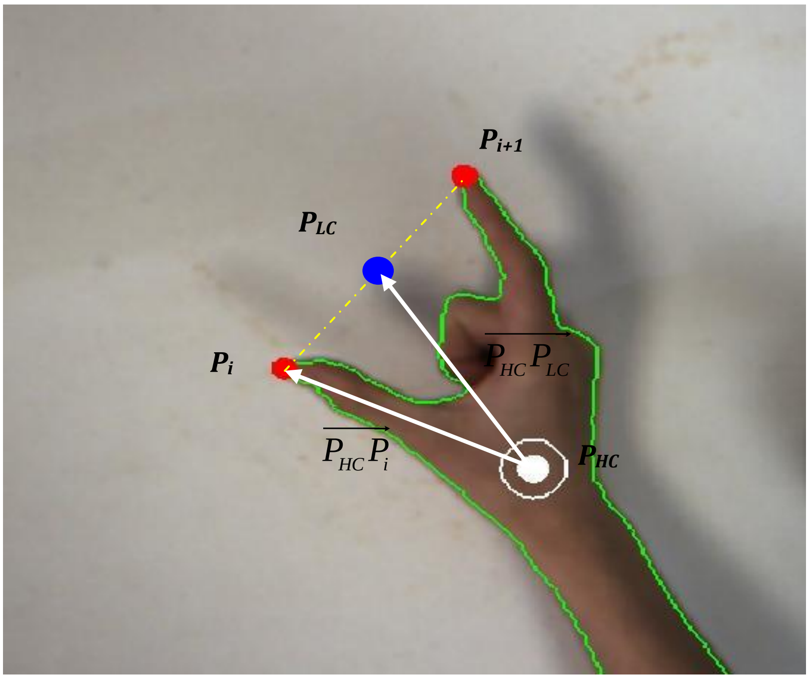


Figure 4-12 Fingertips Differentiation

By checking the direction of $\overrightarrow{D(P_i)}$, the thumb and the index fingertip can be differentiated for both right and left hands. The vector $\overrightarrow{D(P_i)}$ can be represented in Equation (4.10).

$$\overrightarrow{D(P_i)} = f(i)_x \vec{i} + f(i)_y \vec{j} + f(i)_z \vec{k} \tag{4.10}$$

The differentiation result is shown in Figure 4-13. The details of the differentiation algorithm are illustrated in Figure 4-14.

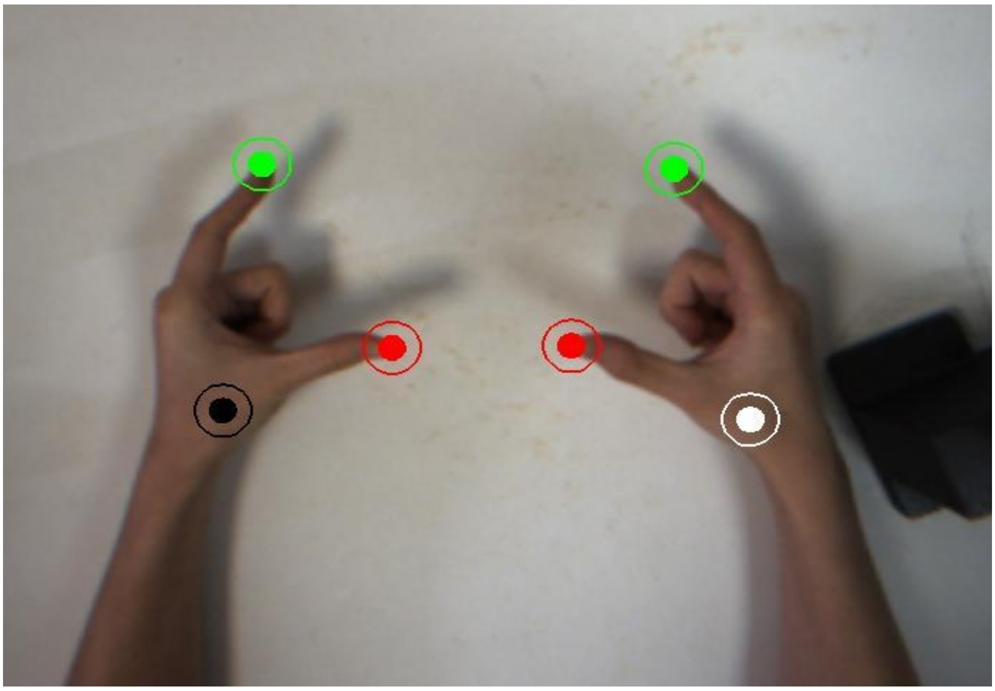
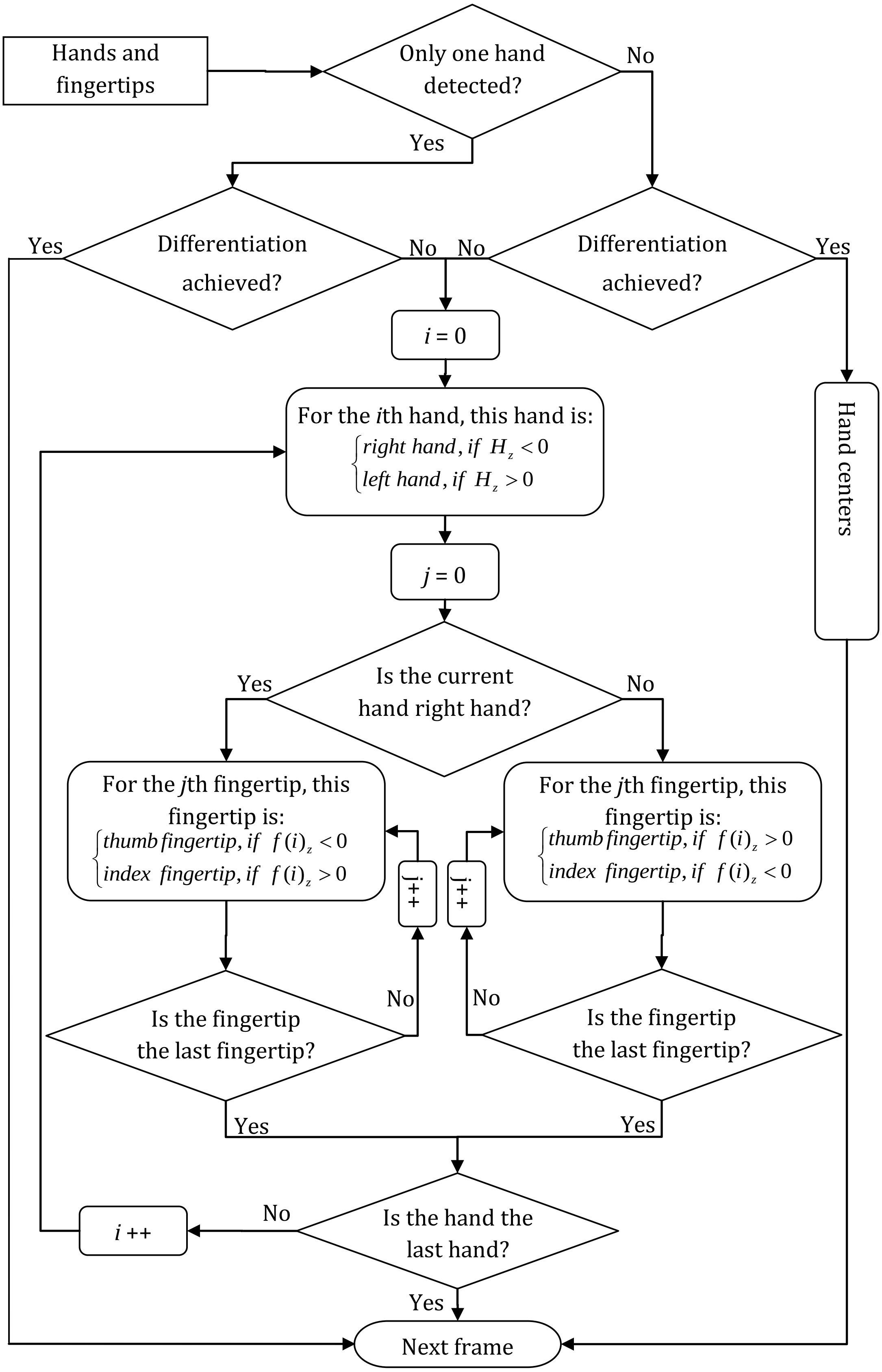


Figure 4-13 Differentiation: right hand with white centre point and left hand with black centre point; thumb tips are in red colour and index fingertips are in green colour



Hands and fingertips
Only one hand detected?
No
Yes
Yes
Differentiation achieved?
No
No
Differentiation achieved?
Yes
$i = 0$
For the ith hand, this hand is:
$\begin{cases} right\ hand, if\ H_z < 0 \\ left\ hand, if\ H_z > 0 \end{cases}$
Hand centers
$j = 0$
Yes
Is the current hand right hand?
No
For the jth fingertip, this fingertip is:
$\begin{cases} thumb\ fingertip, if\ f(i)_z < 0 \\ index\ fingertip, if\ f(i)_z > 0 \end{cases}$
j++
j++
For the jth fingertip, this fingertip is:
$\begin{cases} thumb\ fingertip, if\ f(i)_z > 0 \\ index\ fingertip, if\ f(i)_z < 0 \end{cases}$
No
No
Is the fingertip the last fingertip?
Is the fingertip the last fingertip?
Yes
Yes
No
i ++
Is the hand the last hand?
Yes
Next frame

Figure 4-14 Flowchart of the hand and fingertip differentiation algorithm

4.7 Correct Occlusion between fingers and virtual objects

Depth buffering, also known as z-buffering, is the solution for hidden surface elimination in OpenGL. Depth buffering provides a promising approach for solving AR occlusion problems. In the proposed system, depth buffering is adopted to achieve the correct occlusion between the fingers and virtual objects.

4.7.1 Raw data processing

Depth buffer is a two-dimensional array that always keeps record of the closest depth value to the viewer of each pixel. Depth buffer value is not the actual distance between the object and the viewpoint but the distance after a projection transformation, division and normalization to the range [0, 1]. Therefore, the distance from the virtual and real objects to the viewpoint cannot be compared directly since they are represented in different forms.

In an OpenGL pipeline, the transformation between the 3D scene and a 2D image is described as a projection matrix *M*, which is used for converting from the eye coordinates Z_e (real distance from vertices to the viewer) to the clip coordinates Z_c (distance in the clip space where objects outside the view volume are clipped away). *M* is obtained using the ARToolKitPlus library in the proposed system. *M* can be represented in Equation (4.11).

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \tag{4.11}$$

Next, the Z_c is transformed to the normalized device coordinates Z_n by division with the homogenous component in the clip space. Since the range of Z_n is [-1, 1], it needs to be offset and scaled before it is sent to the depth buffer with range [0, 1]. The depth value of a pixel in the depth buffer Z_d can be obtained from Equation (4.12) [Martx, 2006].

$$z_d = -0.5 * (\frac{m_{33} * z_e + m_{34}}{z_e}) + 0.5 \tag{4.12}$$

In Equation (4.12), z_e is the real depth value in the eye coordinate system. z_e is obtained by using stereo vision technologies which will be described in Appendix A. In order to achieve correct occlusion between the real hand and the virtual object, all the depth values are written into the depth buffer before display. If the size of a 2D array containing the depth values is too large, the process of writing this array into the depth buffer will be slow. In the proposed system, only the occlusion between the fingers and the virtual objects is handled as the pinch operation.

4.7.2 Occlusion handling

Inspired by the occlusion detection system [Hayashi et al, 2005], a contour-based approach is adopted to achieve correct occlusion between the user' hands and the virtual objects. The correct occlusion result is shown in Figure (4-16). The proposed system can handle the occlusion issue according to following procedures.

1) Segment fingers from the hand region. The convexity defect points along the hand contour are extracted, as shown in Figure (4-15). The contour between the convexity defect point and the fingertip is divided into segments of equal length for each finger. Next, the contour of another side of each finger is also sampled into segments of the same length.
2) The real depth value z_e of each segmentation point is retrieved using stereo vision technologies.
3) Each segment on the contour is sampled, as shown in Figure (4-15). Interpolate the depth of the sampled contour points by the depth values of the neighbouring segmentation points.
4) Two matching sampled contour points on both sides of each finger are linked as an inner segment. The inner segment is sampled, as shown in Figure (4-15). The depth value of each sampled inner point is obtained by interpolating the depth values of the matching sampled contour points.
5) All the real depth values are sent to the depth buffer after they have been converted into the depth buffer value using Equation (4.12).

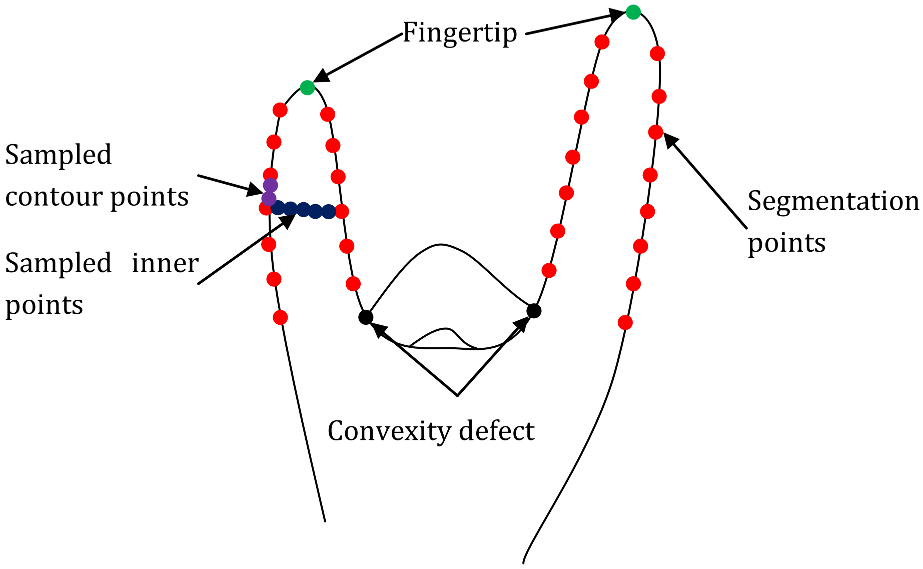


Figure 4-15 Contour segmentation and depth interpolation

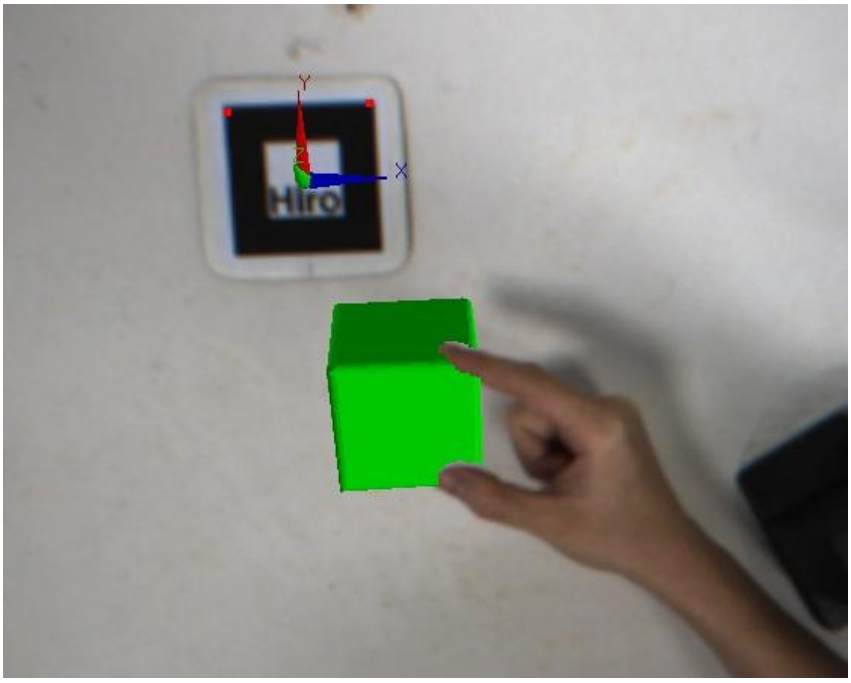


Figure 4-16 Correct occlusion between finger and virtual object

4.8 3D Bare-hand Interaction

In the proposed bare-hand interaction interface, a marker-based tracking method is used to detect the camera pose and render the virtual objects onto the real scene in an AR environment. The ARToolKitPlus library is used for marker-based tracking. The tips of the thumbs and index fingers are used to achieve a 3D pinch operation. Using stereo vision technologies, the 3D information of the fingertips and a fiducial marker can be obtained easily. A BumbleBee2 stereo camera is used for hand region detection and depth information calculation. The 3D stereo vision technologies and the application of BumbleBee2 camera in depth calculation can be found in Appendix A.

4.8.1 Coordinate Systems

During a 3D pinch operation process, two coordinate systems, *i.e.*, the WCS based on the fiducial marker and the hand coordinate system (HCS) established on the hand.

4.8.1.1 Hand Coordinate System

The HCS is established on the 3D triangle that is formed by the thumb tip, index fingertip and the hand centre, as shown in Figure 4-17. In the HCS, the origin is the grasp point, which is the centre of the line segment between the thumb tip and the index fingertip. The X-axis is along the direction of the vector from the thumb tip to the index fingertip. The Z-axis is along the direction of the normal vector of the triangle. Next, the Y-axis is obtained as a cross product of Z-axis and X-axis. The coordinates in the HCS are denoted as (X_H, Y_H, Z_H).

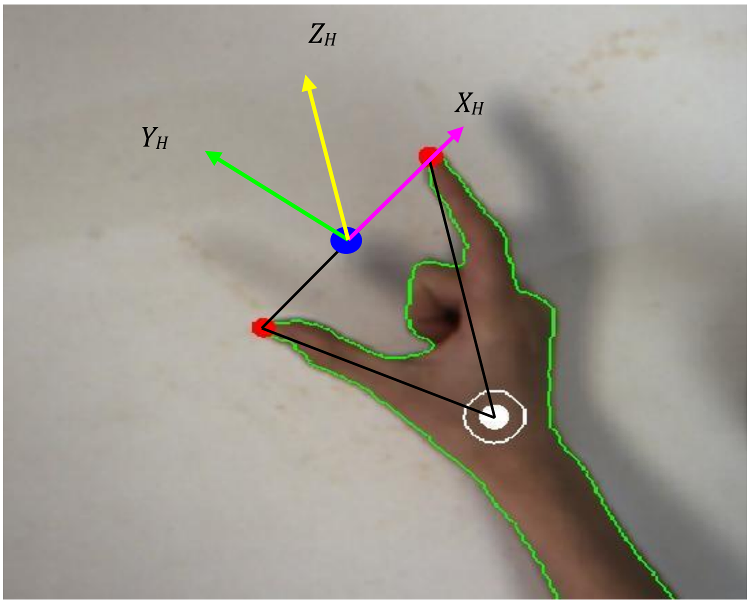


Figure 4-17 The hand coordinate system

4.8.1.2 Transformation between WCS and BCCS

The transformation matrix ${}^{W}_{B}T$ between the WCS and the BCCS can be determined so that the 3D information of the real world can be projected easily onto the WCS according to Equation (4.13), where P_W is a point in the WCS and P_B is the same point in the BCCS. The relationship between these two coordinate systems is shown in Figure 4-18.

$$P_W = {}^{W}_{B}T \, P_B \tag{4.13}$$

${}^{W}_{B}T$ is a 4×4 matrix which can be expressed as in Equation (4.14):

$$ {}^{W}_{C}T = \begin{bmatrix} {}^{W}_{C}r & {}^{W}_{C}t \\ 0 & 1 \end{bmatrix} \tag{4.14} $$

In this research, ${}^{W}_{C}r$ is obtained from the ARToolKitPlus library and ${}^{W}_{C}t$ is constructed from the 3D information of the centre of the fiducial marker.

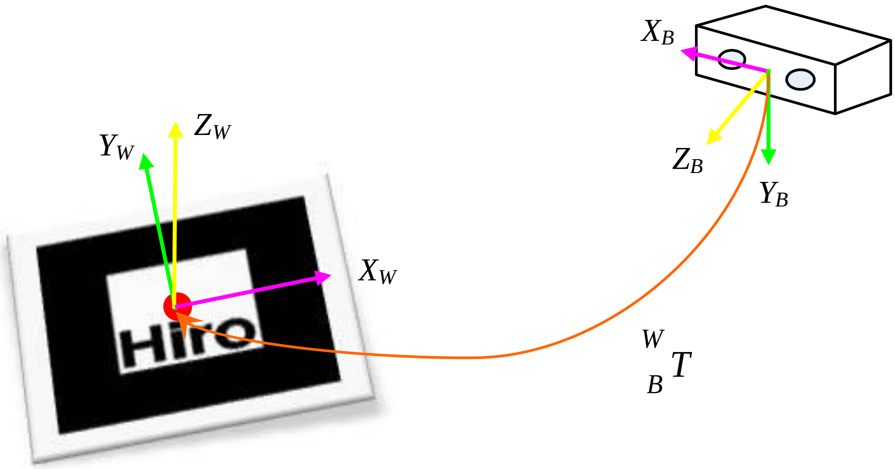


Figure 4-18 Coordinate systems transformation between WCS and BCCS

4.8.1.3 Transformation between WCS and HCS

The transformation matrix ${}^{W}_{H}T$ between the WCS and the HCS is 4×4 a matrix which can be expressed as in Equation (4.15).

$$ {}^{W}_{H}T = \begin{bmatrix} {}^{W}_{H}r & {}^{W}_{H}t \\ 0 & 1 \end{bmatrix} \tag{4.15} $$

${}_{H}^{W}t$ is constructed from the 3D information of the origin of the hand coordinate system. The unit vectors giving the principal directions of the HCS in terms of the WCS are denoted as ${}^{W}X_{H}$, ${}^{W}Y_{H}$ and ${}^{W}Z_{H}$. ${}_{H}^{W}r$ is obtained from Equation (4.16). The relationship between these two coordinate systems is shown in Figure 4-19.

$$ {}_{H}^{W}r = \left({}^{W}X_{H} \quad {}^{W}Y_{H} \quad {}^{W}Z_{H} \right) \tag{4.16} $$

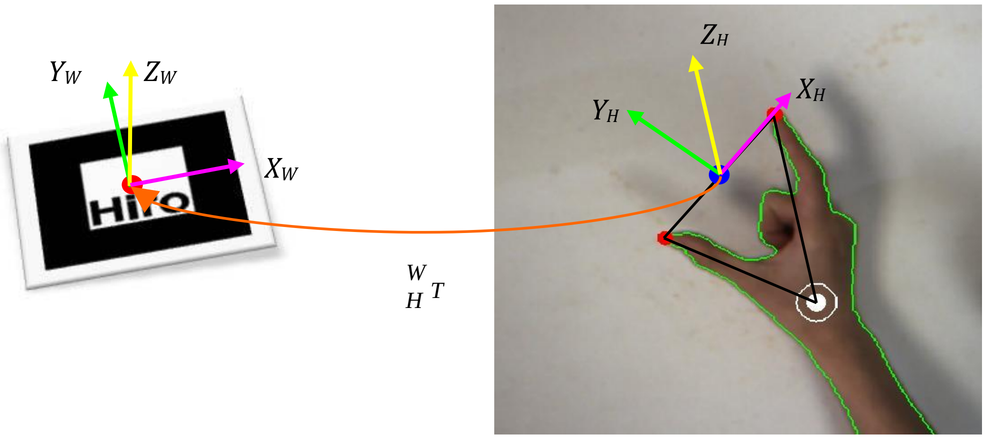


Figure 4-19 Coordinate systems transformation between WCS and HCS

4.8.2 3D Pinch Operation

The transformation matrix of the virtual object in the AR environment is calculated for each frame. The virtual object is being manipulated by the user only when two virtual spheres registered on the fingertips are both in contact with the virtual object. The user can deselect a virtual object by enlarging the distance between two fingertips to make sure that one or two virtual spheres are not in contact with the virtual object. When the virtual object is being manipulated by the user, the spatial relationship between the virtual object and the hand coordinate system is recorded and maintained. Based on the recorded spatial relationship, the transformation matrix of the virtual object manipulated by the user in the current frame is calculated using Equation (4.17).

$$ {}^{W}_{O}T = {}^{W}_{O'}T \, {}^{O'}_{H'}T \, {}^{H'}_{H}T \, {}^{H}_{O}T \tag{4.17} $$

where ${}^{W}_{O}T$ is the transformation matrix of the virtual object in the current frame, ${}^{W}_{O'}T$ is the transformation matrix of the virtual object in the previous frame, ${}^{O'}_{H'}T$ and ${}^{H}_{O}T$ are the transformation matrix between the hand coordinate system and the object coordinate system (OBS), and ${}^{H'}_{H}T$ is the transformation matrix between the hand coordinate systems in these two successive frames.

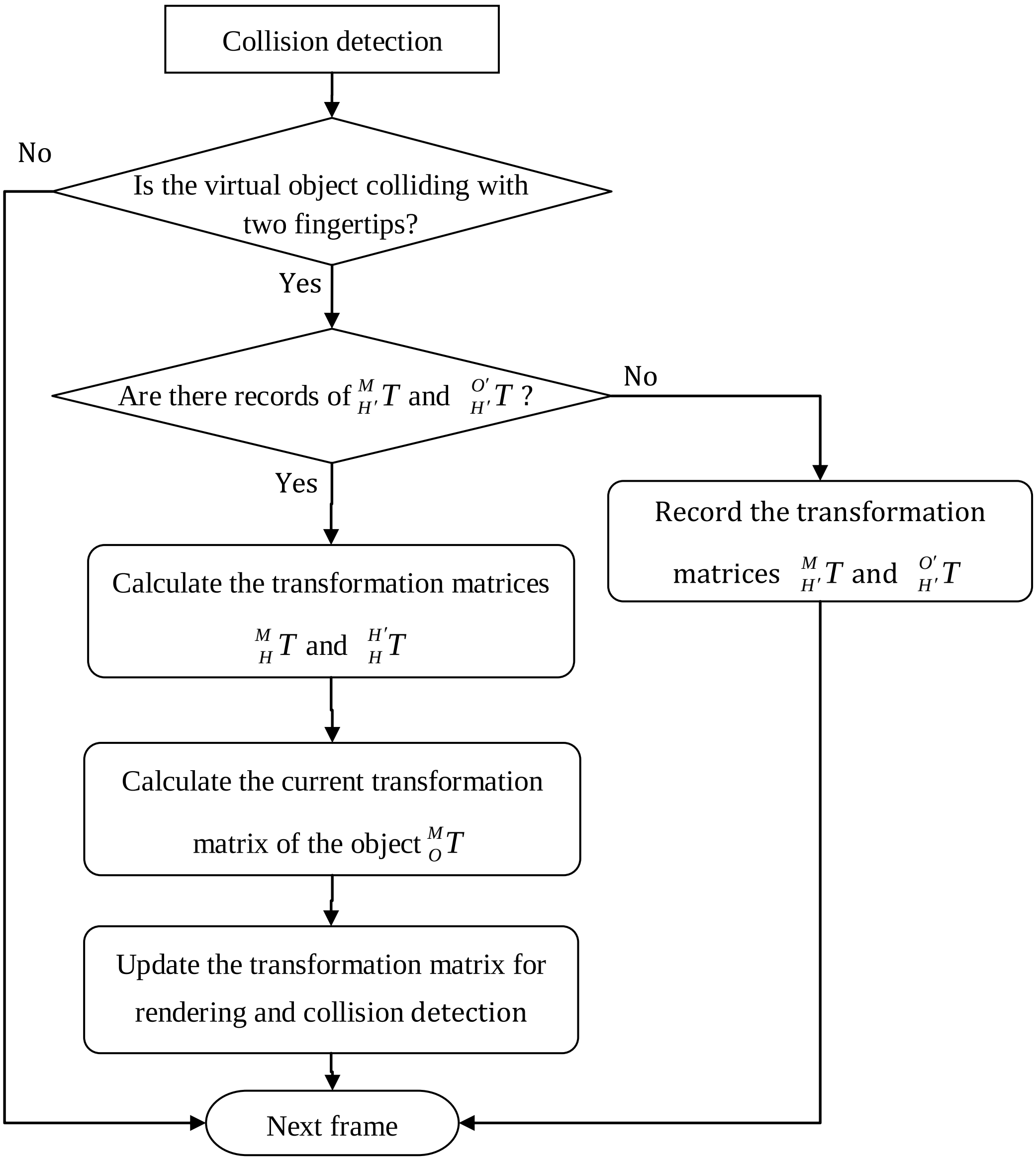
${}^{O'}_{H'}T$ is a 4×4 matrix which contains only translation information and can be expressed as Equation (4.18):

$$ {}^{O'}_{H'}T = \begin{bmatrix} 1 & {}^{O'}_{H'}t \\ 0 & 1 \end{bmatrix} \tag{4.18} $$

${}^{O'}_{H'}t$ is constructed from the distance information between the grasp point and the centre of the virtual object. ${}^{H}_{O}T$ is the inverse matrix of ${}^{O'}_{H'}T$. The transformation matrix ${}^{H'}_{H}T$ is calculated using Equation (4.19). ${}^{H'}_{W}T$ is the inverse matrix of ${}^{W}_{H'}T$, which is the transformation matrix between the hand coordinate system and the world coordinate system in the previous frame. ${}^{W}_{H}T$ is the transformation matrix between the hand coordinate system and the world coordinate system in the current frame.

$$ {}^{H'}_{H}T = {}^{H'}_{W}T \, {}^{W}_{H}T \tag{4.19} $$

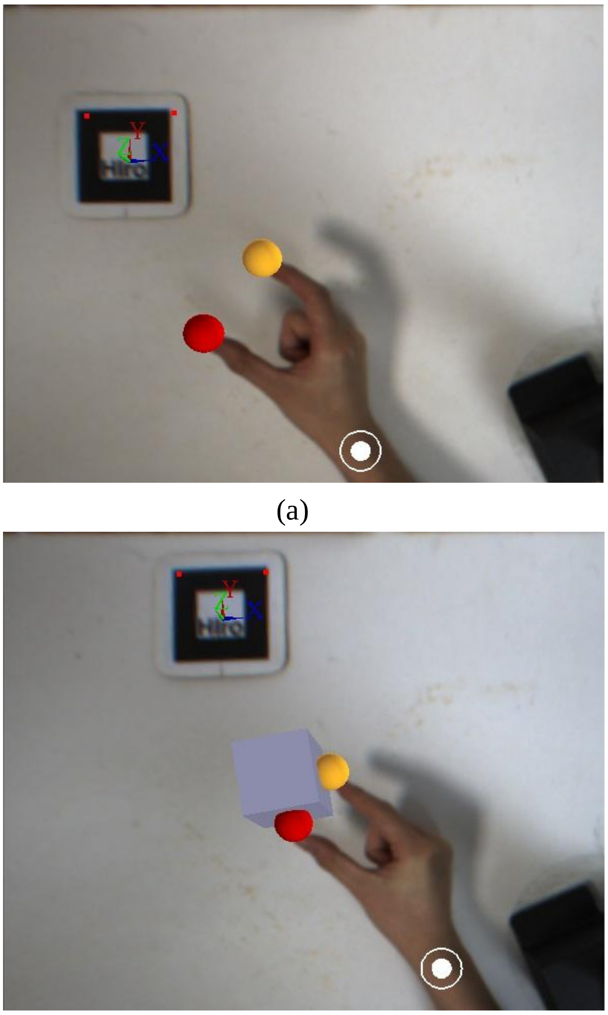
In order to achieve the interaction between the hands and the virtual objects, a small virtual sphere is rendered on each fingertip. The V-Collide algorithm [Thomas et al, 1997] has been adopted in this research to detect collisions between the spheres on the fingertips and the virtual objects. The detail information of V-Collide information can be found in Appendix B. The 3D interaction algorithm is illustrated in Figure 4-20.



Collision detection
Is the virtual object colliding with two fingertips?
No
Yes
Are there records of ${}^{M}_{H'}T$ and ${}^{O'}_{H'}T$?
No
Yes
Record the transformation matrices ${}^{M}_{H'}T$ and ${}^{O'}_{H'}T$
Calculate the transformation matrices ${}^{M}_{H}T$ and ${}^{H'}_{H}T$
Calculate the current transformation matrix of the object ${}^{M}_{O}T$
Update the transformation matrix for rendering and collision detection
Next frame

Figure 4-20 Interaction algorithm performed for each acquired image frame

Figure 4-21(a) shows the registration of a virtual red ball on the thumb tip and a virtual yellow ball on index fingertip. Figure 4-21(b) shows the pinch operation of a virtual cube using the fingertips of the thumb and the index finger. More detailed discussion of the use of the 3DNBHI for manual assembly simulation and design will be presented in Chapter 6.



(a)

(b)

Figure 4-21 Rendering virtual balls and pinch operation of a virtual cube

4.9 Accuracy of the Interaction Tool

The root mean square (RMS) errors are used to estimate the accuracy of the proposed interaction method. For a set of n values $\{x_1, x_2, \cdots, x_n\}$, the RMS value is calculated using Equation (4.20).

$$x_{rms} = \sqrt{\frac{x_1^{\ 2} + x_2^{\ 2} + \cdots + x_n^{\ 2}}{n}} \qquad (4.20)$$

4.9.1 Accuracy of Fingertip Detection

To estimate the accuracy of the fingertip detection method for 2D images, an RMS error estimation method [Shen et al, 2011] is adopted in this research. The accuracy is calculated by measuring the difference of the position of a specific fingertip and a pre-defined reference point. The user points at the reference point using the tip of his/her index finger during the error estimation method, as shown in Figure 4-22. The black point is the reference point and the red point is the 2D position of the fingertip. These two points have the same radius.

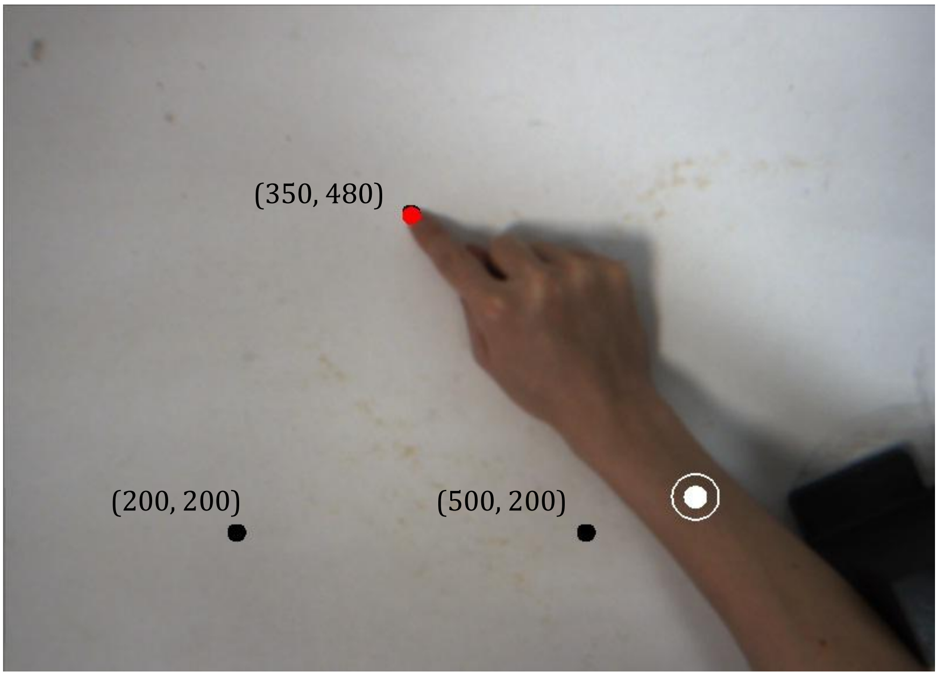


Figure 4-22 Accuracy estimation of the fingertip detection method

The image points (200, 200), (350, 480) and (500, 200) are selected as the reference points. During the error estimation process, 1000 fingertip positions were collected for each reference point. The RMS errors are calculated based on the deviations between the coordinates of the index fingertip and the coordinates of the reference points. The RMS errors of the fingertip detection method proposed in this research are 1.23 pixels and 1.46 pixels in the x- and y-axis respectively.

4.9.2 Accuracy of the Registration of Virtual Balls on Fingertips

During the estimation the accuracy of the registration of the virtual balls on the fingertips, the user is required to point his/her fingertip at a pre-defined 3D point, as shown in Figure 4-23. The deviations between the 3D coordinates of the fingertip and the 3D coordinates of the reference point are used to calculate the RMS errors. A rigid cuboid with the size of $50 \times 50 \times 86$ (mm) is chosen as a small platform for the user to locate his/her fingertip. 1000 fingertip positions were collected for each of the three reference points, which are (100, -150, 86), (0, -140, 50) and (-100, -100, 50). The RMS errors of the fingertip registration in this research are 3.26mm, 13.28mm and 14.80mm in the x-axis, y-axis and z-axis respectively.

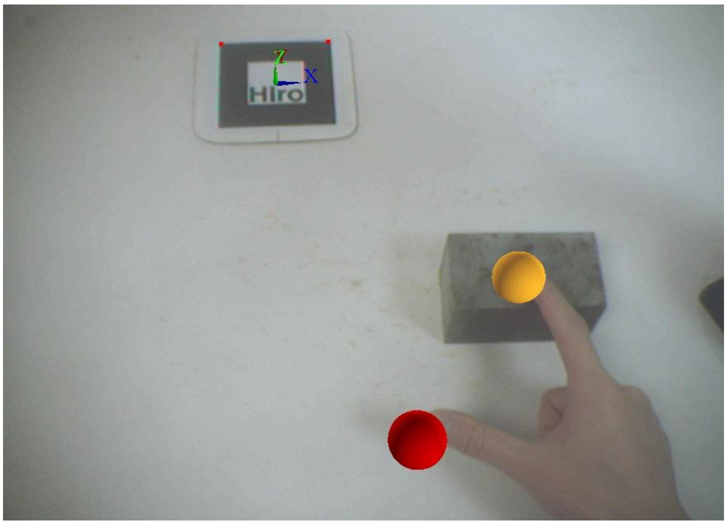


Figure 4-23 Accuracy estimation of the fingertip registration

The variation errors of the fingertips registration are mainly caused by two factors:

(1) The fingertip detection errors which are described in Section 4.9.1; and

(2) The registration errors of the ARToolKitPlus [Malbezin et al, 2002].

4.10 Hand Strain

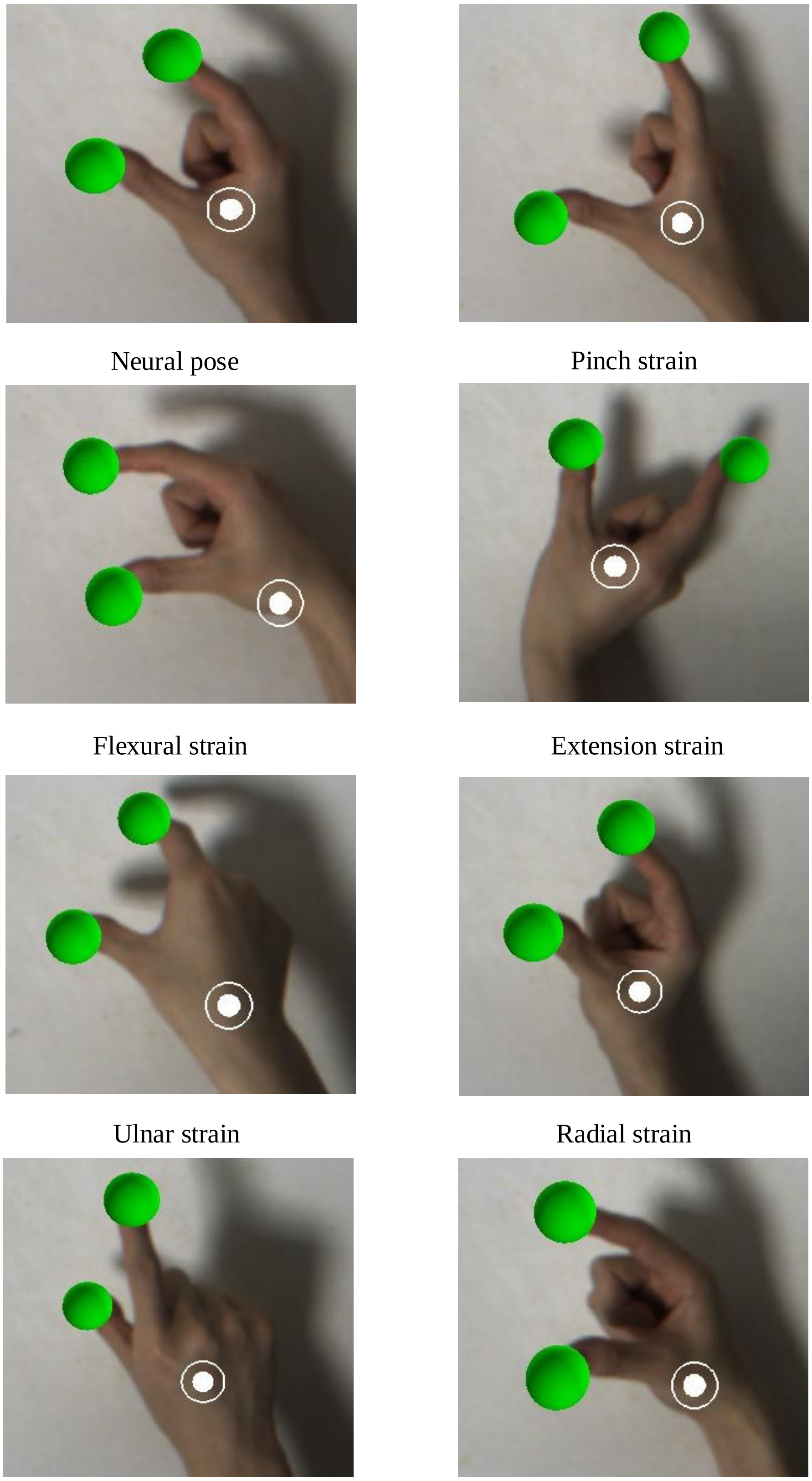
Hand strain is defined as the discomfort a user experiences at certain hand postures during an assembly process. Two types of hand strains can be captured during assembly evaluation, namely, pinch width strain and wrist angle strain. The pinch width strain will be captured when the pinch width exceeds a discomfort threshold. The wrist angle strain will be captured when the deviation of the wrist angle has reached a range where discomfort is experienced by the user. The range of the different wrist angles where discomfort is experienced is listed in Table 4-1. The various hand strain postures that can be captured are shown in Figure 4-24.

Table 4-1 Discomfort range for different wrist angles [Carey and Gallwey, 2002; Khan et al, 2010]

Deviation types	Range of motion (ROM)	Discomfort range	
Flexural	95 °	>45% of ROM	>43 °
Extension	85 °	>45% of ROM	>38 °
Radial	45 °	>45% of ROM	>20 °
Ulnar	70 °	>45% of ROM	>32 °
Pronation	130 °	>45% of ROM	>59 °
Supination	145 °	>45% of ROM	>65 °

(1) Pinch width strain

The width of the pinch is the distance between the tip of the thumb and the index fingertip. The user will have only 60% of the pinch strength when the pinch width exceeds 110mm [Imrhan and Rahman, 1995]. Therefore, when the width of the pinch is more than 110mm for more than 1 second, a strain is identified and recorded along with the duration of the strain.



Neural pose

Pinch strain

Flexural strain

Extension strain

Ulnar strain

Radial strain

Pronation strain

Supination strain

Figure 4-24 Hand strain postures

(2) Wrist angle strain

The deviation of the wrist angles of the hand is calculated based on the transformation of the HCS from the neutral posture to the current posture. Neutral pose is defined as the posture where the bones of the fingers and forearm are roughly parallel [Carey and Gallwey, 2002; Khan et al, 2010]. The user is allowed to define the neutral posture at the beginning of evaluation. The methods of wrist angle strain calculation [Ng et al, 2012] are adopted in this research.

The terminating condition for a strain capturing event is when the hand is no longer in strain, i.e., when the width of the pinch and the wrist deviation angles are less than their respective thresholds. There are certain postures where the tracking of the fingertips will fail as they are occluded by other parts of the hand. These postures mainly happen when hand strains are captured. Loss of tracking of the fingertips will be deemed as a continuation of the hand strain event. Several strains may be captured simultaneously during an assembly evaluation process. It is difficult to obtain a formula to calculate the total strain of combined strains [Khan et al, 2010]. In this research, hand strains are treated independently and the average of them is used to derive the hand strain during an assembly operation.

4.11 User Study

A user study was performed to evaluate the system and compare the proposed bare-hand interaction with two traditional interactions for AR and/or VR applications, namely, keyboard-based interaction and sensor-based interaction. Six researchers (ages from 25 to 31) were invited to test the system. All participants use computers regularly and have experience in the use of AR-based systems. The user study consists of two parts, namely, a system test and a questionnaire-based survey. At the beginning, a training session, which takes approximately 10 minutes, is conducted to allow the participants to familiarize themselves on the use these three different interactions. The questionnaire can be found in Appendix C.

4.11.1 Tasks

In this user study, all participants are required to move one virtual object from its original position (drawn in solid) to the target position (drawn in semi-transparent mode) by using three different interactions, as shown in Figure 4-25. The keyboard-based interaction allows the users to move and rotate the virtual object step by step while certain keys are being pressed. The sensor-based interaction uses two data-gloves for the purpose of gesture recognition. It allows the users to move and rotate the virtual object step by step while certain gestures are being recognized.

4.11.2 Discussion

During the user study, both quantitative and qualitative data have been collected to analyse the performance and usability of the proposed system and compare the bare-hand interaction with keyboard-based interaction and sensor-based interaction.

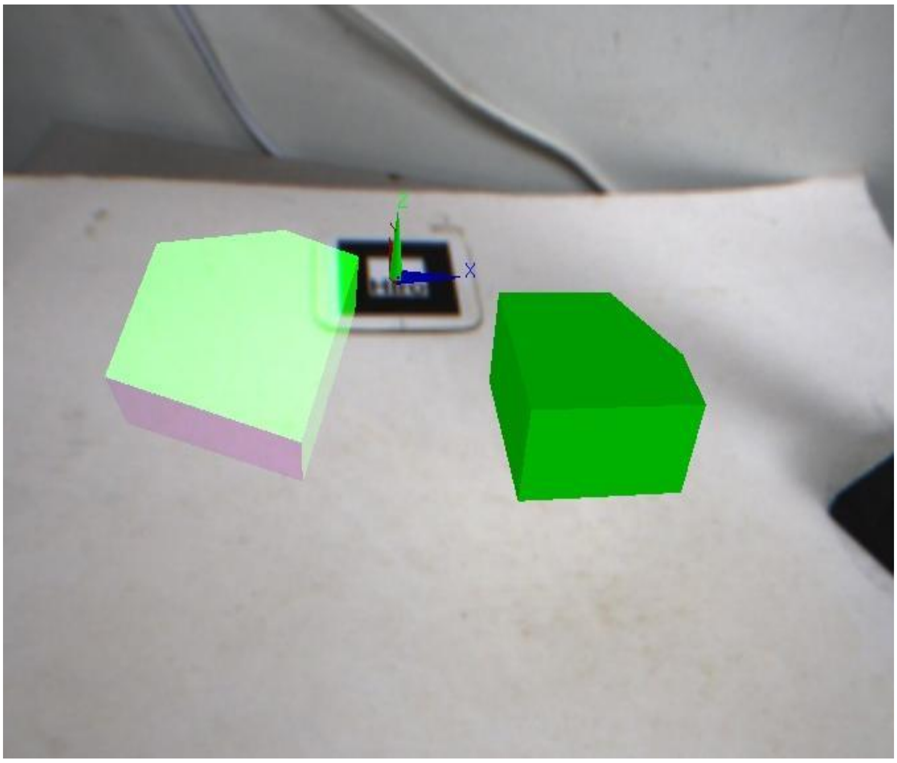


Figure 4-25 The task of the user study

The average completion time of the task and the accuracy of the result are shown in Table 4-2. The accuracy of the final result is represented as the RMS error which is Table 4-2, and they are calculated based on the deviations between the current and target position and orientation of the virtual object.

Table 4-2 Quantitative analysis of the user study

Method Parameter	Bare-hand interaction	Keyboard-based interaction	Sensor-based interaction
Average completion time (s)	35.0	50.6	80.8
Accuracy	91.7%	100%	100%

The user study indicates that the proposed system has enabled faster and easier interaction than keyboard-based interaction and sensor-based interaction but with less accuracy. As keyboard-based interaction and sensor-based interaction allow the user to translate and rotate the virtual object step by step, the virtual object can be moved to the exact position. As the user must adjust the position and orientation of the virtual object based on his/her visual feeling, it is very difficult to complete the task without any error.

Table 4-3 Evaluation criteria

Index	Evaluation Criteria
1	Ease of learning the interaction method (1~5: 1 = very difficult and 5 = very easy)
2	Ease of use of the system (1~5: 1 = very difficult and 5 = very easy)
3	Usefulness of the system in AR applications (1~5: 1 = not useful at all and 5 = very useful)
4	How immersive the experience is (1~5: 1 = not immersive at all and 5 = very immersive)

In this user study, all participants were asked to complete a post-experiment questionnaire. Four evaluation criteria as shown in Table 4-3 are used to evaluate the performance and the ease of use of the system.

As shown in Table 4-4, the result indicates that the proposed bare-hand interaction method is intuitive and can satisfy the requirements and needs of the participants. The participants also provided a few suggestions and comments on the proposed system, such as “It would be better to resolve self-occlusion issue”, “Haptic feedback would be useful through which user could have a sense of manipulate or interact with the virtual object”, and “Apply manipulation constraints to improve the user experience”.

Table 4-4 Qualitative analysis of the user study

Criteria	**Ease of learning**	**Ease of use**	**Potential usefulness**	**Immersive feeling**
Result	4.5	4	4.3	4.3

4.12 Summary

Human hands are more robust and provide user-friendly HCI. In this chapter, a natural and robust HCI approach for AR applications using bare hands is proposed. The methods and algorithms for 3D bare-hand interaction are discussed and illustrated. The hand segmentation method in this methodology uses an efficient colour segmentation algorithm. The thumb tip and index fingertip are used for the pinch operations of the virtual objects. Using stereo vision technologies, 3D information of the fingertips are retrieved and used for 3D pinch operations. Hands and fingertips differentiation algorithms are developed to achieve a dual-hand interaction interface. Correct occlusion relationship between the fingertips and the virtual objects can be achieved. The dual-hand interaction interface allows the user to manipulate and orientate virtual objects in an augmented environment simultaneously. The dual-hand interface is robust and provides direct and intuitive interactions between the user and the virtual objects in an AR environment.

Chapter 5 Constraint-based Augmented Assembly Design

5.1 Introduction

In the previous chapter, a dual-hand bare-hand interaction interface was presented. With this interface, the users can manipulate virtual assembly parts and complete an assembly process. In this chapter, a constraint-based augmented assembly system is presented. Algorithms for assembly constraint recognition, assembly location refinement are proposed. A hybrid approach based on these methods is formulated and implemented to allow the user to simulate the manual assembly process without the need for auxiliary CAD information in an AR environment. After the assembly model is available, assembly planning can be conducted to select a near-optimal sequence from all the feasible sequences.

5.2 Augmented Assembly Process

With the 3DNBHI interface, the user can manipulate and assemble two different parts more intuitively and realistically. When these two parts are sufficiently close to each other, the user can adjust the positions and orientations of these parts easily and efficiently to trigger the assembly constraint recognition and confirmation function for the augmented assembly process. This system provides a new perspective to assembly operations through recognizing assembly constraints based on geometric constraint analysis and contacting surface analysis of the feature-based models of the assembly parts. Once the list of surface contacts has been generated, the system analyses the types of geometric features that are in contact with each other and predicts what a user is trying to achieve. The flowchart of the constraint-based augmented assembly process is shown in Figure 5-1.

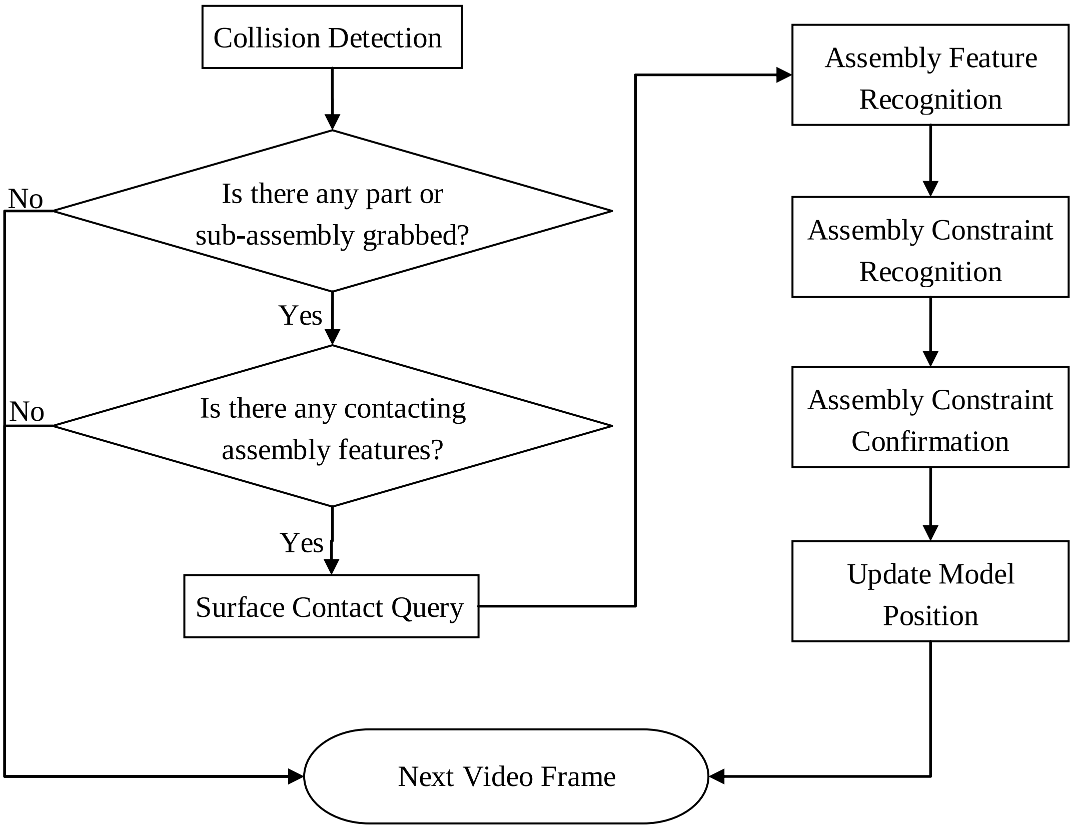

Collision Detection
Is there any part or sub-assembly grabbed?
No
Yes
Is there any contacting assembly features?
No
Yes
Surface Contact Query
Assembly Feature Recognition
Assembly Constraint Recognition
Assembly Constraint Confirmation
Update Model Position
Next Video Frame

Figure 5-1 The constraint-based augmented assembly process

5.2.1 Assembly Feature Recognition

As discussed in Section 2.4.2, only the geometric attributes of a part are considered as assembly features. In this research, the assembly environment does not need to be prepared *a-priori*. From the CAD solid models that are loaded into the ARIMADP system, the geometric information required for assembly modelling is extracted using the API of a commercial CAD system (SolidWorks^{TM}).

5.2.1.1 Geometric Information Extraction

During initialization, the CAD models of the assembly components are loaded into the ARIMADP system. The surfaces of these CAD models are enumerated and discretized into triangle tessellations. Information of these tessellations is stored and used for display using OpenGL and collision detection using V-Collide. After all the triangle tessellations of an assembly component have been extracted, a continuous geometric entity of the component is reconstructed using the SolidWorks API. The surfaces of the CAD models will be treated as individual objects in the assembly process.

A surface consists of a series of triangle tessellations indexed consecutively. The surface type is obtained using several API function of the SolidWorks surface object model to determine whether the surface is planar, cylindrical or conical, *i.e.*, “IsPlane”, “IsCylinder”, “IsCone” or “IsSphere” respectively. The parameters of these surfaces can be obtained using other API functions of the SolidWorks surface object model. In ARIMADP, only regularly-shaped surfaces are treated as reference surfaces.

5.2.1.2 Surface Contact Query

The bounding box collision detection method has been widely used to obtain the list of potential assembly features that need to be analysed for possible constraints [Marcelino et al, 2003; Yang et al, 2007]. However, this method would result in a large list of potential assembly features to be analysed. In this research, a surface contact query process is conducted to obtain an exact list of geometric entities that are currently in contact. This will result in fewer number of potential assembly features to be analysed.

Geometric constraints help the users to achieve precise part manipulations through reducing the DOFs of the parts. The assembly features considered include planar surfaces, cylindrical surfaces, conical surfaces, spherical surfaces and surface groups which consist of planar or cylindrical surfaces with parallel intersection lines between the adjacent surfaces. The spatial mating constraints considered include co-planar fit, cylindrical fit, conical fit and spherical fit, as shown in Figure 5-2. The mating direction vectors are the unit normal vectors of surfaces for co-planar fit, the axis direction vectors for cylindrical fit and conical fit. There is no specified direction vector for spherical fit.

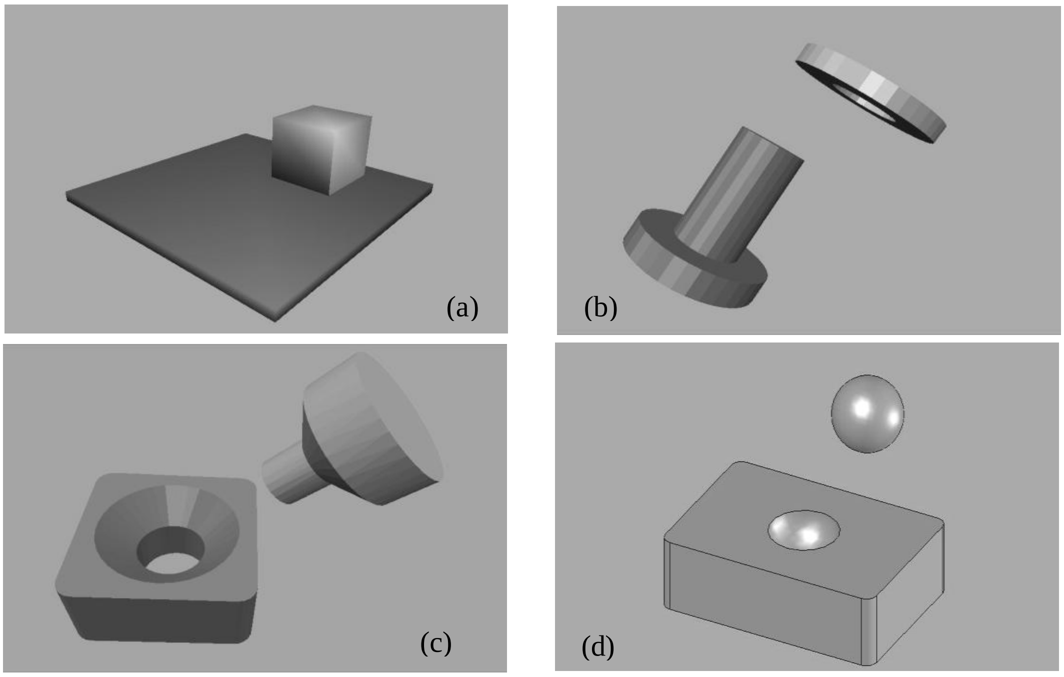


Figure 5-2 Mating constraints: (a) co-planar fit (b) cylindrical fit (c) conical fit (d) spherical fit

Based on the types of mating constraints, for a list of surface contacts, the ARIMADP system will query the surface types and parameters of these surfaces from PTS in TADS to obtain the exact list of contacting surfaces. The surface contact query is as follows:

1) *Check the types of the surface pairs in contact.* Assembly constraints are only possible between surfaces with the same face types (planar, cylindrical, conical and spherical); all other contacts between surfaces with different face types or between other types of surfaces are ignored.
2) *Check the parameters of the surface pairs in contact.* For a planar surface pair, this pair of surfaces remains in the list of surface contacts. For a cylindrical surface pair, the system checks the diameters of these contacting

surfaces. For a conical surface pair, the system checks the diameters and the half angle of these contacting surfaces. For surface groups, the system checks the diameters of the corresponding cylindrical surfaces in the group. In a CAD environment, the basic size, which is specified for each part, is the exact theoretical size of each part. The basic sizes of both connected parts must be the same. In this research, geometric tolerance has not been considered. If there is no difference between the basic sizes of both surfaces, the pair of surfaces remains in the list of surface contacts; otherwise, this pair of surfaces will be removed from the list of surface contacts.

5.2.2 Geometric Constraint Recognition

An automatic constraint recognition process is implemented to incorporate the information of these constraints into the augmented assembly process. An assembly constraint is recognized when the spatial relationship between the assembly features satisfies a particular condition within a threshold value.

5.2.2.1 Rules for Constraint Recognition

An assembly constraint has a parameter t, which is an angle t_θ when the constraint is an orientation constraint or distance value t_d when this constraint is a position constraint. The rules for constraint recognition [Li et al, 2009] can be classified into three categories according to the assembly features of the parts to be assembled.

1) *Plane-plane*. There are two parameters. The first parameter is the distance ε_{pd} between a point on one plane and the projected point of this point on the second plane. The second parameter is the angle $\varepsilon_{p\alpha}$ between the normals of the two planes.
2) *Line-line*. There are two parameters. The first parameter is the distance ε_{ld} between a point on one line and the projected point of this point on the second line. The second parameter is the angle $\varepsilon_{l\alpha}$ between these two lines.
3) *Point-point*. There is one parameter which is the distance ε_{td} between these two points.

5.2.2.2 Constraint Recognition for Groups of Surfaces

For two groups of surfaces (such as a prismatic joint and a dove-tail connection) in contact, two pseudo-axes are calculated based on the intersection lines between the adjacent surfaces in each group and these groups of surfaces are considered as feature candidates to be established with a cylindrical fit, as shown in Figure 5-3. Given a surface group with N parallel intersection lines ($l_{1,}l_{2,}...,l_N$), the pseudo-axis is determined in three steps:

1) Obtain the orientation vector $\vec{v}$ of these parallel lines.

2) Randomly choose a plane that is perpendicular with these parallel lines and obtain all the intersection points ($p_{1,} p_{2,} ..., p_N$); obtain the average point p_{ave} of these intersection points according to Equation (5.1).

$$p_{ave} = \frac{p_1 + p_2 + ... + p_N}{N} \tag{5.1}$$

3) Calculate the pseudo-axis based on the orientation vector $\vec{v}$ and the average point p_{ave}.

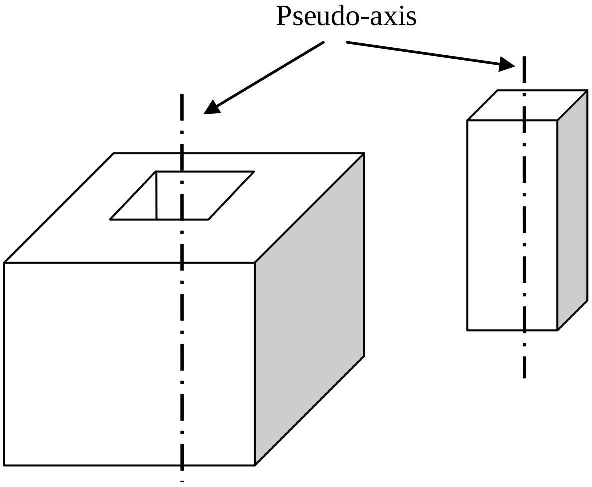


Figure 5-3 Pseudo-axes of a prismatic joint

Once a pseudo-axis has been determined for a surface group, it will be stored for use in future assembly tasks. The rules for constraint recognition in ARIMADP are listed in Table 5-1.

Table 5-1 Rules for constraint recognition

Assembly features	Constraint type	Recognition Rules
Planar surfaces	Co-planar fit	*Plane-plane* $\lvert\varepsilon_{pd}\rvert \le t_d, \lvert\varepsilon_{p\alpha}\rvert \le t_\theta$
Cylindrical surfaces	Cylindrical fit	*Line-line* $\lvert\varepsilon_{ld}\rvert \le t_d, \lvert\varepsilon_{l\alpha}\rvert \le t_\theta$
Conical surfaces	Conical fit	*Line-line* $\lvert\varepsilon_{ld}\rvert \le t_d, \lvert\varepsilon_{l\alpha}\rvert \le t_\theta$
Spherical surfaces	Spherical fit	*Point-point* $\lvert\varepsilon_{td}\rvert \le t_d$
Surface group	Cylindrical fit	*Line-line* $\lvert\varepsilon_{ld}\rvert \le t_d, \lvert\varepsilon_{l\alpha}\rvert \le t_\theta$

5.2.3 Geometric Constraint Confirmation

In constraint confirmation, for each constraint that has been recognized, the system can adjust the position and orientation of the components automatically to ensure that the mating features meet the current constraint precisely. The position and orientation of the component in the user's hand is adjusted for one-hand assembly operation. If the assembly operation is a dual-hand operation, the position and orientation of the component in the user's right hand will be adjusted. If the component in the user's right hand cannot be adjusted due to the established constraints, the position and orientation of the component in the user's left hand will be adjusted. During this process, a location-solving transformation matrix M_{LS} is constructed based on the minimal position and orientation refinement principle for the part [Yang et al, 2007]. The matrix is calculated according to Equation (5.2).

$$M_p^{'} = M_{LS} \times M_p \tag{5.2}$$

In Equation (5.2), M_p and $M_p{'}$ are the position and orientation matrices of the part in the user's hand before and after constraint confirmation, respectively. In this research, a single quaternion is used for rotation calculating of the part to assure the part can be rotated freely and correctly in 3D space during the constraint confirmation process. The detailed information of quaternions and spatial rotation can be found in Appendix D.

The main intention of this process is to construct M_{LS} . The constraint confirmation steps are described as follows:

1) Obtain the quaternion Q from the angle ε_α and the revolution axis l_α : For coplanar fit, ε_α is the angle between two plane normals and l_α is a cross product of two plane normals; for cylindrical fit and conical fit, ε_α is the angle between two axes and l_α is a cross product of the two axes.
2) Construct the rotation transformation matrix T_R from Q, so that the orientation between the two assembly features satisfies the constraint when the part is subject to T_R .

3) Obtain the current ε_d and construct the translation transformation matrix T_T based on ε_d, so that the position between two assembly features satisfies the constraint when the part is subject to T_T.

4) Construct the location-solving matrix $M_{LS} = T_R \times T_T$.

The geometric calculation equations for constraint confirmation can be found in Appendix D. After the constraint has been refined in the constraint confirmation process, the position and orientation of the parts in contact will be adjusted to satisfy the constraint precisely. As a result of this process, the DOFs of the parts will be reduced and the motion of parts is constrained. The parts can only be translated in the aligned plane and rotated about the normal of the aligned plane for the co-planar fit constraint, rotated and translated about the aligned axis for the cylindrical fit constraint, rotated about the aligned axis for the conical fit constraint, and rotated about the aligned centre point for the spherical fit. The assembled sub-assembly will be treated as a single part for manipulation and collision detection in further bare-hand assembly operations. These established constraints will remain in effect for further bare-hand assembly operations when the system is in the Assembly Mode, which is the default mode. The assembled parts can be disassembled when the system is switched to the Component Mode, where the established constraints are ignored for further bare-hand assembly operations. The user can toggle between the Assembly Mode and the Component

Mode during the assembly or sub-assembly of a product using a virtual interaction panel shown in Figure 5-4. Several constraints may be applied to a part in a real assembly process. Each constraint will affect the relative motion of the two assembly features. When the entire assembly has been achieved, the assembly information will be stored in ATS in TADS for future use.

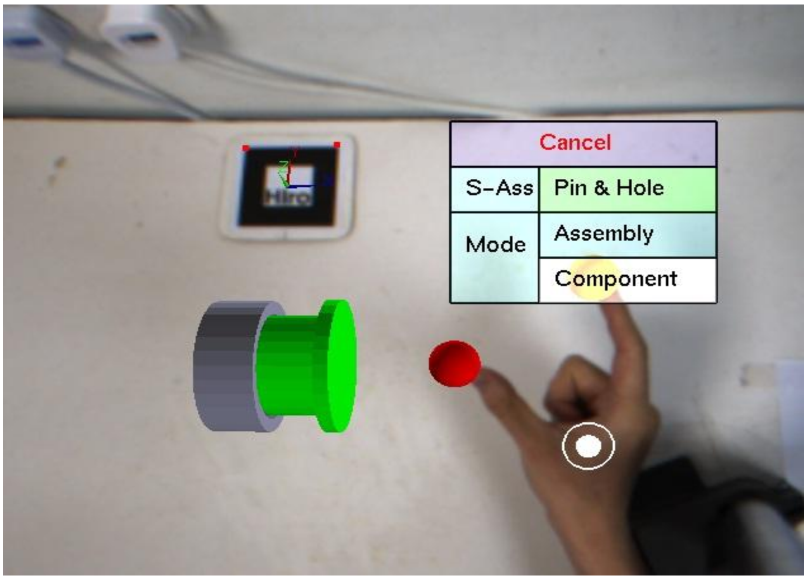


Figure 5-4 Mode switching of a sub-assembly (pin-hole) using a virtual interaction panel

5.2.4 Tool-assisted Assembly Operation

Assembly tools and tool-assisted assembly operations are considered during assembly design and sequence planning in the ARIMADP system. Fastening and unfastening operations are the most common tool-assisted operations in assembly and disassembly. In this research, the tool-assisted assembly operation considered is that of fastening a part using a fastening tool, such as a wrench. In the ARIMADP system, assembly tools are stored in a database. The user can select a proper tool from this database to carry out an assembly operation. The tool-assisted assembly operation process is carried out in three steps.

1) *Identification*: In this step, the contacting faces of the tool and the part are obtained and their attributes are analysed to determine whether the tool can be used to operate the part. This step is similar to the assembly process discussed in Sections 5.2.1-5.2.3 with the difference that the geometric constraint can only be a cylindrical fit constraint between the part and the fastening tool (for example between a nut and a wrench). If the selected tool is suitable for the fastening operation and the geometric constraint is confirmed, the motion of the tool will be constrained along the axis of the part.
2) *Operation*: In this step, as the motion of the tool has been constrained along the axis of the part, the part will be tightened when the tool is turned counter-clockwise. Similarly, the part will be loosened when the tool is turned clockwise.
3) *Withdrawal*: The tool operation will be completed when the part is fastened and a geometric constraint which is usually a co-planar fit is confirmed between the part and another part. After an operation has been completed, the

tool is removed and the constraints and placement relationships between the tool and the part would be released.

5.3 Assembly Planning and Evaluation

All parts with related tools for an assembly operation of a to-be-assembled product are designed in a CAD system. By defining a series of mating constraint relationships, the system can achieve the assembly and relative position constraint for two to-be-assembled parts. All the assembly constraint information is obtained by using the proposed on-line assembly modelling method which is introduced in Section 5.2. In this research, the simulate-constraint-replan method [Yin et al, 2004] is adopted for assembly planning in an AR environment.

In this research, assumptions for assembly planning are as follows:

1. All assembly plans are linear. One assembly operation must be completed before another can be carried out, which means two operations cannot be executed simultaneously.
2. The disassembly sequence is exactly the reverse of the assembly sequence.
3. The assembly workplace has been well designed. Assembly tools and parts are located close to the point of use.

An assembly sequence is valid only if all the assembly operations are geometrically feasible, which means there is no interference between parts during the assembly process. Precedence expressions (PEs), which prevent the occurrence of unwanted ordering of assembly operations, are used widely in assembly planning.

In this research, an algorithm to generate all feasible linear sequences is implemented using PEs, which follows the assembly-by-disassembly strategy. After an assembly model is ready, the user needs to disassemble the assembly using different sequences to generate several disassembly sequences. PEs are acquired from collision information derived from the disassembly operations. For example, if collision is detected between part B (which is being disassembled) and part A (which is assembled before part B) according to a certain disassembly sequence, a PE is recognized which indicates that part B should be assembled before part A. Next, a number of feasible assembly sequences are obtained by searching all possible assembly sequences with all PEs.

Once all feasible sequences are available, a few good assembly sequences are obtained in the assembly evaluation stage. The user can verify each feasible assembly sequence by assembling the product according to the sequences with the bare-hand interface in the AR environment. The number of feasible disassembly sequences grows exponentially when the number of parts of an assembly increases. Therefore, it is difficult to obtain all the PEs of an assembly after it has been disassembled. Thus, the user is allowed to impose additional PEs during the evaluation process and the replanning process is performed to search for a new set of feasible assembly sequences that also satisfies these newly imposed constraints. If the instability of a subassembly is detected during the evaluation of a feasible sequence, this sequence will be deemed as impractical and the evaluation process for this sequence will be terminated. With several goodness criteria, the assembly cost of a sequence will be calculated and recorded. Next, the user can select a near-optimal assembly sequence by comparing the assembly cost of different feasible sequences. The above process will be repeated until one or a few good or near-optimal sequences are found. The flowchart of the proposed planning and evaluation method is shown in Figure 5-5.

5.3.1 Precedence Constraint Acquisition

The PEs of a part are usually obtained based on collision detection between the part and the remaining components when the part is moving in either an assembly or disassembly process. The sweep volume method is often used for collision detection between a moving part and static remaining parts for generating PEs [Lin and Chang, 1993].

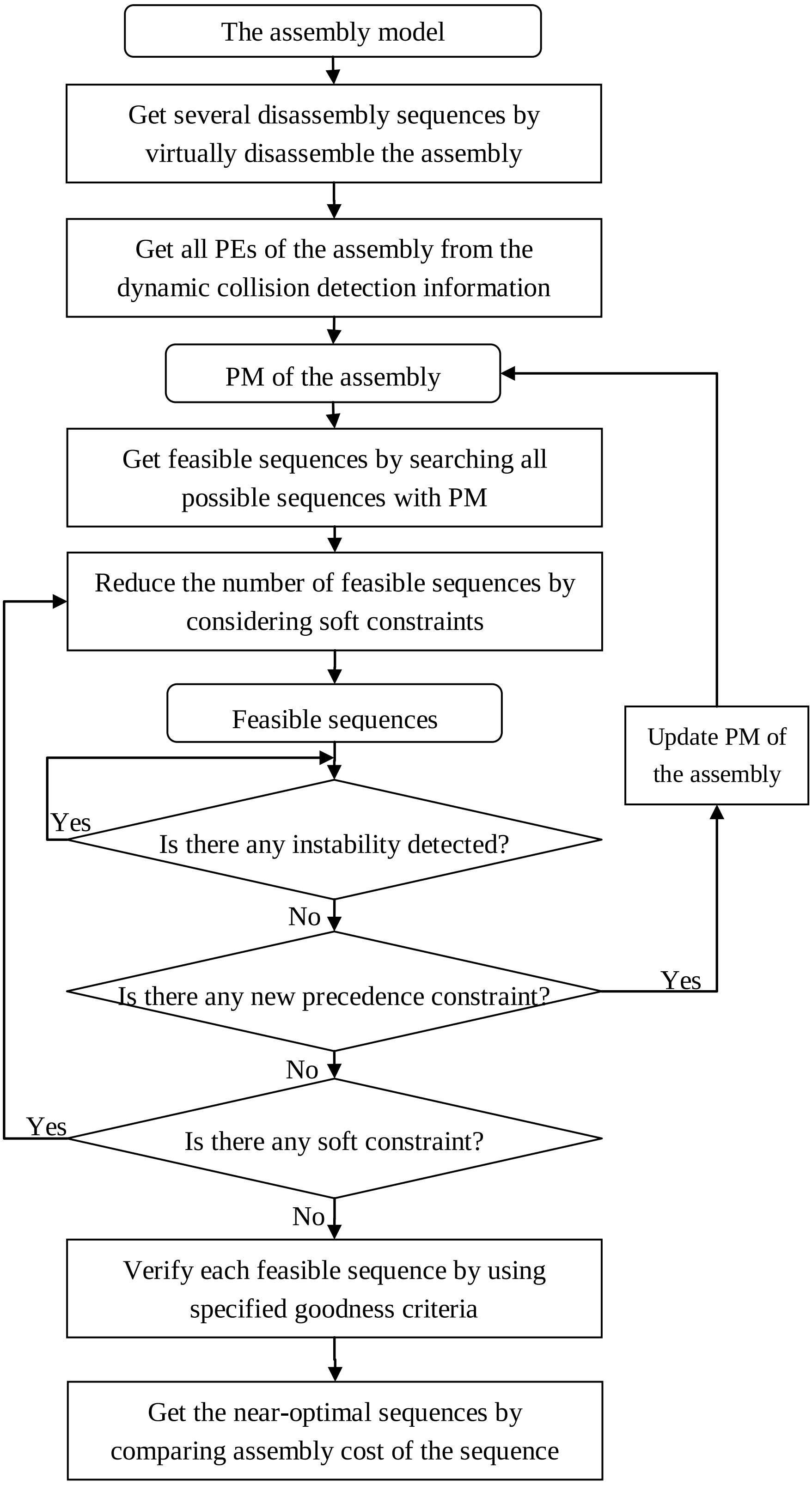

The assembly model
Get several disassembly sequences by virtually disassemble the assembly
Get all PEs of the assembly from the dynamic collision detection information
PM of the assembly
Get feasible sequences by searching all possible sequences with PM
Reduce the number of feasible sequences by considering soft constraints
Feasible sequences
Update PM of the assembly
Yes
Is there any instability detected?
No
Is there any new precedence constraint?
Yes
No
Yes
Is there any soft constraint?
No
Verify each feasible sequence by using specified goodness criteria
Get the near-optimal sequences by comparing assembly cost of the sequence

Figure 5-5 Flowchart of the assembly planning and evaluation approach

In this research, the assembly model is disassembled several times and all the disassembly sequences are recorded. The part is said to be disassembled only if all its mating relations are released, which means there is no contact between the part and all remaining parts. Each part is disassembled along its corresponding mating direction and collision detection is conducted between these parts and the parts that should be disassembled before it. The V-Collide collision detection algorithm is adopted for dynamic collision detection. Given an assembly containing *N* parts with disassembly sequence $\{t_1, t_2 \cdots, t_{N-1}\}$, when part P_i involved in task t_i is disassembling along its mating direction, collision detection is conducted only between part P_i and any part P_j involved in task t_j ($j = 1, 2, \cdots, i-1$). If collision is detected, a PE is defined between relevant parts.

In the proposed method, a precedence matrix (PM) [Choi et al, 2009] is used to describe the PEs between parts in an assembly. Given an assembly containing *N* parts, PM is a $N \times N$ matrix which elements are either 1 or 0. For example, if part P_j must be assembled after part P_i, $PM(p_i, p_j) = 1$, otherwise $PM(p_i, p_j) = 0$. Therefore, PM can be used to describe all PEs for a product. For the example assembly shown in Figure 5-6, the cap must be assembled after the plate. Therefore, $PM(p_{plate}, p_{cap}) = 1$. Table 5-2 shows all the PEs among the three parts of the assembly shown in Figure 5-6.

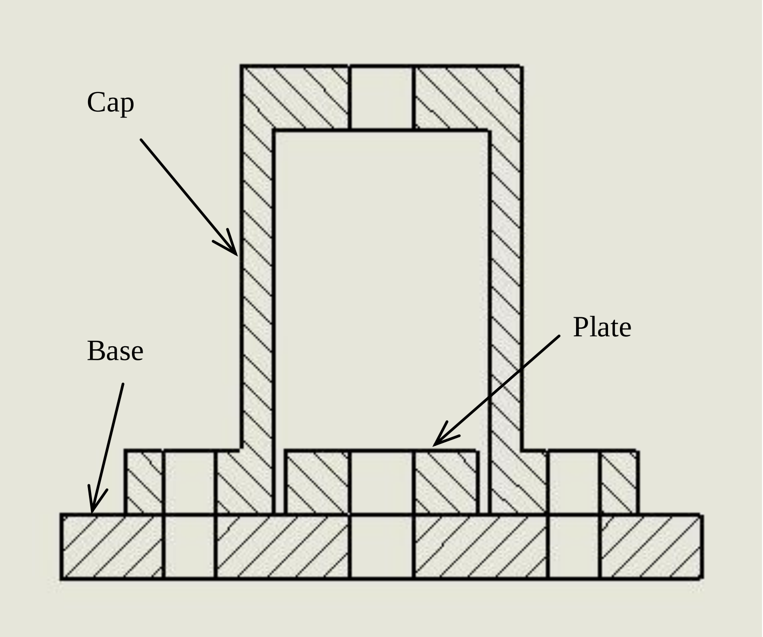


Figure 5-6 A sample assembly

Table 5-2 PM for the sample assembly shown in Figure 5-6

j / *i*	Cap	Plate	Base
Cap	0	0	0
Plate	1	0	0
Base	0	0	0

It is difficult to obtain all the PEs of an assembly as not all the feasible disassembly sequences may be considered in this process. During the evaluation stage, the PM of an assembly can be updated when additional PEs are found by the user. At the beginning of the evaluation stage, a set of feasible assembly sequences which satisfy all the PEs in PM are obtained by searching all the possible sequences. Next, all the feasible assembly sequences are provided to the user for evaluation.

5.3.2 Stability Examination

Instability problems caused by gravity are common for subassembly systems. To ensure that each subassembly system as well as the final assembly is stable, feasible assembly sequences that may lead to unstable intermediate subassemblies must be discarded. A virtual ground plane which is coincident with the *X*-*Y* plane of the WCS is established, as shown in Figure 5-7. Each part and subassembly is placed on the ground plane. Generally, the stability examination of a subassembly involves checking the DOF in the gravitation direction for each component of the subassembly [Eng et al, 1999]. This method is sufficiently accurate as instability may be caused when the centre of gravity of the subassembly falls outside its base. In this research, stability estimation of a subassembly is conducted according to the following three steps:

(1) Check the attribute of the base of each subassembly. The subassembly is said to be unstable if its base plane is not planar.

(2) Check the DOF in the gravitation direction for each component of the subassembly. The subassembly is said to be unstable if there is any component with allowed motion in the gravitation direction.

(3) Check geometric relation between the centre of gravity of the subassembly and its base. For a subassembly with *n* components, the centre of gravity of a subassembly is calculated using Equation (5.3).

$$a_{gc} = \frac{m_1 a_1 + m_2 a_2 + \ldots + m_n a_n}{m_1 + m_2 + \ldots + m_n} \tag{5.3}$$

where m_x ($x = 1, 2, ..., n$) is the xth component's mass and a_i ($i = 1, 2, ..., n$) is the coordinate value of the xth component's centre of gravity.

The subassembly is said to be unstable if its centre of gravity falls outside of its base. The relation between a_{gc} and the base is checked using Equation (5.4).

$$\begin{cases} (B|_{x\min} < a_{gc} | x < B|_{x\max}) & \\ \&(B|_{y\min} < a_{gc} | y < B|_{y\max}) & \textit{Stable} \\ \\ \textit{Otherwise} & \textit{Instable} \end{cases} \tag{5.4}$$

where, $B|_{x\min}$ and $B|_{x\max}$ are base limitations in X direction of WCS. $B|_{y\min}$ and $B|_{y\max}$ are base limitations in Y direction of the WCS.

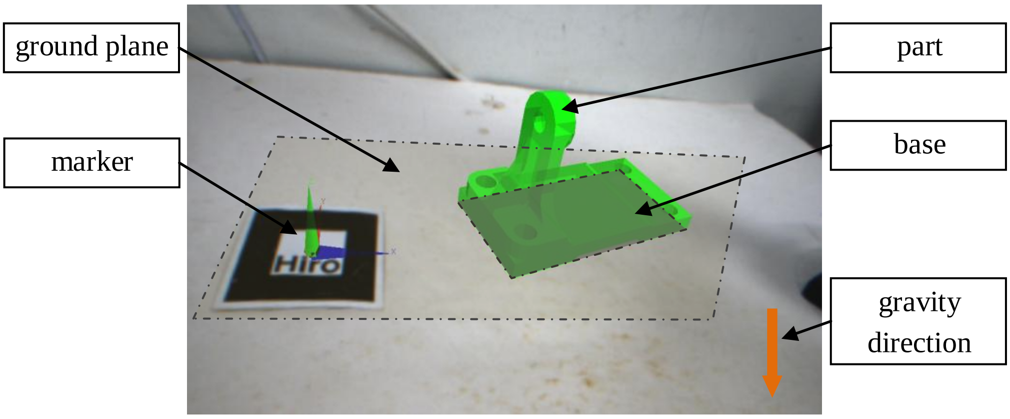


Figure 5-7 The stability examination of a subassembly

5.3.3 Goodness Criteria for Assembly Evaluation

After the feasible sequences that satisfy the PEs of the assembly have been obtained, the user can assemble the product according to certain feasible sequences by direct manipulation using the bare-hand interface. In this research, a set of criteria used to choose a good or near-optimal assembly sequence. The criterion set includes the handling of a certain component which is represented as hand strains and operation continuity which is influenced by orientation changes and tool changes. An overall assembly index is calculated based on these criteria to rank the goodness of different assembly sequences.

5.3.3.1 Ease of Handling

In this research, the ease of handling an assembly operation is determined by the hand posture used for completing an assembly operation and the connection type of the assembly. According to the assembly mating constraints adopted in this research, the connections can be classified into three types, namely, Mate (co-planar fit), Insert (cylindrical fit, conical fit and spherical fit) and Fasten (using screws to secure certain components). Two selection indices are used to evaluate the ease of handling, and they are the Hand Strain Index (HSI) and the Operation Preference Index (OPI).

(1) HSI

A HSI is derived from the Strain Index [Moore and Vos, 2004] used to evaluate jobs for exposure to increased risk of developing musculoskeletal disorder of elbow, forearm, wrist and hand. Strain events are captured during the assembly operation and the parameters, namely, maximum deviation, duration of strain and duration of the assembly operation, will be used to evaluate the hand strains of each assembly operation. In this research, Strain Strength (SS), Strain Duration (SD) and Strain Efforts per minutes (SE) are used to calculate the HSI for each assembly operation [Ng et al, 2012]. SS is defined as a percentage of the difference between the maximum strain deviation and the threshold over the threshold, as shown in Equation (5.5).

$$SS = \frac{Maximum\ Deviation - Strain\ Threshold}{Strain\ Threshold} \times 100\% \tag{5.5}$$

SD is defined as the percentage of the total durations of all the strain events over the total duration of the assembly operation, as shown in Equation (5.6). SE is the number of strain events detected per minute.

$$SD = \frac{Duration\ of\ all\ strain\ events}{Duration\ of\ assembly\ operation} \times 100\% \tag{5.6}$$

In this research, the HSI is calculated using Equation (5.7).

$$HSI = M_s \times M_D \times M_E \tag{5.7}$$

where M_s , M_D and M_E are multiplier values for SS , SD and SE respectively. The multiplier values are listed in Table 5-3.

The 3DBHNI allows the user to assemble or disassemble a product with his/her two hands simultaneously. According to the principles of motion economy in manual assembly [Crowson, 2006], two hands working together at the same time is more desirable than working with one hand while holding the object being worked on with the other hand. In this research, it is recommended for the user to complete an assembly operation with his/her two hands moving symmetrically. The symmetrical movement of the user's two hands enables the user to perform an operation with less physical effort while capturing small HSI. In this research, the sum of the HSI of both right hand and left hand is considered as the HSI of an assembly operation. A hand posture with HSI of 5.0 is considered to be hazardous [Moore and Vos, 2004]. If the HSI of either right hand or left hand exceeds 5.0 while evaluating a feasible sequence, the sequence will be deemed as impractical and the evaluation process will be terminated.

Table 5-3 The multiplier values for the calculation of HSI

SS		*SD*		*SE*	
Value	M_s	Value	M_D	Value	M_E
-	1.0	<10	0.5	<4	0.5
-	1.0	10%-29%	1.0	4-8	1.0
0-9%	1.5	30%-49%	1.5	9-14	1.5
10-19%	2.0	50%-79%	2.0	15-19	2.0
$\geq$20%	3.0	$\geq$80%	3.0	$\geq$20	3.0

(2) OPI

The ease of handling an assembly operation is affected by the connection type involved in this operation. In this research, the OPI indicates the priority of the connection type. The operation complexity of each connection type is different [Dong et al, 2005]. For instance, a Fasten type connection is more difficult to be assembled or disassembled than a Mate type. As shown in Table 5-4, the OPI is determined by the connection types.

Table 5-4 OPI of different connection types

Connection type	OPI
Mate	0.3
Insert	0.5
Fasten	0.7

The ease of handling of an assembly operation is represented as the Manipulability Index (MI), which is calculated using Equation (5.8).

$$MI = k_H \times HSI + k_C \times OPI \tag{5.8}$$

where k_H and k_C are coefficients and $k_H + k_C = 1$.

5.3.3.2 Continuity

Less physical effort will be required when assembly operations with the same mating direction are grouped together and conducted successively. Therefore, an assembly plan where assembly operations with similar directions are grouped together and performed consecutively is a more efficient plan. In this research, the Continuity Index (CI) [Dong et al, 2005] is used to evaluate the assembly operations. A smaller CI indicates fewer additional movements the user has to make by exchanging the connection types, mating directions and tools during the assembly process. In this research, the CI consists of three indices, which are the

Connection Change Index (CCI), Direction Change Index (DCI) and Tool Change Index (TCI). If the connection type of a current operation is different from the previous one, $CCI = 1$; Otherwise, $CCI = 0$. If the mating direction of a current operation is different from the previous one, $DCI = 1$; Otherwise, $DCI = 0$. If the tool used in a current operation is different from the previous operation, $TCI = 1$; Otherwise, $TCI = 0$.

The CI is calculated using Equation (5.9).

$$CI = k_C \times CCI + k_D \times DCI + k_T \times TCI \tag{5.9}$$

where, k_C, k_D and k_T are coefficients and $k_C + k_D + k_T = 1$.

5.3.3.3 Overall Assembly Index

The Overall Assembly Index (OAI) is calculated using Equation (5.10) [Dong et al, 2005].

$$OAI = e^{k_1 \times MI + k_2 \times CI} \tag{5.10}$$

where, k_1 and k_2 are the coefficients and $k_1 + k_2 = 1$.

The assembly sequence with a lower OAI value is more efficient than the one with higher value. All the coefficients used in Equations (5.8-5.10) can be assigned by the user based on the relative significance of each selection index on the OAI. By comparing the OAI values of different assembly sequences, the user can select a good or near-optimal sequence with the lowest OAI value.

5.3.4 Planning with Soft Constraints

The use of soft constraints can ease the assembly planning as it can reduce the search space to obtain a good or near-optimal sequence [Yin et al, 2001]. In this research, two soft constraints, similar to REQ-CLUSTER and REQ-ORDER-FIRST reported by Jones and Wilson [Jones and Wilson, 1996] are used for reducing the number of feasible sequences.

The first soft constraint is parts-clustering, which requires that a particular group of parts be added to the assembly successively without interruption by other parts or subassemblies. The second soft constraint is first-part-definition, which requires that the assembly plan starts with a certain part as the first part. It is not easy for the user to specify certain soft constraints at the beginning of an assembly evaluation. Many of these constraints will become obvious for the user during the assembly evaluation process. Therefore, the user is allowed to impose new soft constraints to the assembly. Then, the feasible sequences will be re-searched from all the possible sequences. With fewer feasible sequences, it is easier for the user to obtain a good or near-optimal sequence.

5.4 Summary

In this chapter, a hybrid approach is proposed to develop the ARIMADP system. Using the hybrid approach, the system allows the user to simulate the manual assembly process of a product. The hybrid approach consists of several algorithms including a robust feature-based automatic constraint recognition algorithm and constraint-based assembly constraints recognition and refinement have been developed. Assembly tools and tool-assisted operations have been considered to make the assemblies in an AR environment more realistic. By integrating the 3DNBHI, ARIMADP can achieve an intuitive recognition of the mating relations between the components and the functions of the assembly. By analysing the disassembly process, precedence constraints are captured. The precedence constraints are used to search for feasible assembly sequences from all the sequences. Finally, practical or good sequences are selected from the feasible sequences by determining an assembly cost function based on the ease of handling of the assembly and its components and the continuity of the assembly operations. A case study implementing the developed approach for assembly planning is reported in Chapter 6.

Chapter 6 Implementation and Case Studies

6.1 Introduction

This chapter first presents a 2D assembly simulation system. The system architecture of ARIMADP is introduced. Two case studies are discussed to illustrate the simulation of manual assembly using the bare-hand interface in an AR assembly environment based on the methodologies presented in Chapter 4 and Chapter 5.

6.2 2D Assembly Simulation

A 2D manual assembly simulation system is developed for the user to simulate the assembly using a 2D bare-hand interaction tool. The architecture of this system is shown in Figure 6-1, including the CAD module in the orange box, the AR module in the blue box and the interaction module in the red box. This system implements a marker-based method to set up the AR environment. A standard web camera is used to capture the video sequence of the real assembly scene. The hand segmentation and the hand feature extraction methods discussed in Sections 4.3.2 and 4.4 respectively are implemented to realize the 2D bare-hand interaction. In the AR-based assembly simulation system, users can perform manual assembly simulation in a real environment.

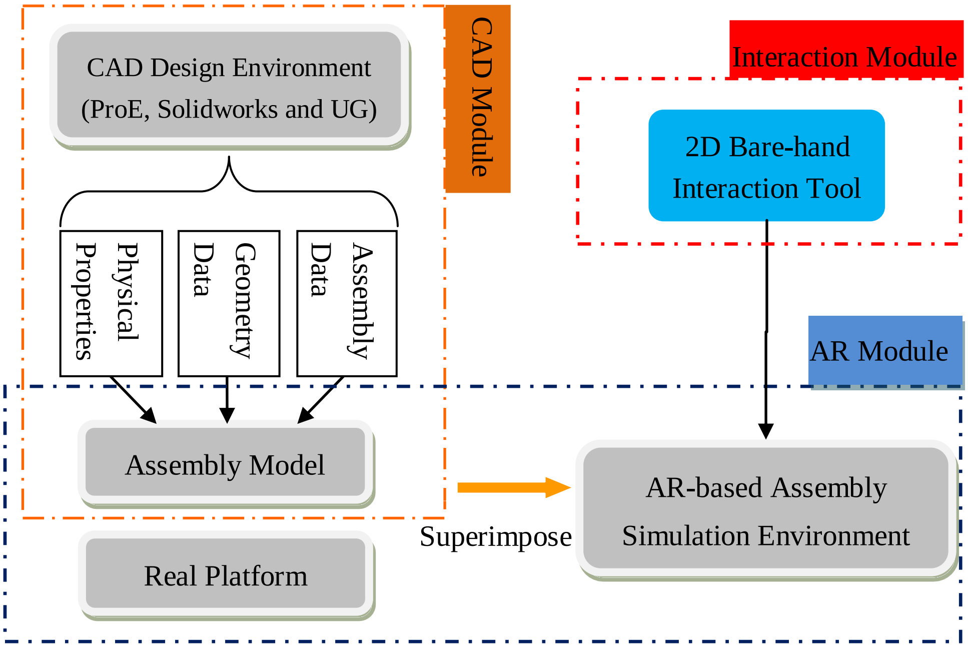


Figure 6-1 2D Architecture of the manual assembly simulation system

6.2.1 CAD Module

Modern CAD systems have excellent manipulation and sophisticated assembly creation capabilities that allow the modelling of different parts which can be assembled together to create a complete unit. Most of the information required for simulating and evaluating an assembly can be generated with a CAD system. In this proposed system, a CAD system (SolidWorks^{TM}) is integrated with the AR environment and three types of data are exported from the CAD system, namely,

1) Assembly data: assembly hierarchy and assembly constraints;
2) Geometry data: geometry data of each part or sub-assembly;
3) Physical properties: mass, texture, etc., of each part or sub-assembly for physical modelling purposes.

These three types of data are defined as an assembly model. The assembly models are stored in the system memory and are transferred into the AR environment for further design modelling and evaluation.

In the system that has been developed in this research, the virtual model is a triangle-based polygonal approximation, *i.e.*, an STL model, of the original CAD model. The STL model can be created from a CAD model directly using the existing export function of CAD systems. However, this conversion process has several limitations, such as the loss of geometric accuracy, topological information and assembly structure. After importing CAD models into the AR module, the users can render the virtual models using OpenGL to draw all the individual triangles listing in the ASCII .stl file.

6.2.2 AR Module

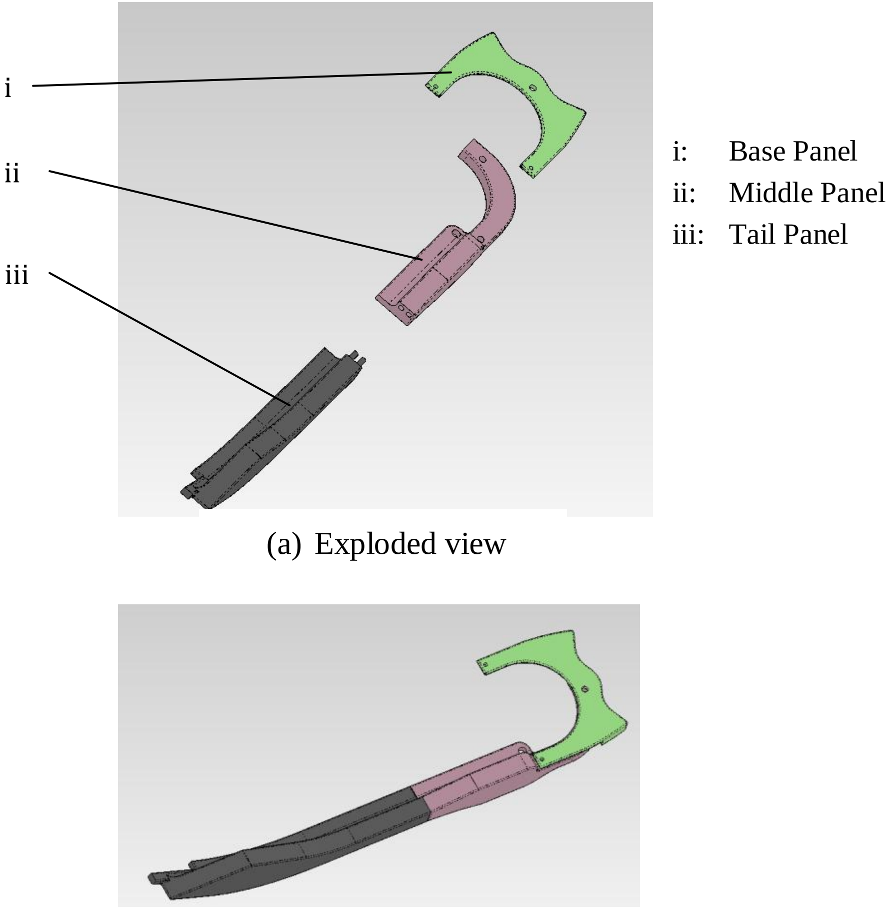
The AR module combines a user’s real world with virtual components, such as geometric models or assembly constraints, with the objective to create the sensation that virtual objects are integrated with the real environment. The ARToolKit library is used to set up the AR environment. A “Hiro” marker is used to set up the WCS. Using AR technologies, the users can see and operate the virtual models with lower cost and effort.

6.2.3 Interaction Module

Hand segmentation is realized using the RCE neural network. During the RCE network training process, the user needs to specify a region on his/her hand to obtain the training data in the L*a*b* colour space. The system executes the training process to obtain the parameters required by the RCE network. Two fingertips, namely the fingertips of the thumb and index finger, are used to control virtual models in a 2D space. When a virtual object is being controlled by the user's hand, the translation information of the virtual model is obtained by comparing the coordinates of centre point of a line segment between two fingertips. The 2D rotation information of the virtual model is calculated by comparing the direction of the line segment between two fingertips.

6.2.4 2D Assembly Simulation

In the case study of the 2D assembly simulation system, a simple product consisting of three motorcycle shield panels is used to illustrate the proposed methodology. The exploded and assembled views of the motorcycle shield panels are shown in Figure 6-2.

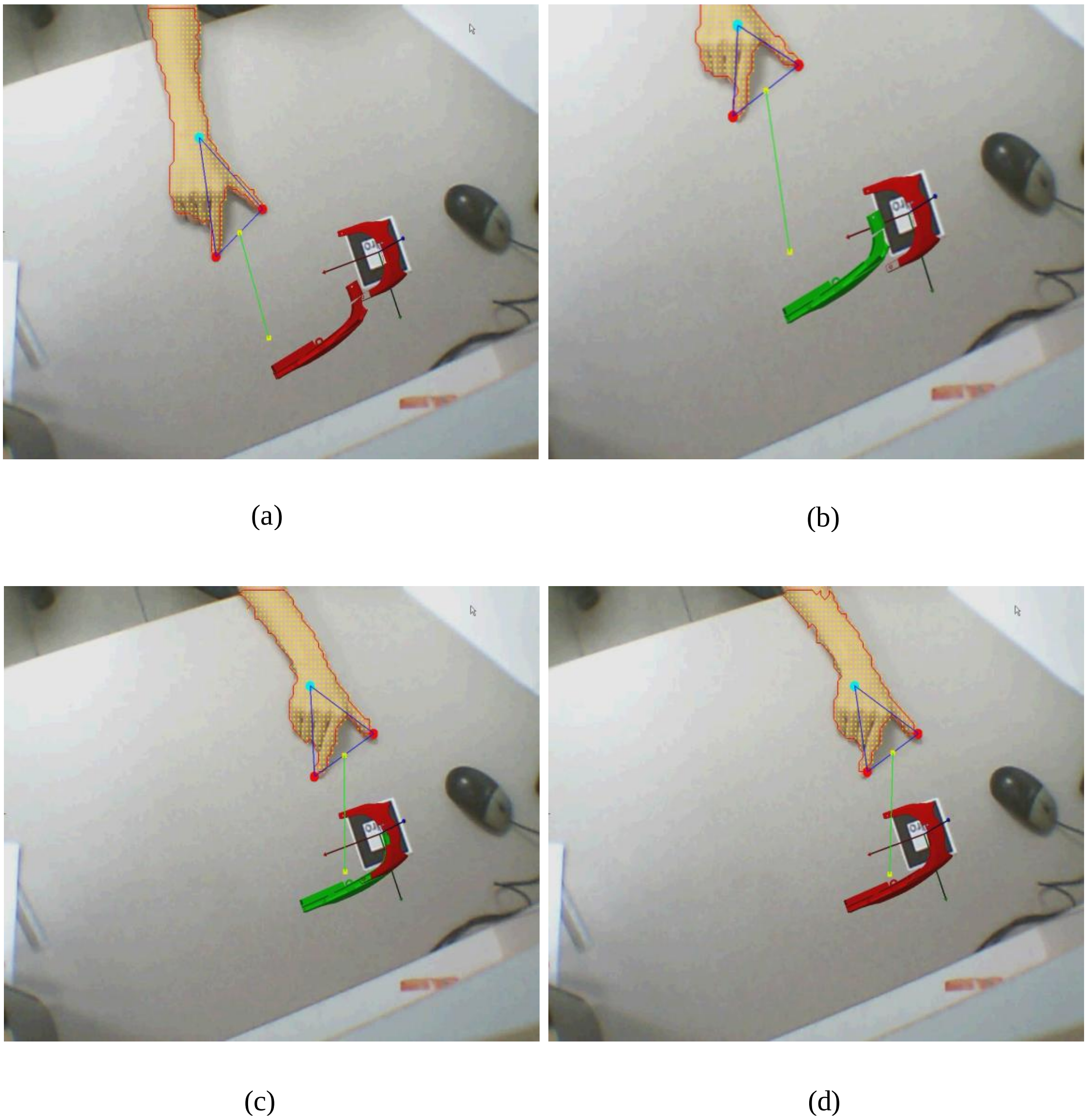


(a) Exploded view

(b) Assembled View

Figure 6-2 The motorcycle shield panels

The base panel is located stably on the marker. The middle panel is loaded into the AR environment and controlled by the user using the bare-hand interaction tool, as shown in Figures 6-3 (a-d). The colour of the virtual model is changed to green when it is being controlled by the user. The middle panel is snapped to its final position based on the geometric mating constraints stored in the computer memory when the distance between the centres of the middle panel and the base panel is smaller than a specified threshold. Next, the tail panel is assembled in the same way, as shown in Figures 6-4 (a-d).



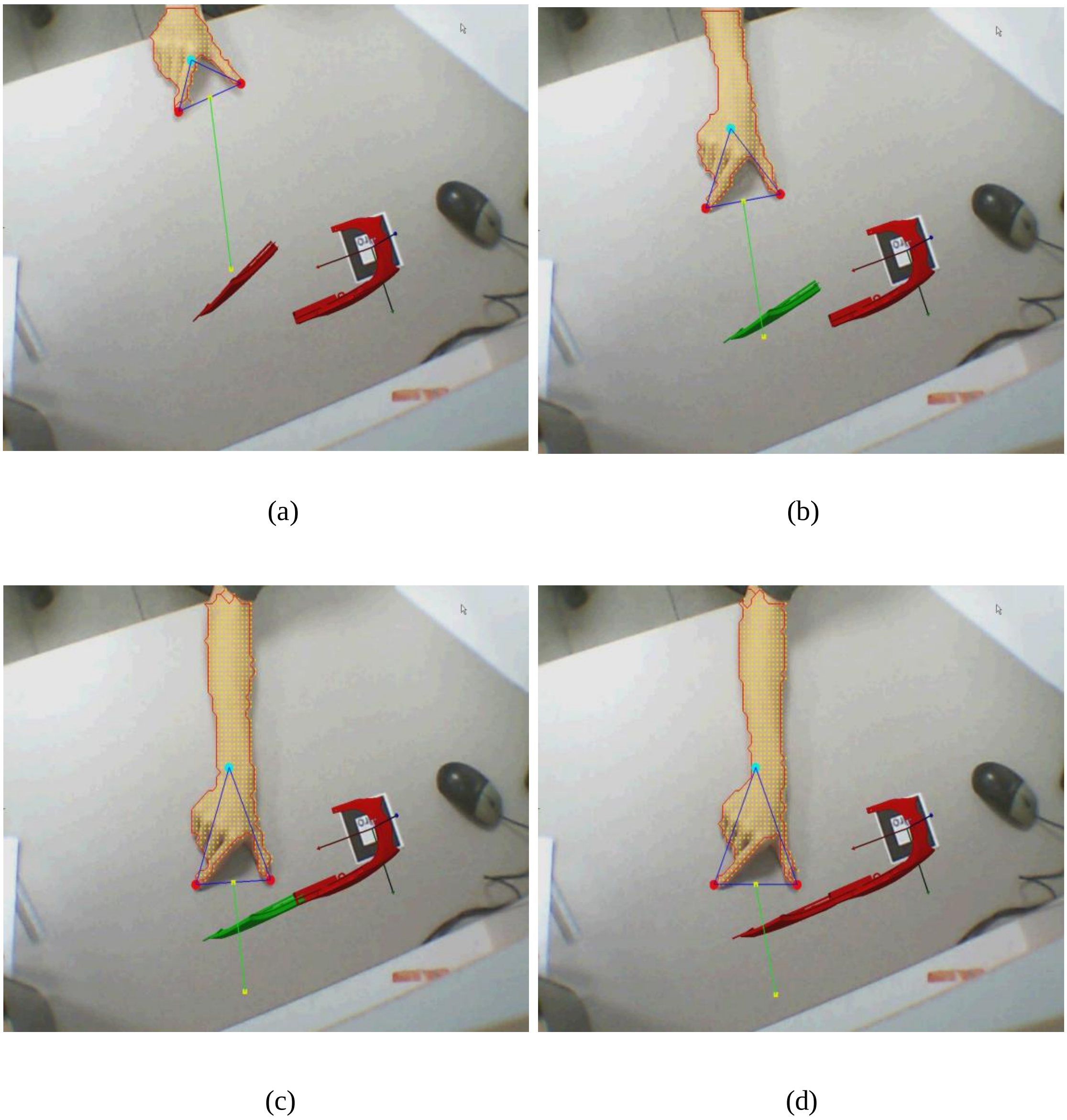
(a)

(b)

(c)

(d)

Figure 6-3 2D assembly: assemble the middle panel with the base panel



(a) (b)

(c) (d)

Figure 6-4 2D assembly: assemble the tail panel with the middle panel

6.2.5 Discussion

This case study shows that bare-hand interaction is robust and effective for manual assembly simulation in an AR environment. The user can perform the assembly in a 2D space to have cognition on the manual assembly process.

However, the hand segmentation and fingertips tracking algorithms are executed in a 2D space, meaning that there is no depth information in this system. A 3D bare-hand interaction tool is more robust and useful for the manual assembly simulation. Case studies using the 3DNBHI for manual assembly simulation and design will be discussed in the next section.

6.3 3D Manual Assembly Design and Planning

The 3DNBHI is implemented in the ARIMADP system to aid the user to perform the manual assembly simulation and design in a 3D space. The system configuration is shown in Figure 6-5.

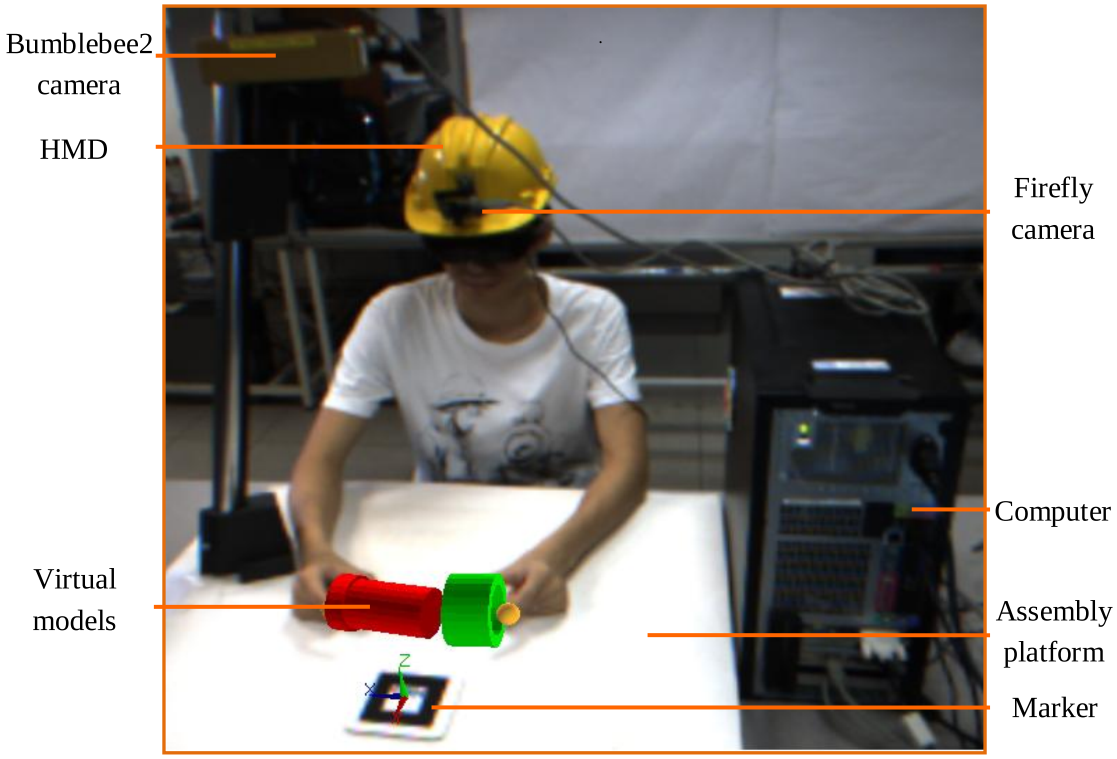
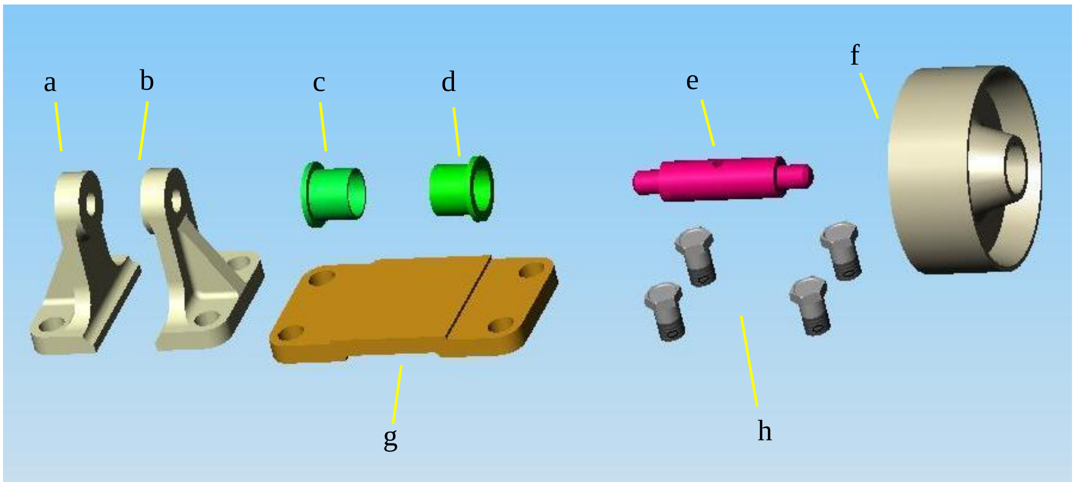


Figure 6-5 Configuration of the ARIMADP system

6.3.1 Pulley Bracket Assembly

In this case study, the user can assemble a pulley bracket using the ARIMADP system. The user can manipulate different parts of the pulley bracket with the 3DNBHI interface and assemble them together. The algorithms discussed in chapter 5 are used to set up all the geometric constraints between different parts to complete the assembly. The exploded view of the pulley bracket is shown in Figure 6-6.



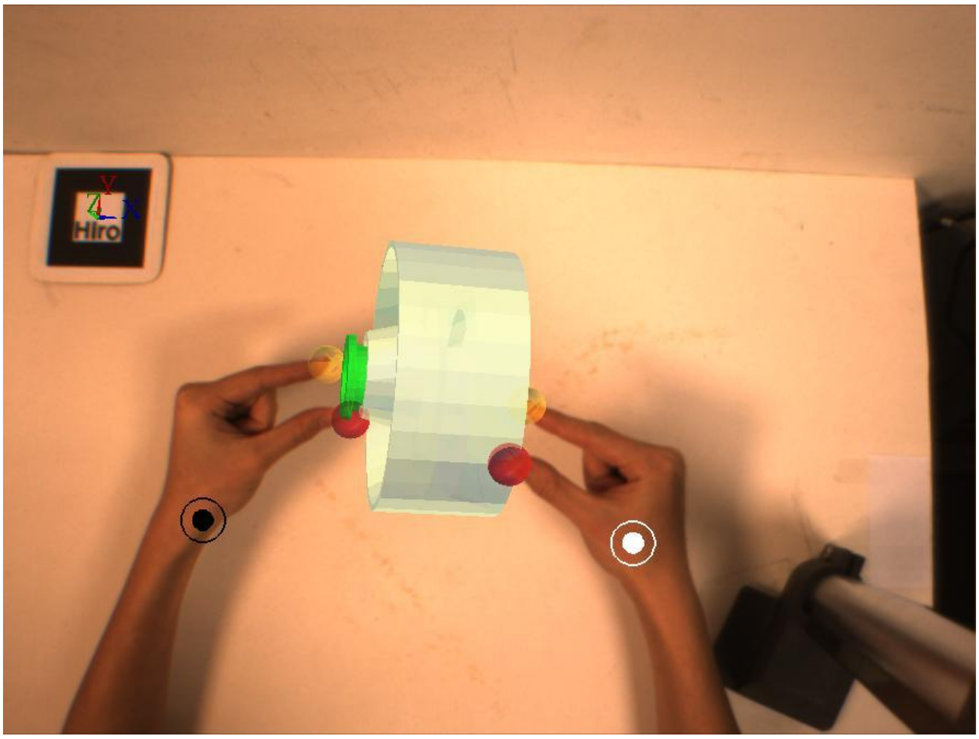
a: Left bracket
b: Right bracket
c: Left Bush
d: Right Bush
e: Spindle
f: Roller
g: Base
h: Bolts

Figure 6-6 Exploded view of the pulley bracket

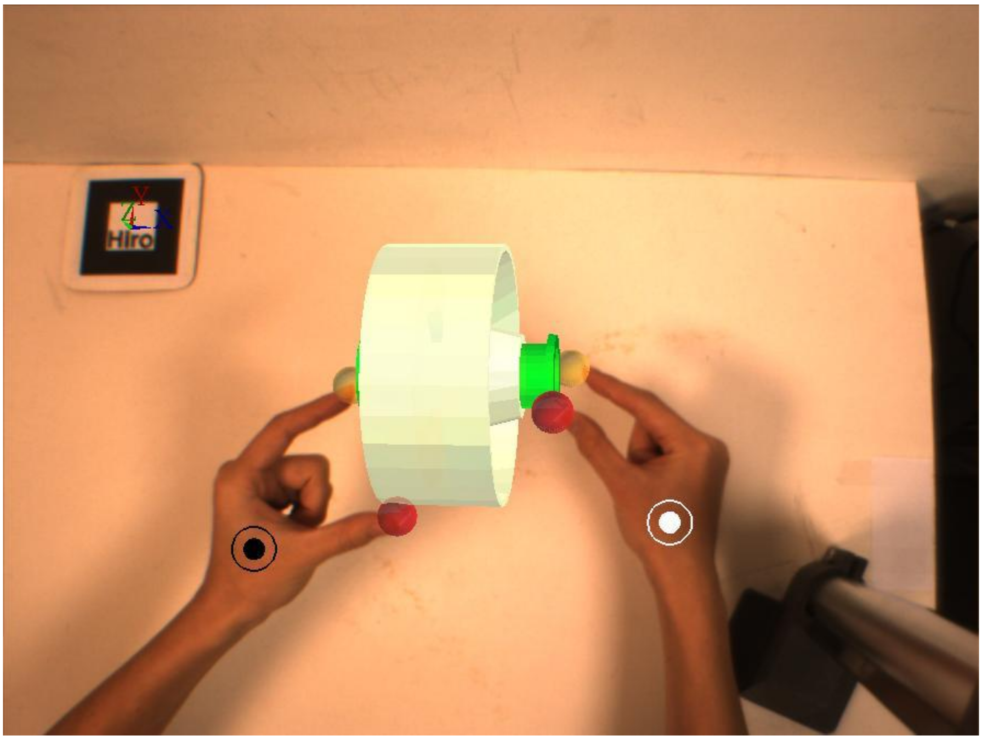
In Figure 6-7(a), the user grasps the pulley with his right hand and the left bush with his left hand and assembles these components. When the two components are colliding, the system analyses the information of the surfaces in the contact list and detects the possible constraints. A cylindrical fit constraint is recognized. Next, the position and orientation of the pulley in the user's right hand is adjusted automatically to ensure the cylindrical fit constraint is met precisely. The DOF of these two components is constrained, so that these two components can only rotate and translate along their aligned axis. This assembly process is completed when the planar surfaces of the pulley and the left bush are in contact and a co-plane constraint is recognized and confirmed. Then the user manipulates the assembled pulley and left bush and tries to assemble the whole part with the right bush in the user's right hand, as shown in Figure 6-7(b). The constraint recognition and confirmation process is similar to the process shown in Figure 6-7(a).

In Figure 6-7(c), the base and right bracket are loaded into the AR environment. The user tries to assemble these two components together. This assembly process is completed when two co-planar constraints are recognized and confirmed. In Figure 6-7(d), the user fastens two bolts using an open-end wrench to fix the right bracket onto the base. In Figure 6-7(e), the user tries to assemble the pre-assembled pulley and bracket with the spindle. After a cylindrical constraint is recognized and confirmed, the spindle can be only translated and rotated along the aligned axis. This assembly process is completed when a co-planar constraint is recognized and confirmed between the spindle and the right bracket. In Figure

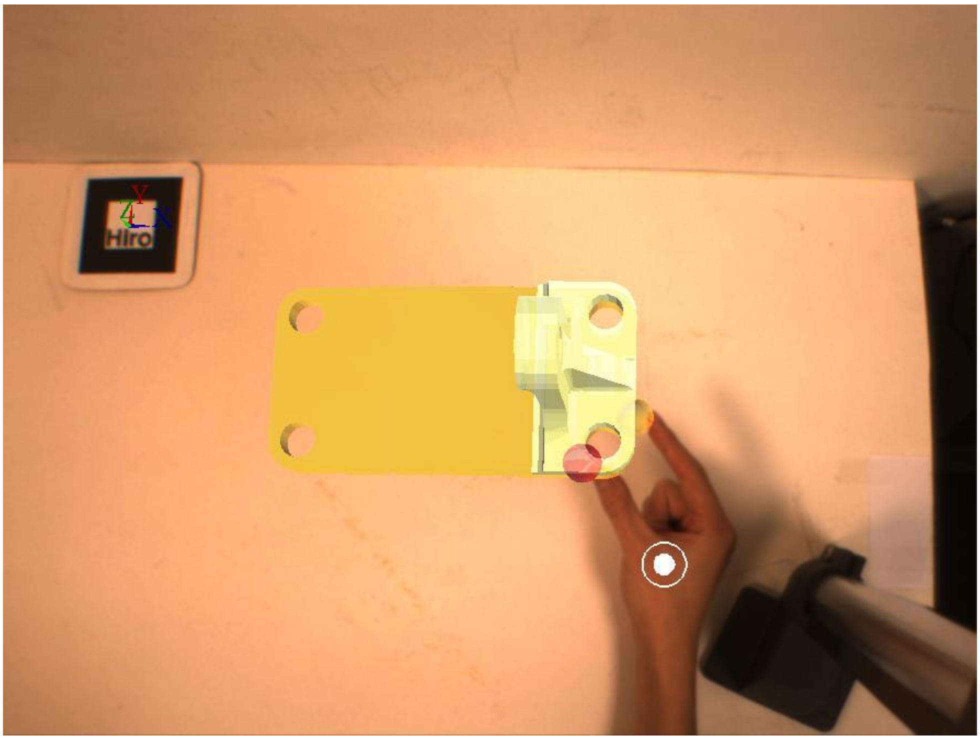
6-7(f), the user tries to assemble the sub-assembly of the pulley and bushes with the sub-assembly of base and bracket. During this process, a cylindrical constraint is recognized and confirmed. The last step is to assemble the left bracket to finish the assembly. As shown in Figure 6-7(f), the user grasps the left bracket and assembles it to the base firstly. After a co-planar constraint is recognized and confirmed, the motion of the left bracket is constrained in the planar surface of the base. Next, the left bracket on the user's left hand is moved to the spindle. A cylindrical constraint is recognized and confirmed to constrain the motion of the left bracket further. This assembly operation is completed when another co-planar constraint is confirmed between the left bracket and the base. The whole assembly process is completed when the two bolts are assembled to fix the left bracket onto the base. The whole assembly is shown in Figure 6-7(h).



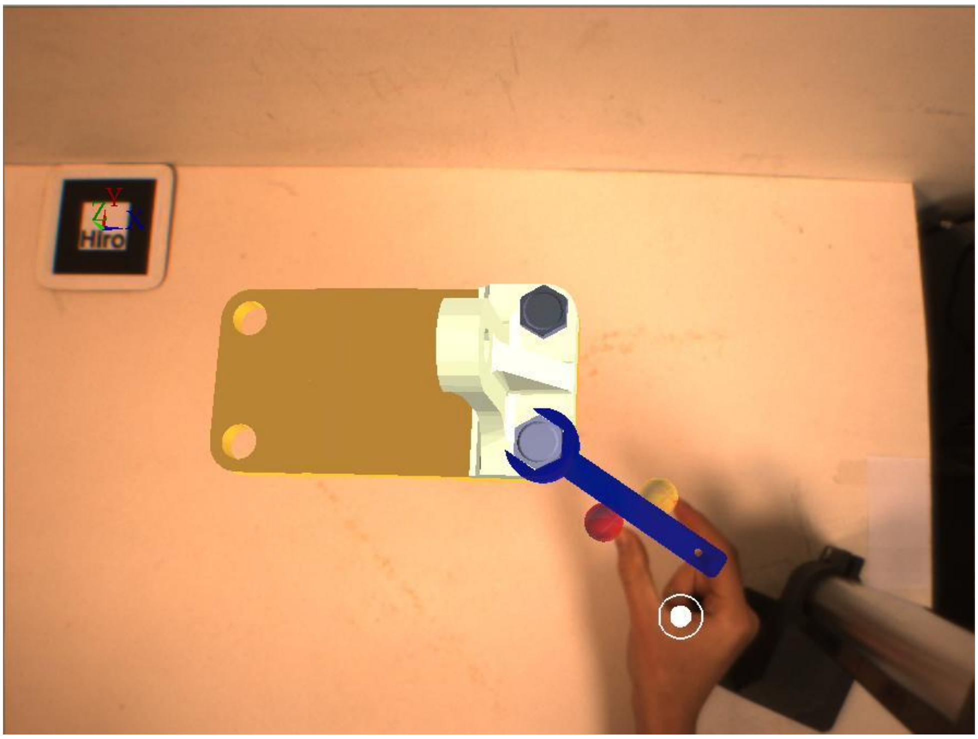
(a)



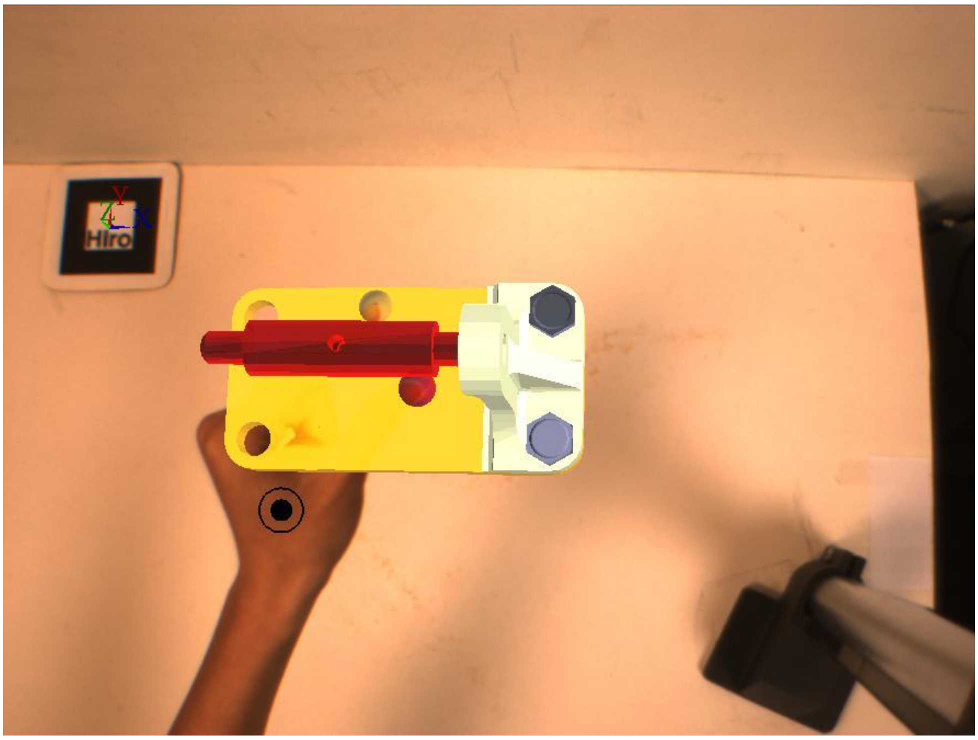
(b)



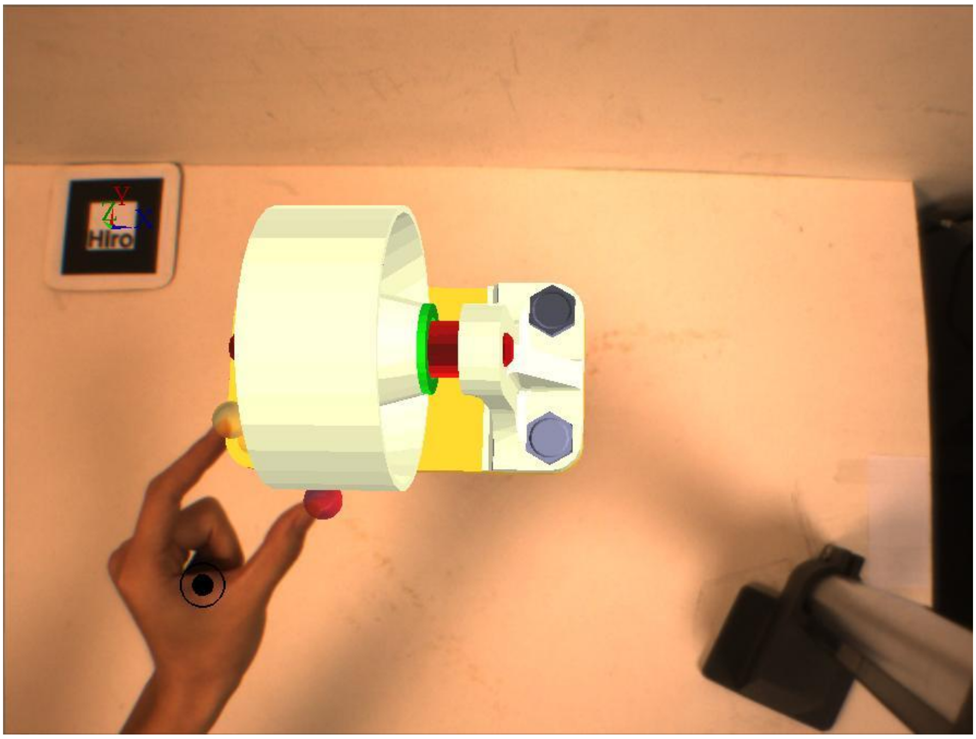
(c)



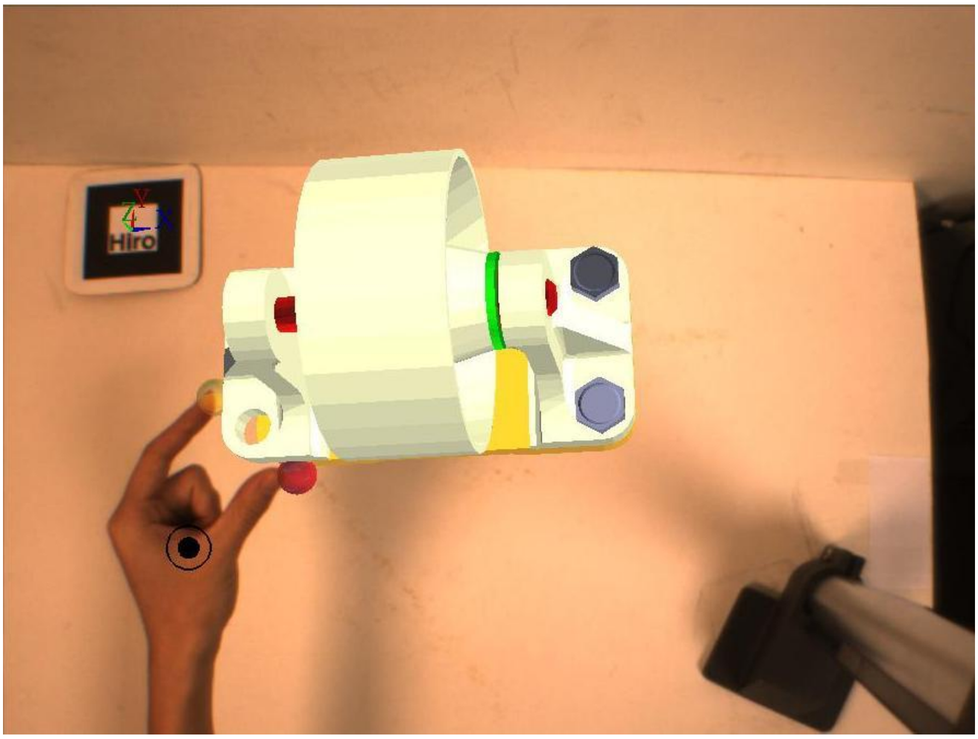
(d)



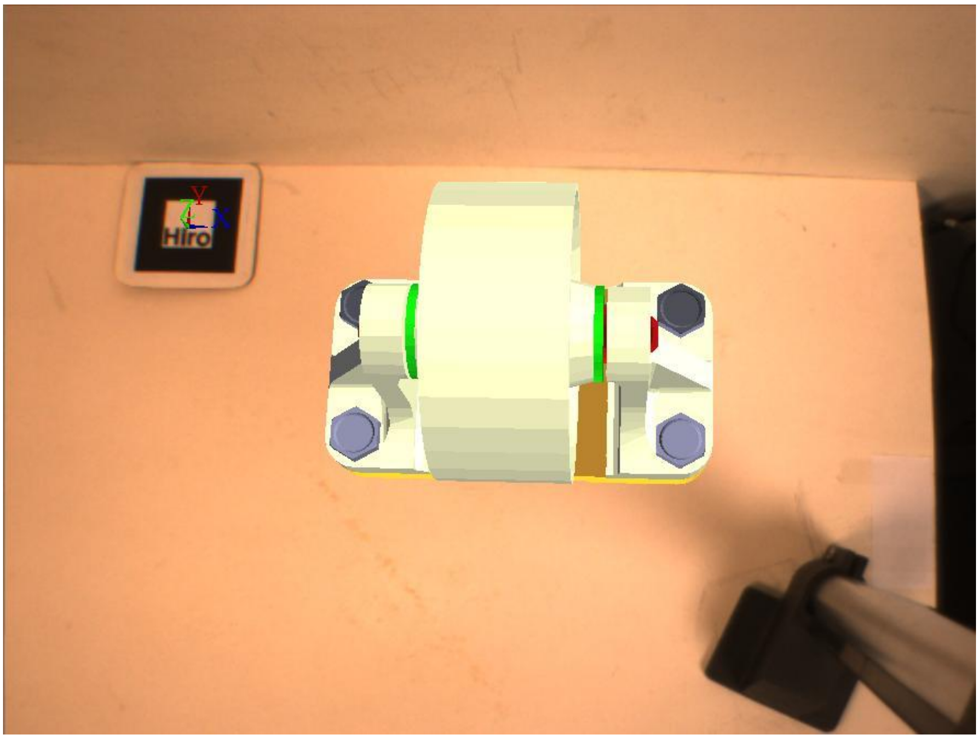
(e)



(f)



(g)

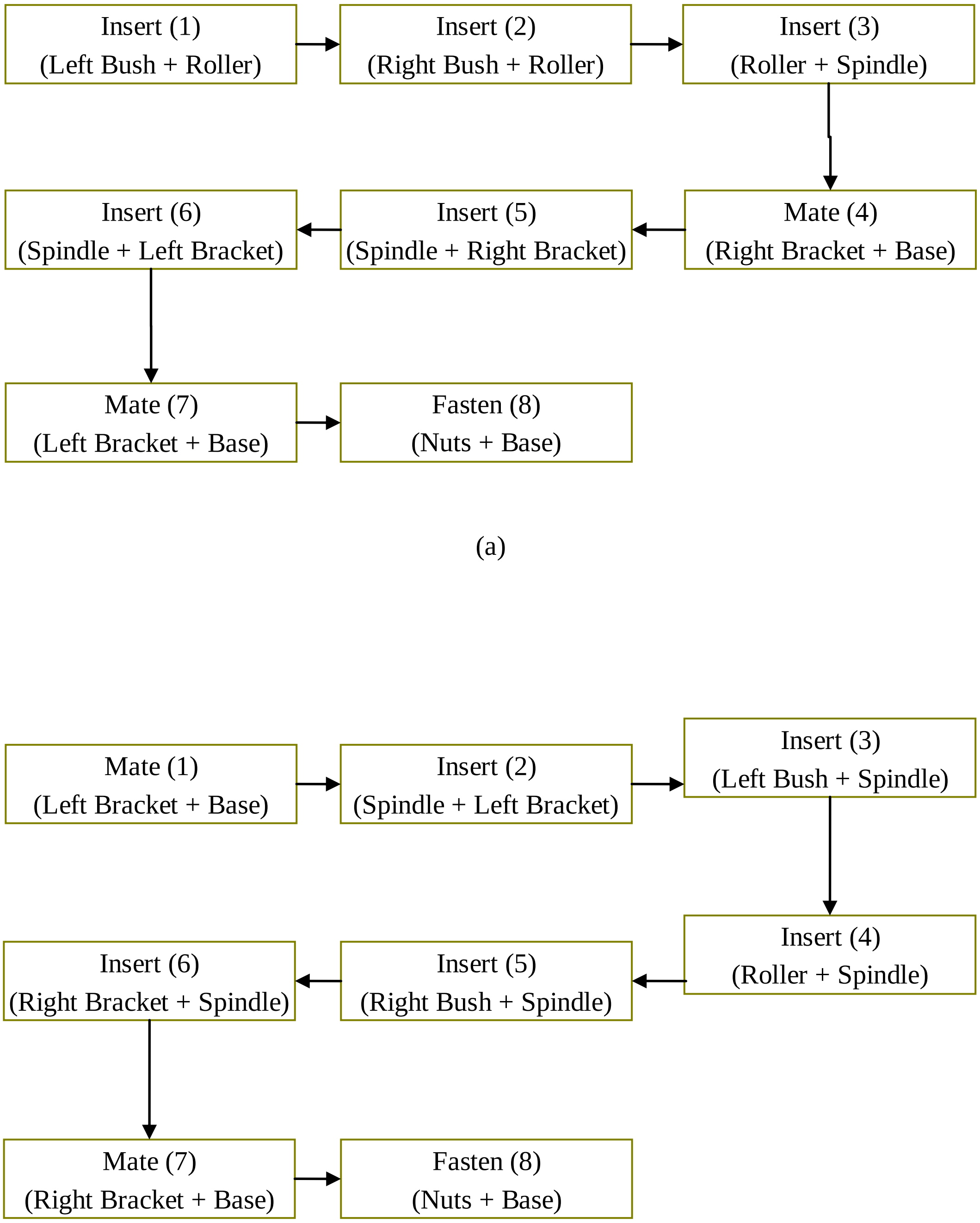


(h)

Figure 6-7 Screen Captures during the demonstration of pulley assembly

6.3.2 Pulley Assembly Planning

After the pulley assembly is available, the user can plan and evaluate the assembly sequences to obtain a near-optimal sequence. During the disassembly process, several precedence constraints can be obtained, e.g., the Roller cannot be assembled after both the Left Bush and Right Bush have been assembled to the Roller. Based on the PEs, a number of feasible sequences are generated. The soft constraint of parts-clustering is imposed to make sure all the four Nuts are assembled sequentially first before proceeding to the next assembly operation. Two feasible sequences which are depicted in Figure 6-8 are evaluated. The evaluation result is shown in Table 6-1. The coefficients used for calculating OAI are listed in Table 6-2.



(a)

(b)

Figure 6-8 Two feasible assembly sequences

Table 6-1 Evaluation result

Criteria		Sequence (a)	Sequence (b)
MI	HSI	1.5	0
	OPI	3.8	3.8
CI	CCI	3	3
	DCI	5	3
	TCI	0	0
OAI		12.7	5.1

Table 6-2 Coefficients used for calculating OAI

Coefficients	k_H	k_O	k_C	k_D	k_T	k_1	k_2
Value	0.7	0.3	0.3	0.4	0.3	0.5	0.5

The main cause for the hand strain detected in sequence (a) is that the Roller is blocking the Spindle to the Right Bracket in operation Insert (5). As shown in Table 6-1, sequence (b) has a lower OAI value, which means sequence (b) is more efficient than sequence (a). Considering another soft constraint of first-part-definition that the assembly sequence must start with the Base part, the user concluded that sequence (b) is the optimal assembly sequence.

6.4 Summary

This chapter presents the architecture of a 2D manual assembly system and the configuration of the ARIMADP system. The case studies illustrate how the presented methodology can be used for manual assembly simulation and design in an AR environment. Based on the experiments and results, the bare-hand interaction interface developed can aid the user in manual assembly simulation in an efficient manner. A user can obtain an intuitive investigation of all the mating relations between the components and the functions of the assembly and components.

Chapter 7 Conclusions and Recommendations

7.1 Overview

The main objective of this research, *i.e.*, to explore the application of AR in manual assembly simulation and design using natural bare-hand interaction, has been fulfilled. This chapter summarizes the thesis and identifies the major contributions from this research work. Future research directions are then presented.

7.2 Conclusions and Contributions

The overall objective of this research is to develop an assembly simulation and design system in an AR environment, allowing users to design and plan an assembly robustly and efficiently. In this thesis, to make assembly faster, easier and less costly, a systematic methodology for manual assembly simulation and design in an AR environment has been developed. To achieve natural and intuitive HCI, the human hands can be used as the interaction devices. A robust 3D bare-hand interaction tool has been developed in this thesis to make it possible for users to manipulate and assemble virtual components in an intuitive way.

This thesis makes contributions in two major areas. The first area is introducing a 3D robust bare-hand interaction tool into the AR environment. The second area is developing a manual assembly simulation and design methodology in an AR environment.

7.2.1 Bare-hand Interaction Tool

A 3D bare-hand tool has been developed to improve the HCI for performing the assembly operations in the AR environment. The tips of thumbs and the index fingers of both of the user's hands are used for achieving 3D pinch operations. In comparison with data-gloves, the interaction tool developed is less expensive and more convenient for users to operate. It can be used by the user to manipulate and orientate parts or subassemblies simultaneously. This is important for the manual assembly task in an AR environment, because it not only reduces the number of assembly steps but also makes assembly simulation and design more realistic, by closely replicating real world interactions.

7.2.2 Methodology for Manual Assembly Design

A methodology for assembly simulation and design in an AR environment has been discussed in this research. A new hybrid method for augmented assembly based on constraint analysis is developed, which predicts users' assembly intent robustly without access to auxiliary CAD information, such as predefined constraints and assembly hierarchy. This method improves manual assembly process significantly by avoiding time-consuming and cumbersome model pre-processing requirements whenever a new assembly scenario is imported into the augmented environment.

The methodology adopted a hand strain determination method (put reference here) and a method to determine an overall assembly cost for an assembly sequence considering the ease of handling of the assembly parts in this sequence and the continuity of the operations in this sequence (put reference here). By adopting these methods, the ARIMADP system can assist a user to determine a near to optimal assembly sequence for a product assembly.

7.3 Recommendations for Future Work

In terms of system implementation and integration, there are several aspects that have not been covered in this thesis which require further studies and investigations.

Currently, the 3D bare-hand interaction interface can only provide pinch operations for the users, and this constrains the flexibility of the human hands. Further research will be necessary to develop an effective algorithm for providing 3D manipulative gesture recognition in real time so that the user can interact with the virtual objects more naturally and intuitively using his/her whole hand. The 3D manipulative gestures can be achieved by projecting a 3D hand model to an image space and matching it with the observed image features. The challenge is how this process can be done in real time. The wrist angle strain calculation result will be more accurate as the user can mimic real hand operations more closely. Therefore, The HSI evaluation will also benefit from adopting 3D manipulative gestures. In addition, it will be very helpful to introduce tactile and force feedback into the bare-hand interface to allow users "feel" not only the virtual component but also the assembly processes.

In this research, only virtual components are used for manual assembly simulation and design. Introducing real parts into the system will make the system more flexible and effective. 3D model-based object tracking is a process of locating moving 3D objects using a camera. Future study can focus on assembly simulation and design using both real and virtual assembly components.

The assembly simulation and design methodology developed in this thesis provides the users a possibility to simulate the assembly. Currently, penetrations between virtual components are not prevented which may make the assembly process less intuitive. To address this problem, future studies should use collision detection package or physical simulation engine to enhance the realism of the augmented assembly environment. In addition, the methodology mainly focuses manual assembly. Further study is needed to extent the developed methodology to different industrial automated assembly.

Publications from This Research

1. Wang, Z.B., Shen, Y., Ong, S.K. and Nee, A.Y.C., 2009, Assembly design and evaluation based on bare-hand interaction in an augmented reality environment, *International Conference on CyberWorlds*, pp.21-28
2. Ong, S.K. and Wang, Z.B., 2011, "Augmented assembly technologies based on 3D bare-hand interaction", *CIRP Annals - Manufacturing Technology*, Vol. 60, No. 1, pp. 1-4
3. Z.B. Wang, L.X., Ng, S.K. Ong and A.Y.C. Nee, October 2012, "Assembly Planning and Evaluation in an Augmented Reality Environment", International Journal of Production Research, submitted October 2012.
4. Z.B. Wang, Y. Shen, S.K. Ong and A.Y.C. Nee, Oct 2012, A 3D Bare-Hand Interaction Methodology for Augmented Reality Applications, International Journal of Human Computer Interaction, submitted Oct 2012.
5. Z.B. Wang, S.K. Ong and A.Y.C. Nee, October 2012, Augmented Reality Aided Interactive Manual Assembly Design, International Journal of Advanced Manufacturing Technology, submitted Oct 2012.

References

[Anantha et al, 1996]

Anantha R., Kramer G.A. and Crawford R.H., 1996, Assembly modelling by geometric constraint satisfaction, Computer-Aided Design, Vol. 28, No. 9, pp. 707-722

[ARToolKit, 1998]

ARToolKit, A software library for building AR applications, http://www.hitl.washington.edu/artoolkit/, last accessed on Jul, 2011

[ARToolKitPlus, 2006]

ARToolKitPlus, A software library for building AR applications, http://handheldar.icg.tugraz.at/artoolkitplus.php, last accessed on Jul, 2011

[Azuma, 1997]

Azuma R., 1997, "A Survey of Augmented Reality", Presence: Teleoperators and Virtual Environments, Vol. 6, No. 4, pp. 355-385

[Balcisoy et al, 2000]

Balcisoy S., Kallmann M., Fua P. and Thalmann, D., 2000, "A Framework for Rapid Evaluation of Prototype with Augmented Reality", Proceedings VRST 2000, pp. 61-66

[Baldwin et al, 1991]

Baldwin, D. F., Abell, T. E., Lui, M. C. M., De Fazio, T. L. and Whitney, D. E., 1991, "An integrated computer aid for generating and evaluating assembly sequences for mechanical products", IEEE Trans Robotics Automat, Vol. 7, No. 1, pp. 78-94

[Bane and Hollerer, 2004]
Bane, R., and Hollerer, T., 2004, "Interactive Tools for Virtual X-Ray Vision in Mobile Augmented Reality", Proceedings of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 231-239

[Bradski, 1998]
Bradski, G.R., 1998, "Computer vision face tracking for use in a perceptual user interface", Proceedings Fourth IEEE Workshop on Applications of Computer Vision, pp. 214-219.

[Buchmann et al, 2004]
Buchmann, V., Violich, S., Billinghurst, M., and Cockburn, A., 2004, "FingARtips – Gesture Based Direct Manipulation in Augmented Reality", Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, pp. 212 – 221

[Carey and Gallwey, 2002]
Carey, E.J. and Gallwey, T.J., 2002, "Effects of wrist posture, pace and exertion on discomfort". International Journal of Industrial Ergonomics, Vol. 29, No. 2, pp. 85-94

[Caudell and Mizell, 1991]
Caudell, T.P. and Mizell, D.W., 1991, "Augmented reality: an application of heads-up display technology to manual manufacturing processes", Paper

Presented at the Twenty-Fifth Hawaii International Conference on System Sciences, Vol. 2, pp. 659-669

[Chai and Ngan, 1999]
Chai, D., and Ngan, K.N., 1999, “Face segmentation using skin-color map in videophone applications,” IEEE Transaction on Circuits and Systems for video Technology, Vol. 9, No. 4, pp. 551-564

[Chakrabarty and Wolter, 1997]
Chakrabarty, S. and Wolter, J., 1997, “A structure-oriented approach to assembly sequence planning”, IEEE Trans Robotics Automat, Vol. 13, No. 1, pp.258-263

[Chang and Wang, 2007]
Chang, S.W., Wang, M.J.J., 2007, “Digital human modeling and workplace evaluation: using an automobile assembly task as an example”, Human Factors and Ergonomics in Manufacturing, Vol. 17, No. 5, pp. 445-455

[Chari and Bouzerdoum, 2000]
Chai, D. and Bouzerdoum, A., 2000, “A Bayesian approach to skin colour classification in YCbCr colour space”, Intelligent Systems and Technologies for the New Millennium, pp. 421–424

[Chen et al, 2005]
Chen, X. W., Xu, N. and Li, Y., 2005, "A Virtual Environment for Collaborative assembly", Second International Conference on Embedded Software and Systems (ICESS'05), pp. 441-448

[Chen et al, 2007]

Chen, Q., Rahman, A.M., Shen, X.J., El Saddik, A. and Georganas, N.D., 2007, "Navigating a 3D virtual environment of learning objects by hand gestures", International Journal of Advanced Media and Communication, Vol. 1, No. 4, pp. 351-368

[Chimienti et al, 2010]

Chimienti, V., Iliano, S., Dassisti, M., Dini, G. and Failli, F., 2010, "Guidelines for Implementing Augmented Reality Procedures in Assisting Assembly Operations", IFIP Advances in Information and Communication Technology, Vol. 315, pp. 174-179

[Chintamani et al, 2010]

Chintamani, K., Cao, A., Ellis, R.D. and Pandya, A.K., 2010, "Improved tele-manipulator navigation during display-control misalignments using augmented reality cues", IEEE Transactions on Systems, Man and Cybernetics, Part A (Systems and Humans), Vol. 40, No. 1, pp. 29-39

[Choi et al, 2009]

Choi, Y.K., Lee, D. and Cho, Y., 2009, "An approach to multi-criteria assembly sequence planning using genetic algorithms", International Journal of Advanced Manufacturing Technology, Vol. 42, No. 1, pp. 180-188

[Crowson, 2006]

Crowson, R., 2006, Handbook of manufacturing engineering, CRC Press, USA

[CyberGrasp, 2010]

http://www.vrlogic.com/html/immersion/cybergrasp.html, last accessed Jun, 2010.

[De Crescenzio et al, 2011]

De Crescenzio, F., Fantini, M., Persiani, F., Di Stefano, L., Azzari, P. and Salti, S., “Augmented Reality for Aircraft Maintenance Training and Operations Support”, 2011, Computer Graphics and Applications, pp. 96-101

[De Fazio and Whitney, 1987]

De Fazio T.L. and Whitney D.E., 1987, “Simplified generation of all mechanical assembly sequences”, IEEE Trans Robot Automat, Vol. 3, No.6, pp.640–658

[Deneux, 1999]

Deneux D., 1999, Introduction to assembly features: an illustrated synthesis methodology, Journal of Intelligent Manufacturing, Vol. 10, pp. 29-39

[Dhawale et al, 2006]

Dhawale, P., Masoodian, M. and Rogers, B., 2006, “Bare-hand 3D gesture input to interactive systems”, Proceedings of the 7th Int. Conf. NZ Chapter of the ACM's SIG on Human-Computer Interaction, Vol. 158, pp. 25-32

[Doil et al, 2003]

Doil, F., Schreiber, W., Alt, T., and Patron, C., 2003, “Augmented Reality for Manufacturing Planning”, Proceedings of the Workshop on Virtual Environments, Vol. 39, pp. 71-76

[Dong et al, 2005]

Dong, T.Y., Tong, R.F., Zhang, L. and Dong, J.X., 2005, “A collaborative approach to assembly sequence planning”, Advanced Engineering Informatics, Vol. 19, No. 2, pp. 155-168

[Du and Charbon, 2007]

Du, H. and Charbon, E., 2007, "3D Hand Model Fitting for Virtual Keyboard System", Proceedings - IEEE Workshop on Applications of Computer Vision, pp. 31

[Eng et al, 1999]

Eng, T.H., Ling, Z.K., Olson, W. and McLean, C., 1999, "Feature-based assembly modelling and sequence generation", Computers and Industrial Engineering, Vol. 36, No. 1, pp. 17-33

[Fan and Dong, 2001]

Fan, J. and Dong, J.X., 2001, "KVAS: a Knowledge-based Virtual Assembly System", IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, pp.1041-1046

[Fang et al, 2011]

Fang, H.C., Ong, S.K. and Nee, A.Y.C., 2011, "Interactive Robot Trajectory Planning and Simulation using Augmented Reality", Robotics and Computer-Integrated Manufacturing, in press

[Feiner et al, 1993]

Feiner, S., MacIntyre, G. and Seligmann, D., 1993, "Knowledge-Based Augmented Reality", Communications of the ACM, Vol. 36, No. 7, pp. 52-63

[Fischer et al, 2007]

Fischer, J., Eichler, M., Bartz, D. and Strasser, W., 2007, "A hybrid tracking method for surgical augmented reality", Computers & Graphics, Vol. 31, No. 1, pp. 39-52

[Fiorentino et al, 2002]

Fiorentino, M., Amicis, R., Monno, G. and Stork, A., 2002, “Spacedesign: A Mixed Reality Workspace for Aesthetic Industrial Design”, Proceedings of the 1st International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, 30 Sept.-1 Oct. 2002, pp. 86 – 96

[Fong et al, 2010]

Fong, W.T., Ong, S.K. and Nee, A.Y.C., 2010, “Marker-less Computer Vision Tracking for Augmented Reality”, IADIS Computer Graphics, Visualization, Computer Vision and Image Processing 2010 (CGVCVIP 2010) Conference, 27-29 July, Freiburg, Germany, pp. 85-92

[Fritzsche, 2010]

Fritzsche, L., 2010, “Ergonomics risk assessment with digital human models in car assembly: Simulation versus real life”, Human Factors and Ergonomics in Manufacturing, Vol. 20, No. 4, pp. 287-299

[Gottschlich et al, 1994]

Gottschlich, S., Ramos, C. and Lyons, D., 1994, “Assembly and task planning: A taxonomy,” IEEE Robotics Automation & Magazine., Vol. 1, No. 3, pp. 4–12.

[Gottschalk et al, 1996]

Gottschalk, S., Lin, M.C. and Maocha, D., 1996, “OBB-Tree: A Hierarchical Structure for Rapid Interference Detection”, Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, pp. 171-180

[Grabowik et al, 2005]

Grabowik, C., Kalinowski, K.and Monica, Z., 2005, “Integration of the CAD/CAPP/PPC systems”, Journal of Materials Processing Technology, Vol. 164-165, pp. 1358-1368

[Greenspan et al, 2001]

Greenspan, H., Goldberger, J. and Eshet, I., 2001, “Mixture Model for Face Colour Modelling and Segmentation,” Pattern Recognition Letters, Vol. 22, No. 14, pp. 1525-1536

[Hakkarainen et al, 2008]

Hakkarainen, M., Woodward, C. and Billinghurst, M., 2008, “Augmented Assembly using a Mobile Phone”, International Symposium on Mixed and Augmented Reality, pp. 167-168

[Haller et al, 2006]

Haller, M., Brandl, P., Leithinger, D., Leitner, J., Seifried, T. and M. Billinghurst, 2006, "Shared Design Space: Sketching Ideas Using Digital Pens and a Large Augmented Tabletop Setup," in ICAT 2006, Lecture Notes in Computer Science 4282, pp. 185-196

[Halttunen and Tuikka, 2000]

Halttunen, V. and Tuikka, T., 2000, “Augmenting virtual prototype with physical objects”, Proceedings of the Working Conference on Advanced Visual Interfaces, ACM 2000, pp. 305-306

[Hamer et al, 2009]

Hamer, H., Schindler, K., Koller-Meier, E. and Van Gool, L., 2009, “Tracking a hand manipulating an object”, Proceedings of the IEEE International Conference on Computer Vision, pp. 1475-1482

[Harders et al, 2007]

Harders, M., Bianchi, G. and Knoerlein, B., 2007, “Multimodal Augmented Reality in Medicine”, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 4555, pp. 652-658

[Hartley and Zisserman, 2003]

Hartley, R. and Zisserman, A., 2003, Multiple View Geometry in Computer Vision, Cambridge University Press

[Hayashi et al, 2005]

Hayashi, K., Kato, H., Nishida, S., 2005, “Occlusion detection of real objects using contour based stereo matching”, In proceedings of the 2005 International Conference on Augmented Tele-Existence, Vol. 157, pp. 180-186

[Henderson and Feiner, 2009]

Henderson, S.J. and Feiner, S., 2009, “Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret”, Mixed and Augmented Reality, pp. 135-144

[Hilliges et al, 2009]

Hilliges, O., Izadi, S., Wilson, A. D., Hodges, S., Garcia-Mendoza, A., and Butz, A., 2009, “Interactions in the air: adding further depth to interactive tabletops”, Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, pp. 139-148

[Homem de Mello and Sanderson, 1991]

Homem de Mello, L.S. and Sanderson, A.C., 1991, “Representation of Mechanical Assembly Sequences”, IEEE Transactions on Robotics and Automation, Vol. 7, No. 2, pp.211-227

[Homem and Sanderson, 1991]

Homem de Mello, L.S. and Sanderson, A.C., 1991, “A correct and complete algorithm for the generation of mechanical assembly sequences”, IEEE Trans Robotics Automat, Vol. 7, No. 2, pp. 228-240

[Imrhan and Rahman, 1995]

Imrhan, S.N. and Rahman, R., 1995, “The effects of pinch width on pinch strengths of adult males using realistic pinch-handle coupling”, International Journal of Industrial Ergonomics, Vol. 16, No. 2, pp.123-134

[Jayaram et al, 1997]

Jayaram, S., Connacher, H. I. and Lyons, K. W., 1997, “Virtual assembly using virtual reality techniques”, Computer-Aided Design, Vol. 29, No. 8, pp.575-584

[Jayaram et al, 2007]

Jayaram, S., Jayaram, U., Kim, Y. J., DeChenne, C., Lyons, K. W., Palmer, C. and Mitsui, T., 2007, “Industry case studies in the use of immersive virtual assembly”, Virtual Reality, Vol. 11, No, 4, pp.217-228

[Jayaram et al, 1999]

Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., and Hart, P., 1999, “VADE: A Virtual Assembly Design Environment”, IEEE Computer Graphics and Applications, Vol. 19, No. 6, pp. 44–50

[Jones and Wilson, 1996]

Jones, R.E. and Wilson, R.H. (1996), A survey of constraints in automated assembly planning, IEEE International Conference on Robotics and Automation, Vol. 2, pp. 1525-1532

[Kasson and Plouffe, 1992]

Kasson, J.M. and Plouffe, W., 1992, "Analysis of selected computer interchange colour spaces", ACM Transactions on Graphics, Vol. 11, No. 4, pp. 373-405

[Kato and Billinghurst, 1999]

Kato, H. and Billinghurst, M., 1999, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system", Proceedings of The 2nd International Workshop on Augmented Reality, San Francisco, CA, USA, October, pp. 85-99

[Kato et al, 1999]

Kato, H., Billinghurst, M., Blanding, R. and May, R., 1999, ARToolKit Manual, PC version 2.11. Human Interface Technology Laboratory, University of Washington

[Klein and Drummond, 2003]

Klein, G. and Drummond, T., 2003, "Robust visual tracking for non-instrumented augmented reality", International Symposium on Mixed and Augmented Reality, pp. 113-122

[Khan et al, 2010]

Khan, A.A., O'Sullivan, L. and Gallwey, T.J., 2010, "Effect on discomfort of frequency of wrist exertions combined with wrist articulations and forearm rotation", International Journal of Industrial Ergonomics, Vol. 40, No. 5, pp. 492-503

[Kim and Dey, 2010]
Kim, S.J. and Dey, A.K., 2010, "AR interfacing with prototype 3D applications based on user-centered interactivity", CAD Computer Aided Design, Vol. 42, No. 5, pp. 373-386

[Kim and Fellner, 2004]
Kim, H. and Fellner, D.W., 2004, "Interaction with hand gesture for a back-projection wall", Proceedings of Computer Graphics International, pp. 395-402

[Klinker et al, 2002]
Klinker, G., Dutoit, A. H., Bauer, M., Bayer, J., Novak, V. and Matzke, D., 2002, "Fata Morgana – A Presentation System for Product Design", IEEE and ACM International Symposium on Mixed and Augmented Reality, Darmstadt, Germany, September 30-October 1, pp. 76 – 86

[Kojima et al, 2001]
Kojima, Y., Yasumuro, Y., Sasaki, H., Kanaya, I., Oshiro, O., Kuroda, T., Manabe, S. and Chihara, K., 2001, "Hand manipulation of virtual objects in wearable augmented reality", Proceedings Seventh International Conference on Virtual Systems and Multimedia, pp. 463-469

[Kölsch, 2004]
Kölsch, M., 2004, "Vision Based Hand Gesture Interfaces for Wearable Computing and Virtual Environments", PhD Thesis, University of California

[Kölsch et al, 2004]

Kölsch, M., Turk, M. and Höllerer, T., 2004 “Vision-Based Interfaces for Mobility”, International Conference on Mobile and Ubiquitous Systems, pp. 86-94

[Lee and Höllerer, 2007]

Lee, T. and Höllerer, T., 2007, “Handy AR: Markerless inspection of augmented reality objects using fingertip tracking”, International Symposium on Wearable Computers, pp. 83-90

[Lee et al, 2008]

Lee, G.A., Kang, H. and Son, W., 2008, “MIRAAGE: A touch screen based mixed reality interface for space planning applications”, In Proc. IEEE Virtual Reality 2008, pp. 273-274

[Lee et al, 2011]

Lee, J.W., Han, S.H. and Yang, J.S., 2011, “Construction of a computer-simulated mixed reality environment for virtual factory layout planning”, Computers in Industry, Vol. 62, No. 1, pp. 86-98

[Lepetit et al, 2003]

Lepetit, V., Vacchetti, L., Thalmann, D. and Fua, P., 2003, “Fully Automated and Stable Registration for Augmented Reality Applications”, International Symposium on Mixed and Augmented Reality, pp. 93-101

[Li et al, 2009]

Li, S.Q., Peng, T., Xu, C., Fu, Y. and Liu, Y., 2009, “A mixed reality-based assembly verification and training platform”, Proceedings of the Third International Conference (VMR 2009), pp.576-585

[Lin and Chang, 1993]

Lin, A.C. and Chang, T.C., 1993, “3D maps: three-dimensional mechanical assembly planning system”, Journal of Manufacturing Systems, Vol. 12, No. 6, pp.437-456

[Liverani et al, 2004]

Liverani, A., Amati, G. and Caligiana, G., 2004, “A CAD-augmented Reality Integrated Environment for Assembly Sequence Check and Interactive Validation”, Concurrent Engineering: Research and Applications, Vol. 12, No. 1, pp. 67-77

[Malbezin et al, 2002]

Malbezin, P., Piekarski, W. and Thomas, B. H., 2002, “Measuring ARToolKit Accuracy in Long Distance Tracking Experiments”, Proceedings of 1st International Augmented Reality Toolkit Workshop, Darmstadt, Germany, September, pp. 28-29

[Marcelino et al, 2003]

Marcelino, L., Murray, N. and Fernando, T., 2003, “A Constraint Manager to Support Virtual Maintainability”, Computers and Graphics, Vol. 27, No. 1, pp.19-26

[Martx, 2006]

Martx, P., 2006, “OpenGL Distilled”, Addison-Wesley

[Menser and Wien, 2000]

Menser, B. and Wien, M., 2000, "Segmentation and Tracking of Facial Regions in Colour Image Sequences," Proceedings of the SPIE - The International Society for Optical Engineering, Vol. 4067, pt.1-3, pp. 731-740

[Milgram and Kishino, 1994]

Milgram P. and Kishino F., 1994, "A Taxonomy of Mixed Reality Virtual Displays", Information and Communication Engineers (IEICE) Transactions on Information and Systems, Vol. E77-D, No. 9, pp. 1321-1329

[Mine et al, 1997]

Mine, M.R., Brooks, F.P., Sequin, C.H., 1997, "Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction", ACM SIGGRAPH, pp. 19-26

[Mistry et al, 2009]

Mistry, P., Maes, P. and Chang, L., 2009, "WUW - wear Ur world: a wearable gestural interface", Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems, pp. 4111-4116

[Moore and Vos, 2004]

Moore, J.S. and Vos, G.A., 2004, The Strain Index, Handbook of Human Factors and Ergonomics Methods (edited by Stanton, N., Hedge, A., Brookhuis, K., Eduardo Salas, E. & Hendrick, H.) , CRC Press, 9-1 – 9-5

[Ng et al, 2012]

Ng, L.X., Wang, Z.B., Ong, S.K. and Nee, A.Y.C., 2012, “Integrated Product Design and Assembly Planning in an Augmented Reality Environment”, Assembly Automation, in press

[Nielsen et al, 2004]

Nielsen, M., Störring, M., Moeslund, T.B. and Granum, E., 2004, “A procedure for developing intuitive and ergonomic gesture interfaces for HCI”, Lecture Notes in Artificial Intelligence, Vol. 2915, pp. 409-420

[Nikolakis et al, 2004]

Nikolakis, G., Tzovaras, D., Moustakidis, S. and Strintzi, M.G., 2004, “CyberGrasp and PHANTOM Integration: Enhanced Haptic Access for Visually Impaired Users”, 9th Conference Speech and Computer, September

[Ohmori and Sakamoto, 2010]

Ohmori, K. and Sakamoto, K., 2010, “Automatic Mobile Robot Control and Indication Method Using Augmented Reality Technology”, Entertainment Computing - ICEC 2010, pp. 464-467

[Ong and Shen, 2009]

Ong, S.K. and Shen, Y., 2009, “A Mixed Reality Environment for Collaborative Product Design and Development”, CIRP Annals, Vol. 58, No. 1, pp. 139-142

[Ong et al, 2006]

Ong, S.K., Yuan, M.L. and Nee, A.Y.C., 2006, “Markerless augmented reality using a robust point transferring method”, Advances in Multimedia Modelling, pp. 258-268

[Ong et al, 2007]

Ong, S.K., Pang, Y. and Nee, A.Y.C., 2007, “Augmented reality aided assembly design and planning”, CIRP Annals, Vol. 56, No. 1, pp.49-52

[OpenCV, 2011]

OpenCV, Open Souce Computer Vision Library, http://sourceforge.net/projects/opencvlibrary/, last accessed on Jul, 2011

[OpenGL, 2011]

OpenGL, Open Source Graphic Library, http://www.opengl.org/, last accessed on Jul, 2011

[Pang, 2006]

Pang, Y., 2006, “Manual Assembly Design and Planning in an Augmented Reality Environment”, PhD Thesis, National University of Singapore

[Pang et al, 2006]

Pang, Y., Nee, A.Y.C., Ong, S.K., Yuan, M.L. and Youcef-Toumi, K., 2006, “Assembly feature design in an augmented reality environment”, Assembly Automation, Vol. 26, No. 1, pp.34-43

[Park et al, 2009]

Park, H.J., Moon, H.C. and Lee, J.Y., 2009, “Tangible augmented prototyping of digital handheld products”, Computers in Industry, Vol. 60, No. 2, pp. 114-125

[Pasman et al, 2004]

Pasman, W., Woodward, C., Hakkarainen, M., Honkamaa, P. and Hyväkkä, J., 2004, “Augmented Reality with Large 3D Models on a PDA-Implementation, Performance and Use Experience", In Proc. 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Application in Industry, pp. 344-351

[Pentenrieder et al, 2008]
Pentenrieder, K., Bade, C., Doil, F. and Meier, P., 2008, "Augmented reality-based factory planning – an application tailored to industrial needs", Proceedings of the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 76-84

[PHANToM, 2011]
http://www.sensable.com/, last accessed Jul, 2011

[Piekarski and Smith, 2006]
Piekarski, W. and Smith, R., 2006, "Robust gloves for 3D interaction in mobile outdoor AR environments", International Symposium on Mixed and Augmented Reality, pp. 251-252

[Raghavan et al, 1999]
Raghavan, V., Molineros, J. and Sharma, R., 1999, "Interactive Evaluation of Assembly Sequences Using Augmented Reality", IEEE Transactions on Robotics and Automation, Vol. 15, No. 3, pp. 435-449

[Rolland et al, 2001]
Rolland, J.P., Davis, L.D. and Baillot, Y., 2001, "A Survey of Tracking Technology for Virtual Environments", Fundamentals of Wearable Computers and Augmented Reality, Barfield, W. and Caudell, T. (Eds), pp. 67-112

[Salonen et al, 2007]

Salonen, T., S ääski, J., Hakkarainen, M., Kannetis, T., Perakakis, M., Siltanen, S., Potamianos, A., Korkalo, O. and Woodward, C., 2007, “Demonstration of Assembly Work using Augmented Reality”, International Conference on Image and Video Retrieval, pp. 120-123

[Seth et al, 2010]

Seth, A., Vance, J. M. and Oliver, J. H., 2010, “Virtual reality for assembly methods prototyping: a review”, Virtual Reality, Vol. 15, No. 1, pp.5-20

[Schall et al, 2009]

Schall, G., Mendez, E., Kruijff, E., Veas, E., Junghanns, S., Reitinger, B. and Schmalstieg, D., 2009, “Handheld augmented reality for underground infrastructure visualization”, Personal and Ubiquitous Computing, Vol. 13, No. 4, pp. 281-291

[Scheweighofer and Pinz, 2006]

Schweighofer, G. and Pinz, A., 2006, “Robust pose estimation from a planar target”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 12, pp. 2024-2030

[Segen and Kumar, 1998]

Segen, J. and Kumar, S., 1998, “Gesture VR: vision-based 3D hand interface for spatial interaction”, Proceeding ACM Multimedia 98, pp. 455-464

[Shen et al, 2011]

Shen, Y., Ong, S.K. and Nee, A.Y.C., 2011, "Vision-based hand interaction in augmented reality environment", International Journal of Human-Computer Interaction, in press

[State et al, 1996]

State, A., Livingston, M. A., Hirota, G., Garrett, W. F., Whitton, M. C., Fuchs, H. and Pisano, E. D., 1996, "Technology for Augmented Reality Systems: realizing ultrasound-guided needle biopsies", Proceedings of SIGGRAPH 96, New Orleans, LA, USA, August 4–9, pp. 439-446

[Schlattmann et al, 2009]

Schlattmann, M., Broekelschen, J. and Klein, R., 2009, "Real-time are-hands-tracking for 3D Games", Proceedings of IADIS International Conference Game and Entertainment Technologies, pp. 59-66

[Shah and Rogers, 1993]

Shah J.J. and Rogers M.T., 1993, Assembly modelling as an extension of feature-based design, Research in Engineering Design., Vol. 5, pp. 218-237

[Shen et al, 2011]

Shen, Y., Ong, S.K. and Nee, A.Y.C., 2011, "Vision-based hand interaction in augmented reality environment", International Journal of Human-Computer Interaction, pp. 523-544

[Song and Chung, 2009]

Song, I.H. and Chung, S.C., 2009, "Synthesis of the digital mock-up system for heterogeneous CAD assembly", Computers in Industry, Vol. 60, No. 5, pp. 285-295

[Sung et al, 2001]

Sung, R.C.W.，Corney, J.R. and Clark, D.E.R., 2001, Automatic assembly feature recognition and disassembly sequence generation, International Journal of Computing and Information Science in Engineering, Vol. 1, No. 4, pp. 291-299

[Swaminathan and Barber, 1996]

Swaminathan A. and Barber S., 1996, "An experience-based assembly sequence planner for mechanical assemblies", IEEE Trans Robot Automat, Vol. 12, No. 2, pp. 252–266

[Thomas et al, 1997]

Thomas, C.H., Lin, M.C., Cohen, J., Gottschalk, S. and Manocha, D., 1997, V-COLLIDE: Accelerated collision detection for VRML, Proceedings in 2nd Symposium on the Virtual Reality Modelling Language, pp. 117-125

[Thomas and Piekarski, 2002]

Thomas, B.H. and Piekarski, W., 2002, "Glove based user interaction techniques for augmented reality in an outdoor environment", Virtual Reality, Vol. 6, No. 3, pp. 167-180

[Valentini, 2009]

Valentini, P.P., 2009, "Interactive virtual assembling in augmented reality", Interactive Design and Manufacturing, Vol. 3, No. 2, pp.109-119

[Vallino, 1998]

Vallino, J.R., 1998, "Interactive Augmented Reality", PhD Thesis, University of Rochester

[van Holland and Bronsvoort, 2000]

van Holland, W. and Bronsvoort, W.F., 2000, Assembly features in modelling and planning, Robotics and Computer-Integrated Manufacturing, Vol. 16, No. 4, pp.277-294

[VUZIX, 2011]

VUZIX, 2011, Products Information Obtained through the Internet: http://www.vuzix.com/consumer/products_wrap920.html, last accessed on Sep, 2011

[Wagner et al, 2006]

Wagner, D., Schmalstieg, D. and Billinghurst, M., 2006, "Handheld AR for Collaborative Edutainment", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 4282, ISBN 978-3-540-49776-9, pp. 85-96

[Wang and Jen, 2006]

Wang, X. and Jen, H., 2006, "Designing Virtual Construction Worksite Layout in Real Environment via Augmented Reality", Proceedings of ISARC2006, pp. 757-761

[Wang et al, 2011]

Wang, R.Y., Paris, S. and Popovic, J., 2011, "6D hands: Markerless hand tracking for computer aided design", Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 549-557

[Wang et al, 2010]

Wang, Y.T., Shen, Y., Liu, D.C., Wei, S., Zhu, C.A., 2010, "Key technique of assembly system in an augmented reality environment", Proceedings 2010 Second

International Conference on Computer Modelling and Simulation (ICCMS), pp. 133-137

[Wilson, 1995]

Wilson R.H., 1995, "Minimizing user queries in interactive assembly planning", IEEE Trans Robot Automat, Vol.11, No. 2, pp. 308–312

[Wiedenmaier et al, 2003]

Wiedenmaier, S., Oehme, O., Schmidt, L. and Luczak, H., 2003, "Augmented reality (AR) for assembly processes design and experimental evaluation", International Journal of Human-Computer Interaction, Vol. 16, No. 3, pp. 497-514

[Xin et al, 2008]

Xin, M., Sharlin, E. and Sousa, M.C., 2008, "Napkin Sketch – Handheld Mixed Reality 3D sketching", In Proc. 15th ACM Symposium on Virtual Reality Software and Technology, pp. 223-226

[Yang et al, 2007]

Yang, R.D., Fan, X.M., Wu, D.L. and Yan, J.Q., 2007, "Virtual assembly technologies based on constraint and DOF analysis", Robotics and Computer-Integrated Manufacturing, Vol. 23, No. 4, pp.447-456

[Yang et al, 2007]

Yang, P., Wu, W.Y. and Moniri, M., 2007, "A hybrid marker-based camera tracking approach in augmented reality" IEEE/ACS International Conference on Computer Systems and Applications, pp. 662-667

[Ye et al, 1999]

Ye, N., Banerjee, P., Banerjee, A., Banerjee, A. and Dech, F., 1999, “A comparative study of assembly planning in traditional and virtual environments”, IEEE Trans Systems, Man and Cybernetics, Vol. 29, No. 4, pp.546-555

[Yin et al, 2001]

Yin, X., Guo, D. and Xie, M., 2001, “Hand image segmentation using colour and RCE neural network”, Robotics and Autonomous Systems, Vol. 34, No. 4, pp. 235-250

[Yin et al, 2001]

Yin, Z.P., Ding, H. and Xiong, Y.L., 2004, “A virtual prototyping approach to generation and evaluation of mechanical assembly sequences”, Proceedings of the Institution of Mechanical Engineers, Vol. 218, No. B1, pp. 87-102

[Yoon et al, 2006]

Yoon, J.H., Park, J.S. and Sung, M.Y., 2006, “Vision-based bare-hand gesture interface for interactive augmented reality applications”, Lecture Notes in Computer Science, Vol. 4161, pp. 386-389

[Yuan et al, 2004]

Yuan, M. L., Ong, S. K. and Nee, A. Y. C., 2004, “The Virtual Interaction Panel: An Easy Control Tool in Augmented Reality Systems”, Computer Animation and Virtual Worlds, Vol. 15, No. 3-4, pp. 425-432

[Yuan et al, 2008]

Yuan, M.L., Ong, S.K. and Nee, A.Y.C., (2008) “Augmented reality for assembly guidance using a virtual interactive tool”, International Journal of Production Research, Vol.46, No.7, pp.1745-1767

[Zaeh and Vogl, 2006]

Zaeh, M.F. and Vogl, W., 2006, “Interactive laser-projection for programming industrial robots”, IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 125-128

[Zhang et al, 2010]

Zhang, J., Ong, S.K. and Nee, A.Y.C., 2010, “RFID-Assisted Assembly Guidance System in an Augmented Reality Environment”, International Journal of Production Research, in press

[Zhong et al, 2005]

Zhong, Y.M., Shirinzadeh, B., Ma, W.Y. and Liaw, H.C., 2005, “Assembly Modelling Through Constraint-based Manipulations in A Virtual Reality Environment”, TENCON 2005 2005 IEEE Region 10, pp.1-6

[Zhou et al, 2007]

Zhou, X.H., Qiu, Y.J., Hua, G.G., Wang, H.F., and Ruan, X.Y., 2007, “A feasible approach to the integration of CAD and CAPP”, Computer Aided Design, Vol. 39, No. 4, pp. 324-338

[Zorriassatine et al, 2003]

Zorriassatine, F., Wykes, C., Parkin, R. and Gindy, N., 2003, “A survey of virtual prototyping techniques for mechanical product development”, Engineering Manufacture, Vol. 217, No. B4, pp. 513-53

Appendix A. Depth Calculation using 3D Vision Technologies

It is important to realize seamless integration and interaction of the virtual and real worlds in an AR system. Depth information recovered from a stereo image pair can be used to reconstruct the 3D model of the real object and realise the mutual occlusion and interaction problem between real and virtual objects in an AR system. The first step is to recover depth information from two images captured from two different views. The epipolar geometry [Hartley and Zisserman, 2003] is the intrinsic projective geometry between two views. It is independent of scene structure, and only depends on the cameras' internal parameters and relative pose (Figure A-1).

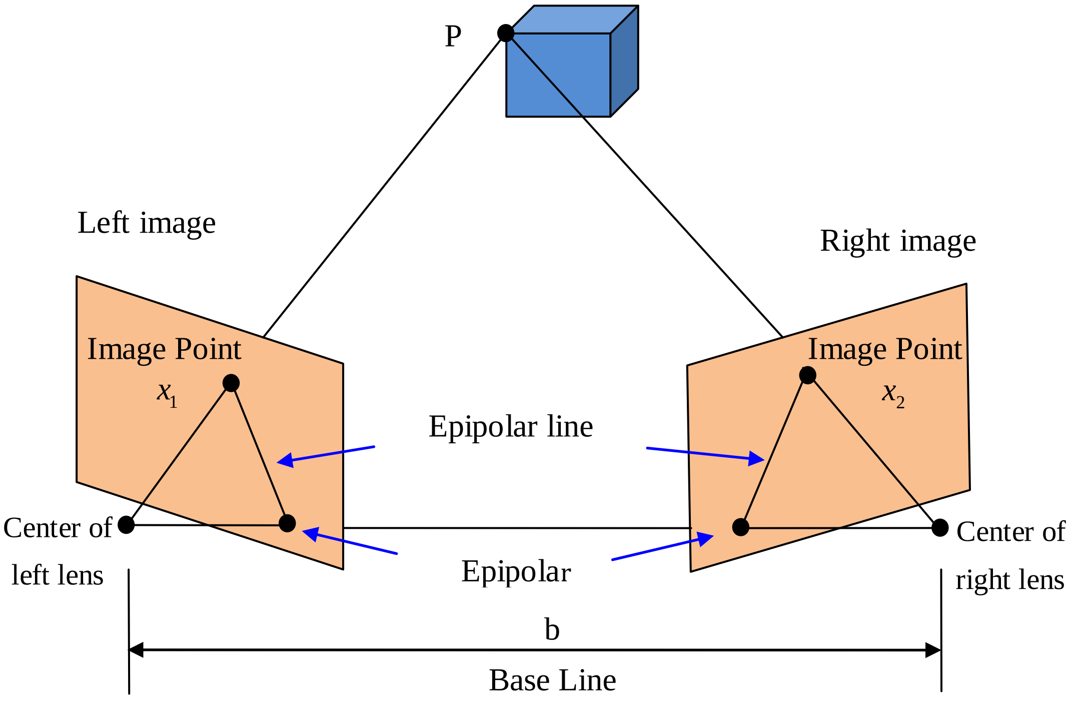


Figure A-1 Epipolar geometry

According to the pinhole perspective model, the depth of a point P in the world coordinate can be calculated using Equation (A.1).

$$Z = \frac{bf}{x_1 - x_2} \qquad \text{(A.1)}$$

where, *b* is the distance between the optical centres of these two cameras, *f* is intrinsic parameter of the camera, $x_1 - x_2$ is the disparity. To compute the disparity of two correspondence points in two images, the camera pose which is the rotation and translation between two cameras should be obtained first. In epipolar geometry, Equation (A.2) shows the relationship between the two image points as follow:

$$x_1 * F * x_2 = 0 \qquad \text{(A.2)}$$

where, *F* is the fundamental matrix. The fundamental matrix describes the map relationships from one point to a line (epipolar line) between two images. RANdom SAmle Consensus (RANSAC) method can be used to calculate the fundamental matrix from more than 7 pairs of correspondence points. The essential matrix *E* can be determined from the fundamental matrix *F* and the camera calibration matrices M_l, M_r using Equation (A.3)

$$E = M_r^T * F * M_l \qquad \text{(A.3)}$$

Finally, the camera pose, R and T, can be calculated using Equation (A.4):

$$\begin{cases} E = U * S * V^T \\ R = U * W * V^T \ or \ U * W^T * V^T \\ T = v_3 \ or \ -v_3 \end{cases} \tag{A.4}$$

where, *U*, *S* and *V* are the single-value decomposition (SVD) of *E*, v3 is the last column of *V*. *W* is a constant matrix (Equation (A.5)).

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

To calculate the disparity, the two images are usually rectified to make epipolar lines collinear and parallel to horizontal axis, as shown in Figure A-2. After the transformation, the disparity along the column is 0.

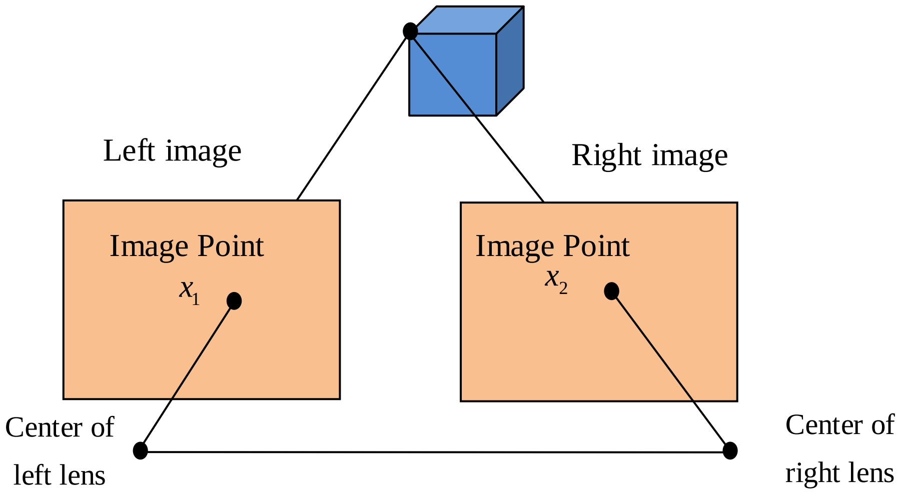


Figure A-2 Image plans after rectification

Appendix B. V-Collide Collision Detection Algorithm

The V-Collide algorithm [Thomas et al, 1997] is implemented in the AR-based assembly system to detect the collision between virtual models. V-Collide is a collision detection library for large virtual environments. It is designed to operate on large numbers of polygonal objects and it is suitable for assembly scenario. It makes no assumptions about input data structure and works on arbitrary models. It uses a three-stage collision detection architecture:

1) An *n*-body test finds possibly colliding pairs of objects;
2) A hierarchical oriented bounding box test finds possibly colliding pairs of triangles;
3) An exact test determines whether or not a pair of triangles actually overlaps.

The *n*-body routine uses coherence between successive time steps of a simulation to perform well in animations and moving simulations. The hierarchically oriented bounding boxes (OBBs) and exact collision routines are taken from RAPID which is a component of V-COLLIDE which is also available as a stand-alone package [Gottschalk et al, 1996]. The differences between V-Collide and Rapid are shown as follows:

1) V-COLLIDE keeps track of where objects are, so that if objects do not move between queries their locations need not be resupplied to the collision detection system.

2) V-COLLIDE supports many simultaneous objects, where RAPID only handles two.

3) RAPID can report exactly which pairs of triangles collide, but V-COLLIDE only reports to object precision.

Appendix C. Questionnaire of the User Study on 3D Bare-hand Interaction

Name: ____

Email Address: ____

Age: ____

Date: ____

1. **Is it easy to fully understand and learn the hand-based interaction for manipulating virtual objects in an AR environment? (1 is different, 5 is easy)**

   1 2 3 4 5

   □ □ □ □ □

2. **Is it easy to manipulate and rotate the virtual objects? (1 is different, 5 is easy)**

   1 2 3 4 5

   □ □ □ □ □

3. **Do you have the immersive feeling while using this interaction interface? If yes, how strong is it? (1 is weak, 5 is strong)?**

   1 2 3 4 5

   □ □ □ □ □

4. **Is this interaction interface useful of VR and AR applications? If yes, how useful is it? (1 is weak, 5 is strong)?**

1 2 3 4 5

□ □ □ □ □

5. **Other suggestions toward improving the current system, in terms of display equipment, interaction mechanism, accuracy and haptic feedback?**

Appendix D. Quaternions and Spatial Rotation

Quaternions provide a convenient mathematical notation for representing orientations and rotations of objects in a 3D space. Compared to Euler angles, they are simpler to compose and avoid the problem of gimbal lock which is the loss of one DOF in a 3D space that occurs when the axes of two of the three gimbals are driven into a parallel configuration. Compared to rotation matrices, they are more numerically stable and more efficient.

A quaternion is a 3D rotation about a unit vector by a specific angle, which is defined by four parameters. From example, a rotation of an angle θ around an axis directed by a normalized vector $\vec{N}(x_0, y_0, z_0)$ is represented by the quaternion using Equation (C.1).

$$Q = (\cos(\theta / 2),\ x_0 \square \sin(\theta / 2),\ y_0 \square \sin(\theta / 2),\ z_0 \square \sin(\theta / 2)) \qquad \text{(C.1)}$$

Using a single quaternion for rotation representation, the computation cost is reduced and the object can be rotated smoothly in a 3D space. Assuming a quaternion has been created in the form $Q = [w, (x, y, z)]$, the corresponding rotation matrix M_R can be calculated according to Equation (C.2).

$$M = \begin{bmatrix} 1-2*(y*y+z*z) & 2*(x*y+z*w) & 2*(x*z-y*w) & 0 \\ 2*(x*y-z*w) & 1-2*(x*x+z*z) & 2*(y*z+x*w) & 0 \\ 2*(x*z+y*w) & 2*(y*z-x*w) & 1-2*(x*x+y*y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{C.2}$$

Given two quaternions $Q_1 = [w_1, (x_1, y_1, z_1)]$ and $Q_2 = [w_2, (x_2, y_2, z_2)]$, the combined quaternion Q_r can be calculated using Equation (C.3).

$$\begin{cases} Q_r = Q_1 * Q_2 = [rw, (rx, ry, rz)] \\ rw = w_1 * w_2 - x_1 * x_2 - y_1 * y_2 - z_1 * z_2 \\ rx = w_1 * x_2 + x_1 * w_2 + y_1 * z_2 - z_1 * y_2 \\ ry = w_1 * y_2 + y_1 * w_2 + z_1 * x_2 - x_1 * z_2 \\ rz = w_1 * z_2 + z_1 * w_2 + x_1 * y_2 - y_1 * x_2 \end{cases} \tag{C.3}$$

Appendix E. Geometric Calculation Equations

1. *Point-point*

Problem Definition: Consider two points in 3D space P_1 and P_2, as shown in Figure D-1. The objective is to calculate the rotation matrix T_R and translation matrix T_T in order to make these two points coincident when P_2 is subjected to T_R and T_T.

$P_2(x_2,y_2,z_2,1)$
$P_1(x_1,y_1,z_1,1)$

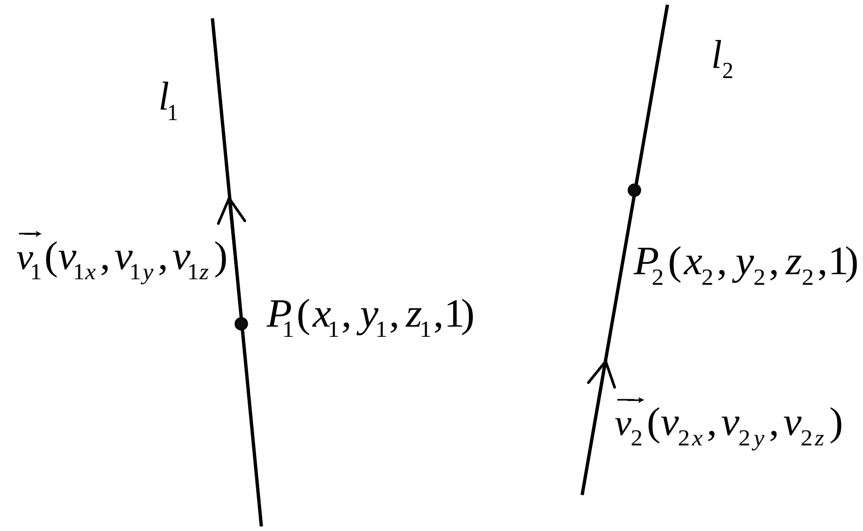
Figure D-1 Constraint recognition for *Point-point*

Thus,

$$T_R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } T_T = \begin{bmatrix} 1 & 0 & 0 & x_1 - x_2 \\ 0 & 1 & 0 & y_1 - y_2 \\ 0 & 0 & 1 & z_1 - z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(D.1)}$$

2. *Line-line*

Problem Definition: Consider two lines in 3D space l_1 and l_2 such that l_1 passes through point P_1 and is parallel to normalized vector $\overrightarrow{v_1}$ and l_2 passes through point P_2 and is parallel to vector normalized $\overrightarrow{v_2}$, as shown in Figure D-2. The objective is to calculate the rotation matrix T_R and translation matrix T_T in order to make these two lines aligned when l_2 is subjected to T_R and T_T.



l_1
l_2
$\overrightarrow{v_1}(v_{1x}, v_{1y}, v_{1z})$
$P_1(x_1, y_1, z_1, 1)$
$P_2(x_2, y_2, z_2, 1)$
$\overrightarrow{v_2}(v_{2x}, v_{2y}, v_{2z})$

Figure D-2 Constraint recognition for *Line-line*

Let $\vec{n}(n_x, n_y, n_z)$ be a vector perpendicular to both lines. Then

$$\vec{n} = \begin{bmatrix} \vec{i} & \vec{j} & \vec{k} \\ v_{2x} & v_{2y} & v_{2z} \\ v_{1x} & v_{1y} & v_{1z} \end{bmatrix} \tag{D.2}$$

Let θ be the angle between these two lines. Then

$$\theta = a\cos(\overrightarrow{v_1} \cdot \overrightarrow{v_2}) = a\cos(v_{1x}v_{2x} + v_{1y}v_{2y} + v_{1z}v_{2z}) \tag{D.3}$$

The rotation that can make these two lines parallel can be represented by a quaternion Q'

$$Q' = (\cos(\theta/2), n_x \cdot \sin(\theta/2), n_y \cdot \sin(\theta/2), n_z \cdot \sin(\theta/2)) \qquad \text{(D.4)}$$

Assume l_2 has a original quaternion Q_1. Then, the current quaternion Q_2 of l_2 is

$$Q_2 = Q_1 * Q' \qquad \text{(D.5)}$$

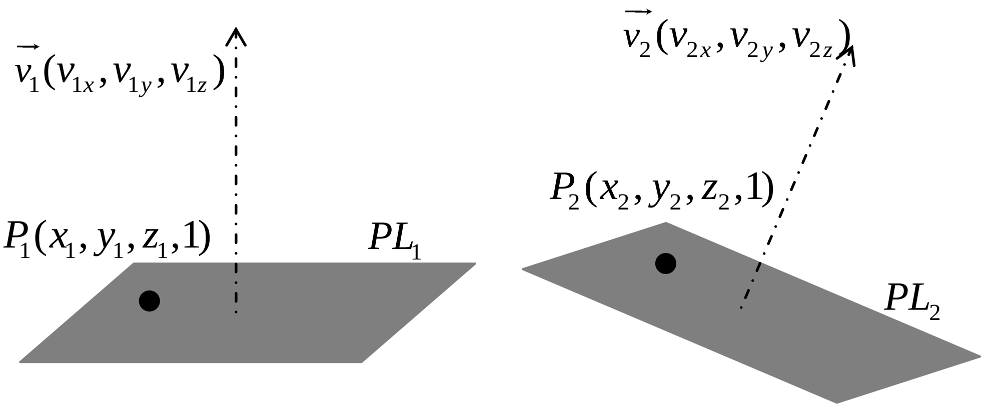
The rotation matrix T_R can be calculated using the Equation (C.2) introduced in Appendix C. l_2 is parallel to l_1 and P_2 is updated to P_2' when l_2 is subjected to T_R.

$$P_2' = T_R \times P_2 \qquad \text{(D.6)}$$

The translation matrix T_T can be calculated using Equation (D.1) based on P_1 and P_2'.

3. *Plane-plane*

Problem Definition: Consider two planes in 3D space PL_1 and PL_2 such that PL_1 passes through point P_1 and has a normalized normal $\vec{v_1}$ and PL_2 passes through point P_2 and has a normalized normal $\vec{v_2}$, as shown in Figure D-3. The objective is to calculate the rotation matrix T_R and translation matrix T_T in order to make these two planes coincident when P_2 is subjected to T_R and T_T.



$\vec{v_1}(v_{1x}, v_{1y}, v_{1z})$
$\vec{v_2}(v_{2x}, v_{2y}, v_{2z})$
$P_2(x_2, y_2, z_2, 1)$
$P_1(x_1, y_1, z_1, 1)$
PL_1
PL_2

Figure D-3 Constraint recognition for *Plane-plane*