Distributed Augmented Reality for Visualizing Collaborative Construction Tasks

Hui Wang

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

January 2008

Your file Votre référence
ISBN: 978-0-494-40956-5
Our file Notre référence
ISBN: 978-0-494-40956-5

ABSTRACT

Distributed Augmented Reality for Visualizing Collaborative Construction Tasks

Hui Wang

Augmented Reality (AR) can provide better user experience of real time interaction. This technology can be used to support different engineering applications, such as construction, manufacturing and inspection. However, AR engineering applications have to overcome many challenges such as tracking and visulazation. In addition, outdoor mobile AR engineering applications bring in more issues, such as scalable tracking, mobility, etc. Furthermore, engineering constraints and reasonable user interaction in practice need to be considered and integrated.

In this research, we describe a new approach to simulate construction activities in outdoor environment using AR for multi-user collaboration. This approach intergrates several tracking techniques to suit the requirements of mobile outdoor AR systems. We also describe and evaluate system design challenges and possible solutions for simulating multi-user collaborative construction activities within a distributed AR environment. These challenges include tracking and registration, dynamic object modeling and constraints and user interaction.

The proposed approach is demonstrated by developing a prototype system which integrates a head-mounted display, a GPS receiver, a 3D motion sensor, and a portable computer to carry out a construction simulation which can be run in the outdoors. The prototype system is implemented in Java language. The results of the case study are discussed to evaluate the usefulness of the proposed approach.

ACKOWNLEGEMENTS

First, it is a great pleasure for me to express my sincere appreciation and thanks to my supervisors, Dr. Amin Hammad and Dr. Sudhir P. Mudur who introduced me to an exciting topic, and always gave me extremely useful advices and directions letting me achieve great results.

During these demanding years of study, the support of our group was of extreme importance. I would like to thank Qing Zhe Huang, and Ran Wang for their cooperation and contribution in the discussion of mathematics and helps of 3D scanning. In addition, the help of all other friends in this project is greatly appreciated.

Finally, I would like to thank my wife Qian Zhou, my parents, and my family for their continuous encouragement and love. They are my most important motivation to pursue my graduate study. Without their help and contributions, it was impossible for me to succeed in this study. I deeply appreciate them for inspiring me to accomplish this work. To them, I dedicate this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

Abbreviation	Description
2D	Two-dimensional
3D	Three-dimensional
AR	Augmented Reality
AV	Augmented Virtuality
CCD	Cyclic Coordinate Descent
CDED	Canadian Digital Elevation Data
CT	Computed Tomography
DEM	Digital Elevation Model
DGPS	Differential GPS
DOF	Degrees of freedom
FK	Forward Kinematics
GIS	Geographic Information System
GPS	Global Positioning System
GUI	Graphical User Interface
HMD	Head-Mounted Display
IC3	InertiaCube3
IK	Inverse Kinematics
Java 3D API	Java 3D Application Programming Interface
JDBC	Java Database Connectivity
JNI	Java Native Interface
JVM	Java Virtual Machine
MEMS	Micro-Electro-Mechanical systems
MR	Mixed Reality
MRI	Magnetic Resonance Imaging
NMEA	National Marine Electronics Association
RTK-GPS	Real-time kinematics GPS
SQL	Structured Query Language
SXGA	Super eXtended Graphics Array
UWB	Ultra-wideband
VE	Virtual Environment
VESA	Video Electronics Standards Association
VR	Virtual Reality
VRML	Virtual Reality Modeling Language
WIMP	windows, icons, menus, and pointing
WLAN	Wireless local area network

CHAPTER 1
INTRODUCTION

1.1 GENERAL BACKGROUND

Augmented Reality (AR) is a growing area in computer graphics research. With recent advances in computer graphics performance and the evolution of new software technologies, we experience a switch in architectural visualizations from still images and pre-rendered animations to interactive three-dimensional (3D) models. AR is one of the techniques for presenting 3D digitally represented objects combined with the physical environment. It enables complex details of the real world to be augmented by virtual objects and information and can support full interaction with various types of human actions.

AR has been found useful for many engineering applications such as urban planning and construction. In order to realize an AR system, enormous amount of time and money has to be invested to design the models and to acquire the necessary tracking. Most of these models cannot fulfill the requirement of outdoor environment and specific engineering tasks. The recent improvement of high-end hardware, enabling mobile computing, different ways of real world sensing, wireless access to information, GPS receivers, etc. makes it possible to access information in ways which was not possible earlier. This creates a great opportunity to enhance most AR systems in terms of their technical capability, in interaction and usability, and finally, in applications. The complexity of integrating multi high-end equipment becomes an important issue of AR. Therefore, a practical and economic hardware and software integration method for creating AR systems for engineering applications in outdoor environment needs to be developed.

Another issue related to outdoor AR is that the users of these applications may have severe problems in exploring and interacting with the real environments to accomplish specific tasks. For accomplishing their tasks in these environments, they need to be able to interact either to accomplish specific tasks or to become more familiar with the environments. Properly designed user interfaces for operation can make that experience successful and enjoyable. There are many research works on the general principles of 3D interaction in AR. However, little work has been done about providing operation support in engineering applications in outdoor environments.

1.2 RESEARCH OBJECTIVES

In order to overcome the above mentioned limitations, this research focuses on investigating a framework for AR applications in outdoor construction environments. The framework includes a practical method for developing an AR, several tracking methods and user interactions which are suitable for outdoor AR engineering applications. Hence, the research objectives are:

(1) To design a mobile AR system that integrates various high-end equipment and interactive rendering methods to facilitate on site construction simulation.

(2) To investigate a practical method for creating AR in outdoor environment by integrating multiple tracking methods.

(3) To develop a framework which considers user collaboration and engineering constraints for advanced real time visual simulation.

(4) To investigate the user interaction suitable for the above AR applications.

1.3 THESIS ORGANIZATION

This study will be presented as follows:

Chapter 2 A Review of Augmented Reality Technology and Applications: This chapter presents the major Mixed Reality applications developed using different approaches, current technology challenges and important issues. Then, mobile augmented reality technologies used in civil engineering area are introduced. In addition, several tracking and display technologies are compared and discussed.

Chapter 3 Augmented Reality for Visualizing Collaborative Construction Tasks: This chapter introduces the detailed design of mobile augmented reality for construction simulation. AR is discussed within a generic framework. The framework embodies the general functionalities of AR applications in civil engineering and can be used to suit the requirement of mobile infrastructure management systems. Mobile computing is used to construct the computing platform in outdoor environment. The usage of multi-tracking methods for finding the user's position and orientation, and a 3D crane model with interaction functions are discussed. An integration method is proposed for creating AR by synthesizing information from different sources. In addition, engineering constraints are considered. Furthermore, two display methods used in this framework are introduced.

Chapter 4 Implementation and Case Studies: In this chapter, the implementation is demonstrated and several case studies are considered to validate the features of this prototype system developed using the proposed mobile AR approaches.

Chapter 5 Summary, Conclusions, Contributions, and Future work: This chapter summarizes and concludes the present research work, highlights its contributions, and suggests recommendations for future research.

CHAPTER 2

A REVIEW OF AUGMENTED REALITY TECHNOLOGY AND APPLICATIONS

2.1 INTRODUCTION

Mixed Reality (MR) is a virtuality continuum integrating the virtual and real worlds to generate new environments where physical and virtual objects can co-exist and interact in real-time (Milgram et al., 1994). It represents a continuous spectrum between the completely virtual reality (VR) environment, and the completely real environment. It includes augmented reality (AR), wherein the virtual augments the real, and augmented virtuality (AV), where the real augments the virtual. Figure 2.1 illustrates these different levels of MR.

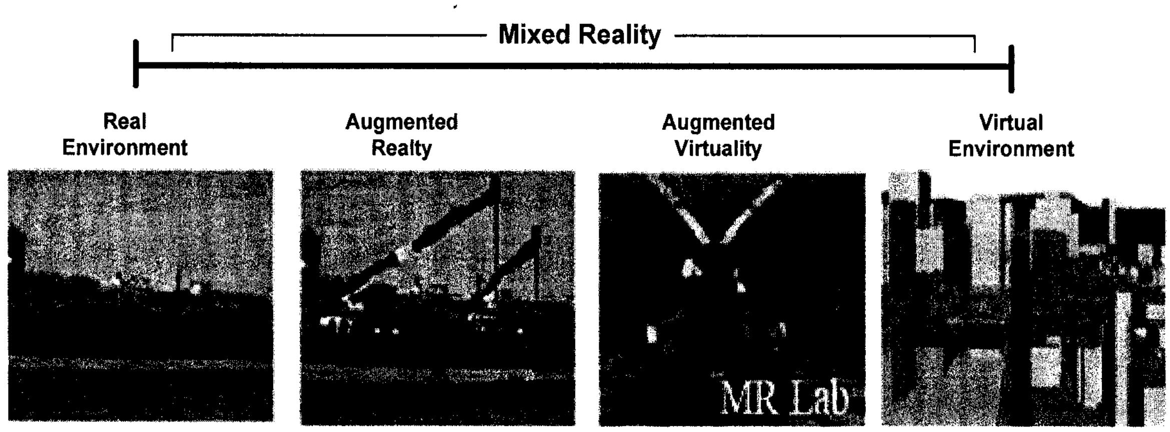


Figure 2.1: Mixed Reality Continuum (adapted from Isdale, 2003)

A virtual environment (VE) is a computer-generated, 3D environment in which users can interact in real-time (Capps, 2002). It builds a virtual 3D model in a computer to visually reproduce the shape, texture and movement of objects (Burdea and Coiffet, 2003). The VE could represent either reality or fiction. For a real world, it might be a big 3D model of real objects, such as a group of buildings in the downtown of a real city; it also could be a totally imaginary world or an abstract model that never exists in the real world but can be understood by people, such as a 3D representation of a set of data. Under all these circumstances, the VE serves a common and crucial role that could make the user believe that he/she is actually experiencing a different and compelling world. The VE also has another important feature, i.e., the visual cues must be what the user expects in a real world when doing some actions such as moving his/her hands or head. In other words, the VE should support navigation and interaction in an immersive environment (Bowman and Billinghurst, 2002).

AR stays near the real environment end of the MR continuum with the view of the real world mixed with computer generated graphics. The technology supplements the real world with composite 3D virtual objects that are integrated into the real world. AR is widely researched and rapidly evolving, as can be found in a survey paper by Azuma (1997). Azuma et al. (2001) defined three main properties of AR: (1) It combines real and virtual objects in a real environment; (2) It allows interaction with users in real time; and (3) It registers (aligns) real and virtual objects with each other. It enhances user immersion and experience of human-computer interaction and provides an innovated avenue of user's perception of the real world. Because of many technical challenges and emerging methods in this area, AR remains an active and interesting research area of computer science and information technology.

Azuma (1999) classified AR systems as indoor or outdoor systems. Indoor AR provides a prepared environment, predictable or controllable user movement, and stable lighting, but it limits user's actions, and therefore it limits the possibilities of the usage and the types of AR applications. For example, construction simulation in an actual outdoor construction site cannot be realized using indoor AR.

AV is a term created by Milgram and Kishino (1994) to identify systems which are captured in real-time and texture mapped to corresponding synthetic surfaces of virtual objects. Chen (1995) described the approach of VE navigation with real images. The advantage of AV is that it creates a virtual world that has the relevant data of the real world. Thus the virtual world is not affected by the physical location and can keep the flexibility of virtual world without temporal, spatial, and physical constraints from the physical world.

2.2 APPLICATIONS OF AUGMENTED REALITY

The advantages of AR have been discussed in many engineering applications such as design perception (Dunston et al., 2002). AR arguably provides the most natural means to communicate with a computer by allowing users to use their inherent 3D spatial skills, such as walking, gesturing, looking, grabbing, etc. There is a vast range of potential applications of AR, such as medicine, manufacturing and repair, urban planning, architecture and archaeology, journalism, navigation and path finding, and civil engineering. The next paragraphs describe a few representative applications which have been explored in different areas.

(1) Medicine

AR shows many advantages for medicine because medicine has high visualization requirements for surgeons. For example, doctors and nurses can get important information about patient's status from their AR glasses when they interview and check their patients (Hasvold, 2002). Meanwhile, many research groups (Lorensen et al., 1993; Grimson et al., 1995; Mellor 1995) focused on registering magnetic resonance imaging (MRI) or computed tomography (CT) data onto the patient body with live overlays which need accurate registration. Figure 2.2 shows a medical AR example developed at the University of North Carolina at Chapel Hill (State et al., 1996). The figure shows that a needle has been inserted into the breast phantom.

Figure 2.2: Real-time stereo HMD view with ultrasound volume display (State et al., 1996)

(2) Manufacturing, maintenance and repair

AR applications can provide support in the manufacturing, maintenance and repair of complex machinery. In the real work space, AR could simply display 3D virtual instructions to train and help assembly staff. The 3D virtual instruction can provide a step by step tutoring with animation designed for a better user experience. It would greatly improve efficiency of the training process for manufacturing because the instruction using AR can clearly provide the user with working contents, processes and routines. Feiner et al. (1993) developed a laser printer maintenance application (cf. Figure 2.3).

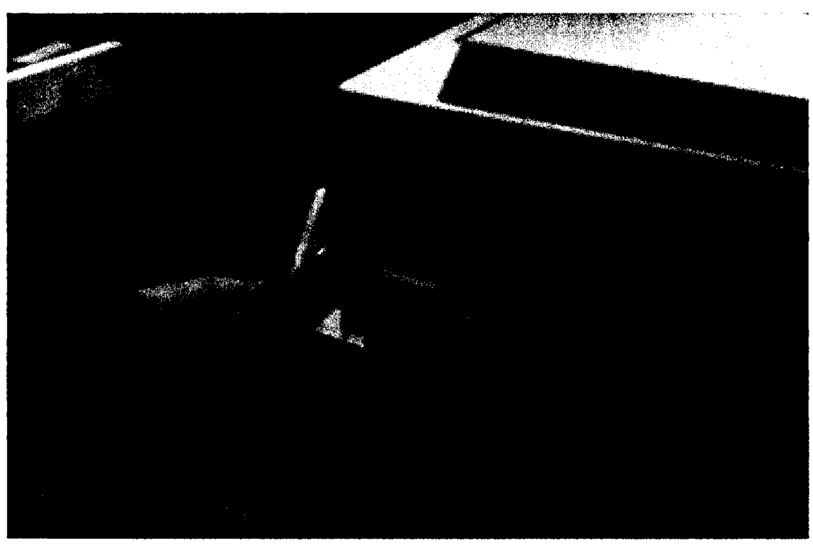


Figure 2.3: Prototype laser printer maintenance application of AR, displaying how to remove the paper tray (Feiner et al., 1993)

Meanwhile, the industry has undoubtedly presented its interests for this kind of applications. For example, a research group in Boeing built an AR-supported assembly of electrical wire bundles of an airplane (Mizell, 2001).

(3) Urban planning

AR systems promise a possible replacement for the media and systems that designers traditionally use for supporting collaborative work. They also provide a good environment for 3D rendering of buildings and other facilities. Penn et al. (2004) represented a research project in which optical see-through AR displays have been developed together with a prototype decision-support software for architectural and urban design (Figure 2.4).

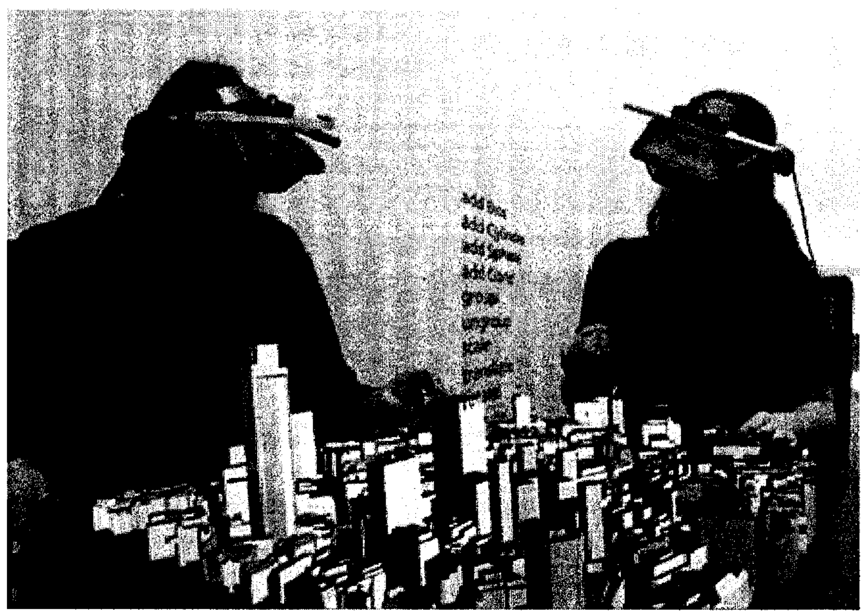


Figure 2.4: Prototype urban planning system on a table (Penn et al., 2004)

(4) Architecture and archaeology

Architecture gets impact from AR technology because AR can easily visualize the architect's design on a real site or building. Webster et al. (1996) describe an AR application to improve methods for the construction, inspection, and renovation of architectural structures.

In addition, AR also promises benefits for the reconstruction of archaeology. Vlahakis et al. (2002) started a European sponsored project ARCHEOGUIDE that focuses on reconstructing a cultural site with AR and helps visitors to view and understand the ancient architecture knowledge. They selected the ancient town of Olympia in Greece for the first test with the AR application (Figure 2.5).

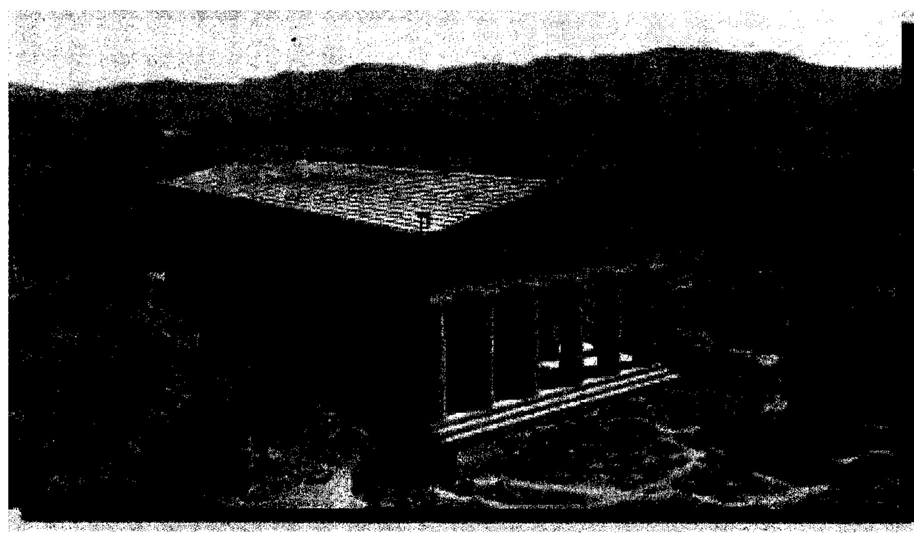


Figure 2.5: ARCHEOGUIDE project reconstructed a virtual hall on a real location (Vlahakis et al., 2002)

(5) Journalism

Journalism is another approach in which mobile AR has an obvious role. The advantage of wireless technology for the mobile journalism is described by Pavlik (2001). Mobile AR systems can record and document news as it develops in real time. For example, a mobile AR system can add messages to the news scene, and other collaborating journalists can view and follow these messages while following the news. Hollerer et al. (1999) presented the Situated Documentaries project combining journalism and AR technology. Figure 2.6 shows the system for presentation of historical information and storytelling. Figure 2.6 (a) shows a user with the backpack AR system. Figure 2.6 (b) represents the view through head-wear display. The virtual flags in the view represent points of interest in the scene. Figure 2.6 (c) shows the related information displayed on the hand held computer.

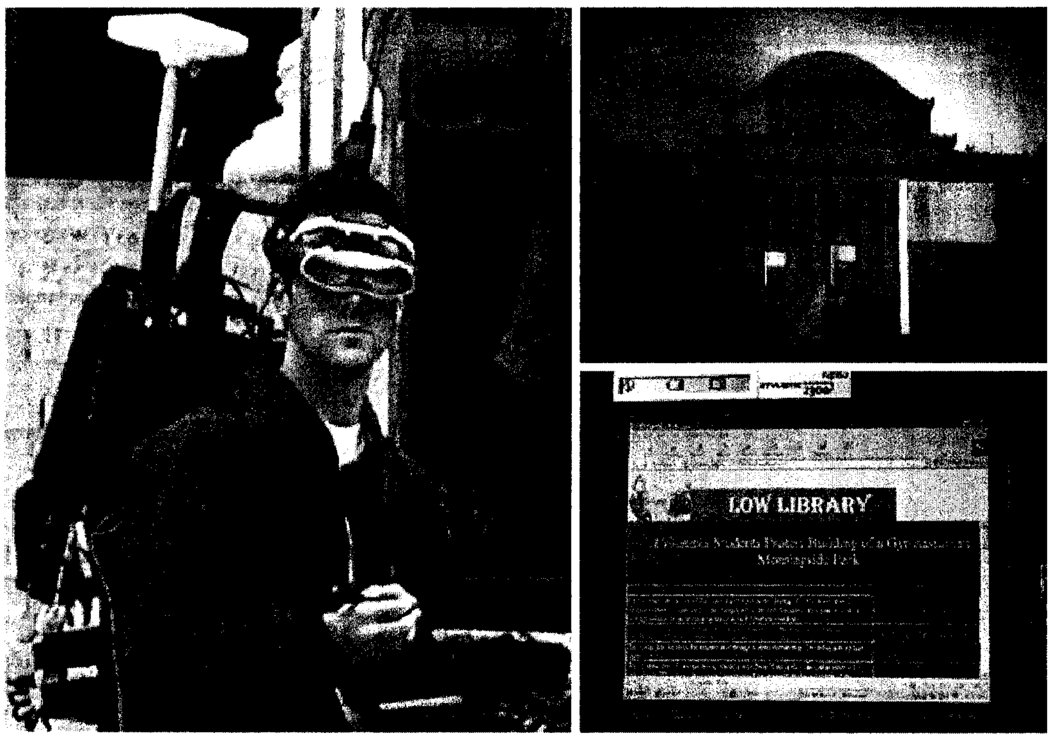


Figure 2.6: Situated documentaries project in Columbia University (Hollerer et al., 1999)

(6) Navigation and Tourism

Because of the advantages of mobile AR in outdoor environment, navigation and tourism aids become an important area for mobile AR application. In outdoor environment, mobile AR systems can directly describe locations or landmarks in the user's field of view for aiding navigation. For example, Furmanski et al. (2002) used directional annotations such as arrows and trails to show the current path. In addition, a research group at Microsoft overlays a two dimensional (2D) map on the display of current environment for providing assistance in navigation. The AR system of Microsoft can also display related history information of an object, which is currently viewed by a user for quick tourism. The tourism function can be naturally and easily developed based on navigation functions supported by AR technology. On the other hand, AR application is not only used to find a path, but also to display related information, such as the history of the place or comments. Figure 2.7 shows an example of adding a navigation map to the video scene (Zitnick et al., 2004).

Figure 2.7: View of a garden with a superimposed map (Zitnick et al., 2004)

(7) Civil Engineering

Wang and Dunston (2006) investigated the usability of an AR-based pipe design system. The AR-based collaborative system helps designers to arrange pipes on the desktop. The prototype is only used in the indoor environment. Behazdan and Kamat (2006, 2007) demonstrated the concept of scheduled simulation of construction activities with AR.

They applied 3D graphical simulation of construction activities in the outdoor environment. However, the visualized simulation is pre-scheduled and it does not have real-time user interaction as required by Azuma's definition of AR (Azuma, 1997). In addition, their work does not take into consideration the engineering constraints and multi-user collaboration. Thus the current AR applications in civil engineering still need further improvement.

2.3 CHALLENGES OF MOBILE AR

Using a variety of 3D modeling, tracking, user interaction, rendering and display techniques, AR enhances user's immersion by allowing them to view the AR environment while moving in the real world. In addition, mobile AR provides a new approach for simulating multi-person collaborative environment. It fulfills the requirements of real-time visual simulation and the ability to move freely on site. However, mobility in the outdoor environment brings several challenges for developing outdoor AR applications, such as tracking, displays and user interaction methods. We will look at current technologies addressing these challenge areas in the following section.

2.3.1 Tracking and Registration

Tracking and registration are among the most important technological challenges of AR. Accurately tracking the user's position and viewing orientation is crucial for AR applications (Azuma et al., 2001). We cannot convince people that computer generated virtual objects actually exist in the current real environment if we cannot correctly and accurately align or register these virtual objects. Azuma (1997) described three crucial demands of tracking: (1) input variety and bandwidth, (2) high accuracy, and (3) long range. Many useful applications of AR must be applied in dynamic environment. With the options of dynamic environment, we have to apply a variety of tracking sensors at the same time. Meanwhile, the requirement of performance for real-time interaction presents the demands of high tracking frequency and update rate. In addition, for AR environment, registration is mainly relying on tracking. In many situations, the accuracy of registration is same as the accuracy of tracking. Therefore, the accuracy requirement of tracking should be seriously considered. Furthermore, scalable tracking methods are required for most outdoor and some indoor environment. For example, if a user may freely move in a large scale space such as a city, the tracking method should be available in any required range. Moreover, portability is also necessary for mobile AR; i.e., the equipment should be easy to wear and ready for unprepared environment.

To fulfill the above requirements, we have to consider the following important question: How much tracking do we need for AR registration? To represent an object in 3D space, six degrees of freedom are required. Three degrees of freedom are used for positioning; and the other three degrees of freedom are used for the orientation. Figure 2.8 shows the 6 degrees for freedom assuming that the user is looking along the X axis.

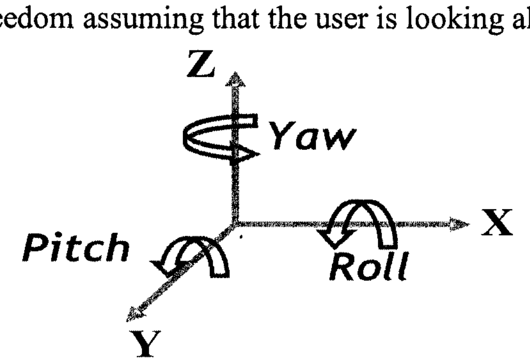
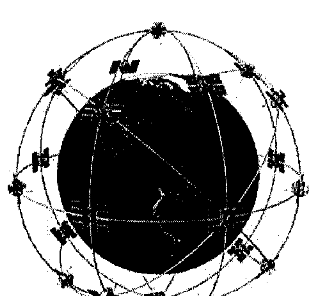


Figure 2.8: Six degrees of freedom for representing in 3D space

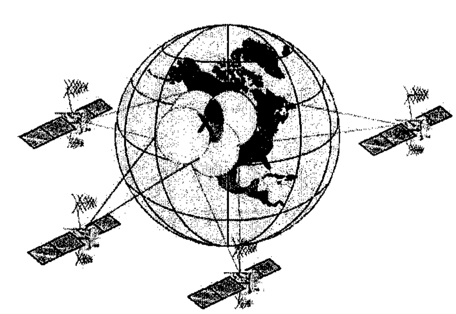
In the last thirty years, tracking technology has been improving steadily (Azuma et al., 2006). Raab et al. (1979) introduced the Polhemus magnetic tracker. It obviously improved the tracking of VR and AR researches. Many researchers still use the same technology today. After that, many hybrid tracking systems have been investigated based on different technologies, such as ultrasonic, magnetic, and optical position tracking. In addition, the accuracy of position tracking is increased by advanced techniques. Some of the current position tracking methods are discussed below:

(1) Global Position System (GPS)

GPS has been completed in 1993. GPS is one positioning technology which is available anywhere within certain conditions and it measures the horizontal and vertical positions of the receiver from the GPS satellites (Hofmann-Wellenhof et al., 2004). The GPS consists of 24 earth-orbiting satellites so that it can guarantee that there are at least 4 of them above the horizon for any point on global space at any time (Figure 2.9). The factors that affect GPS accuracy include ionospheric and tropospheric distortion of the radio signals from the satellites, orbital alignment and clock errors of the satellites, and signal multi-path errors (reflections and bouncing of the signal near buildings). In addition, GPS is easily blocked in urban areas, near hills, or under highway bridges. The accuracy of a position is also a function of the geometry of the GPS constellation visible at that moment in time, i.e., when visible satellites are well separated in the sky, GPS receivers compute positions more accurately. One method to increase the accuracy of GPS is by using Differential GPS (DGPS). DGPS is based on correcting the effects of the pseudo-range errors caused by the ionosphere, troposphere, and satellite orbital and clock errors by placing a GPS receiver at a precisely known location (Bossler et al., 2001). The pseudo-range errors are considered common to all GPS receivers within some range. Multi-path errors and receiver noise differ from one GPS receiver to another and cannot be removed using differential corrections. DGPS has a typical 3D accuracy of better than three meters and an update rate of 0.1-1 Hz. The DGPS corrections can be sent to the mobile GPS receivers in real time, or added later by post-processing of the collected data. Real-time kinematics GPS (RTK-GPS) receivers with carrier-phase ambiguity resolution can provide accuracy of about 1 cm, but are computationally intensive to run in real time (Kaplan, 1996). The DGPS systems of North America usually use the long wave radio frequencies for broadcasting. In some situations, the streaming of DGPS or RTK correction data can be sent by Internet when the radio infrastructure is not available in the local area (Lenz, 2004). The European Union is planning to launch another separate global tracking system with 30 satellites, called Galileo. Therefore, using GPS for location tracking is a reasonable choice for mobile AR in the outdoor.



(a) 24 satellites



(b) 4 satellites used in calculating the location of the receiver

Figure 2.9: Satellites used for GPS

(2) Ultra wideband (UWB)

UWB provides another alternative solution for indoor position tracking. It is a short-range radio technology that can provide advantages for precision localization service. Meanwhile it uniquely employs a recently legalized frequency spectrum with low-power (Intel, 2007). Depending on different positioning methods such as the angle of arrival, the signal strength, or time delay information, the accuracy of UWB tracking can be up to 1 cm (Gezici et al., 2005; Pahlavan et al., 2002).

(3) Wireless local area network (WLAN)

WLAN also has been explored for tracking. Due to the demand of wireless access point, it only can work in an environment which has WLAN infrastructure. However, it has strong robustness feature because of its robust radio frequency signal propagation (Xiang et al., 2004). Several research and some commercial products investigated the approach (Wang et al., 2003; Youssef et al., 2003; Ganu et al., 2004; Ekahau, 2007). The Ekahau position engine, a commercial WLAN positioning API, can provide 1 meter accuracy in indoor environment with three or more access points (Ekahau, 2007). The signal strength and the setting of the access points are two main constraints for accuracy. Thus WLAN is mainly used for indoor applications now.

Orientation tracking has been also greatly improved by new techniques. Gyroscopes, accelerometers and magnetometers are widely applied for orientation tracking. These methods are discussed below:

(1) Gyroscopes

A gyroscope is a device to measure orientation using the method of conservation of angular momentum. It determines the rotational forces applied to the device and calculates relative orientation values. In its early stages, a gyroscope was a mechanical device which was made by suspending a relatively massive rotor inside three rings called gimbals (Welch and Foxlin, 2002). The motion perpendicular to the axis of rotation was measured when it obtained an external force. Gyroscopes are widely used for direction measurements in airplanes and ships. However, they are typically very large and not portable. Today, with the development of micro-electro-mechanical systems (MEMS) technology, they can be constructed without moving parts. Digital gyroscopes include an internal vibrating resonator. If the vibrating resonator measures rotational forces along the appropriate axis, the tines of the fork will vibrate in a perpendicular direction by Coriolis forces. These perpendicular forces are proportional to the angular velocity. Thus they can be evaluated to produce the output results. One gyroscope can measure one axis of rotation. Thus, we can measure all degrees of freedom of rotation by orthogonally mounting three gyroscope sensors (Foxlin and Durlach, 1994).

(2) Accelerometers

An accelerometer is a device to measure acceleration, and to detect and measure vibrations. It measures the linear forces and then produces relative-position values. It is often used to measure absolute pitch and roll by measuring the acceleration caused by gravity.

Foxlin et al. (1998) introduced the details of how the accelerometers measure acceleration forces applied to an object along a single axis. Currently accelerometers are often used along with gyroscopes in inertial measurement systems. They are implemented by MEMS technology that integrates all parts which are not moveable in a small IC sized component. Many commercial companies produce a variety of accelerometers for all kinds of applications, such as airbag deployment for automobiles, video game console (e.g. Nintendo Wii), etc.

(3) Magnetometers

A magnetometer is a device to measure the strength and direction of the magnetic field around the device. If a magnetic field passes through a coil of wire, it produces an induced current to calculate the strength of the field and the incident angle to the coil. We also can calculate the direction to magnetic north by aligning three magnetometers orthogonally. Current magnetometers are solid state electronic devices, so that they do not have inertia, and then they can produce fast and accurate outputs.

Advanced calibration techniques are applied in real-time tracking and registration of AR. Each tracking device will bring some errors and reduce the total accuracy of orientation tracking. Kalman (1960) presented an efficient recursive algorithm that estimates the state of a dynamic process from a series of discrete, incomplete and noisy measurements. Kalman filters are widely applied for digital computing for reduce error and improve the accuracy of measurements.

InterSense (2002) released the InertiaCube2 and IS300 orientation sensor that combine three micro-electromechanical gyroscopes, three accelerometers, and an electromagnetic compass using Kalman filter. Azuma et al. (1999) introduced a hybrid tracker that combines a calibrated compass and tilt sensor with three gyroscopes for orientation tracking. The system is extended with a move-matching vision algorithm by You et al. (1999). In addition, Hoff and Azuma (2000) applied a Kalman-like estimator for auto-calibration of an electronic compass in an outdoor AR system.

Computer vision approach can also be used for tracking of 6 DOF. The move-matching algorithm can dynamically find key points in images and track relative orientation and position. For example, ARToolKit is a popular and widely applied API which provides vision-based tracking solution for indoor AR applications (Kato and Billinghurst, 1999). Meanwhile, the combination of vision-based correction for tracking is also investigated. Behringer (1999) presented a vision-based correction method based on comparing the silhouette of the horizon line with a model of local geography. Satoh et al. (2001) used a template-matching technique on manually selected landmarks in a real-time algorithm that corrects for the orientation drift of an accurate fiber optic gyroscope.

However, current computer vision algorithms have strict requirements for the application environment. For example, the dynamic lighting condition and view distance of outdoor environment obviously affect the precision of vision-based solutions. Malbezin et al. (2002) found that the error of ARTooKit in position increases with the distance from the target. The robustness and applied range of computer vision based tracking techniques still need improvement. Table 2.1 shows the comparison of above tracking techniques.

In summary, in indoor environments, research shows that one can easily track a person's head and body with sufficiently high accuracy and resolution, low latency and high update rates for accurately registering realistic computer graphics. On the other hand, mobile AR in outdoor environments has much more challenges for accurate tracking because the environment is usually unprepared with the sensing infrastructure. In addition, tracking equipment has to be wearable, resistant to shock and qualified for wide spectrum of conditions such as lighting, weather, and temperate. Currently it is difficult to get a perfect solution for tracking in outdoor environment.

Table 2.1 Comparison of tracking techniques

Technology	Applied Environment	Range	Accuracy	Robustness
GPS	Outdoor	Global space	10 mm (RTK)	High
UWB	Indoor	Location based range	1 cm	High
WLAN	Indoor	Location based range	1 meter	High
Vision-based (ARToolKit)	Indoor	Marker-based range	depends on distance 27mm for 3M	Low (depends on lighting, view angle and distance)
3D orientation tracker (InteriaCube3)	Indoor and outdoor	30 meters	0.03 degree	High

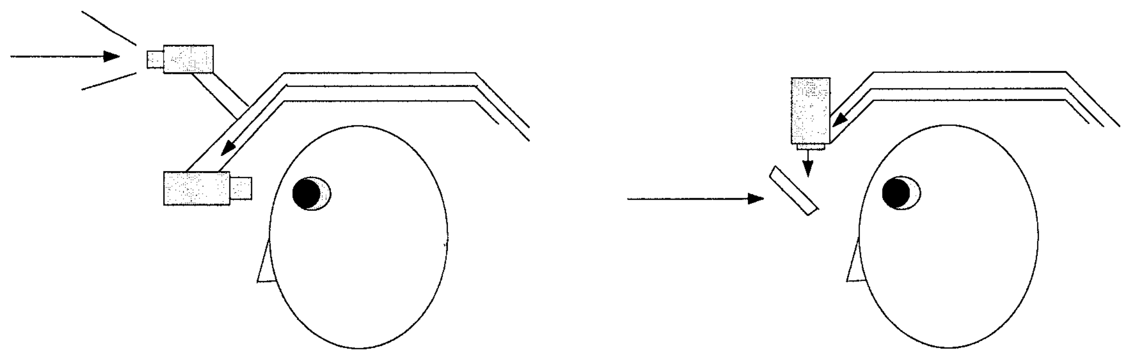
2.3.2 Displaying techniques

How to view the mixing of real and virtual is another fundamental design challenge for AR. Azuma et al. (2001) classified displays for viewing AR into three categories: head-mounted displays, hand-held displays and projection displays.

(1) Head-mounded displays (HMD)

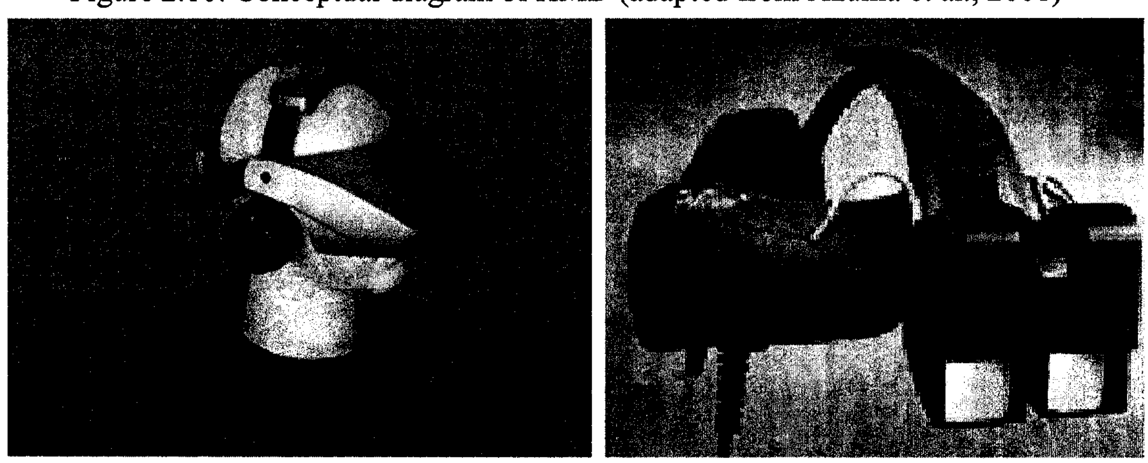
An HMD is a display device worn on the head that superimposes virtual and real scenes. The user mounts the display on his/her head and views the image in front of his/her eyes. It includes two types: optical see-though and video see-though.

Optical see-through HMD includes optical combiners. The optical combiners, such as mirror beam-splitters, are placed in front of user's eyes for viewing. It is transparent, so that the user can see through it the real environment. In contrast, video see-through has a video camera mounted on the head and a non-transparent display. It captures the real view in front of the user from the video camera, and then sends the mixed view which blends virtual and real to the display which is placed in front of users' eyes. Bell et al. (2001) present more details of the two kinds of HMD. Figure 2.10 shows the conceptual diagram of optical see-through and video see-through HMD. The main difference of optical see-through and video see-through is that the real world is directly shown by optical see-through. Figure 2.11 shows examples of an optical see-through and a video see-through HMD.



(a) Video see-through (b) Optical see-through

Figure 2.10: Conceptual diagram of HMD (adapted from Azuma et al., 2001)



(a) Video see through HMD: HMD 800 (5DT, 2007)

(b) Optical see through: ProView XL40/50 STm (Kaiser Electro-Optics, 2007)

Figure 2.11: Examples of two kinds of HMD

The decision of choosing optical or video see-through HMD is a main issue in AR system. For outdoor environments, the optical see-through HMD gives more flexibility and advantages because the optical combiner is transparent, while the video camera of video see-through constrains resolution of the images of the real world. Thus the optical see-through HMD provides better quality of view and user immersion. In addition, the video camera of video see-through causes some delay of the refresh rate. It increases the processing time and reduces the real-time performance of the whole AR system. Furthermore, in video see-through HMD, the video camera is mounted on the head with an offset to the real position of the eye. This offset may cause some displacements when we register computer-generated images with the real world. For outdoor usage, the real-time performance and accuracy of registration are important. Considering these issues, optical see-through is preferable for outdoor applications.

(2) Hand-held displays

Hand-held displays are applied in some AR applications such as Tablet PCs, PDAs, and cell phones because of their convenience (Fründ et al., 2001; Geiger et al., 2001; Wagner and Schmalstieg, 2003; Moehring et al., 2004). Most hand-held displays usually have flat-panel LCD and integrated video camera for mobile handling. The video camera captures images from the environment, and then the LCD displays the augmented image which overlays the virtual object and the real picture. This concept is similar to that of video-see though HMD. However, it reduces the realistic experience for user immersion, although it provides more convenience for mobile usage. Figure 2.12 shows an example of a cell phone based AR.

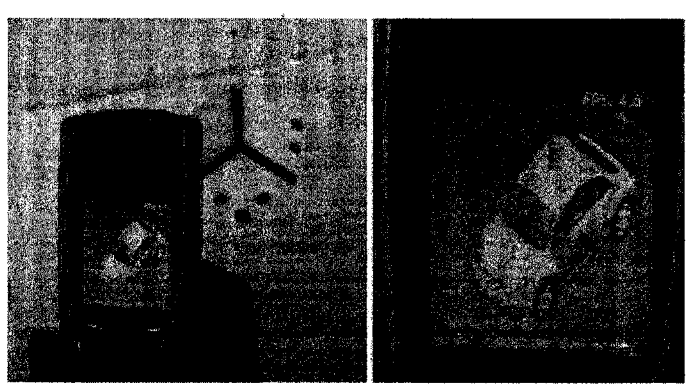
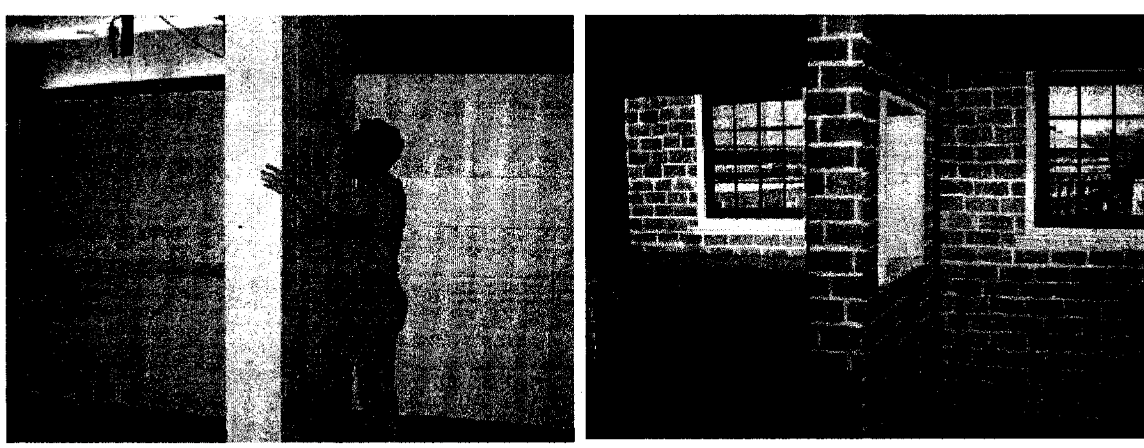


Figure 2.12: Video see-though example on a cell phone (Moehring et al., 2004)

(3) Projection displays

Projection displays directly project virtual images on the real physical objects' surface. Because of their excellent user experience and the scalable display range, they were applied in many AR applications (Bimber et al., 2002; Low et al., 2001; Raskar et al., 1999; Pinhanez, 2001; Unerkoffler, 1999). Figure 2.13 shows a projector-based AR of a test room. The display does not need any mounting and it can cover a large irregular physical object with multi-overlap. Most applications of the display are limited to the indoor environment because of the requirements of lighting and space conditions. Although it is an interesting and valuable approach which provides strong user immersion and support for multi-users, it is still difficult to apply for outdoor environments.



(a) Physical surface (b) Augmented with projection

Figure 2.13: Projector-based augmented reality (Low et al., 2001)

2.3.3 User interaction

AR provides an innovative method for human-computer interaction. Thus, one important issue of AR is how to effectively interact with the users of the AR environment. On the other hand, usability decides user experience of AR applications. Hix and Gabbard (2002) presented usability as ease of use and usefulness involving quantifiable characteristics such as subjective user satisfaction, speed and accuracy of user task performance, user error rate, etc. Shneiderman (2000) stated that usability is a part of "usefulness" and includes efficiency of use, memorizability, learnability and satisfaction.

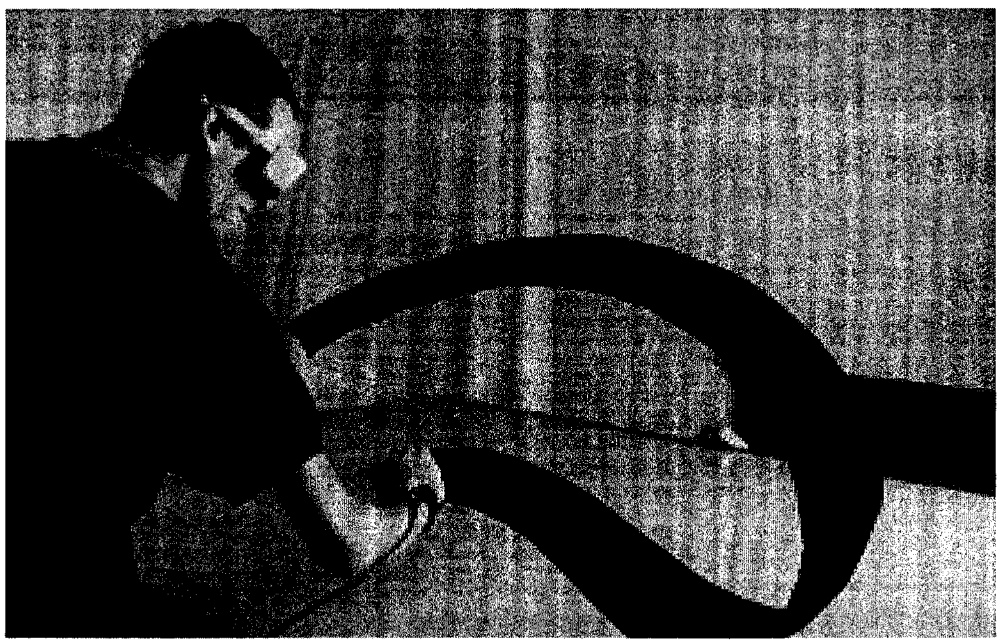


Figure 2.14: Drawing in 3D space with hand-tracking (Schkolne et al., 2001)

The typical user interaction methods such as WIMP (windows, icons, menus, and pointing) may not be suitable for AR applications. For example, it is difficult to use a typical mouse to click on an icon for a mobile person in an outdoor AR application. In addition, AR has to deal with combination of virtual objects and real physical world, so that the selection of input devices is crucial for AR applications. Many researchers applied a variety of input devices and user interfaces for better AR usability. Foxlin and Harrington (1998) presented a "sourceless" prototype system of ultrasonic tracking with acoustic emitters worn on hand and microphones mounted on head, so that the users can use gestures for user control and input. Another interesting prototype system for creating organic 3D shapes in 3D space is developed using sensors for hand tracking as shown in Figure 2.14 (Schkolne et al., 2001).

Many new techniques such as handwriting recognition, speech recognition, gaze tracking, etc. can be integrated with AR for usability as long as they can improve the usability for practical applications. We have to carefully select and use input and output devices and techniques based on the requirements and environment of AR applications.

2.3.4 Collaboration and distributed AR

Collaborative tasks present the requirement of sharing the view of the real workspace and virtual objects (Ahlers et al., 1995). Distributed computation provides a promising solution for this issue. Many AR researches explored the collaborative tasks in AR with distributed computation methods (Zhong et al., 2002; Hamza-Lup et al, 2007; Rolland et al., 2003; Hicks et al, 2003). For performance issue, those researches only distributed the data which cannot be acquired from local computing and are necessary for generating the augmented view. In the case of construction tasks in outdoor environment, the collaboration activities are mainly affected by multi users operations. Therefore, this research is focused on distributing the states of multi user operations for visualization.

2.4 SUMMARY AND CONCLUSIONS

In this chapter, the literature of the current research, applications and challenges of AR have been reviewed. There are many different approaches of AR, but all of them have high demand for accurate tracking and registration. In addition, displaying also needs solutions for efficiently integrating virtual and real environment. Furthermore, the user interaction can use a variety of equipment for the benefits of usability. Little research has been undertaken on engineering applications of AR in the outdoor environment because of many technical difficulties and design challenges as discussed above.

AR is especially useful to visualize complex engineering tasks. For example, the construction industry always uses heavy equipment for construction activities with space and time limitation. This situation creates many challenges such as the need for detailed erection simulation, support for decision-making, safety analysis, layout planning, logistics planning, equipment selection, etc. AR can be used to address some of these challenges. However, applying in construction simulation brings many issues, such as outdoor tracking, dynamical object modeling, real-time interactive operation, physical simulation, etc.

Based on the literature review, the main focus of the research is on the development of a practical method for designing and creating an AR system for construction simulation in outdoor environment and improving the efficiency of interaction in the system.

CHAPTER 3

AUGMENTED REALITY FOR VISUALIZING COLLABORATIVE CONSTRUCTION TASKS

3.1 INTRODUCTION

As discussed in Chapter 2, AR is useful for many engineering applications, such as construction simulation, manufacturing and maintenance. However, for outdoor environment AR systems have many technological challenges and specific requirements, such as accurate tracking, mobility and display quality, etc. A large amount of time and money has to be invested to design and integrate several advanced techniques and high-end equipment. Therefore, there are needs and opportunities for developing a generic framework for outdoor mobile AR applications in engineering areas.

In this chapter, we introduce an integrating method for creating AR systems to simulate outdoor construction activities by integrating position and orientation tracking, engineering constraints, dynamical modeling, and real-time user interaction. In addition, we will describe a framework for using these techniques to suit the requirements of mobile AR systems.

3.2 DESIGN CHALLENGES

The basic design decision in building an outdoor AR system is how to accomplish the combination of the real and virtual worlds (Azuma, 1997). Figure 3.1 provides an overview of the design space of the proposed outdoor AR system. The figure identifies the main components needed in an AR system for visualizing construction simulation, considering multi-user collaboration and engineering constraints. These components are: tracking and registration, dynamic object modeling and constraints, user interaction, and communication. Each component technology is discussed in greater detail in the following.

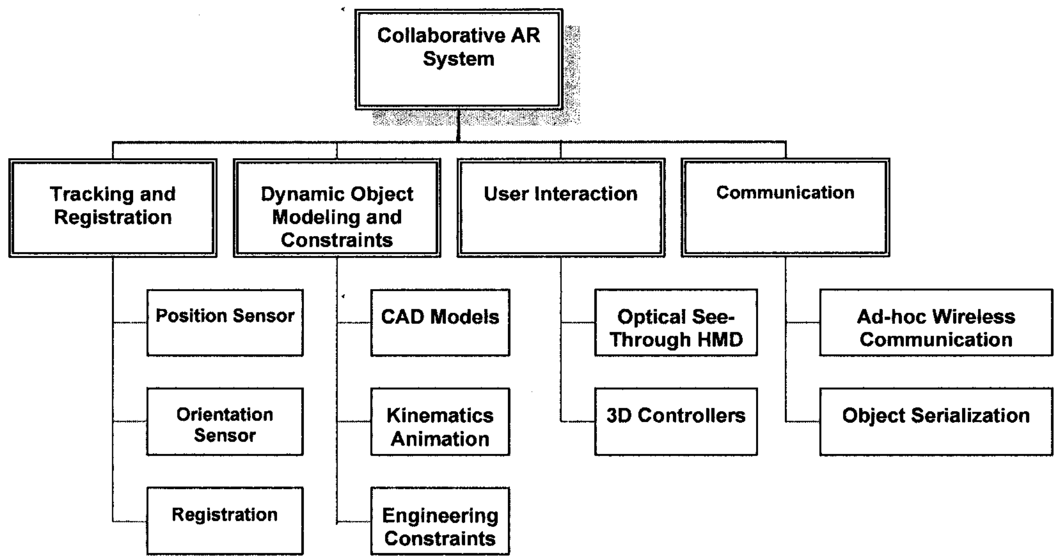


Figure 3.1: Design overview of collaborative AR system

3.3 TRACKING AND REGISTRATION

Tracking and registration are among the most important technological challenges of AR. Accurately tracking the user's position and viewing orientation is crucial for AR applications (Azuma et al., 2001). Outdoor AR brings more difficulties for accurate tracking because it is usually unprepared with the tracking infrastructure. In recent years, tracking technology in the outdoor environment has been improving steadily (Azuma et al., 2006). Figure 3.2 shows the abstract model of the workflow of tracking and registration. The model includes two tracking parts: position and orientation tracking of the user using special tracking devices. The workflow is described as follows: (1) The computing unit calculates the related position and orientation of 3D objects based on the tracking result; (2) The render unit renders the virtual 3D objects using the computed result; and (3) The resulting video frame is sent to the user's HMD.

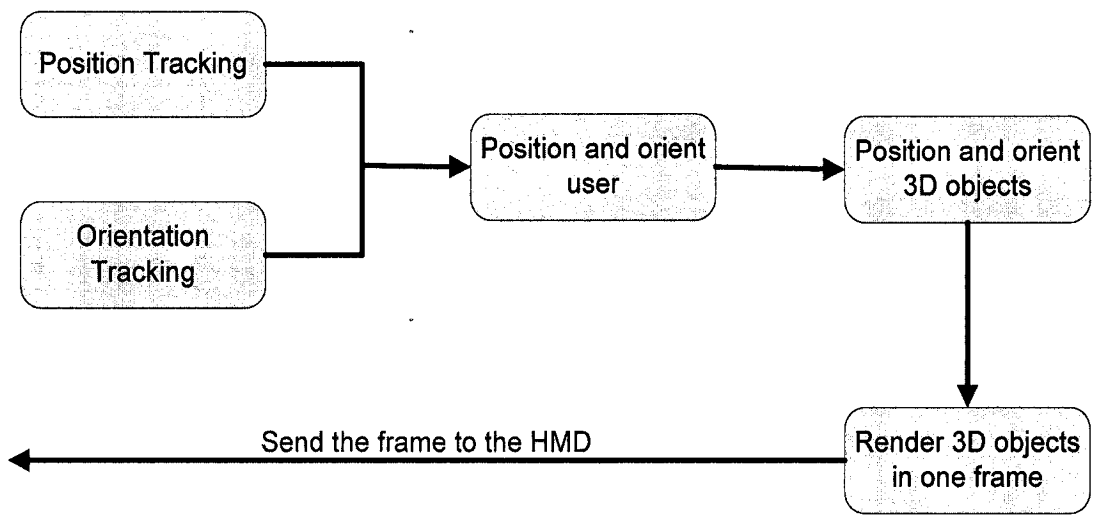


Figure 3.2: The workflow of tracking and registration

3.2.1 Position tracking

Based on the discussion in Chapter 2, GPS represents competitive features for outdoor position tracking. Many researches explored the AR applications with GPS tracking (Hollerer et al., 2001; Piekarski and Thomas, 2002; Azuma et al., 2006). Trimble (2005) provided Trimble 5700 GPS receiver as a flexible and cost-effective solution for GPS-based tracking. It is a 24-channel dual-frequency RTK GPS receiver which has the Trimble Maxwell technology for superior tracking of GPS satellites, increased measuring speed, longer battery life through less power use, and optimal precision in outdoor environments. It can receive available differential correction signals with accuracy of approximately 1 centimeter. Therefore, Trimble 5700 RTK GPS receiver is selected as a suitable solution for position tracking in our research because of its availability and relatively high accuracy and low cost.

3.3.2 Orientation Tracking

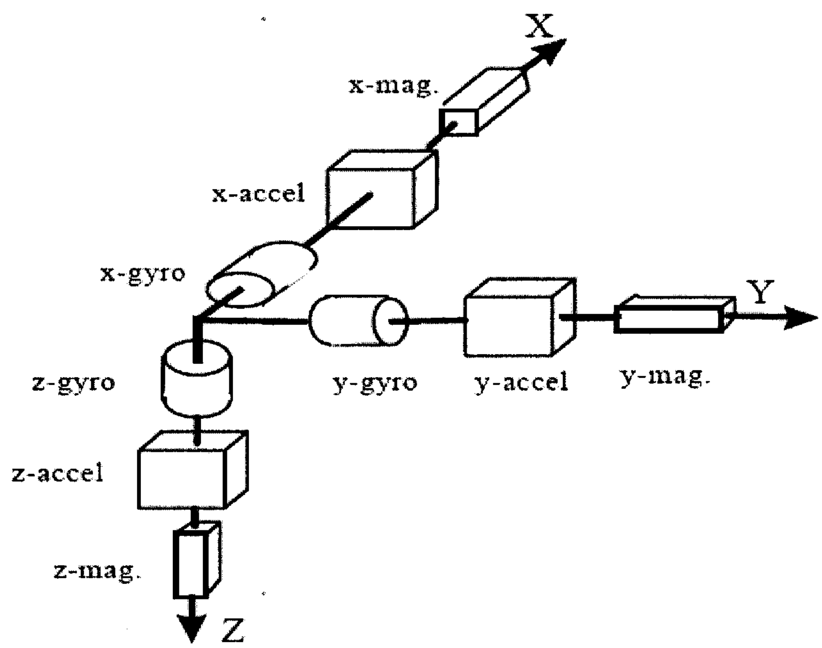


Figure 3.3: Functional diagram of InertiaCube3 (InterSense, 2007)

The orientation tracking methods of gyroscope, magnetometer and accelerometer provide a promising and robust solution for real-time tracking in indoor or outdoor environments. Based on the discussion in Chapter 2, it is difficult to find a perfect standalone method for orientation tracking. Azuma et al. (1998) discussed that hybrid tracking systems are required since each technology has limitations that cannot otherwise be overcome. InterSense (2007) provided a hybrid tracking solution integrating gyroscopes, accelerometers, magnetometers using Kalman filter techniques. Figure 3.3 shows the functional diagram of the product. It can measure 9 physical properties, namely angular rates, linear accelerations, and magnetic field components along all 3 axes by combining gyro, accelerometer and magnetometer sensors on each axis.

The orientation tracking architecture has several compelling advantages: (1) It is simple and suitable for tracking with scalable-range capabilities; (2) It is possible to wear the whole orientation tracking system including all of the tracking devices and the computational unit; and (3) It provides high update rates and accuracy (Table 4.1). Therefore, this tracking method has been adopted for our implementation.

3.3.3 Registration

The crucial issue of registration is how to accurately align virtual objects with real ones (Azuma, 1997). To align virtual objects, we firstly should calculate the transformation matrix of user's viewpoint. Due to the dependency of registration on tracking, we divide the calculation into two parts: position transformation and orientation transformation.

(1) Position transformation

To translate an object by a vector *v*, each homogeneous vector *p* (written in homogeneous coordinates) would need to be multiplied by this translation matrix:

$$T_v = \begin{pmatrix} 1 & 0 & 0 & V_x \\ 0 & 1 & 0 & V_y \\ 0 & 0 & 1 & V_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

Based on the GPS tracking, the translation vector v can be calculated by the difference between current position and last position. Then we can easily build the translation matrix T_v for registration.

(2) Orientation transformation

As discussed in Section 3.3.2, the InertiaCube3 sensor is chosen as the orientation tracking unit. The sensor outputs the Euler angles as the orientation tracking results. We convert the angles to the matrix form for transform computation. The Euler angles include yaw, pitch and roll for rotations of three axis. A yaw is a counter-clockwise rotation of α about the Z axis. The rotation matrix is given by

$$R_z(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

A pitch is a counter-clockwise rotation of β about the Y axis. The rotation matrix is given by

$$R_y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

A roll is a counter-clockwise rotation of γ about the X axis. The rotation matrix is given by

$$R_z(\gamma) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma & 0 \\ 0 & \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

The yaw, pitch, and roll rotations can be used to place a 3D body in any orientation. A single rotation matrix can be formed by multiplying the yaw, pitch, and roll rotation matrices to obtain:

$$R(\alpha,\beta,\gamma) = R_z(\gamma)R_y(\beta)R_x(\alpha) =$$

$$\begin{pmatrix} \cos\alpha\cos\beta & -\sin\alpha\cos\beta & \sin\beta & 0 \\ \cos\alpha\sin\beta\sin\gamma+\sin\alpha\cos\gamma & -\sin\alpha\sin\beta\sin\gamma+\cos\alpha\cos\gamma & -\cos\beta\sin\gamma & 0 \\ -\cos\alpha\cos\gamma\sin\beta+\sin\gamma\sin\alpha & \sin\alpha\sin\beta\cos\gamma+\cos\alpha\sin\gamma & \cos\beta\cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

It is important to note that $R(\alpha,\beta,\gamma)$ performs the roll first, then the pitch, and finally the yaw if the axes taken as fixed. If the axes move with the object, the rotations are performed in the opposite order: yaw, then pitch, then roll (Buss, 2003).

In addition, the calculations of the translation and rotation matrix are not fully synchronized procedure because of the different update frequencies of different tracking equipment. Thus we apply the two rotation and translation matrices to change the current view point, respectively.

3.3.3.1 View model

Influences of the physical environment on the view model have to be considered when we adopt HMD. For example, different users have different view parameters, such as the distance between the left and right eyes. The predefined or automatically computed head position may not exactly specify the user's view. In practice, the factors that may affect the user's view include the actual field-of-view of the HMD, the HMD's pixels per inch, user's physical height and other such parameters. The policy of our view model is to ensure that the user's eye position in the real world corresponds to the virtual eye position in the virtual world. This is realized by positioning the view platform at the same location the user's eye relative to a geo-referenced world coordinate system. In other words, the view that the renderer draws on the HMD's display must match what the user would see had the experience occurred in the real world. To provide this experience, what the user can see will be changed by applying several transformation on virtual coordinates when the user moves in the real world. The view model is discussed in the following.

You et al. (1999) described the principal coordinate systems of AR. Figure 3.4 shows the five principal coordinate systems: Geo-referenced coordinate system, W:(X_w, Y_w, Z_w); Camera or eye centered, C:(X_c, Y_c, Z_c); orientation-centered, I:(X_i, Y_i, Z_i); position-centered, V:(X_v, Y_v, Z_v) and 2D screen coordinates, U:(X_u, Y_u).

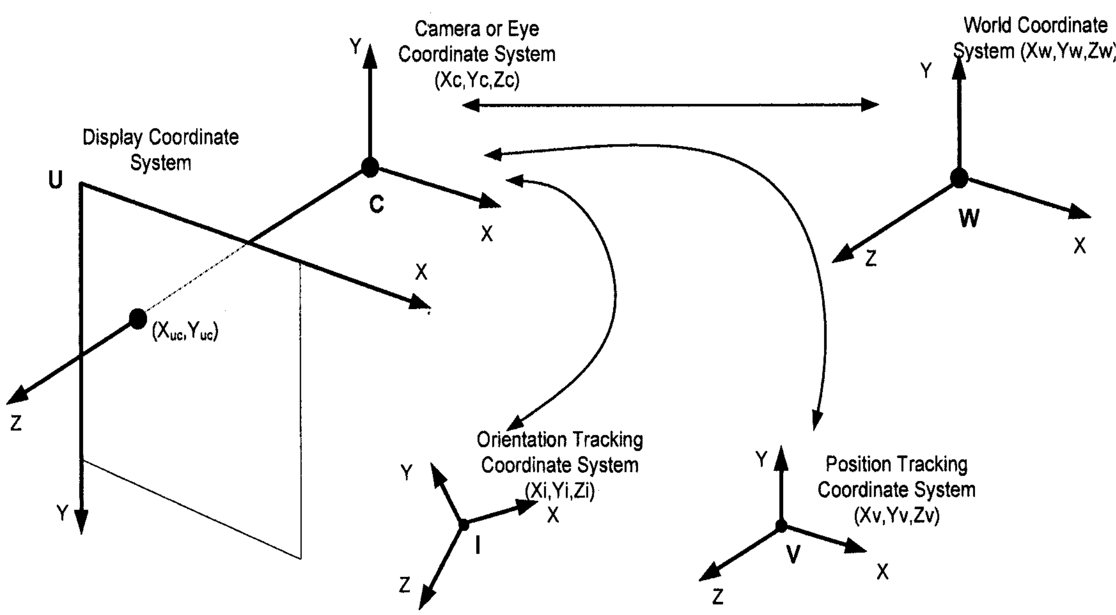


Figure 3.4: View model and related coordinate systems (adapted from You et al., 1999)

The transformation from O and P to the C are introduced to calibrate the tracking results considering that the tracking devices are not centered at the eye or camera viewpoint. However, this calibration is outside the scope of the present research and the three systems are considered identical.

A camera or eye models the imaging process. The origin of C lies at the camera's projection center. The transformation from W to C is

$$W \to C: \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = R_{wc} T_{wc} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{6}$$

The rotation matrix R_{wc} and the translation vector T_{wc} characterize the camera's orientation and position with respect to the world coordinate system. Under perspective projection, the transformation from W to U is

$$W \to U: \begin{bmatrix} X_u \\ Y_u \\ Z_u \\ 1 \end{bmatrix} = K R_{wc} T_{wc} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{7}$$

where the matrix K represents the intrinsic parameters of the camera used in video see-through HMD, such as the focal length of the camera and the horizontal and vertical pixel sizes on the imaging plane. Because this research uses an optical see-through, the rest of this section will focus on obtaining the matrices R_{wc} and T_{wc} based on tracking information.

In this research, the X and Z axes of world coordinate system W composing the planar surface are matched with the tow Cartesian axes used in small scale urban geographic information systems (GIS) maps, while the Y axis of W is matched with the altitude. This decision is very important because it allows us to use widely available GIS maps for locating virtual objects on the construction site. Consequently, the position information of the virtual objects and the tracking information can be both geo-referenced against the same world coordinate system W. The V_x, and V_z components of the translation vector are calculated by applying the Modified Transverse Mercator (MTM) projection algorithm on the latitude and longitude values obtained from the GPS receiver. The V_y component of the translation vector is equal to the altitude value obtained from the GPS receiver. Then, the translation matrix T_{wc} and rotation matrix Rwc can be obtained from Equations (1) and (5).

3.4 DYNAMIC OBJECT MODELING AND ENGINEERING CONSTRAINTS

Based on the development in equipment hardware, development in simulation software is making it possible to train crane operators using VR (Simlog, 2006) and to visualize the results of construction simulation (Kamat and Martinez, 2001). Furthermore, much research has been done to study the selection of cranes and simulating the working processes using integer programming and optimization techniques (Lin and Haas, 1996; Law and Kelton, 2000) and three-dimensional graphics (Hornaday et al., 1993; Dharwadkar et al., 1994; Watt 2000; Watt and Policarpo, 2003). However, these works do not consider run time calculation of engineering constraints and task-orientated dynamic object modeling.

3.4.1 Dynamic Object Modeling

Each object stores the parameters needed to represent and solve the equations of motion in an object model. The description of the object model is divided into three categories: kinematics, geometric, and graphical. Figure 3.5 shows an example of an object data structure for crane manipulator. The structure of object modeling is explained in following.

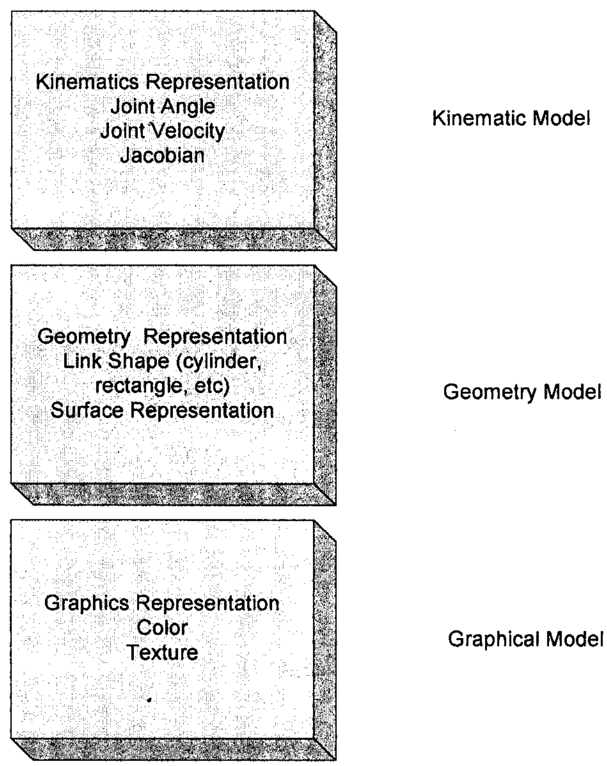


Figure 3.5 Abstract object structure of 3D equipment (adapted from Lee et al., 1994)

The graphical layer will describe color, texture, etc. It determines the visual image of the objects in a graphical rendering of the scene representing such qualities as color, lighting, and shading.

The geometric model stores the geometric description of each object, such as shape and surface features, etc. This information is used to find if contact between objects has occurred and, if so, the location of that contact. The information can then used to resolve the constraint force equations. CAD software is used to create 3D models so that these models are saved separately in CAD files. Therefore these pre-existing models can be reused to populate the scene with virtual objects for real time rendering.

The kinematic model incorporates data such as joint angles, the Jacobian, and transformation. These data may be used to calculate parameters such as the position and orientation of the model parts. Homogeneous transformation matrices are usually chosen to express this information in most researches (Denavit and Hartenberg, 1995; Fu et al., 1987; Foley et al., 1990).

In order to build a general model of a crane that can be used in interactive AR simulation, a hierarchical crane model should be created to represent the different components of the crane and their topological relationships. Furthermore, kinematics animation is essential for producing the interaction needed to operate a crane in the AR environment. Figure 3.6 shows the hierarchical structure. Figure 3.7 (Davis and Sutton, 2003) shows an example of a hydraulic crane that has four degrees of freedom (DoF) for the movements of the boom and hook. To represent the four degrees of freedom, the kinematics actions have to be considered in modeling.

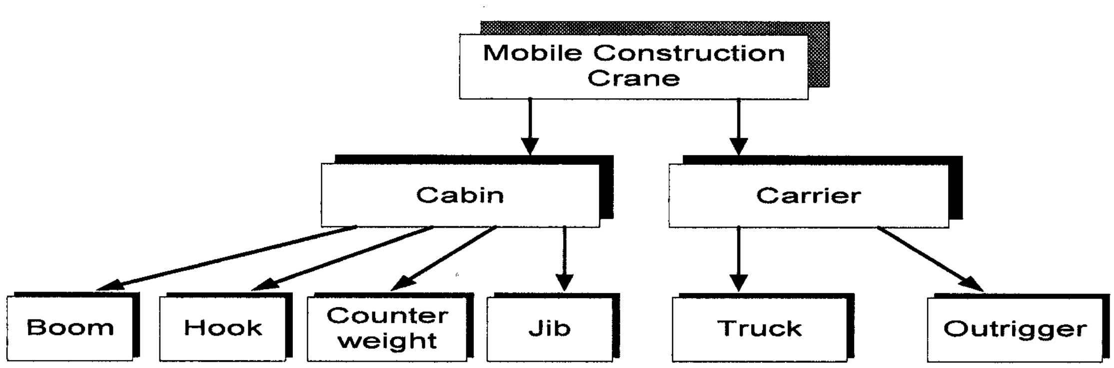


Figure 3.6: The hierarchical structure of a mobile crane

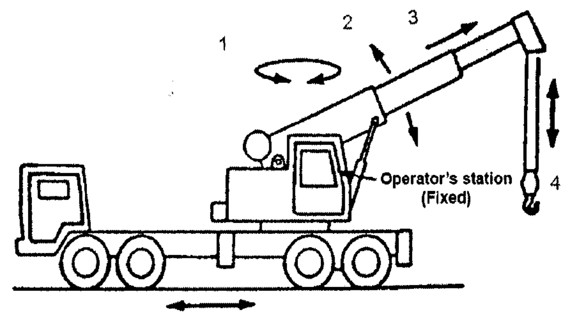


Figure 3.7: The DoF of a crane (Davis and Sutton, 2003)

Kinematics focuses the geometric properties of the motion of points without consideration of their masses or the forces applied on them (McCarthy, 1990). It emphasizes how the position of an object changes with time. An object in which the distance between any two given points of the object always remains constant is called a rigid body. Kinematics can be classified as forward kinematics and inverse kinematics. Forward kinematics (FK) focuses on retrieving a system's pose based on the position and orientation of each component of the system. The mathematical statement of FK is generally written as: X=f (Θ), where X is the final position and orientation vector of the free end of the chain (called the end effector) and $\Theta = (\theta_1, \theta_2, \ldots \theta_n)$ is the state vector that describes the configuration of the chain by encapsulating the position and orientation vectors of each constituent link.

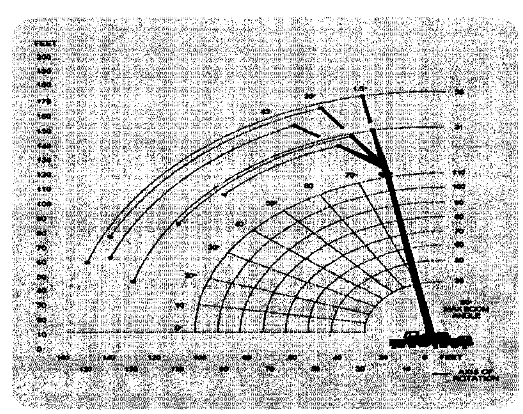
Inverse Kinematics (IK) can be explained as the reverse of FK and is widely applied in the robotics industry (Buss, 2004). It is the process of calculating the parameters of the kinematics chain to achieve a given pose. The mathematical statement of IK can be described as $\Theta = f^{-1}(x)$. The more difficult solution to the inverse kinematics problem is to find the joint angles given the desired configuration of the figure (i.e., end effector). In the general case, we cannot find an analytical solution for the inverse kinematics problem. However, inverse kinematics may be solved via nonlinear programming techniques (Buss, 2004). Since AR requires real time user interaction, we should consider FK in most situations. However, if some specific applications need to realize inverse kinematics, the Cyclic Coordinate Descent (CCD) method can be used for this purpose (Wang and Chen, 1991). For example, the initial and final positions of the lift can be used to calculate the inverse kinematics transformations to be applied on the crane model using the CCD method.

3.4.2 Engineering Constraints

In recent years, there has been much interest for real time animation with dynamic constraints. However, to the best knowledge of the author, previous research did not implement engineering constraints with kinematics actions. For most engineering applications, the accuracy and efficiency of the simulation are affected by engineering constraints such as the strength of boom, loading capacity, etc. In order to achieve realistic simulation, the engineering constraints related to the equipment's mechanical behavior should be considered in the simulation. For example, the simulated operation of a crane should respect the constraints imposed by the working ranges and load charts of the crane (Figure 3.8). Recently, Al-Hussein et al. (2001, 2005) developed a system that can assist in selecting and locating cranes on construction sites using the information of load charts and working range. This system uses a crane database, named D-Crane that has the load charts of different manufacturers and the key dimensions of each crane including its carrier, main boom, jibs/extensions, and accessories. The system ensures that the selected crane has the required lift capacity and can fit on site by satisfying a set of constraints described with detailed equations. However, the system supports a set of predefined configurations of cranes. Therefore, for dynamic calculation of loading and real-time animation, the system may not be able to adequately represent the engineering constraints.

To represent engineering constraints in real-time 3D construction simulation, several rules are developed to represent these constraints which are stored in a database. The working range shows the minimum and maximum boom angle to the ground according to the length of the boom and the counterweight. Load charts give the lifting capacity based on the boom length, boom angle and the counterweight. After interactively selecting a crane from the database and positioning it in the AR environment at a suitable location, the user of the system can specify the lift weight of a virtual object before operating the crane to simulate the lifting task. The system can dynamically detect the changes of the boom length and angle and check whether the maneuvering operation is allowed by querying the database.

(Feet)	36	50	*60	70
10	+140,000 (68)	109,500 (75)	84,200 (78)	**56,450 (80)
12	110,500 (64)	104,500 (72.5)	79,850 (76)	56,450 (78.5)
15	96,800 (58.5)	91,400 (69)	73,900 (73)	56,450 (76)
20	78,750 (47)	75,300 (62)	59,600 (67.5)	56,450 (71.5)
25	59,800 (32.5)	59,750 (55)	50,000 (62.5)	48,900 (67)
30		47,300 (47)	42,300 (56.5)	41,900 (62.5)
35		38,550 (37.5)	38,950 (50)	36,400 (57.5)
40		28,450 (24.5)	28,450 (43)	29,700 (52)
45			23,400 (34.5)	24,650 (46.5)
50			19,450 (23)	20,700 (39.5)



(a) Load chart (b) Working range

Figure 3.8: Example of load chart and working range (Groove Crane, 2006)

3.4.3 Integration method

This section identifies the main steps needed in our AR system for visualizing crane selection, operation and activity analysis. Figure 3.9 diagrammatically illustrates the method of the conceptual process. The method will integrate information from, and add information to, the following databases and models (Hammad et al., 2006):

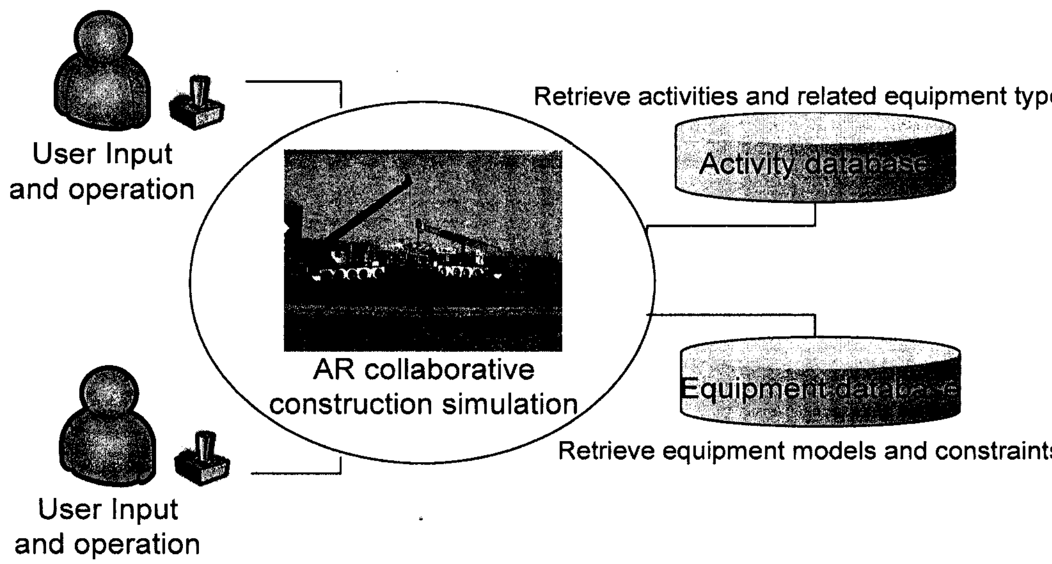


Figure 3.9: General procedure of analysis of crane operation
(adapted from Hammad et al., 2006)

(1) Activity database

This database includes information about all the activities in the construction project, such as the start and finish times of each activity, the target physical components and their attributes, and the types of equipment required in that activity. For example, in the project of the replacement of a bridge deck that is discussed in the case study, a typical activity is the replacement of an old section of the deck with a prefabricated panel. In this example, each activity will include the start and finish times, the ID number of the target section, and the required equipment such as cranes, trucks, etc.

(2) Equipment databases

The equipment database has the specifications about the different models of certain types of construction equipment including the related constraints for using this equipment. Equipment manufacturers and large construction companies usually have databases of different equipment used in their work. D-Crane is a good example of such databases (Al-Hussein, 2006). The equipment database used in the system includes the 3D models of the equipment, which are used for dynamic object modeling as explained in Section 3.4.1.

Using the above databases and the 3D models, the following procedure can be applied:

(1) Each user starts by moving to a location at the construction site at which he/she want to observe the construction activity from a third-person viewpoint.

(2) The main activity to be considered in the construction simulation is selected by one of the users. The system retrieves the information about this activity and all other overlapping activities from the activity database. The information includes the related objects and the required equipment types.

(3) Then, feasible equipment are selected for each required type from the corresponding equipment database. It should be noted that selecting the optimal equipment is beyond the scope of this research.

(4) The next step is to retrieve the equipment 3D models from the database. Other important parameters can be input by the user, such as the lift weight of a crane which is used in the simulation.

(5) Locating virtual construction equipment and other objects on the construction site in the system is done interactively using a geographic information sub-system. The users

click on a digital map of the construction site to approximately locate the virtual objects which are used in the selected activity. This digital map has the same MTM projection used in processing the position tracking for registration. The altitude of the virtual object is extracted automatically from the digital terrain model of the site at the specified location.

(6) In this step, the 3D models of the equipment are generated at the specified location on the construction site. The users interactively operate the equipment in the 3D augmented space of the system using the joysticks. They have the freedom to move around the construction site to observe the augmented scene from several viewpoints. The system continuously checks whether the maneuvering operations are allowed by querying the engineering constraints.

The proposed component has the following functionalities:

(1) Based on the information of the abstract model in the database, the system can generate instances of different kinds of cranes combining the graphical representations of their components. A transformation matrix is used to specify the relative location of each component with respect to its parent. For example, the first part of the boom is located relative to the carrier of the crane. After generating the crane model, the prototype system applies the suitable transformation (scaling, rotation, and translation) on the crane so that it is correctly located at the initial position in the AR environment. The parent-child relationships between the components guarantee the consistency of the relative positions of the components when the user simulates the operation of the crane. For example, when the user starts extending the length of the boom from its initial compact length, a translation is

applied on the second part of the boom with a step of unit length up to the total length of the second part. Then, the same transformation is applied on the third part of the boom, and so on, until the boom reaches its full length.

(2) In addition to following the kinematics relationships, the simulated operation of a crane should respect the constraints imposed by the working ranges and load charts. Several rules are developed to represent these constraints which are stored in the database. The working range shows the minimum and maximum boom angle according to the length of the boom and the counterweight. Load charts give the lifting capacity based on the boom length, boom angel and the counterweight.

(3) Furthermore, geometric attributes (e.g., transformations, normal vectors) of the physical objects (e.g., structural elements) are computed based on information extracted directly from the scene graph. This information is used to locate the crane relative to a physical object. The normal vectors of the surfaces of the objects are used to represent the orientation of the crane, and offset distances along those vectors are used to define its relative location.

(4) After interactively selecting a crane from the database and positioning it in the AR environment at a suitable location, the user of the system can specify the lift weight before operating the crane to simulate the lifting task. The system can dynamically detect the changes of the boom length and angle to ground and check whether the maneuvering operation is allowed by querying the database.

3.5 USER INTERACTION

As discussed in Chapter 2, real-time user interaction is crucial in AR system. Controlling the motion of objects in AR implies using classical computer animation techniques to apply realistic interaction with computer graphics objects. AR generally uses the motion of body to control the view that is displayed to the user. While tracking the body is critical to rendering AR and VR displays, fine grained operations typically require using hands to control input devices. Many AR and VR researchers have developed intuitive interfaces that involve the use of hand gestures supplied by high quality tracking systems. While outdoor AR is similar and using these techniques is desirable, there are a number of unique problems that require distinct solutions. Some of the input/output devices are not intuitive. Shneiderman (1983) and Johnson et al. (1989) discussed how adding levels of indirection may make user interfaces less intuitive to operate. In summary, the choice of interface of Mobile AR has to consider the functionalities, safety, and usability of the users in an outdoor, mobile setting. Therefore, we have to consider the design of interaction methods that will improve the performance of the applications based on the framework.

Our user interaction integrates 3D models, tracking technologies, mobile computing and distributed wireless communication in one framework. This integration will result in the following advantages: (1) Visualizing different types of data including equipment 3D models, engineering constraints, etc; and (2) Providing a user-friendly interface which can facilitate interaction with 3D models and reduce data input errors. Our user interaction system includes 4 components: display, user operation, user location display, engineering constraints.

(1) Display component

As discussed in Chapter 2, the optical see-through HMD is a reasonable choice for our mobile AR framework. In order to provide a mixed view, the HMD is considered as our main display equipment for the AR system. However, the selection of the specific optical see-through HMD device should consider the display resolution and refresh rate, and the visibility of the virtual objects under ambient light conditions in the outdoors. Figure 3.10 shows the nVisor ST, the HMD we choose for our system, which has a resolution of

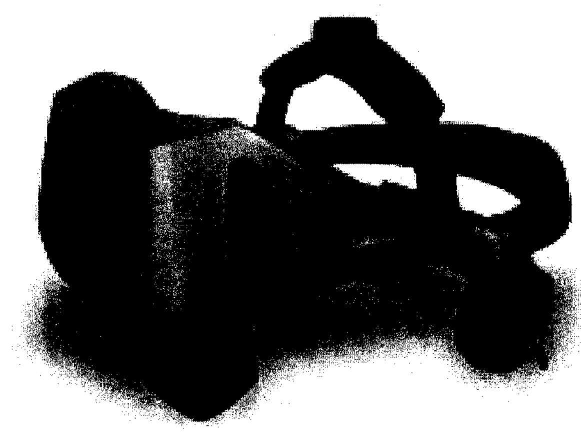
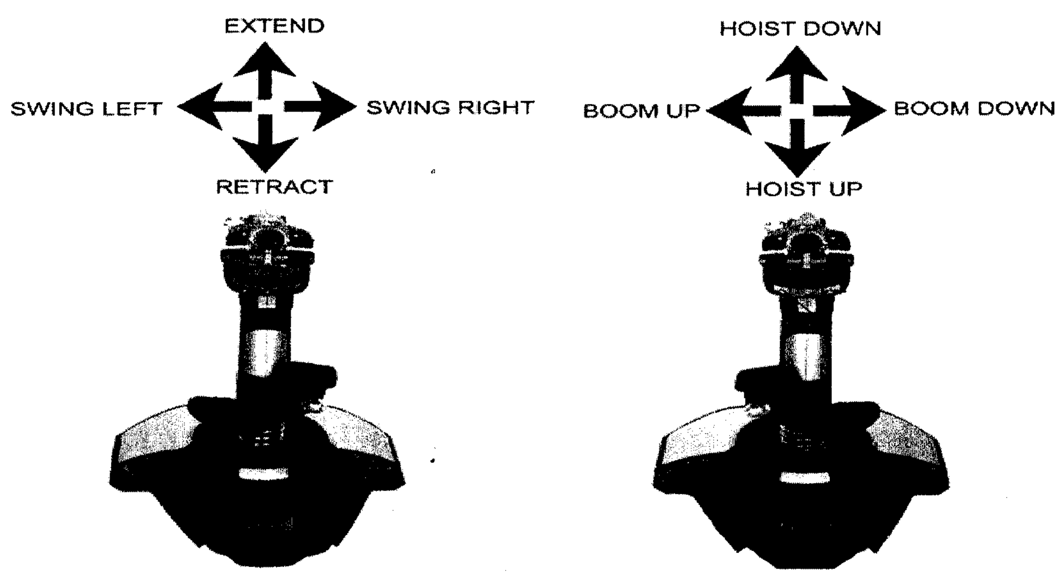


Figure 3.10: nVisor ST HMD (NVIS, 2007)

1280×1024 pixels and 60Hz vertical frequency. Furthermore, the nVisor ST has an ergonomic design which provides a wide field-of-view and its setting can be easily adjusted, such as inter-ocular distance, eye-relief to accommodate users with eyeglasses, etc. The user also can balance and comfortably align the HMD with a robust head-fitting apparatus. In addition, it includes external mounting points for most common head tracking sensors such as InertiaCube. Furthermore, it can receive Video Electronics Standards Association (VESA) standard Super eXtended Graphics Array (SXGA) video in either digital or analog formats. The video control electronics support both stereoscopic and monoscopic viewing by detecting dual or single inputs, respectively. An analog output is provided for driving a repeater monitor for checking and debugging (NVIS, 2007).

(2) User operation component



(a) Left control (b) Right control

Figure 3.11: User control of crane motion using joysticks (Simlog, 2006)

Construction equipment operators usually perform operations using two hands controlling the construction equipment, such as cranes. Many simulation software packages use joysticks to train crane operators because joysticks provide the best operation experience and are easy to use with minimum training (Simlog, 2006). Cranes require the representation of 4 degrees of freedom at least. Only joysticks and some specific 3D controllers may solve the challenge of flexibility of operation simulation. Furthermore, users can easily use joysticks without any additional knowledge and training because of their user friendly design. Thus, to simplify operation tasks for the user of our system, operations such as moving, rotating, and extending the boom of the crane are implemented using two joysticks. Figure 3.11 shows an example of crane control using joysticks. It provides an operation mode similar to the real crane control. Users can perform swing of the boom of the crane (left/right), boom length (extend/retract), boom angle to ground (up/down), and hoist (up/down) using the two joysticks. Other functions are available to drive the equipment, control the outriggers, etc.

(3) User location display component

One of the important issues in developing AR applications for engineering is to use a systematic method to display the tracking information. We developed a data integration method based on synthesizing information from geographic information system (GIS) maps and GPS tracking information. Integrating 2D GIS maps with GPS information helps in providing current position on site. The feature is useful when combined with the AR tracking component. Users can find their current location and related geography information for navigation. An API of embeddable mapping and GIS components, MapObject (Mapobject-Java, 2006), is used for developing this functionality.

3.6 MOBILE COMPUTING AND COMMUNICATION

Mobile computing and communication is critical for AR in outdoor environment because of the limitations of the cognitive performance of users in the outdoors. There are several factors that limit the choices of a mobile computing platform, including the size, graphics and multimedia capabilities, power, etc.

Wireless communication is another important issue of AR in outdoor mobile environment. It is essential for a multi-user collaborative system because it enables the users to exchange information about their collaborative construction activities. For wireless communication in outdoor environment, ad-hoc technology provides potential solution for point-to-point connection without a base station (Xu et al., 2006). It reduces the difficulty of setting wireless stations in outdoor construction sites and provides suitable wireless service for multi-users. The client-server design is chosen to implement the communication considering efficient performance. It is comprised of two logical parts: a client requesting computing services and a server providing the services. It means the client-server computing relates two or more threads of execution using a consumer/producer relationship (Lewandowski, 1998). The clients make requests to servers for computing services or information and then use the response to compute their own task. Each client is served by a thread of the server. The threads compute and send information or computing results to every client. In addition, the object serialization method is used to provide the quality of transmission in network communication. With object serialization, it is easy to take any object, and make it persistent, without writing custom code to save object member variables (Reilly, 2000). The method takes a virtual object's state, and converts it to a stream for cloning another virtual object which is operated by another user. Hence every user can obtain the other user's operations and visualize these operations in their computers.

3.7 CONCEPTUAL CONFIGURATION

Figure 3.12 shows an overview of the hardware configuration for the proposed system. The 3D orientation tracker provides the yaw, pitch, and roll angles which define the orientation of the user, and the GPS receiver provides the current position of the user in the global coordinate system. In addition, the wireless communication unit may provide the information about other equipment's positions and orientations. Furthermore, joysticks send the user's operation information to the scene generator. The scene generator accurately renders and aligns the virtual objects based on the information. Finally, an optical see-through HMD overlays the virtual scene from the scene generator and the real world scene as viewed with the HMD. In other words, the user can see the real world augmented with generated stereoscopic virtual images. The user can naturally experience the AR and operate the virtual objects as if they exist in the real environment. Advanced user interaction and tracking are two major features which are provided by the conceptual configuration.

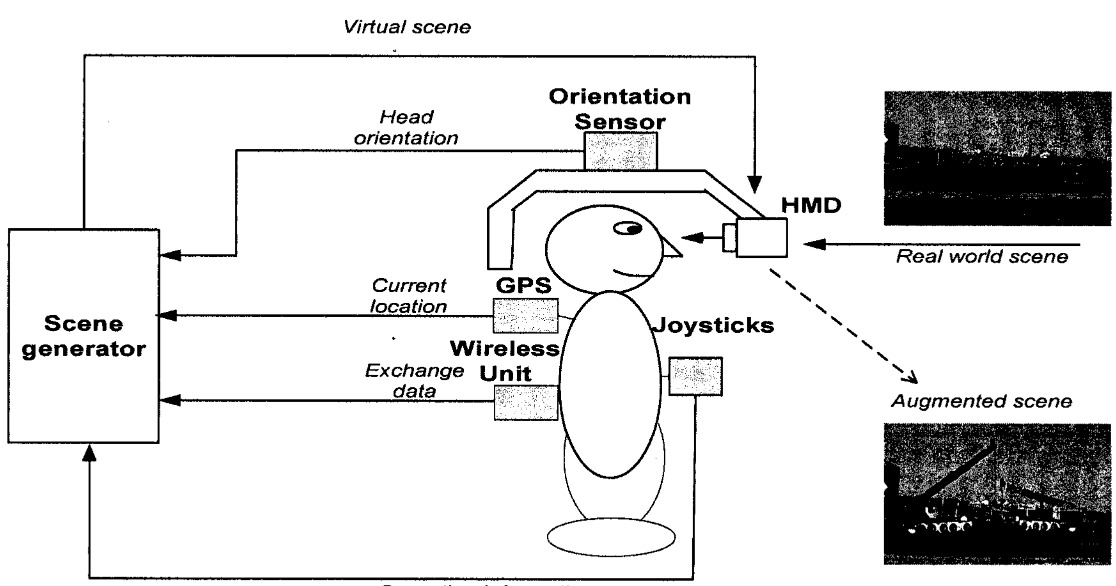


Figure 3.12: Conceptual configuration of the AR system

3.8 SUMMARY AND CONCLUSIONS

In this chapter we have developed the important design issues for outdoor mobile AR systems. We proposed a framework which integrates advanced tracking and visualization techniques using high-end equipment. The generic structure of this framework also includes the engineering constraints and general object modeling methods for construction equipment so that the system design can be extended and customized for more specific applications. The following advantages of the framework can be stated:

(1) A general registration method is proposed for creating accurate and realistic AR environment. It can be updated with other state-of-art tracking equipment or extended with some advanced calibration methods because of the general design;

(2) Two tracking methods which provide 6 degrees of freedom are introduced; and

(3) A user-friendly interaction method which integrates HMD and joysticks for good user experiences in outdoor environment is proposed. By using this specific equipment, user operation and display can be performed efficiently.

The framework is particularly suitable for delivering interactive AR for engineering applications in outdoor environment.

CHAPTER 4
IMPLEMENTATION AND CASE STUDY

4.1 INTRODUCTION

To demonstrate the feasibility and usefulness of the proposed framework, a prototype system is developed following the framework architecture developed in Chapter 3. The prototype system is built using Java programming language and integrates high-end tracking equipment including a GPS receiver and 3D orientation sensor, an HMD, two joysticks, a GIS, a GUI for setting the parameters of the system, and a database. This chapter explains the implementation details of this system and a case study application.

4.2 SELECTION OF DEVELOPMENT TOOLS

Java is a general purpose programming language with a number of advanced features that are suitable for many purposes such as enterprise business computation. Java is chosen as the development tool for the following reasons: object orientation, safety, simplicity, and breadth of the standard library (Horstmann, 2004). Object orientation enables programmers to spend more time on the design of their programs and less time on coding and debugging. Furthermore, graphics, user interface development, database access, multithreads, and network programming are all parts of the standard library. The Java 3D Application Programming Interface (Java 3D API) allows the programmer to describe the 3D scene using coarser-grained graphical objects and to define objects for elements such as appearances, transforms, materials, lights, etc. Compared with Open GL, the code of Java 3D is more readable, maintainable, reusable, and easier to write (Selman, 2002). In addition, MapObject Java edition can be used to build custom applications that incorporate GIS and mapping capabilities (MapObjects-Java, 2006). MapObjects-Java Edition helps programmers build applications that perform a variety of geography-based display, query, and data retrieval. Moreover, Java Native Interface (JNI) is available to communicate with hardware such as tracking equipment. It allows Java code running in the Java virtual machine (JVM) to call and be called by native applications. Most high-end equipment manufacturers only provide API for C++. Therefore, JNI helps Java programmers to connect with native C++ interface with low cost of performance and high productivity.

The database of the 3D objects in the AR system is designed with Microsoft Access to represent the information of engineering constraints for equipment such as cranes. Java Database Connectivity (JDBC) is used to access information stored in the database. Details of software requirements and installation guide of the prototype system are further provided in Appendix C.

4.3 PROTOTYPE SYSTEM CONFIGURATION

A prototype system has been developed to test the proposed design. Table 4.1 shows the hardware devices used in the system including an nVisor ST HMD display, Saitek wireless Joysticks, a Panasonic ToughBook tablet PC, Trimble 5700 RTK GPS receiver, a InertiaCube3 (IC3) sensor and a video camera. These equipments provide high accuracy of tracking, user-friendly interaction and mobile computing in outdoor environment. In addition, an XPower Powerpack 300 is adopted as the mobile power supply in outdoor environment. It can output steady 120V AV and 12V DC power.

The system has five main components: tracking, interaction, visualization, communication, and constraints checking. Figure 4.1 shows the component model of the prototype system using the UML 2 notation.

Table 4.1 Equipment specifications

Device	Type	Specifications
HMD	nVisor ST	Resolution: 1280×1024 pixels
		Vertical frequency: 60 Hz
Joysticks	Saitek Cybord evo wireless Joystick	Range: 30 feet
		Frequency: 2.4 GHz
Tablet PC	Panasonic ToughBook CF-18	CPU: 1 GHz
		RAM: 512 MB
GPS Receiver	Trimble 5700 RTK GPS	Frequency: 10 Hz
		Horizontal accuracy: ±(10 mm + 1 ppm) (×baseline length) RMS
		Vertical accuracy: ±(20 mm + 1 ppm) RMS
3D Orientation Sensor	InertiaCube3	Degrees of Freedom: 3 (Yaw, Pitch and Roll)
		Angular Range: Full 360° (All Axes)
		Maximum Angular Rate: 1200° per second
		Minimum Angular Rate: 0° per second
		RMS Accuracy: 1° in yaw, 0.25° in pitch & roll at 25°C
		RMS Angular Resolution: 0.03°
Video Camera	Logitech-3000	Resolution: 640 x 480 pixels @ 15 fps

The tracking component output the parameters used to update the viewpoint of the virtual scene. The interaction component, engineering constraint component, and communication component output several actions (i.e., OperationAction, ConstraintAction, and SerializationAction, respectively), which are applied on the virtual objects. These actions are represented by an ObjectState. The virtual objects are assembled in the virtual scene. The HMD overlays the virtual and real scenes and displays the augmented scene. The modular and object-oriented software architecture is designed makes the system extensible and easy to modify to accommodate new hardware devices.

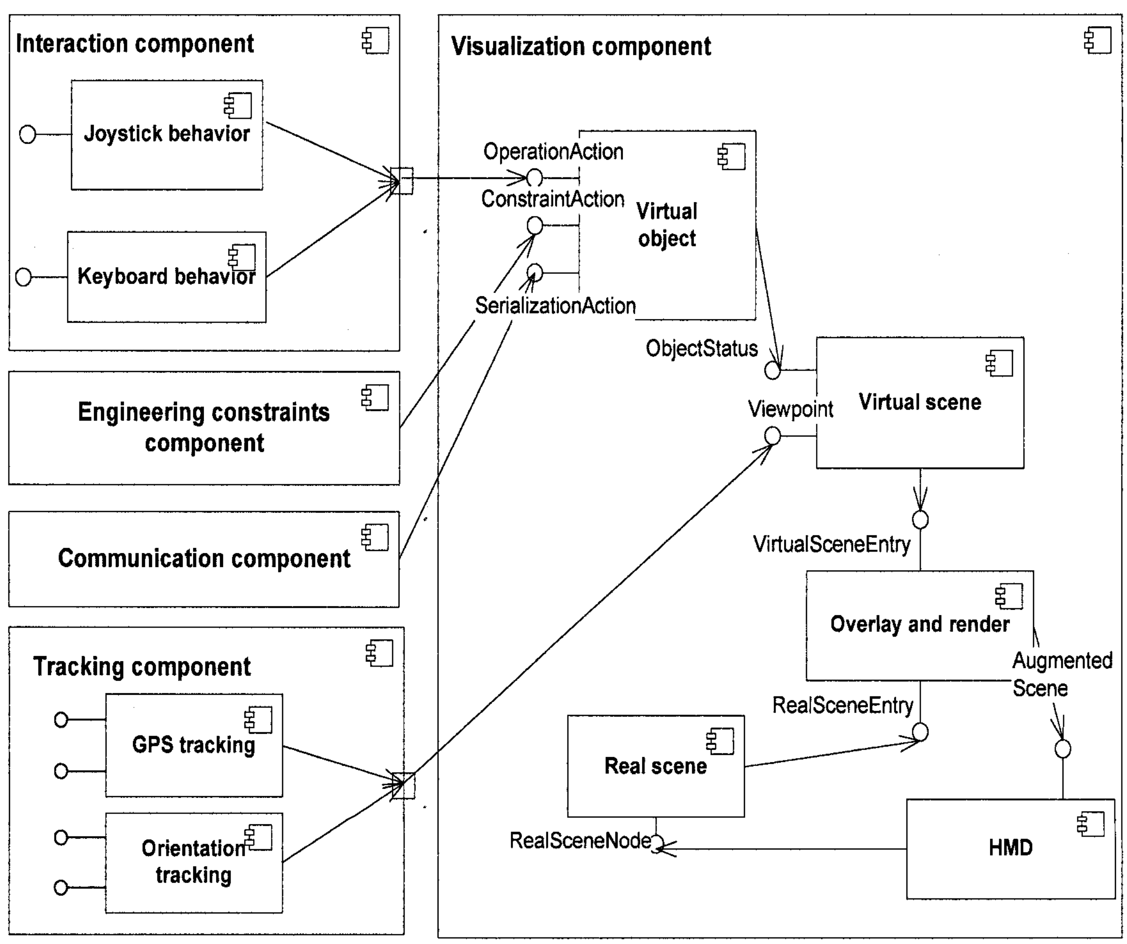


Figure 4.1: Component diagram of system

4.4 TRACKING COMPONENT

4.4.1 Position Tracking

The Trimble 5700 GPS receiver which we adopted in our system only provides serial 232 port communication (Figure 4.2). The accuracy of the GPS unit can be 1 centimeter with differential correction signal. In addition, the update frequency is up to 10 HZ. It also can work with the COM-USB converter when only USB port is available. In this situation, the driver program of the COM-USB converter is required for simulating the COM port. SUN provides Java Communication API which allows 'platform independent' access to communications resources. The API is available for the Solaris and Win 32 platforms and supports both serial and parallel ports communications (SUN, 1999). It includes some powerful features such as event-driven, port monitoring, etc.

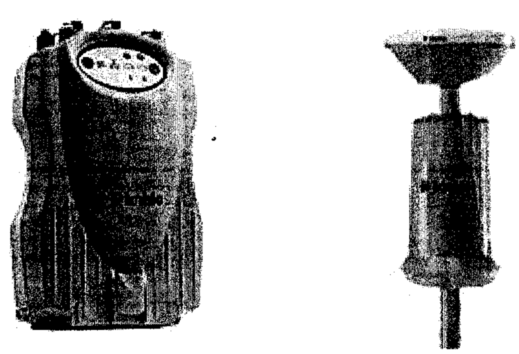


Figure 4.2: Trimble 5700 GPS receiver and an antenna (Trimble, 2005)

When the GPS receiver processes the data sources, it offers many communication protocols. The common data format NMEA is applied for the communication of GPS in our system. The US-based National Marine Electronics Association (NMEA) developed the NMEA format which is a combined electrical and data specification for communication between marine electronic devices (NMEA, 2007). The NMEA uses a simple ASCII, serial communications protocol which defines how data is transmitted in a specific sequence. It also defines the contents of each message type so that all receivers can parse messages correctly. We chose GGA as our protocol of NMEA because it is commonly used for GPS navigation by most GPS companies. The GGA sentence shows an example that provides essential fix data as below:

$GPGGA,101519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,+,*47

1 2 3 4 5 6 7 8

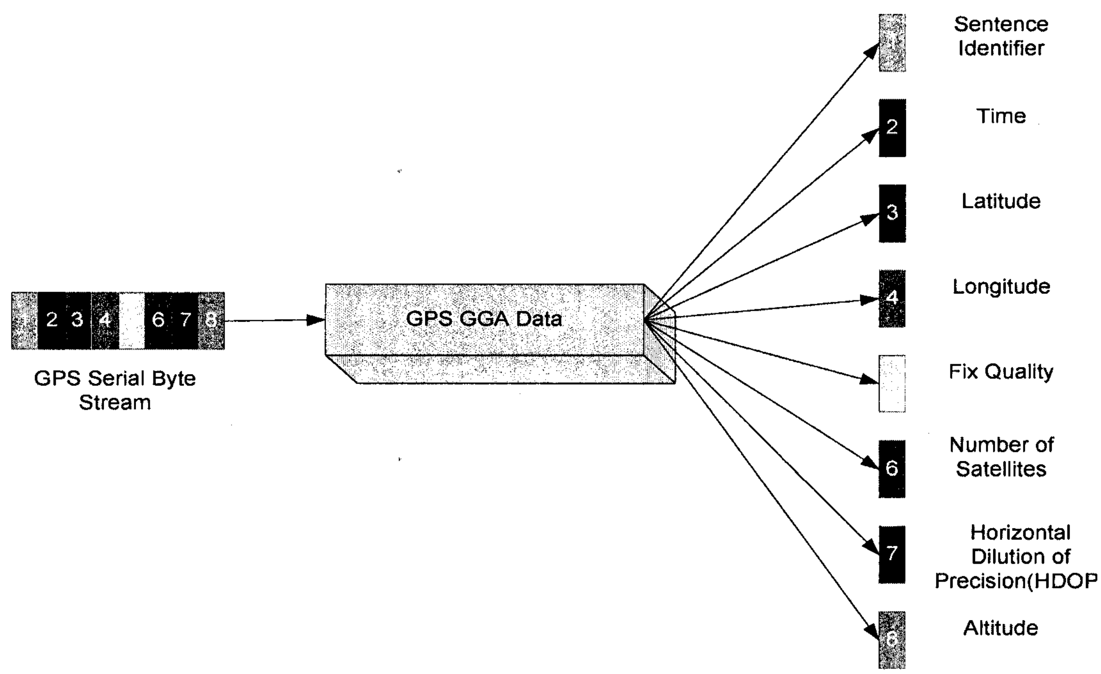


Figure 4.3: GPS CGA message manipulator (adapted from Carroll and Lavin, 2006)

In this example, "$" identifies the beginning of the sentence. The "GP" means that the data is from a GPS device. "GGA" means the GGA protocol. 101519 means the time. 4807.038, N means the latitude is 48 degree 07.038' N. 01131.000, E means the longitude is 11 degree 31.000' E. 1 means the fix quality. 08 means here are 8 tracked satellites. 0.9 means the horizontal dilution of precision. 545.4,M means the altitude is

545.4 meters. We developed a GPS data extraction algorithm to extract GPS tracking position information as shown in Figure 4.3. The serial port listener is defined for monitoring the serial port. Whenever an event on the serial port occurs, the listener will be invoked. The most common example is the arrival of data to the port.

The processing of GPS communication can be explained in the following: (1) wait for data event; (2) read the byte from the serial port; (3) extract and validate messages; (4) extract data from message; and (5) retrieve GPS location. Figure 4.4 shows the flowchart of the process of GPS data communication and extraction.

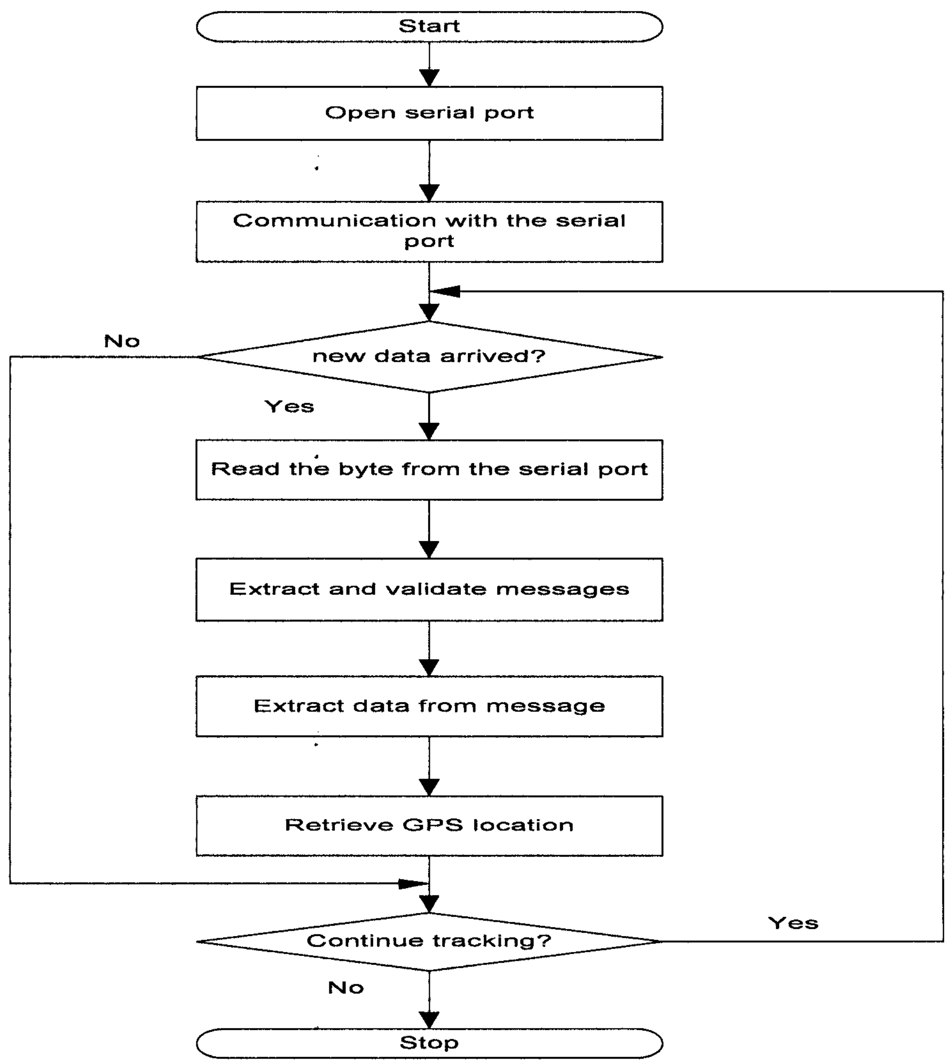


Figure 4.4: Flowchart of GPS data communication and extraction

4.4.2 Orientation Tracking

IC3 is used for orientation tracking in our implementation. The Wireless IC3 sensor is a 3-DOF (Degrees of Freedom) orientation tracking system with wireless communication. It includes an inertial measurement unit, which senses angular rates of gravity, rotation and earth magnetic field along three perpendicular axes. The angular rates are used to compute the orientation of the sensor with low latency and power consumption in real time.

Figure 4.5 shows all components that are integrated into a standard Wireless IC3. It includes two main parts: a wireless IC3 sensor cube and a wireless receiver.

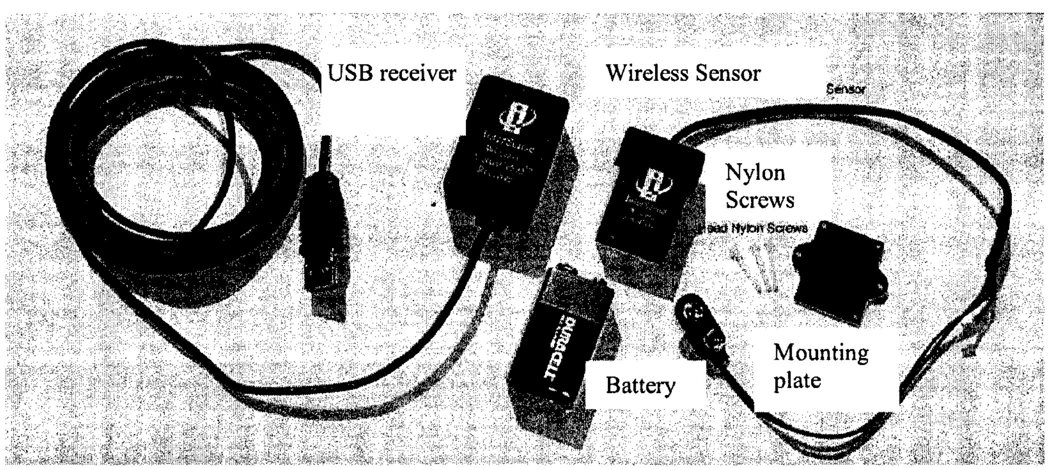


Figure 4.5: Wireless InertiaCube3 components

The sensor includes an inertial measurement unit for computing current orientation. The size of the sensor is 26.2 mm x 39.2 mm x 14.8 mm. The Angular Range is full 360 degree for all axes. The maximum and minimum angular rates are 1200° and 0° per second, respectively. The sensor requires 5.2V to 9V DC at a drain of 60 mA. Voltages below 5.2V DC will reduce the accuracy of the sensor.

The wireless component for the IC3 uses a 2.4 GHz radio module that allows up to 16 different channel selections. The radio module has a very low latency, low power consumption, high bandwidth and wide range. There are two LEDs on the wireless IC3 and the receiver. The green LED on both devices is a power indicator. The amber LED on both devices indicates wireless traffic. It should turn on when the user activates the connection of computer and the receiver. For optimal performance, it is not recommended to have the wireless IC3 or receiver in metal containers or near large metal surfaces.

Because Bluetooth and IEEE 802.11 devices use the same frequency of IC3, some conflict issues would appear if there is heavy wireless traffic in the same area of IC3 (InterSense, 2007). To avoid these conflictions, we should set the wireless IC3 to a different channel from other wireless devices in the same area. Another important issue is power consumption. Although the wireless IC3 is designed to keep power consumption as low as possible, the wireless communication still uses power in every second when operating and the battery provides limited power because of its capacity. In our test, a fully-charged battery can support 4 hours of operation. In most situations, the working time and power consumption are acceptable. However, sometimes we need longer battery life in a mobile environment for special applications. Users should bring more batteries for replacement and check the remaining battery levels during each hour of operation.

In summary, the wireless IC3 supports accurate orientation tracking with wireless communication in real time. It creates a great opportunity to enhance most AR systems in terms of their technical capability, in interaction and usability, and finally, in applications.

IC3 currently only provides API for C++ programming. Hence we have to consider the method to communicate IC3 with Java. As already mentioned earlier, JNI is an available solution that permits Java programs to incorporate native code written in other programming languages, such as C++. The JNI is a powerful feature that allows developer to take advantages of the Java platform, and still employ code written in other languages. As a part of the Java virtual machine implementation, the JNI is an interface that helps Java applications to invoke native code. Figure 4.6 states the role of the JNI.

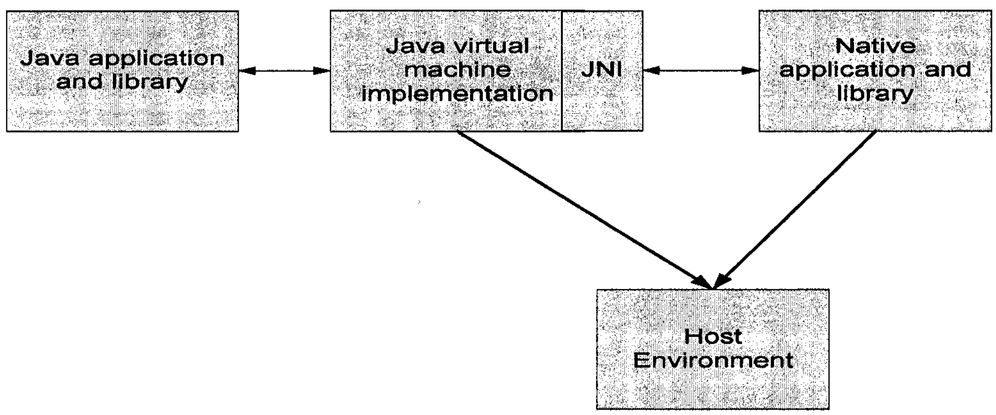


Figure 4.6: The role of JNI (Sun, 1999)

However, applications which use JNI have some disadvantage such as loss of type-safety guarantee when compared with applications written strictly in the Java. Therefore, Sun (Sun, 2003) recommended that JNI should be carefully considered as necessary in the following situations:

(1) The Java API might not support certain required host-dependent features such as special file operations required by an application.

(2) Existing native library has to be accessed for efficiency because the software or hardware provider does not support Java. For example, most high-end equipment only provides C++ API.

(3) Some small portion of time-critical code in a lower level language such as assembly language has to be launched. For example, if a 3D-intensive application spends most of its run time in graphics rendering, it is necessary to write the core portion of a graphics library in assembly code to attain best performance.

In our system, we need to use C++ code in existing native IC3 API. Therefore JNI is chosen as the implementation method. JNI cannot directly invoke an existing C++ library. The solution is building a C++ wrapper firstly. Then JNI will communicate with the wrapper and the wrapper will start the required function from Java side as an agent. Since Java does not have the structure data type which is widely used in C++. SUN suggests to use peer class in Java to connect C++ structure (Sun, 2003). Peer class is a Java class that includes a variety of Java data types corresponding to native data structures. Figure 4.7 shows how a peer class for a data type of IC3 accesses the memory of C++ native IC3 data structure. The peer class gets the data content from the C++ native data structure. Figure 4.8 shows the workflow of our JNI implementation. It is a multi step process as follow: (1) Create Java class that declares native methods in existing C++ library; (2) compile this Java class with JNI; (3) generate C++ wrapper header file for the Java class file; (4) write the wrapper implementation of native method (e.g. in C or C++) following the requirement of JNI; (5) Compile header and implementation files into a shared library file (e.g. DLL on windows systems); and (6) Run the Java program.

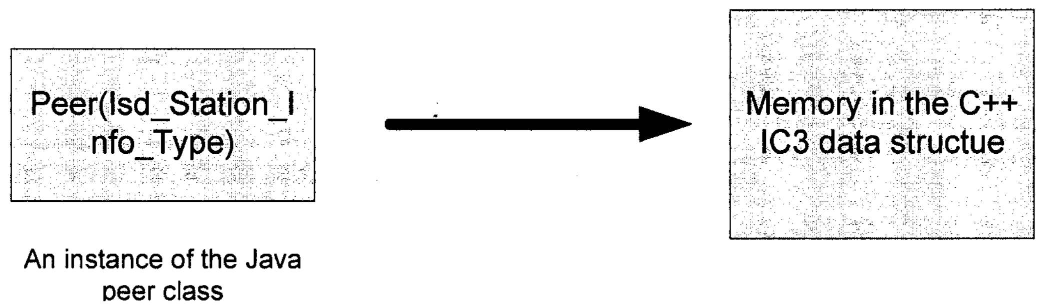


Figure 4.7: Peer class work flow

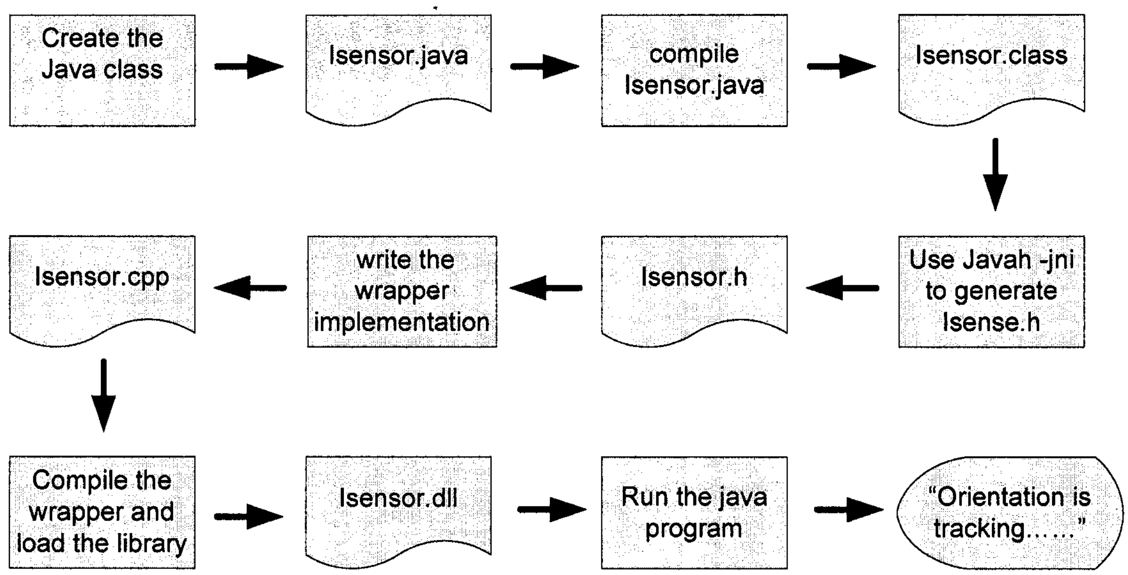


Figure 4.8: The workflow of JNI implementation

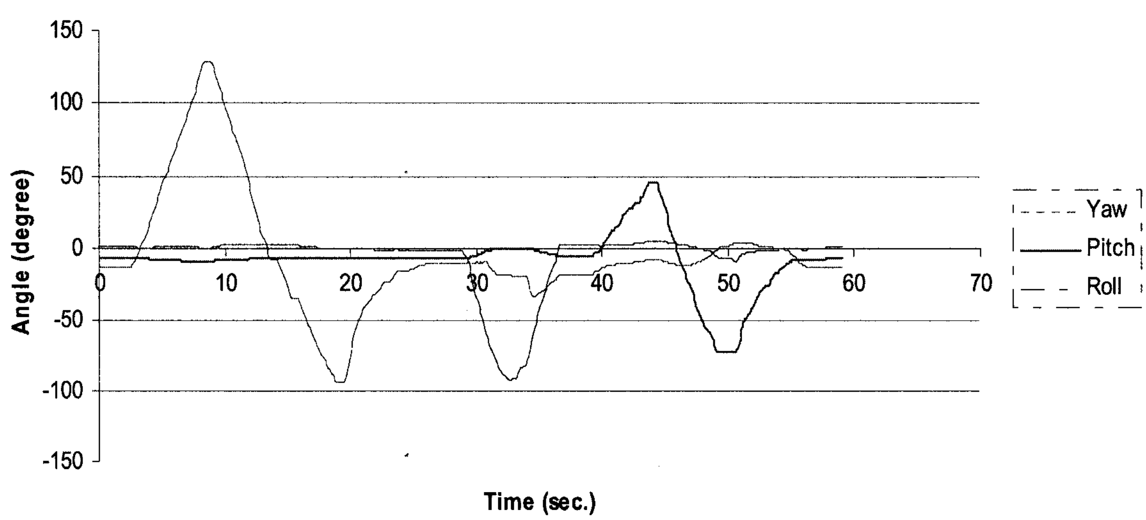


Figure 4.9: Dynamic test of the 3D Sensor

A dynamic test was implemented for verifying the JNI communication and orientation tracking (Figure 4.9). In the test, the IC3 senor was positioned in space, moved in a random motion and finally returned to approximately the same starting position and pose with the 0.16, 0.4, 0.2 degree errors in the yaw, roll, and pitch, respectively. The test evaluated the resolution and response time of the 3D sensor.

4.5 VISUALIZATION OF OBJECTS

Java 3D builds virtual universes from scene graphs. Scene graphs are assemblies of objects to define geometry location, orientation, and appearance of objects. A Java 3D scene graph is a tree-structure which represents a parent-child relationship with nodes (Figure 4.10). For example, TransformGroup objects can be constructed by applying

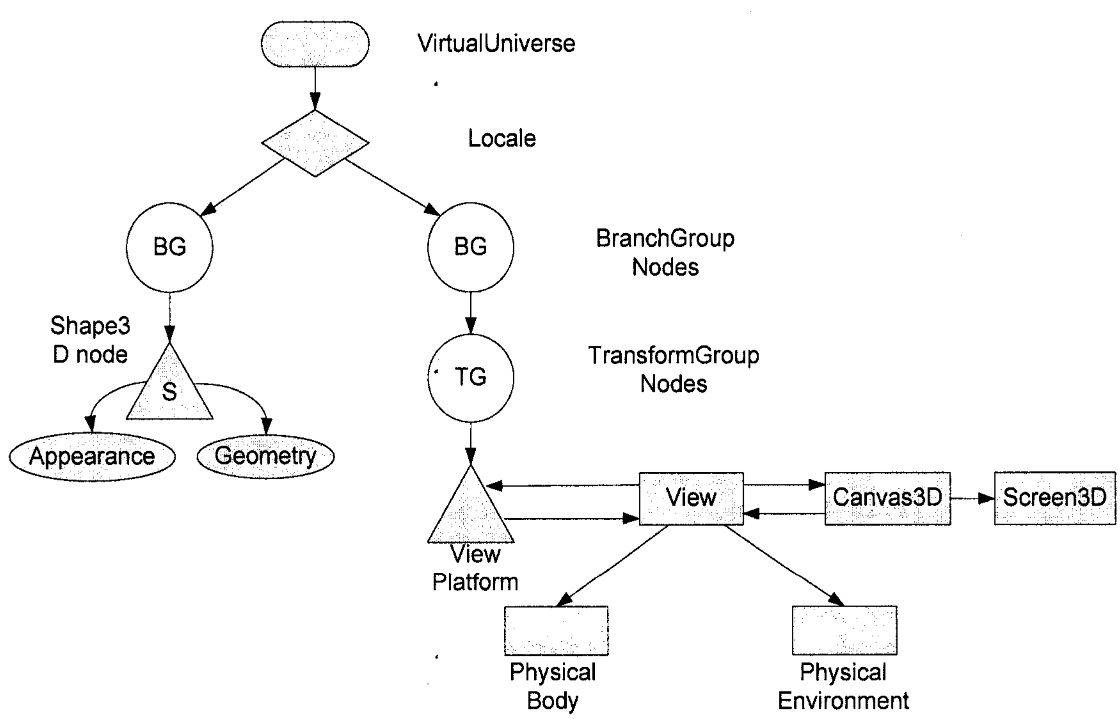


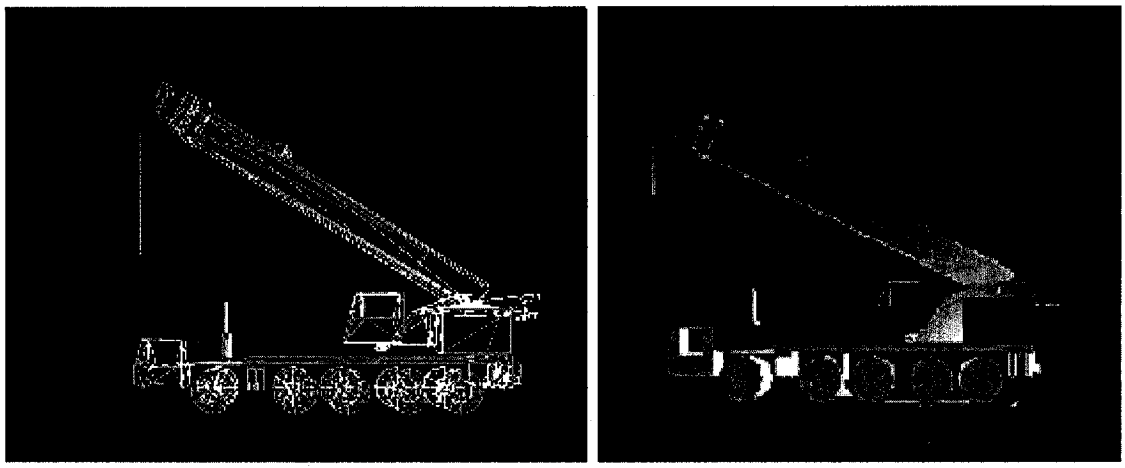
Figure 4.10: Scene graph (Selman, 2002)

Transform3D objects, which correspond to transformations of 3D geometry such as translations, scaling and rotations (Walesh and Gehringer, 2001).

A 3D CAD model of a mobile crane is prepared and translated to CAD files. Every main part of the crane is saved in a separate CAD file following the hierarchical model (Figure 3.7). Microcrowd 3DS Loader API is used to load these 3D models from CAD files (Microcrowd, 2007). In order to assemble the parts in the virtual 3D scene, their coordinates are used to calculate the transformation matrix, including rotation, scaling and translation, from the local coordinate system of the CAD model to the global coordinate system of the virtual model. Another issue to be considered is the orientation of the coordinate axes used in different visualization software. For example, the axis in the height direction is considered as the Y-axis in Java 3D and as the Z-axis in 3D Studio Max.

To represent the degrees for freedom of crane movement and foreword kinematics, the movements of crane bodies can be transferred to the position and orientation of each part of the crane with respect to the global coordinate system in the virtual scene. A Java class is developed to control the transformation of every part of crane. This class provides many functions such as boom-raised, boom-lowered, move-forward, move-backward, etc. Figure 4.11 shows the CAD model and rendered virtual crane.

Information about the available component of the selected crane is instantly retrieved from the database and listed. Once a component is selected, the related information is automatically retrieved from the database and added to the 3D model of the crane in the AR. Figure 4.12 shows an example of dynamically changing the hook of a crane.



(a) Wireframe (b) Solid

Figure 4.11: 3D model of crane

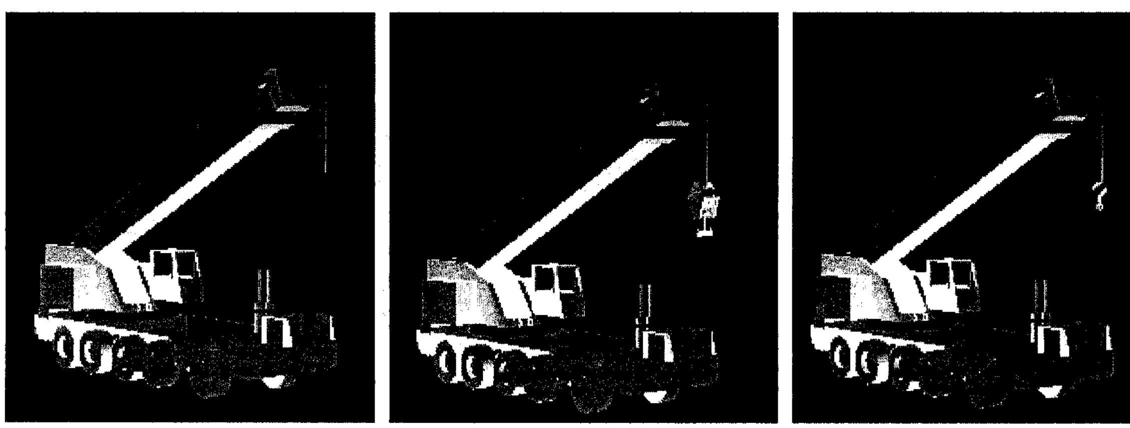


Figure 4.12: Dynamically changing the hook of a crane

The approach of optical see-though HMD directly shows the physical world without any processing combined with the rendering of virtual parts of the AR, while the video see-through HMD combines the virtual parts and the video channel. Our implementation provides an option to switch from optical see-though to video see though (The two approaches are discussed in Chapter 2).

Many of the pictures presented in this thesis of the case study were captured using video based AR. Implementation of video see though is more complex than the optical see though. The computer has to capture the video stream from a head mounted camera, overlay the 3D virtual objects and the 2D image from video stream, and render the composition as the final output to the display.

The video capture function is implemented by JARToolKit (JARToolKit, 2004). It is a marker-based AR development API. The marker-based AR computation, the main function of the API, is not suitable for outdoor environment. However, the video capturing function still is useful for the development of video see-though AR in outdoor environment. We used JARToolKit to connect with current available video camera and capture the video stream from the camera. The rendering part is implemented by Java 3D. It draws a rectangle, and maps the video frame as the texture map of the rectangle. This rectangle is called "background" in our system. The background is placed behind all objects in the Java 3D scene. Thus all virtual objects such as virtual cranes are placed between the background and viewpoint for overlay. Meanwhile the background has to be scaled to exactly fit the entire field of view. In addition, the background follows the movement of the viewpoint. Thus it always provides the user's view for augmentation. Furthermore, the virtual objects have to be scaled with respect to the size of the background for realistic experience.

The final effect is that the 3D virtual objects are overlaid onto the live video stream. It provides the user with the feeling that the real world is being augmented with virtual objects, as shown in Figure 4.13.

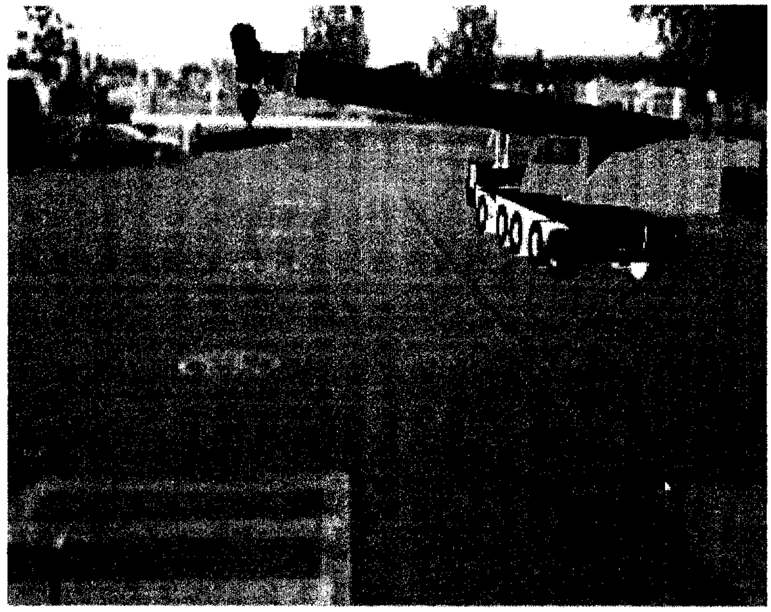


Figure 4.13: Screenshot of a test in Loyola campus

4.6 USER INTERACTION COMPONENT

The nVisor ST HMD is adopted as our display component. It provides 1024×768 pixels resolution and 60 HZ update frequency. It is suitable for mobile environment because of its optical see-though design. In addition, it has a special mounting for the orientation sensor. Its user friendly features reduce the difficulty of physical setting.

We developed an HMD setting panel to allow the user to adjust the parameters for specifying the unique head geometry of the user. We defined the physical body model in Java3D as shown in Figure 4.10. It includes the head model with adjustable parameters. For example, the positions of left and right eyes can be modified in runtime for improving user experience. Figure 4.14 shows a 3D head geometry model and the HMD setting panel. In addition, the stereo view of HMD requires the support of hardware. With the support of two video channels from the video card, the adjustment of eye position will change the two views of left and right eyes, respectively. In the future, the adjustment of left and right ear position can be added for stereo audio subsystem.

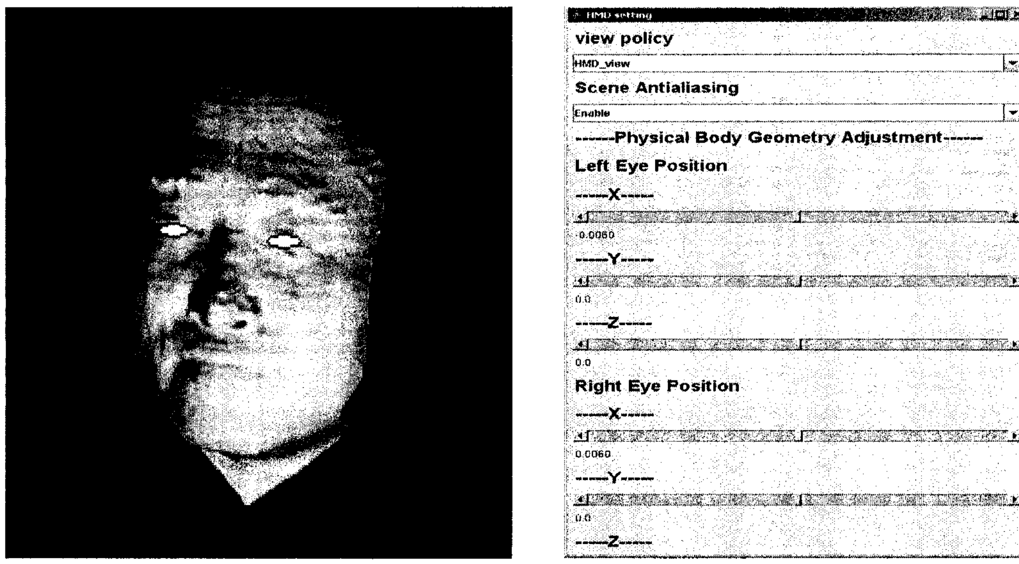


Figure 4.14: HMD parameter setting

The keyboard is a typical input method for user interaction. The user interaction provides the support for keyboard input for convenience. A Java3D keyboard behavior is applied for this purpose. The behavior can be activated when receiving a key-press event. Thus it invokes different functions which are mapped with predefined key-press event. The functions which are called by the behavior include crane operation, manual position and orientation adjustment, etc. It provides a flexible and powerful user input solution for the requirement of simple operation in outdoor environment. More details are described in Appendix G.

Java and Java3D bring many exciting user interaction features such as 3D user interface, and object-oriented scene (Barrilleaux, 2001). However, Java does not provide a direct solution to fulfill the complexity and the possibilities of state-of-the-art input devices we find today. The joystick communication part in our system is implemented by JXInput which uses DirectX to connect common game device such as joysticks, game pads, etc (JXInput, 2007). It supports a set of features of input devices such as: (1) Axis (translational and rotational), (2) Directional, (3) Button, and (4) Sliders. In addition, it allows developers to register a listener for event-driven applications. This feature helps applications to check the user input when the input device changes its status. For the realistic simulation of crane operation, the user will use two joysticks to operate the boom of the virtual crane. Hence the multi-threads programming method is used for operating the two joysticks. Figure 4.15 shows the architecture of joystick operation component. The *CraneOperationAxis*, *CraneOperationDirectional* and *CraneoperationButton* classes are derived from the interfaces of Axis, Directional and Button of JXInput. These classes assign and implement the communication of joysticks (Figure 3.11). Figure 4.16 shows the user interaction with a HMD and two joysticks. A user wears the HMD on the head and carries the GPS receiver on his back. A 3D orientation sensor is mounted with the HMD. Two wireless joysticks are used for controlling virtual crane.

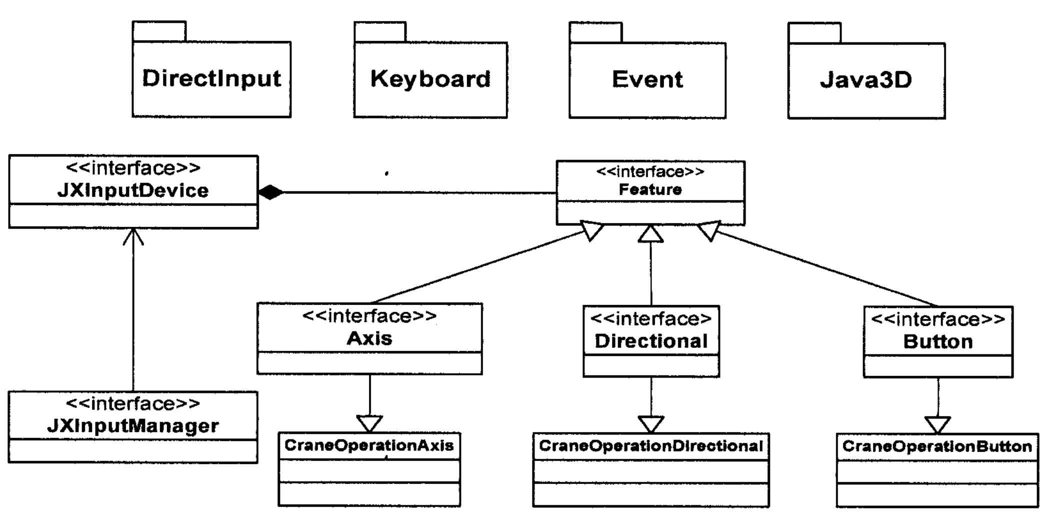


Figure 4.15: The architecture of Joystick operation component

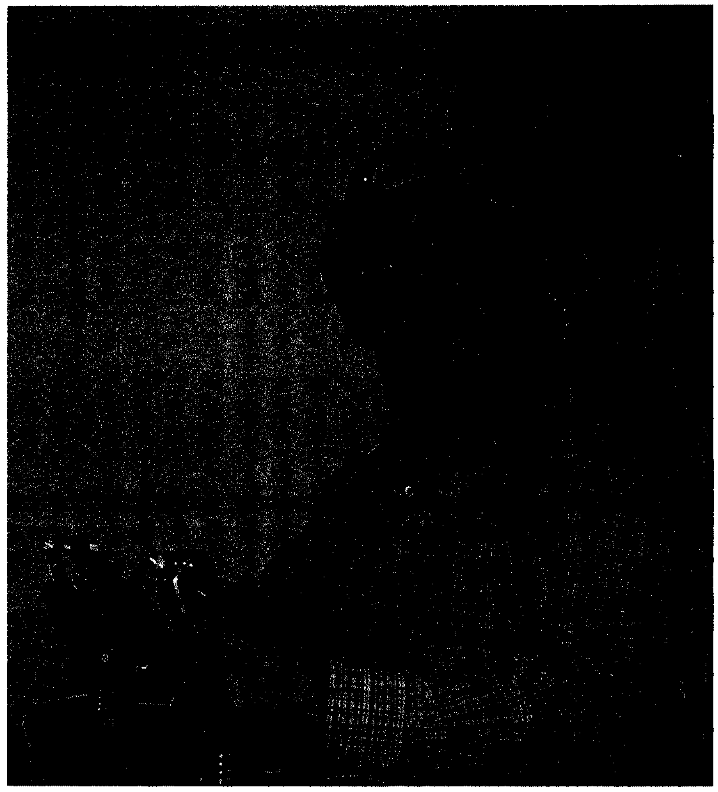


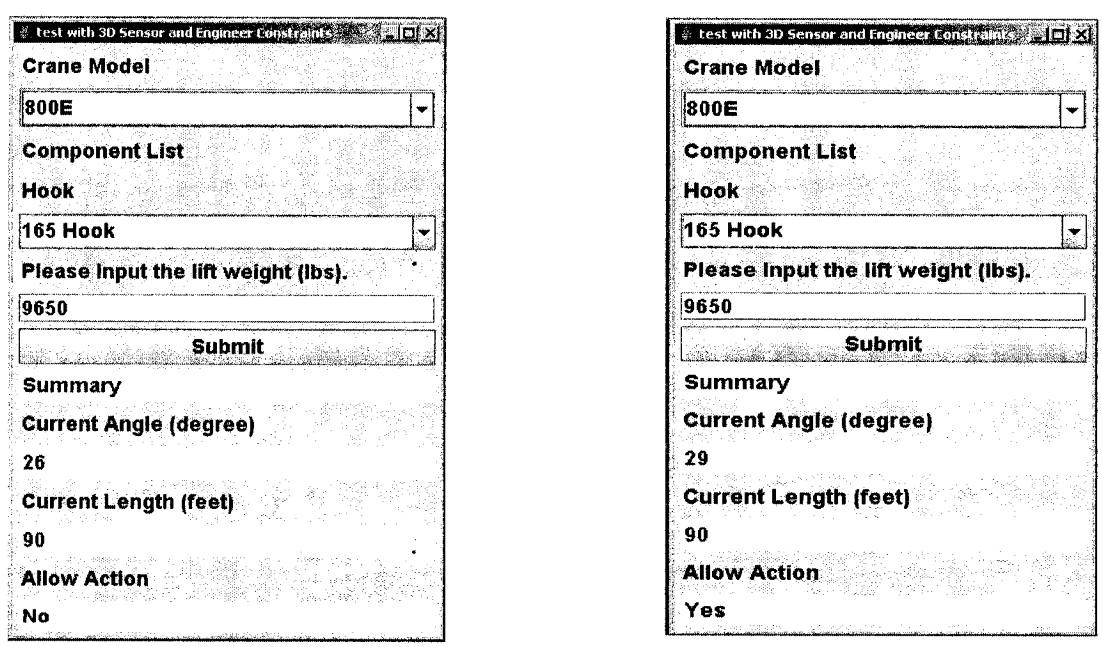
Figure 4.16: User interactions with HMD and joysticks

4.7 ENGINEEING CONSTRAINT COMPONENT

(1) Data store and query: The engineering constraints such as the limitations of crane boom extension are stored in a relational database using Microsoft Access. JDBC is used to access the databases. The data can be queried and updated using Structured Query Language (SQL).

(2) Engineering constraints checking behavior: The basic process of engineering constraints checking is detecting whether the user correctly operates equipment and making sure he/she will not perform dangerous operations, such as extending the boom beyond the safety limitation. The engineering constraint behavior is event driven. Firstly, a listener is reregistered with the virtual crane. When the state of the crane changes, the

listener notifies the behavior. Then, the behavior checks the current status based on the engineering constraints which are stored in the database. The user's dangerous operation will be prohibited with a warning when the behavior detects wrong crane status. For example, the boom of Groove 800E model crane cannot be more than 20 degree if the current load of the crane is more than 38800 lbs and the length of the boom is more than 90 ft. If the user lowers the boom below 20 degrees, the system will stop the action and display the warning. Meanwhile, the system will also display related engineering constraints when the user inputs the assumed load of a virtual crane. This function is important for learning the correct operation of equipment. The algorithms and computational aspects of the engineering constraint behavior are given in Appendix D and shown in Figure 4.17.



(a) Boom configuration violates the constraints

(b) The user adjusted the crane to the safe work range

Figure 4.17: Engineering constraint panel

4.8 GIS INTEGRATION

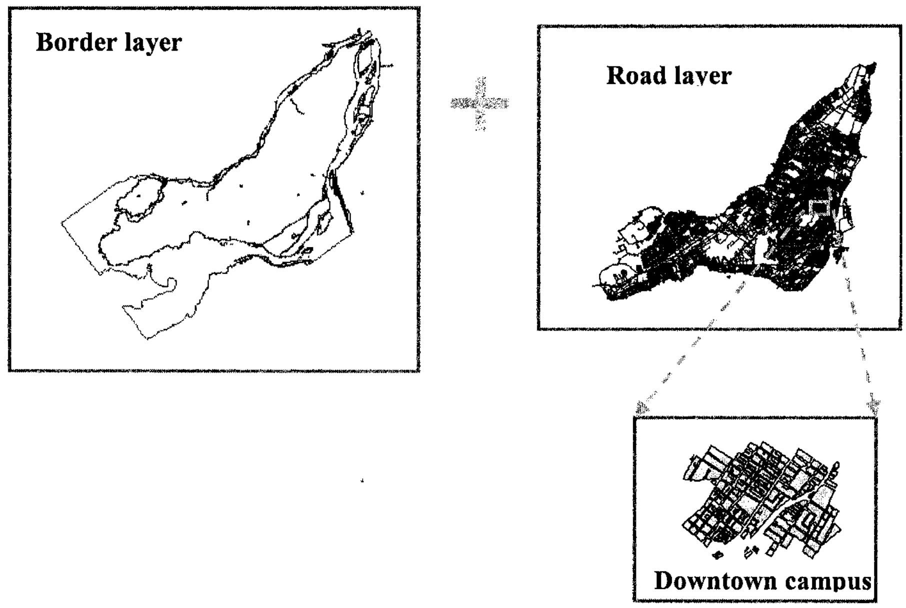


Figure 4.18: GIS Information

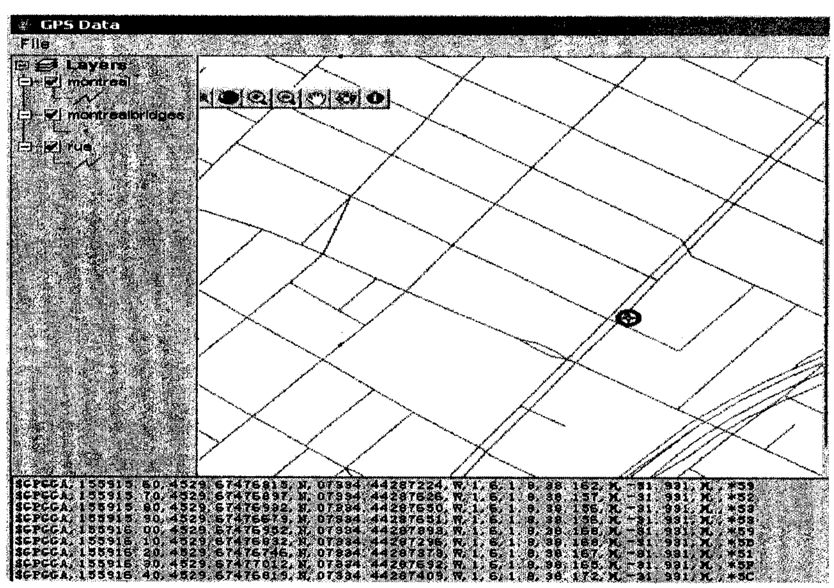


Figure 4.19: GIS Display Panel

A GIS sub-system is created with MapObjects Java Edition (MapObjects-Java, 2006). It displays a current geography map for tracking and navigation. The map includes several layers related to Montreal, such as roads, bridges, blocks and buildings in Concordia downtown campus (Figure 4.18). In addition, the Digital Elevation Model (DEM) model of the area was created for the function of terrain following in the future. The DEM data source is the Canadian Digital Elevation Data (CDED) (Geobase, 2007). The geographic coordinates are transferred to the world coordinates for matching the 2D projected map of Montreal.

The processing of the GIS subsystem is as below: (1) GPS tracking component sends the position information to GIS subsystem in real time; (2) GIS translates the GPS tracking message tracking WGS84 standard to the world coordinates; (3) GIS subsystem displays the map and layers; and (4) the user can interact with the map, e.g., zoom in/out, save current tracking position, etc. Figure 4.18 and 4.19 show the GIS subsystem with GPS tracking in Montreal downtown area.

4.9 MOBILE COMMUNICATION FOR COLLADORATION

A mobile power supply battery, XPower Powerpack 300, is used for supplying energy of the AR system. It provides 110 V·AC and 12V DC outputs for outdoor usages. The run time of the XPower Powerpack is more than 30 minutes when the power consumption of appliances such as computers and monitors is under 150 watt. Since the output of the XPower Powerpack is a non-sinusoidal modified sine wave, some equipment may be damaged by the battery. Hence, the safety of integration of high-end equipment with the battery should be considered before testing. The compatibility of HMD with the XPower Powerpack was confirmed from the manufacturer of the HMD.

As discussed in Chapter 3, the client-server architecture is adopted for the AR system. Java object serialization method is used to transfer the status of virtual objects. Every client will create persistent clones of other virtual objects which are operated by other clients.

Another important issue of collaboration is how to know the status of other user's virtual objects. To solve the issue, observer pattern is adopted for distributed event handling of the AR system. It is a design pattern which is used to observe the state of an object in a system (Erich et al., 1995). It defines a one-to-many relationship between a set of objects. When the state of one object changes, all of its dependents are notified. Figure 4.20 illustrates this structure with a UML diagram. It includes two interfaces: Subject and Observer. Objects use the subject interface to register as observers. All registered objects need to implement the Observer interface. Thus the objects will be called when the Subject's state changes.

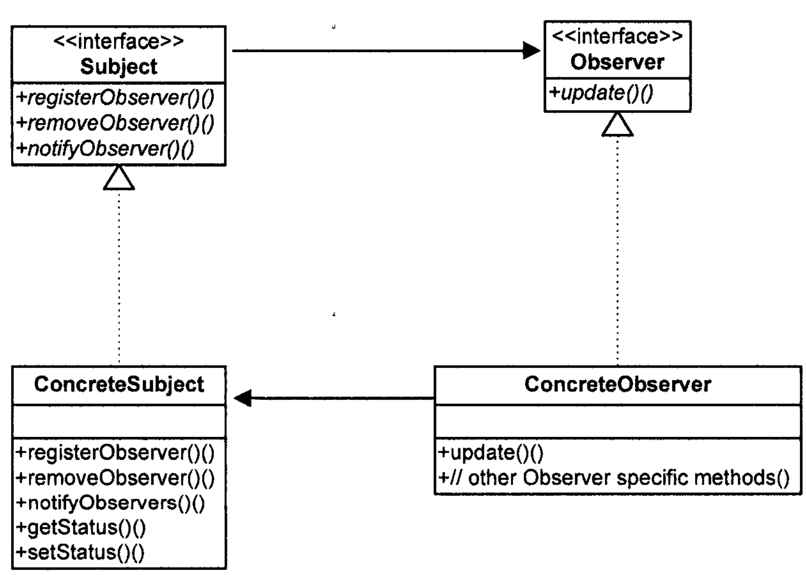


Figure 4.20: The UML diagram of observer pattern (adapted from Erich et al. 1995)

For example, the state changing of the virtual crane which is operated by client A is a Subject. All other clients will be registered by the Subject. When the user of client A operates its virtual crane, client A will take the state of the virtual crane and convert it to a stream which is sent to the server. The server will forward the stream to all clients which registered the observation event of client A. The registered clients (*ConcreteObserver*) will extract the state of the virtual crane from the stream and apply it for changing the state of the clone virtual crane. The processing of registered clients can be concluded as below: (1) register for notify; (2) wait for notify event; (3) receive the object state from the stream; (4) extract the object state; and (5) apply the remote object state on the local clone object.

4.10 CASE STUDY

For evaluating the prototype system, it was used to simulate deck rehabilitation project of a bridge. Deck rehabilitation is a complex and expensive construction activity. It usually needs two cranes to collaboratively move a prefabricated panel of the bridge in limited time. It is a suitable example to test distributed augmented reality system because of its engineering value and collaborative feature. Figure 4.21 shows the deck rehabilitation project of Jacques Cartier Bridge in Montreal, Canada (Zaki and Mailhot, 2003).

In the case study, we used the AR system to simulate the deck rehabilitation project in outdoor environment. Users use HMD, joysticks and tablet PC on the real bridge. GPS and 3D orientation sensor track user's position and orientation, then register the virtual objects, such as virtual cranes and a virtual panel, with the video stream of the HMD.

Figure 4.21: View of the construction in progress (Zaki and Mailhot, 2003)

In the test scenario, two users use the system on the bridge to operate virtual cranes for installing a prefabricated panel. In the tests, we focused on augmenting the scene of the construction site with two virtual cranes. The initial position and orientation of the virtual cranes can be adjusted interactively. The wireless joysticks are used by two operators to operate the virtual cranes.

Figure 4.22 shows two AR scene of an offline test. In the test, an image of a construction site of the bridge is mixed with virtual cranes. 3D orientation sensor tracks user's head direction. The keyboard and two joysticks are used to operate the virtual cranes and adjust the initial position. The result represents the work status of the integrated components. More offline tests can be found in Appendix H.

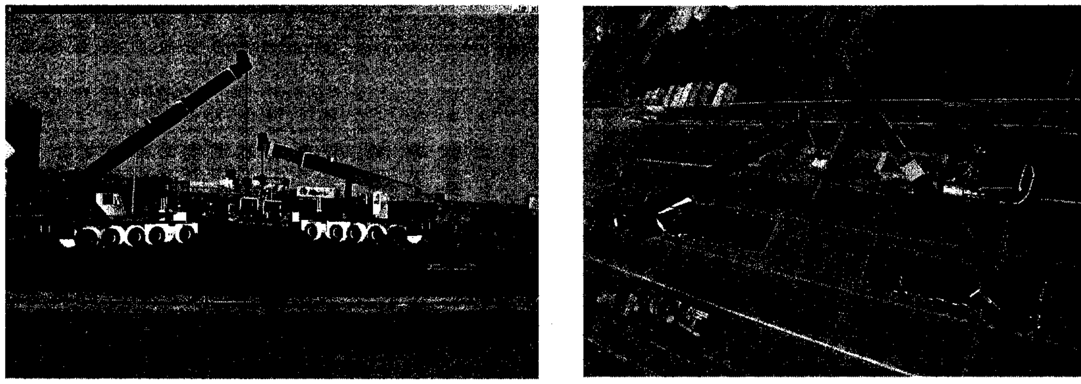


Figure 4.22: Two views of the AR construction simulation

Another outdoor test is implemented in Loyola campus, Concordia University. The test represents the user interaction and orientation tracking ability in outdoor environment. In addition, the GIS and engineering constraints component are evaluated. In the test, a user operates a virtual crane to move a panel. In the processing, the 3D orientation sensor tracks the user's orientation and applies the rotation to the virtual viewpoint. Therefore, the pose of the virtual crane can follow the user's movement for augmentation. Figure 4.23 shows a snapshot of the outdoor test.

Figure 4.23: Screenshot of an outdoor test in Loyola Campus

4.11 SUMMARY

This chapter described the implementation of the proposed framework and methods discussed in Chapters 3. A prototype system, implemented in Java, is developed and a case study about bridge rehabilitation project in Montreal was used to demonstrate the feasibility of our approach and methods. The software development tools and APIs were selected to integrate several information technologies and high-end equipment in the prototype system. The system was demonstrated to engineers responsible for the construction simulation and equipment operation. Furthermore, the preliminary testing of the framework and its user interaction showed that it has good potential for realizing future construction simulation systems because it was carefully designed and implemented to satisfy the specific requirements of these systems.

CHAPTER 5
SUMMARY, CONCLUSIONS, CONTRIBUTIONS, AND FUTURE WORK

5.1 SUMMARY

As introduced in the Chapter 2, AR is an attractive and challenging area. It can be useful for many engineering areas such as construction. However, in order to realize a mobile outdoor AR system, large amounts of time and money have to be invested. Only a few AR research works have explored outdoor engineering applications. Moreover, their research has focused only on the low level integration of 3D models and did not include all the necessary details (e.g., engineering constraints, reasonable user interaction in practice). Therefore, our research focused on developing an outdoor mobile AR system that provides accurate and effective on-site visualization for construction applications.

The literature review of mobile AR systems has described some challenges in this area, which affect the design. For example, it is difficult to find the locations of objects and to simulate a construction scenario in the outdoors without high accurate tracking. These challenges require complex and creative solutions which integrate high-end hardware and advanced software techniques. In this research, we introduced an integration method for tracking 6 degrees of freedom of the user with accurate rotation sensors and a GPS receiver. In addition, we developed interactive 3D models with kinematics actions. We also implemented optical and video see-though image overlay for AR display. Integrating a state-of-art HMD and joysticks provides good user experience. Furthermore, the engineering constraints of equipment operation are implemented for realistic simulation.

Wireless techniques for multi-user communication and collaboration for advanced user interaction in real-time outdoor environment is also discussed.

5.2 CONCLUSIONS AND CONTRIBUTIONS

The contributions of this research are grouped into the following areas:

(1) The design of outdoor AR identified the requirements for selecting accurate tracking devices and interaction devices for AR-based construction simulation. It also integrates novel methods for dynamic object modeling, engineering constraints processing, and multi-user real-time communication in the outdoor environment.

(2) The registration method of outdoor AR is new in that it adopts the coordinate system used in GIS projection as the base for the geo-referenced world coordinate system of the AR registration. This method facilitates the mapping of real world objects and virtual objects based on widely available digital maps.

(3) A new process of integrating outdoor AR with construction activities and equipment databases has been developed. This process can be easily applied in practice because it is based on the available information of construction activities and construction equipment.

(4) A prototype system was developed in Java implementing the design of outdoor AR. This system integrates the state of the art tracking devices for accurate tracking, including an RTK-GPS receiver for position tracking and an accurate wireless 3D orientation sensor using hybrid tracking. A high-end optical see-through HMD and wireless joysticks are integrated in the system for user-friendly interaction. Several software integration problems were solved to integrate the hardware devices while

keeping the system modular and extensible. The favorable initial testing results of the system showed that it has the potential of practical applications.

5.3 FUTURE WORK

While pursuing this research, several limitations have been identified related to the requirements and the performance of the developed methods and techniques.

(1) Improving the modeling of the construction site: Having a 3D model of the construction site would make it possible to consider occlusion and collision problems with existing objects. In addition, a detailed digital terrain model would make it possible to apply terrain following on the virtual equipment.

(2) Improving the accuracy of tracking in outdoor environment: GPS tracking may become unavailable or inaccurate. Therefore, other position tracking methods should be integrated to improve the reliability of the system.

(3) Distributed communication and computation are discussed in our work. Figure 5.1 shows the design of distributed computing. In the design, clients send computation requirement to the action handler. The action handler assigns these requirements to different computation servers based on the feature of computations. These servers provide advanced computation services, such as collision detection, database server, GIS server, etc. However, many issues such as network band width, parallel requests, load balance, fault tolerance, synchronization, etc have to be considered and tested for creating an effective distributed computing system. Further developments are necessary to implement this component.

(4) The integration of dynamic constraints is another important topic for future research (Lee et al., 1994).

(5) Applying more outdoor testing of the system: For example, the stereoscopic view of the HMD needs suitable hardware support and was tested only in the laboratory because the laptop that was used in the outdoor test does not support dual video output.

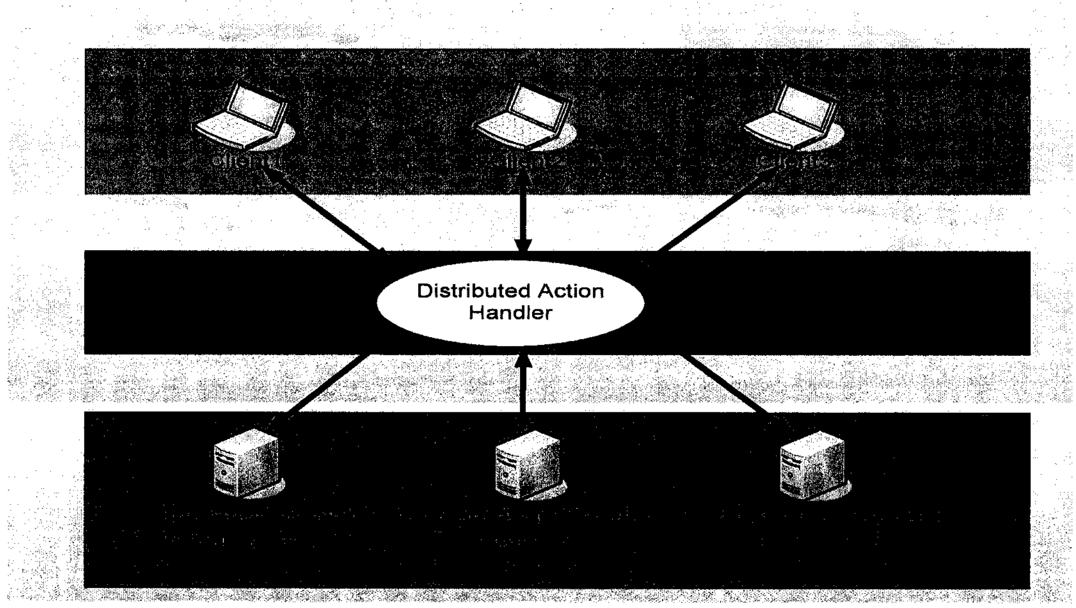


Figure 5.1: Design of distributed system

REFERENCES

5DT (2007) <http://www.5dt.com>.

Ahlers, K. H., Kramer, A., Breen, D. E., Chevalier, P., Crampton, C., Rose, E., Tuceryan, M., Whittaker, R. T., and Greer, D.(1995). Distributed Augmented Reality for Collaborative Design Applications, In Proceedings of Eurographics 1995, Maastricht, The Netherlands, pp. 3-14.

Al-Hussein, M., Alkass, S. and Moselhi, O. (2005). Optimization Algorithm for Selection and on Site Location of Mobile Cranes, Journal of Construction Engineering and Management, 131(5), pp. 579-590.

Al-Hussein, M., Alkass, S. and Moselhi, O. (2001). An algorithm for mobile crane selection and location on construction sites, Construction Innovation Journal, 1(2), pp. 92-105.

Azuma R. (1999). The Challenge of Making Augmented Reality Work Outdoors. In "Mixed Reality: Merging Real and Virtual Worlds" by Yuichi Ohta and Hideyuki Tamura, Springer-Verlag, Chapter 21, pp. 379-390.

Azuma, R. (1997). A Survey of Augmented Reality, Presence: Teleoperators and Virtual Environments, 6(4), pp. 355-385.

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S. and MacIntyre, B. (2001). Recent Advances in Augmented Reality, IEEE Computer Graphics and Applications 21(6), pp. 34-47.

Azuma, R., Hoff, B., Howard Neely III, R. (1999). A Motion-Stabilized Outdoor Augmented Reality System, Proceedings of IEEE VR ' 99, Houston, TX, pp. 252-259.

Azuma, R., Hoff, B., Neely, H. E., Sarfaty, R., Daily, M. J., Bishop, G., Vicci, L., Welch, G., Neumann, U., You, S., Nichols, R., and Cannon, J. (1998). Making augmented reality work outdoors requires hybrid tracking. In Proceedings of the international Workshop on Augmented Reality. Bellevue, Washington, USA, pp. 219-224.

Azuma, R., Neely, H., Daily, M., and Leonard, J. (2006). Performance Analysis of an Outdoor Augmented Reality Tracking System that Relies Upon a Few Mobile Beacons, Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2006), Santa Barbara, CA, pp. 101-104.

Barrilleaux, J. (2001). 3D User Interfaces with Java 3D, 1st Edition. Manning Publications Co.

Behringer, R. (1999). Registration for Outdoor Augmented Reality Applications Using Computer Vision Techniques and Hybrid Sensors, Proc. IEEE Virtual Reality, IEEE CS Press, Los Alamitos, Calif., pp. 244-251.

Behzadan A.H., and Kamat V.R. (2007). Reusable Modular Software Interfaces for Outdoor Augmented Reality Applications in Engineering, ASCE International Workshop on Computing in Civil Engineering, Pittsburgh, PA.

Behzadan, A.H., and Kamat, V.R. (2006). Animation of Construction Activities in Outdoor Augmented Reality, Proceedings of the 11th International Conference on Computing and Decision Making in Civil and Building Engineering (ICCCBE-XI), Montreal, QC, Canada.

Bell, B., Feiner, S. and Höllerer, T. (2001) View Management for Virtual and Augmented Reality, In Proceedings of the 14th annual ACM symposium on User interface software and technology, San Diego, CA, pp. 101-110.

Bimber, O., Gatesy, S.M., Witmer, L.M., Raskar, R. and Encarnação, L.M. (2002). Merging Fossil Specimens with Computer-Generated Information. IEEE Computer, September, pp. 45-50.

Bossler, J. D., Jensen, J. R., McMaster, R. B., Rizos, C. (2001). Manual of Geospatial Science and Technology, 1st Edition, CRC, New York, USA.

Bowman, D. and Billinghurst, M. (2002). Special Issue on 3D Interaction in Virtual and Mixed Realities: Guest Editors' Introduction, Virtual Reality, Vol. 6, pp. 105-106.

Burdea, G. and Coiffet, P. (2003). Virtual Reality Technology, 2nd Edition, John Wiley & Sons, Inc., Hoboken, New Jersey.

Buss, S. R. (2003). 3D Computer Graphics: A mathematical approach with OpenGL, 2nd Edition. Cambridge University Press.

Buss, S. R. (2004). Introduction to inverse kinematics with Jacobian transpose, pseudoinverse, and damped least squares methods. Typeset manuscript, available from http://math.ucsd.edu/~sbuss/ResearchWeb.

Capps, M. (2002). Collaborative virtual entertainment: (if you can't beat 'em, join 'em?), In Proceedings of the 4th international Conference on Collaborative Virtual Environments, Bonn, Germany, pp. 63-63.

Carroll, B. and Lavin, D. (2006). Serial + Java = The World (Mostly), <http://www.embedded.com/192200680?_requestid=35508>.

Chen, S. E. (1995). QuickTime VR: An Image-based Approach to Virtual Environment Navigation, In Proceedings of the 22nd Annual Conference on Computer Graphics and interactive Techniques S. G. Mair and R. Cook, Eds. SIGGRAPH '95. ACM Press, New York, NY, pp. 29-38.

Davis, B.R. and Sutton, S.C. (2003). A Guide to Crane Safety, N.C. Department of Labor, Division of Occupational Safety and Health, <http://www.nclabor.com/osha/etta/indguide/ig20.pdf>.

Denavit, J. and Hartenberg, R.S. (1955). A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices, Trans ASME J. Appl. Mech, 23, pp. 215-221.

Dharwadkar, P.V., Varghese, K., O'Connor, J.T. and Gatton, T.M. (1994). Graphical Visualization for Planning Heavy Lifts, Proceedings of the 3rd Congress on Computing in Civil Engineering, ASCE, New York, pp. 759-766.

Dunston, P., Wang, X., Billinghusrt, M., and Hampson, B. (2002). Mixed Reality Benefits for Design Perception, In Proceedings of 19th International Symposium on Automation and Robotics Construction (ISARC 2002), NIST, Gaithersburg, MD, pp. 191-196.

EKahau (2007). <http://www.ekahau.com/?id=4500>.

Erich, G., Helm, R., Johnson, R., and Vlissides, J. (1995).Design Patterns: Elements of Reusable Object-Oriented Software, 1st Edition. Addision-Wesley Publishing Company Inc.

Feiner, S., MacIntye, B., and Seligmann, D. (1993). Knowledge-based Augmented Reality, Communication of the ACM 36 (7), pp. 52-62.

Foley, James D.,Dam, A.V., Feiner, S.K., Hughes, J.F. (1990). Computer graphics: Principles and Practice, 2nd Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

Foxlin, E. and Durlach, N. (1994). An Inertial Head-Orientation Tracker with Automatic Drift Compensation for Use With HMD's. In Proceedings of the VRST '94, New York, pp. 159-173.

Foxlin, E., Harrington, M., and Pfeifer, G. (1998). Constellation: A Wide-Range Wireless Motion Tracking System for Augmented Reality and Virtual Set Applications, In Int'l Conference on Computer Graphics and Interactive Techniques, Orlando, Fl, pp. 371-378.

Fründ, J., Geiger, C., Grafe, M. and Kleinjohann, B. (2001). The augmented reality personal digital assistant, In proceedings of International Symposium on Mixed Reality, New York, USA.

Fu, K.S., Gonzalez, R.C. and Lee, C.S.G. (1987). Robotics: Control, Sensing, Vision, and Intelligence, McGraw-Hill, New York, USA.

Furmanski, C., Azuma, R. and Daily, M. (2002). Augmented-reality visualizations guided by cognition: Perceptual heuristics for combining visible and obscured information. In Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002).Washington, DC, pp. 215- 320.

Ganu, S., Krishnakumar, A.S. and Krishnan, P. (2004). Infrastructure-based Location Estimation in WLAN Networks, IEEE Wireless Communications and Networking Conference (WCNC 2004), Atlanta, USA.

Geiger, C., Kleinjohann, B., Reimann, C. and Stichling, D. (2001). Mobile Ar4All. In Proceedings of IEEE and ACM International Symposium on Augmented Reality, Washington, DC, pp. 181-182.

Geobase (2007). <http://www.geobase.ca/geobase/en/data/cded/index.html>.

Gezici, S., Tian, Z., Giannakis, G.B., Kobayashi, H., Molisch, A. V., Poor,H.V. and Sahinoglu, Z. (2005). Localization via Ultra-Wideband Radios, IEEE Signal Processing Magazine, 22(4), pp. 70-84.

Grimson, W.E.L., Ettinger, G.J., White, S.J. and Gleason, P.L. (1995). Evaluating and Validating an Automated Registration System for Enhanced Reality Visualization in Surgery, In the proceedings of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine, London, UK, pp. 3-12.

Groove Crane (2006). TMS870/TTS870 Product Guide Brochure, Manitowoc Crane Group, <http://www.manitowoccranegroup.com/MCG_GRO/Products/EN/download.asp?file=../../../../MCG_Downloads/MCG_GRO/EN/TMS800E_ProductGuide.pdf>

Hammad, A., Wang, H., Zhang, C. and Al-Hussein, M. (2006). Visualizing Crane Selection and Operation in Virtual Environment, Proceedings of the 6th International Conference on Construction Applications of Virtual Reality, Florida.

Hamza-Lup, F. G., Santhanam, A. P., Imielinska, C., Meeks, S. L. and Rolland, J. P. (2007). Distributed Augmented Reality With 3-D Lung Dynamics—A Planning Tool Concept, IEEE Transactions on Information Technology in Biomedicine, Jan. 2007, pp. 40-46.

Hasvold, P. (2002). In-the-field health informatics, In The Open Group conference, Paris, France.<http://www.opengroup.org/public/member/q202/documentation/plenary/hasvold.pdf>.

Hicks, J., Flanagan, R., Petrov, P., Stoyen, A. (2003). Eyekeon: distributed augmented reality for soldier teams, In Proceedings of 8th International Command and Control Research and Technology Symposium (ICCRTS'03), Washington D.C, USA.

Hix, D. and Gabbard, J.L. (2002). Usability Engineering of Virtual Environments, Handbook of Virtual Environments: Design, Implementation, and Applications, Lawrence Erlbaum Associates, pp. 681-699.

Hoff, B. and Azuma, R. (2000). Autocalibration of an Electronic Compass in an Outdoor Augmented Reality System, Proceedings of International Symposium on Augmented Reality 2000, Munich, Germany, pp. 159-164.

Hofmann-Wellenhof, B., Lichtenegger, H. and Collins, J. (2004). Global Positioning System: Theory and Practice, 5th Edition, Springer, New York, USA.

Hollerer, T., Feiner, S., and Palik, J. (1999). Situated documentaries: Embedding multimedia presentations in the real world, In Proc. ISWC '99 (third IEEE Int. Symp. on Wearable Computers), San Francisco, CA, pp. 79-86.

Hollerer, T., Hallaway, D., Tinna, N., Feiner, S. (2001). Steps toward accommodating variable position tracking accuracy in a mobile augmented reality system. Proceedings of 2nd International Workshop on Artificial Intelligence in Mobile Systems (AIMS '01), Seattle, WA, pp. 31-37.

Hornaday, W.C., Haas, C.T., O'Connor, J.T. and Wen, J. (1993). Computer-aided planning for heavy lifts, J. Constr. Eng. Manage., 119(3), pp. 498-515.

Horstmann, C. (2004). Big Java, 2nd edition, John Wiley & Sons, Inc. New York, USA.

Intel (2007). <http://www.intel.com/technology/comms/uwb/index.htm>.

InterSense (2007). Product Manual for use with InertiaCube3 and the InertiaCube3 Processor, <http://www.isense.com/support.aspx?id=248&>.

InterSense (2002). InterSense Inc., IS-900 Wide Area Precision Motion Tracker, IS300 and InertiaCube2 orientation sensors. <http://www.isense.com>.

Isdale, J. (2003). Introduction to VR Technology, In Proceedings of the IEEE Virtual Reality 2003, Washington, DC, pp. 302.

JARToolKit (2004). http://sourceforge.net/projects/jartoolkit.

Johnson, J., Roberts, T. L., Verplank, W., Smith, D. C., Irby, C. H., Beard, M., and Mackey, K. (1989). The Xerox Star: A Retrospective, IEEE Computer, Vol. 22, No. 9, pp. 11-29.

JXInput (2007). <http://www.hardcode.de/jxinput>.

Kaiser Electro-Optics (2007). <http://www.rockwellcollins.com>.

Kalman, R.E. (1960). A New Approach to Linear Filtering and Prediction Problems, Transaction of the ASME—Journal of Basic Engineering, pp. 35-45.

Kamat, V.R. and Martinez, J.C. (2001). Visual Simulated Construction Operations in 3D, Journal of Computing in Civil Engineering, 15(4), ASCE, pp. 329-337.

Kato, H. and Billinghurst, M. (1999). Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In Proceedings of the 2nd IEEE and ACM international Workshop on Augmented Reality. Washington, DC, pp. 85-94.

Kaplan, E.(1996). Understanding GPS Principles and Applications, Artech House Publishers, New York, USA.

Law, A.M. and Kelton, W.D. (2000). Simulation Modeling and Analysis, 3rd Edition. McGraw-Hill, New York, NY.

Lee, P. U., Ruspini, D. C., and Khatib, O. (1994). Dynamic simulation of interactive robotic environment, In the proceedings of IEEE Int. Conf. Robot. Automat., San Diego, CA, pp. 1147–1152.

Lenz, E. (2004). Networked Transport of RTCM via Internet Protocol (NTRIP): Application and Benefit in Modern Surveying Systems, FIG Working Week 2004, Athens, Greece.

Lewandowski, S.M. (1998). Frameworks for component-based client/server computing, ACM Computing Surveys, 30(1), pp. 3-27.

Lin, K.L. and Haas, C.T. (1996). Multiple Heavy Lifts Optimization, Journal of Construction Engineering and Management, 122(4), pp. 354-362.

Low, K., Welch, G., Lastra, A., and Fuchs, H. (2001). Life-Sized Projector-Based Dioramas, Symposium on Virtual Reality Software and Technology, Banff, Alberta, pp. 161-168.

Lorensen, W., Cline, H., Nafis, C., Kikinis, R., Altobelli, D. and Gleason, L. (1993). Enhancing reality in the operating room, In the proceedings of IEEE Visualization '93, San Jose, CA, pp. 410-415.

Malbezin, P., Piekarski, W., and Thomas, B. H. (2002). Measuring ARToolKit Accuracy in Long Distance Tracking Experiments. In Proceedings of the 1st International Augmented Reality Toolkit Workshop, Darmstadt, Germany.

MapObjects-Java (2006). <http://www.esri.com/software/mojava>

McCarthy, J.M. (1990). An introduction to theoretical kinematics, MIT Press.

Mellor, J.P. (1995). Enhanced Reality Visualization in a Surgical Environment, MS Thesis, Department of Electrical Engineering, MIT.

Microcrowd (2007). <http://www.microcrowd.com>.

Milgram, P., and Kishino, F. (1994). A Taxonomy of Mixed Reality Visual Displays, IEICE Transactions on Information and Systems, 12(7), pp. 1321-1329.

Milgram, P., Takemura, H., Utsumi, A., and Kishino, F. (1994). Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum, SPIE Proceedings volume 2351: Telemanipulator and Telepresence Technologies, Boston, MA, pp. 282-292.

Mizell, D. (2001). Boeing's wire bundle assembly project. In Barfield, W. and Caudell, T., editors, Foundamentals of Wearable Computers and Augmented Reality, Lawrence Erlbaum Assoc, Mahwah, NJ, pp. 447-467.

Moehring, M., Lessig, C., and Bimber, O. (2004). Video See-through AR on Consumer Cell-phones, In proceedings of IEEE/ACM ISMAR'04, Washington, DC, pp. 252 - 253.

NMEA (2007). NMEA 0183 Standard. <http://www.nmea.org/pub/0183>.

NVIS (2007). <http://www.nvisinc.com/product.php?id=1>.

Pahlavan, K., Li, X., and Makela, J. (2002). Indoor Geolocation Science and Technology, IEEE Communications Magazine, 40(2), pp. 112-118.

Pavlik, J. V. (2001). Journalism and New Media, Columbia University Press, New York, NY.

Piekarski, W. and Thomas, B. (2002). ARQuake: the outdoor augmented reality gaming system. Communications of the ACM 45(1), pp. 36-38.

Penn, A., Mottram, C., Schieck, A.F., Wittkämper, M., Störring, M., Romell, O., Strothmann, A. and Aish, F. (2004). Augmented reality meeting table: a novel multi-user interface for architectural design, In proceedings of 7th International Conference on Design & Decision Support Systems in Architecture and Urban Planning 2004, Sint-Michiels Gestel, Netherlands, pp. 213-220.

Pinhanez, C. (2001). The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces, In proceedings of Ubiquitous Computing, Atlanta, Georgia, USA, pp. 315-331.

Raab, F. H., Blood, E. B., Steiner, T. O., and Jones, H. R. (1979). Magnetic position and orientation tracking system, IEEE Transaction on Aerospace and Electronic Systems, AES-15(5): pp. 709-717.

Raskar, R., Welch, G., and Chen, W.C. (1999). Table-Top Spatially Augmented Reality: Bringing Physical Models to Life with Projected Imagery, In proceedings of Second International IEEE Workshop on Augmented Reality (IWAR'99), San Francisco, CA, pp. 64-71.

Reilly, D. (2000). Object Persistence Made Easy. <http://www.javacoffeebreak.com/articles/serialization/index.html>.

Rolland, J. P., Hamza-Lup, F., Davis, L., Daly, J., Ha, Y., Martin, G., Norfleet, J., Thumann, R. and Imielinska, C. (2003). Development of a training tool for endotracheal intubation: Distributed augmented reality, In Proceedings of Medicine Meets Virtual Reality, Newport Beach, California, pp. 288-294.

Satoh, K., Anabuki, M., Yamamoto, H., and Tamura, H. (2001). A Hybrid Registration Method for Outdoor Augmented Reality, In Proceeding of ISAR'01(Int. Symposium on Augmented Reality), New York, NY, pp. 57-76.

Schkolne, S., Pruett, M., Schröder P. (2001). Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools, In Proceeding of CHI 2001, Seattle, USA, pp. 261-268.

Selman, D. (2002). Java 3D Programming, 1st Edition. Manning Publications Co.

Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming Languages. IEEE Computer, Vol. 16, No. 8, pp. 57-69.

Shneiderman, B. (2000). Creating creativity: User interfaces for supporting innovation, ACM Transactions on Computer-Human Interaction 7(1), pp. 114-138. Reprinted in Carroll, J. M. (Editor), Human-Computer Interaction in the New Millennium, ACM Press, New York, pp. 235-258.

Simlog (2006). http://www.simlog.com/index.html.

State, A., Livingston, M.A., Garrett, W.F., Hirota, G., Whitton,M.C., Pisano, E.D. and FUchs, H.(1996). Technologies for Augmented-Reality Systems: realizing Ultrasound-Guided Needle Biopsies, In Proceeding of SIGGRAPH 96, New Orleans, LA, pp. 439-446.

SUN (2003). <http://java.sun.com/j2se/1.4.2/docs/guide/jni/>.

SUN (1999). <http://java.sun.com/products/javacomm>.

Trimble (2005). <http://www.trimble.com>.

Underkoffler, J., Ullmer, B. and Ishii, H. (1999). Emancipated Pixels: Real-World Graphics in the Luminous Room, In Proceedings of ACM SIGGRAPH, New York, USA, pp. 385-392.

Vlahakis, V., Ioannidis, N., Karigiannis, J., Tsotros, M., Gounaris, M., Stricker, D., Gleue, T., Daehne, P., and Almeida, L. (2002). Archeoguide: An Augmented Reality Guide for Archaeological Sites. IEEE Computer Graphics and Applications, 22(5), pp. 52-60.

Walesh, A. and Gehringer, D. (2001). Java 3D, API Jump-start, Prentice Hall PTR.

Watt, A. (2000). 3D Computer Graphics, 3rd Edition. Addision-Wesley Publishing Company Inc.

Watt, A. and Policarpo, F. (2003). 3D GAMES Animation and Advanced Real-time Rendering. Volume two. Addision-Wesley Publishing Company Inc.

Wagner, D., and Schmalstieg, D. (2003). First Steps towards Handheld Augmented Reality. In proceedings of International Conference on Wearable Computers, Washington, DC, pp. 127-136.

Wang, L. and Chen, C. (1991). A Combined Optimisation Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators. IEEE Transactions on Robotics and Automation, 7 (4), pp. 489-99.

Wang, X.Y. and Dunston, P.S. (2006). Usability Evaluation of a Mixed Reality Collaborative Tool for Mechanical Design. In proceedings of the Third International Conference on Computer Graphics, Imaging and Visualization (CGIV06), IEEE Computer Society, Sydney, Australia, pp. 448-451.

Wang, Y., Jia, X., Lee, H.K. (2003). An Indoors Wireless Positioning System Based on Wireless Local Area Network Infrastructure, Proceedings of the 6th International Symposium on Satellite Navigation (SatNav 2003), Melbourne, Australia.

Webster, A., Feiner, S., MacIntyre, B., Massie, W., and Krueger, T. (1996). Augmented Reality in Architectural Construction, Inspection, and Renovation. In Proceedings of the Third ASCE Congress for Computing in Civil Engineering, Anaheim, California, pp. 913-919.

Welch, G. and Foxlin, E. (2002). Motion Tracking: No Silver Bullet, but a Respectable Arsenal, IEEE Computer Graphics and Applications, special issue on "Tracking", 22(6), pp. 24-38.

Xiang, Z., Song,S., Chen J.,Wang, H., Huang, J., and Gao, X. (2004). A Wireless LAN-based Indoor Positioning Technology, IBM journal of research and development, 48(5-6), pp. 617-626.

Xu, B., Hischke, S. and Walke, B. (2003). The Role of ad Hoc Networking in Future Wireless Communications, In Proceedings of International Conference on Communication Technology, ICCT 2003, Beijing, China, pp. 1353-1358.

You, S., Neumann, U., and Azuma, R. (1999). Orientation Tracking for Outdoor Augmented Reality Registration. IEEE Computer Graphics and Applications, 19(6), pp. 36-42.

You, S.Y., Neumann, U., and Azuma, R. (1999). Hybrid Inertial and Vision Tracking for Augmented Reality Registration, Proceedings of IEEE VR '99 , Houston, TX, pp. 260-267.

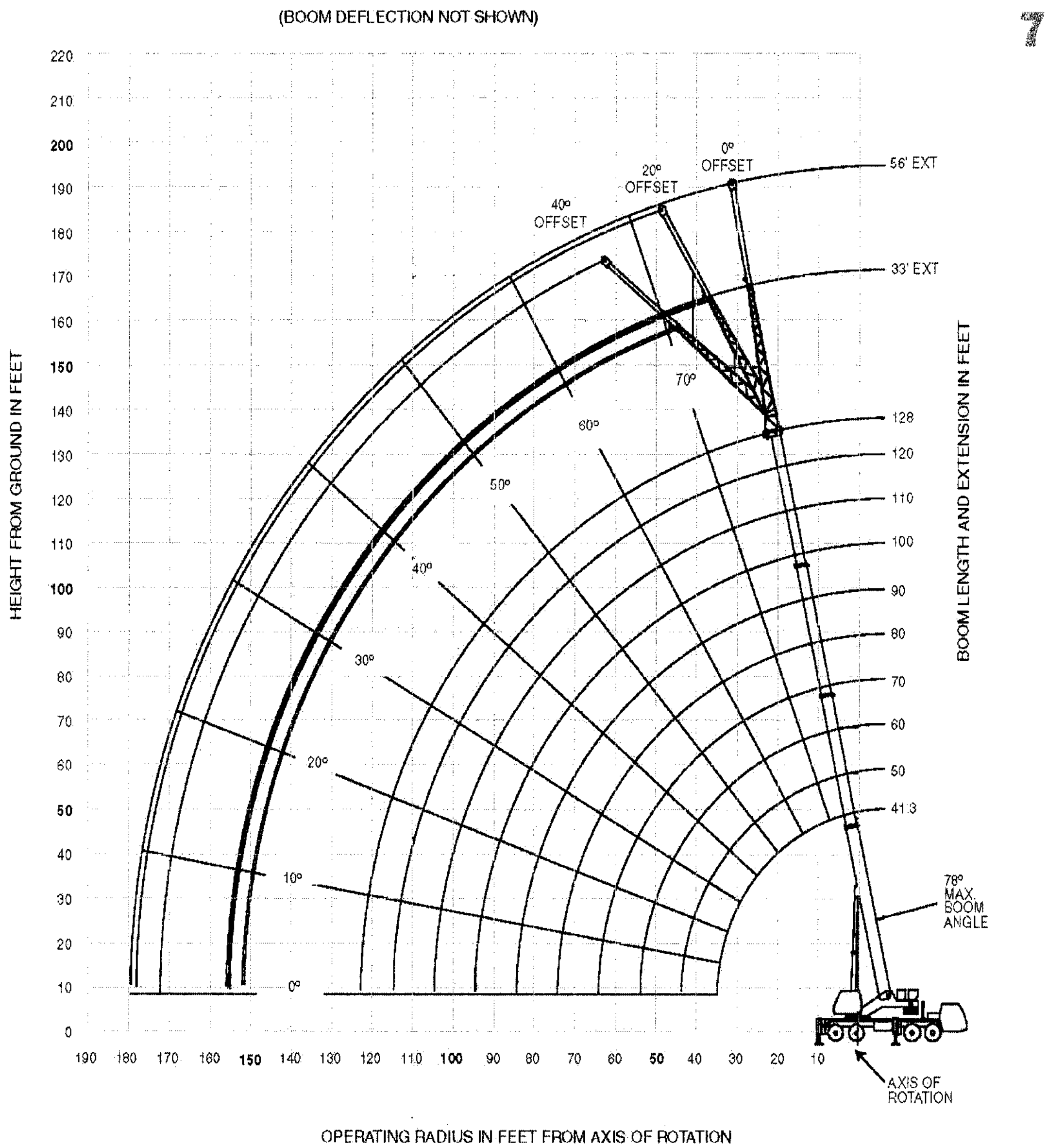
Youssef, M., Agrawala, A., and Shankar, A.U. (2003). WLAN Location Determination via Clustering and Probability Distributions, IEEE International Conference on Pervasive Computing and Communications (PerCom), Fort Worth, Texas.

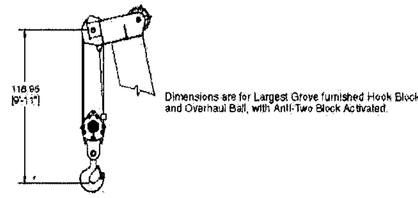
Zaki, A. R. and Mailhot, G.. (2003). Deck Reconstruction of Jacques Cartier Bridge Using Precast Prestressed High Performance Concrete Panels, PCI Journal, pp. 20-33.

Zhong, X. W., Boulanger, P. and Georganas, N. D. (2002) Collaborative Augmented Reality: A Prototype for Industrial Training, In Proceedings of 21st Biennial Symposium on Communications, Kingston, Canada.

Zitnick, C.L., Kang, S. B.,Uyttendaele, M., Winder, S., and Szeliski, R. (2004). High-Quality Video View Interpolation Using a Layered Representation, ACM SIGGRAPH and ACM Transactions on Graphics, Aug. 2004, pp. 600-608.

APPENDIX A
Working Range and Load Chart of Grove Crane TMS 800E (Grove Crane 2006)





TMS800E

41.3-128 ft. | 24,000 lbs | 100% 24' 0" spread | 360°

Pounds

	41.3	50	60	**70	80	90	100	110	120	128
8	+160,000 (73)									
9	++150,000 (71.5)	86,000 (75)								
10	147,000 (70)	86,000 (74)	86,000 (77)							
12	130,500 (67)	86,000 (71.5)	86,000 (75)	41,000 (77)						
15	111,000 (62)	86,000 (67.5)	86,000 (71.5)	41,000 (74.5)	39,000 (76.5)					
20	87,650 (53.5)	86,000 (61)	85,900 (66.5)	41,000 (70)	39,000 (73)	38,800 (75)	*38,700 (78)	*31,950 (78)		
25	67,700 (44)	67,450 (54)	67,250 (61)	41,000 (65.5)	39,000 (69)	38,800 (71.5)	38,700 (74)	31,950 (75.5)	*25,750 (78)	*14,600 (78)
30	50,550 (31)	50,800 (46.5)	50,750 (55.5)	41,000 (61)	39,000 (65)	38,800 (68.5)	36,150 (70.5)	31,950 (72.5)	25,750 (74.5)	14,600 (75.5)
35		38,600 (37)	38,750 (49.5)	38,650 (56.5)	38,150 (61)	34,100 (65)	31,350 (67.5)	29,300 (70)	25,750 (72)	14,600 (73)
40		30,300 (24)	30,500 (42)	30,600 (51)	31,550 (57)	30,050 (61)	27,500 (64.5)	25,650 (67.5)	23,900 (69.5)	14,600 (71)
45			24,550 (33.5)	24,700 (45.5)	25,700 (52.5)	26,500 (57.5)	24,400 (61.5)	22,700 (64.5)	21,450 (67)	14,600 (68.5)
50	See Note 16		20,050 (21.5)	20,250 (39)	21,150 (47.5)	22,050 (53.5)	21,850 (58)	20,250 (61.5)	19,100 (64.5)	14,600 (66)
55				16,750 (31.5)	17,650 (42.5)	18,500 (49.5)	19,300 (54.5)	18,200 (58.5)	17,100 (62)	14,600 (64)
60				13,950 (20.5)	14,800 (36.5)	15,650 (45)	16,450 (51)	16,450 (55.5)	15,450 (59)	14,600 (61.5)
65					12,450 (29)	13,300 (40)	14,150 (47)	14,550 (52)	14,000 (56)	13,350 (59)
70					10,500 (18.5)	11,300 (34)	12,150 (42.5)	12,600 (48.5)	12,700 (53)	12,150 (56)
75						9,650 (27.5)	10,500 (38)	10,950 (45)	11,350 (50)	11,050 (53.5)
80						8,220 (17.5)	9,100 (32.5)	9,530 (41)	9,950 (47)	10,100 (50.5)
85							7,870 (26)	8,300 (36.5)	8,710 (43)	9,090 (47.5)
90							6,800 (17)	7,220 (31)	7,620 (39.5)	8,000 (44)
95								6,260 (25)	6,660 (35)	7,030 (40.5)
100								5,410 (16)	5,810 (30)	6,170 (36.5)
105									5,040 (24)	5,410 (32)
110									4,360 (16)	4,720 (27)
115										4,090 (21)
120										3,530 (10)

Minimum boom angle (deg.) for indicated length (no load)	9
Maximum boom length (ft.) at 0 deg. boom angle (no load)	120

#LMI operating code. Refer to LMI manual for instructions.
*This capacity is based upon maximum obtainable boom angle.
Note: ( ) Boom angles are in degrees.
+ Special equipment is required to lift this capacity.
++9 parts of line required to lift this capacity (using aux. boom nose). Refer to Operator's & Safety Handbook for reeving diagram.

Lifting Capacities at Zero Degree Boom Angle

Boom Angle	Main Boom Length in Feet								
	41.3	50	60	**70	80	90	100	110	120
0°	20,750 (34.1)	15,150 (42.8)	10,500 (52.8)	6,700 (63)	5,100 (72.8)	3,900 (82.8)	2,900 (92.8)	2,000 (102.8)	1,300 (112.8)

Note: ( ) Reference radii in feet.
**This boom length is with inner-mid fully extended and outer-mid & fly fully retracted.

A6-829-103890

APPENDIX B

Software Requirements and Installation Guide of the Prototype System

Software requirements:

(1) Borland JBuilder 2005 Enterprise: used to develop the prototype system of BMS;
(2) MS Access (MS Access XP): used to store the lifecycle data of the bridge;
(3) ArcGIS (ESRI 2004): used to develop GIS application;
(4) JARToolkit: used to develop the video see-though AR;
(5) Windows XP: used as the operation system.

Installation guide:

1. Copy four folders to corresponding driver and change the associated code in the project to match the driver path. The contents in these folders include:
   - *Currproject* or *infra_project* (The folder includes all codes of our project)
   - *Javasoft* (The folder includes all libraries which are required in our project)
   - *Bridgere* (The folder includes all 2d information)
   - *BridgeResources* (The folder includes all 3d information and models)

2. Click Start->Control Panel-> Administrative Tools->Database Source.
   And add ODBC data source as below:
   - Microsoft Access Driver: Name: bridge, Location: C:\ BridgeResources\db1.mdb

   (No password for data source is required, so just leave the password as blank.)

3. Launch Jbuilder, open the project.jpx file. Then click the menu of Jbuilder: Project->Project Properties. Click the tab "Required Libraries". Then edit or add the path of libraries as below:

VRML97 (VRML File Loader API)

- Download the library from: https://j3d-vrml97.dev.java.net/ and install it. Or you can get it from the path: D:/javasoft/loaders/vrml97.jar
- Edit or add the vrml97.jar file to the path of VRML97 library.

JxInput (Game Input Device Library API)

- Download the library from: http://www.hardcode.de/jxinput/ and install it. Or you can get it from the path: D:/javasoft/loaders/jxinput.jar
- Edit or add the vrml97.jar file to the path of VRML97 library.

Jdk3D (Java development Kit 3D) - Version 1.3.1

- Download the library from: http://java.sun.com/products/java-media/3D/downloads/index.html and install it. Or you can get it from the path: D:/javasoft/ JRE/1.3.1_09/lib/ext
- Edit or add all jar files under the Jdk3d directory to the path of Jdk3D library.

Javacomm (Java Communications API) - Version 3.0

- Download the library from: http://www.sun.com/download/products.xml?id=43208d3d and install it. Or you can get it from the path: D:/Javasoft/commapi/
- Edit or add the comm.jar file under the commapi directory to the path of Javacomm library.

DXFLoader (DXF file loader API) - Version 1.0

- Download the library from: http://www.johannes-raida.de/index.htm?cadviewer and install it. Or you can get it from the path: D:/Javasoft/DxFloader/
- Edit or add the dxfloader.jar file under the dxfloader directory to the path of DXFLoader library.

MOJ (MapObject Java API) - Version 2.1

- Download the library from: http://www.esri.com/software/mojava/ and install it.
- Edit or add all jar files under the directory C:/ESRI/MOJ21/lib to the path of MOJ library. Please also add the tutsource.jar and tutorial.jar files that are available at the directory MOJ21\Samples\Tutorial.

Loader3DS (3DS file loader) - Version 1.2

- Download the library from: http://sourceforge.net/projects/java3dsloader/ and install it. Or you can get it from the path: D:/Javasoft/
- Edit or add the path of the directory NeticaJ_Win/bin to the path of Netica library.

JMF (Java Media Framework API) - Version 2.1.1

- Download the library from: http://java.sun.com/products/java-media/jmf/2.1.1/download.html
  and install it. Or you can get it from the path: D:/Javasoft/ jmf211e_scst/build/win32/lib
- Edit or add the path of the directory jmf211e_scst/build/win32/lib to the path of JMF library.

JARToolkit (Java ARToolkit API) - Version 2.0

- Download the library from: http://jerry.c-lab.de/jartoolkit/ and install it. Or you can get it from the path: D:\JavaSoft\JARToolkit Dlls
- JARToolkit need the dll files as below:
  - JARFrameGrabber.dll
  - JARToolkit.dll
  - JARVideo.dll
  - libARvideo.dll
  - libARvideod.dll
  - msvcr70.dll

Make sure you put these dll files in your JARToolkit Dlls directory, then click start-> Control Panel->System->Advanced->Environment Variables, please add the path of JARToolkit Dlls directory to the option"path" of User Variables.
Notes:

* When you use a different account in the same computer, you have to separately set the path of libraries for every account. It means you cannot just set the libraries for all accounts at the same time.

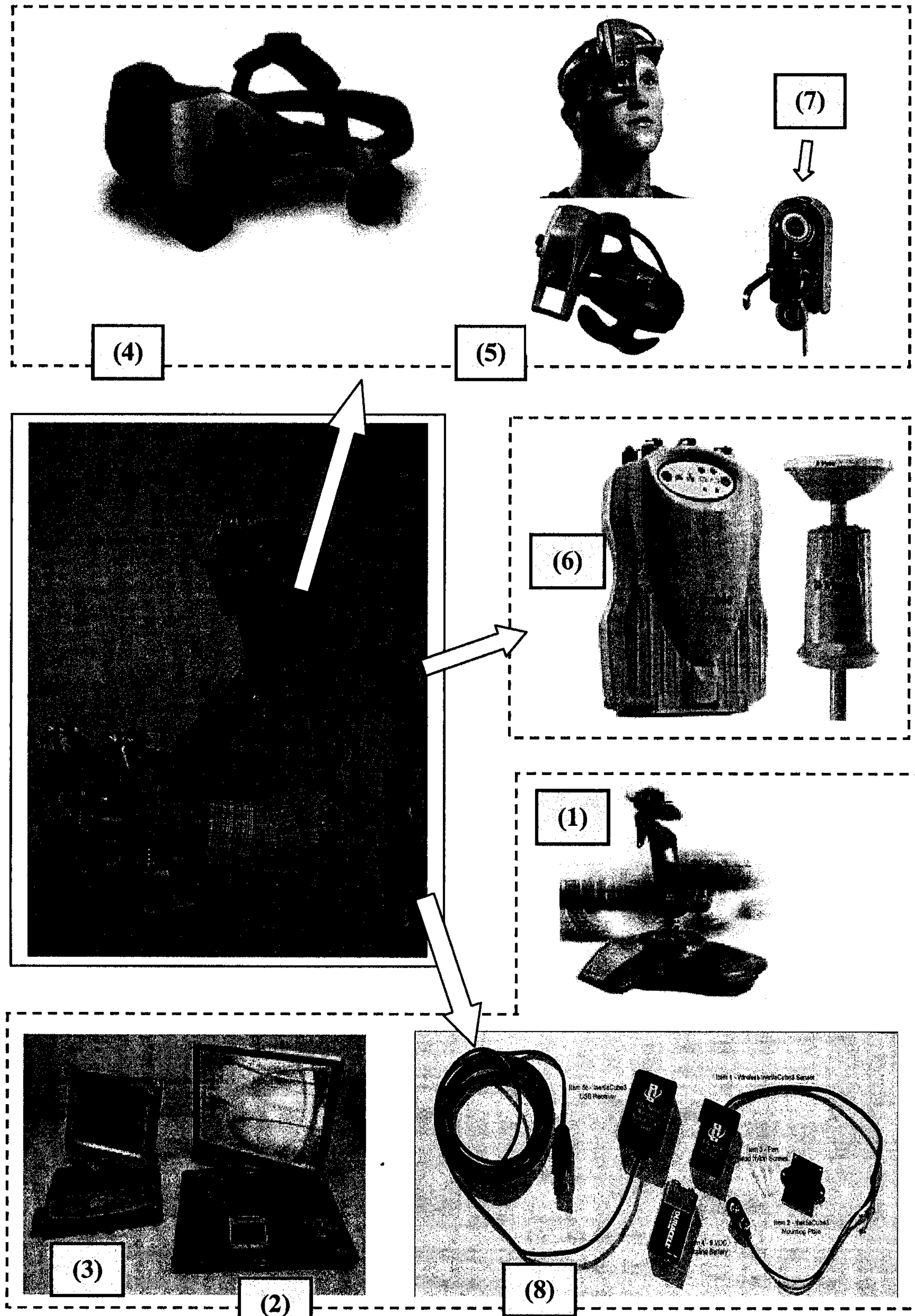
* If your code cannot be compiled following the above instructions, please carefully check the installation instruction. If the error information is about Java 3D, that means your computer does not have Java 3D. You can get the Java 3D package from the Java 3D folder.

Table B.1 Summary of libraries used in the prototype system

Library	Description	Source	Version
VRML97	VRML File Loader API	https://j3d-vrml97.dev.java.net/ D:/javasoft/loaders/vrml97.jar	
JxInput	Game Input Device Library API	http://www.hardcode.de/jxinput/ / D:/javasoft/jxinput.jar	6.1
JDK 3D	Java 3D API	http://java.sun.com/products/javamedia/3D/downloads/index.html D:/javasoft/ JRE/1.3.1_09/lib/ext	1.3.1
Javacomm	Java Communication API	http://www.sun.com/download/products.xml?id=43208d3d D:/Javasoft/commapi/	3.0
DXFLoader	DXF File Loader API	http://www.johannes-raida.de/index.htm?cadviewer D:/Javasoft/DxFloader/	1.0
MOJ	MapObject Java API	http://www.esri.com/software/mojava/ C:/ESRI/MOJ21/lib	2.1
Loader3DS	3DS file loader	http://sourceforge.net/projects/java3dsloader/ D:/Javasoft/ loader3DS	2.17
JMF	Java Media Framework API	http://java.sun.com/products/java-media/jmf/2.1.1/download.html D:/Javasoft/jmf211e_scst/build/win32/lib	2.1.1
JARToolKit	Java ARToolKit API	http://jerry.c-lab.de/jartoolkit/ D:\JavaSoft\JARToolkit Dlls	2.0

APPENDIX C

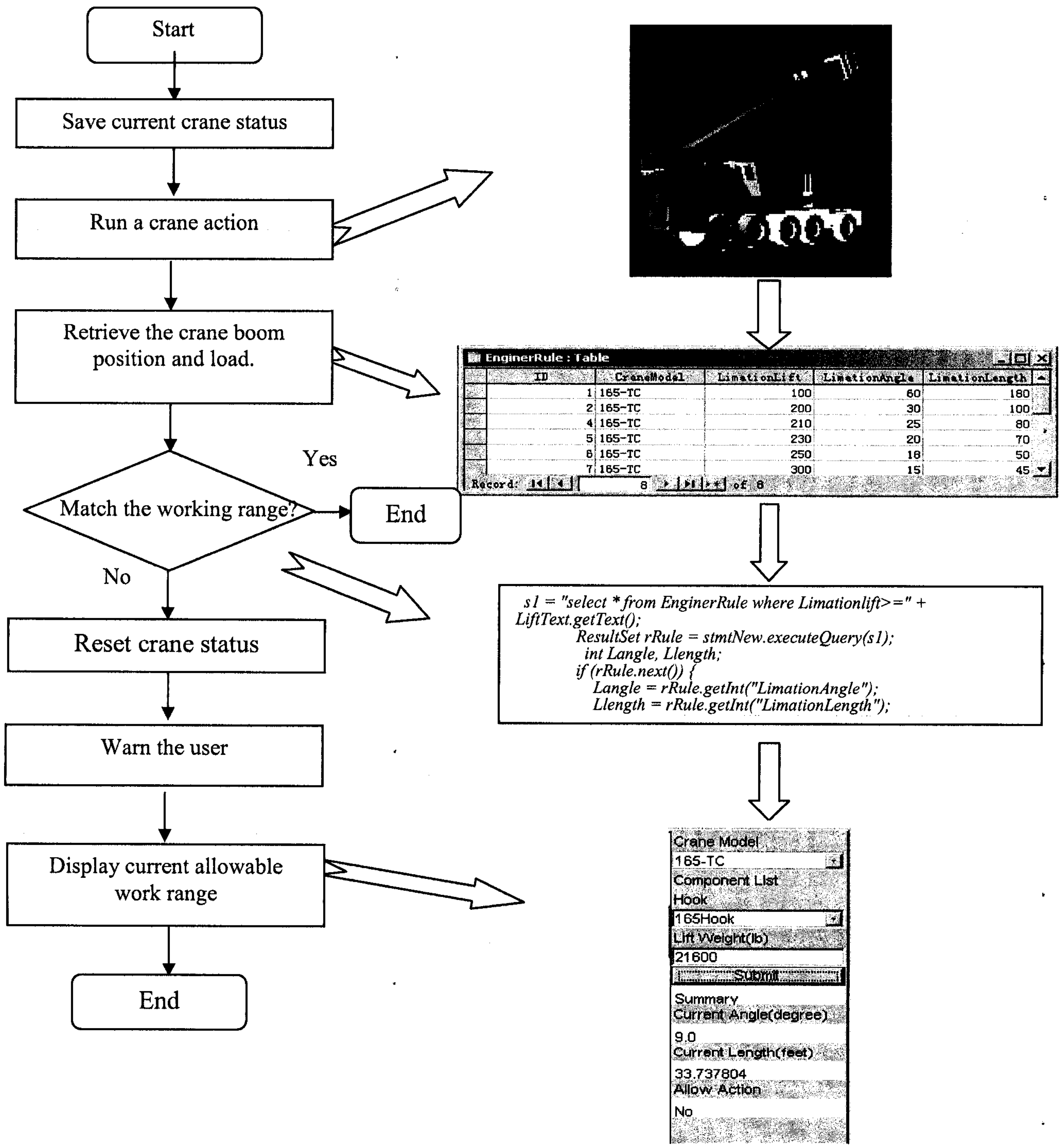
Equipment Used in the Prototype System



Type	No.	Device	Feature
Joystick	(1)	Saitek Cybord evo wireless joystick	Range:30feet Frequency:2.4 GHz
Tablet PC	(2)	Toshiba Tecra M4 Tablet PC	Processor: Mobile Intel® Pentium® M1.86GHz, Memory: 1GB DDR2, Hard Disk Driver: 80GB, Display: TFT Active Matrix colour LCD display w digitizer, Pointing Device: Touchpad + Accupoint, Toshiba Tablet Pen, Battery Life: 3.5 hours
	(3)	Panasonic Toughbook-18	Processor: Intel® Pentium® M Processor ULV 753, 1.20GHz Memory: 512MB+256MB SDRAM standard Hard Disk Driver: 40GB Display: (XGA) transmissive daylight readable TFT Active Matrix Color LCD, Pointing Device: Pressure sensitive touchpad with vertical scrolling support Battery Life: 7.0 hours
HMD	(4)	NVIS ST Optical HMD	Display Format: 1280 x 1024 pixels, 60 Hz refresh rate Display Color: 24-Bit. Field of View: Approximately 48° horizontal, 60° diagonal, Input Signal: SXGA format Brightness (nits): 30 fL max
	(5)	Microvision Nomad ND2000	Display Format: SVGA 800x600 pixels, 60 Hz refresh rate Display Color: Monochrome Red Field of View: 17.25° horizontal, 23° diagonal Input Signal: SVGA format Focus Range: Adjustable from 1 feet to infinity. Continuous Operation: 8 hours on full charge
GPS	(6)	Trimble 5700 RTK GPS	Dual-frequency, RTK, GPS and WAAS/EGNOS receiver, Portable Zephyr antenna, 128 MB of compact flash memory, USB port to transfer data to a PC at speeds of more than 1mb/s.
Digital Camera	(7)	Logitech-3000	Resolution: 640 x 480 (Image), 640 x 480 @ 15fps (Video), Features: Built-in microphone , Connectivity: USB
3D orientati on sensor	(8)	InertiaCube3	Degrees of freedom:3(Yaw, Pitch and Roll) Angular Range: Full 360° Maximum Angular Rate: 1200° per second Minimum Angular Rate: 0° per second

APPENDIX D

The Work Flow of Engineering Constraint Component



APPENDIX E

GPS Programming Manual

1. How to connect GPS with computer

- Use configuration software such as Trimble GPS Configurator, Trimble Configuration Toolbox to set Trimble 5700. Make sure you set the port2 as the output port and output frequency is 10Hz/sec. (You can set other frequency values, but don't set the frequency which less than one message per second). And also set the baud rate of port2 be 9600, add the output protocol as the NMEA-GGK for port2.
- Connect your GPS with your computer in COM port. Note: The com connection cable of GPS is female, so you just can link COM1 in most computer. If your computer doesn't have a COM port, you can link USB port with the USB-COM convert cable. In this situation, you have to install the drive program of USB-COM convert cable and set the number of the simulated COM port.
- I recommend that you use the FindGPS (an extension of ArcPad) to test your GPS and get the parameter of GPS before you try to connect GPS with your own Java program. You can download the FindGPS software from here: http://arcscripts.esri.com/details.asp?dbid=12637
  Install it and launch it, it will automatically scan your COM ports and try to find your GPS. When it finds your GPS, it will display the parameters of your current setting of GPS. We will use the parameters in our Java Program.

2. How to receive data from GPS in Java

2-1 API: We must use the Java Communications API in our GPS Java program. You can download the Java Communications API from here: http://java.sun.com/products/javacomm/index.html . Then install the Java Communications API as fellows:

(1) Copy the platform-specific library (win32com.dll for Win32) to the bin folder of the JDK. For example, if the Java SDK is located in the C:\jdk2 directory, copy win32com.dll into C:\jdk2\bin. This ensures that the native methods found in this library are available at runtime.

(2) Copy the file comm.jar into the lib folder of the JDK.

(3) You must also copy the javax.comm.properties file into the lib folder, C:\jdk2\lib. If the entry in this file is not available to the application, the communications driver class will fail to load and no device operations, such as enumerating or opening ports, will succeed.

2-2 NMEA: NMEA is a standard protocol used by GPS receivers to transmit data. NMEA output is EIA-422A but for most purposes you can consider it RS-232 compatible. NMEA 0183 sentences are all ASCII. Each sentence begins with a dollarsign ($) and ends with a carriage return linefeed (<CR><LF>). Data is comma delimited. All commas must be included as they act as markers. We will use the NMEA protocol to communicate with GPS and extract current position information from the NMEA format message. You can get more information of NMEA format from: http://www.kh-gps.de/nmea.faq

2-3 Programming steps:

Here are the steps for programming with GPS:

1. Read the bytes from the serial port.
2. Define and monitor the event of GPS
3. Extract and parse the data of GPS message.

2-3-1 Read the bytes from the serial port.

Here is the code of example which explains how to get the serial port:

```
commportnumber = System.getProperty("commportnumber");
baudrate = System.getProperty("baudrate");
if (commportnumber == null)
{
trace.append("Failed to load commportnumber
from properties");
enabled = false;
}

try
{
portId = CommPortIdentifier.getPortIdentifier
(commportnumber);
}
catch (NoSuchPortException e)
{
trace.append("Failed to get port identifier!
[" + e.getMessage() + "]");
e.printStackTrace();
return;
}
```

Note: In our test, the code always tries to connect a wrong COM port (COM2). Actually we connected the GPS with COM1 port, so I suggest that we manually set the port as below:

```
commportnumber="COM1";
```

Please take care of this point, sometimes Java cannot automatically get a correct COM port, so you have to set it by yourself.

After we received the correct serial port, we should connect the port, we use the code as below:

```
try
```

```
{
sPort = (SerialPort)portId.open("GPS", 1000);
}
catch (PortInUseException e)
{
trace.append("Port in use??
[" + e.getMessage() + "]");
e.printStackTrace();
return;
}
```

When we open the serial port, we will set the port configuration as below:

```
try
{
trace.append("Setting parameters\r\n");
sPort.setSerialPortParams(9600,SerialPort.DATABITS_8,
SerialPort.STOPBITS_1,SerialPort.PARITY_NONE);
}
catch (UnsupportedCommOperationException e)
{
trace.append("Failed to set port parameters
[" + e.getMessage() + "]");
e.printStackTrace();
sPort.close();
return;
}
// Set flow control.
try
{
sPort.setFlowControlMode(SerialPort.FLOWCONTROL_NONE);
}
catch (UnsupportedCommOperationException e)
{
trace.append("Setting flow control failed
[" + e.getMessage() + "]");
e.printStackTrace();
sPort.close();
return;
}
```

I recommended you use the parameter of FindGPS for adjusting the port configuration in our program. It is a convenience way to put correct parameters for debugging.

2-3-2. Define the event of GPS

In addition to the ActionListener interface, our java class also implements the SerialPortEventListener interface by providing the serialEvent() method. This method is invoked whenever an event on the port occurs. The most common example is the arrival of data at the port. The first portion of serialEvent simply dumps all received data to the TextArea. We use the necessary event to monitor the data of GPS. You can review the code as below:

```
public void serialEvent(SerialPortEvent e)
{
//System.err.println("Serial Event
[" + e.getEventType() + "]");
int newData = 0;
switch (e.getEventType())
```

```
{
// Read data until -1 is returned.
If \r is received substitute
// \n for correct newline handling.
case SerialPortEvent.DATA_AVAILABLE:
if (!nmea)
{
inputBuffer = new StringBuffer();
while (newData != -1)
{
try
{
newData = is.read();
if (newData == -1)
{
break;
}
inputBuffer.append((char)newData);
}
catch (IOException ex)
{
System.err.println(ex.getMessage());
return;
}
}
// Append received data to messageAreaIn.
trace.append(new String(inputBuffer));
}
```

2-3-3. Extract and parse the messages.

When we start to monitor the GPS, we will receive the messages of NMEA format from GPS every second. We have to extract and parse the message for getting the data which we required.

Here is the code of parsing message as below.

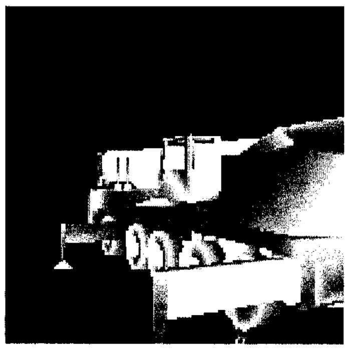
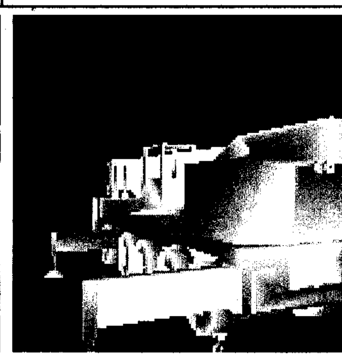
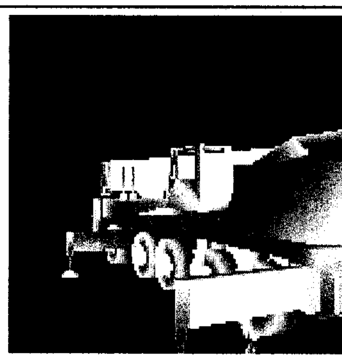
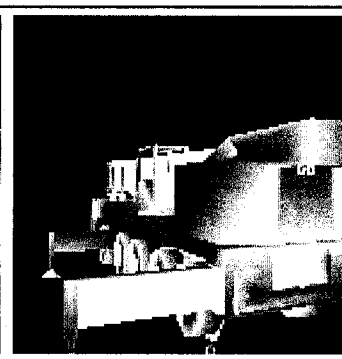
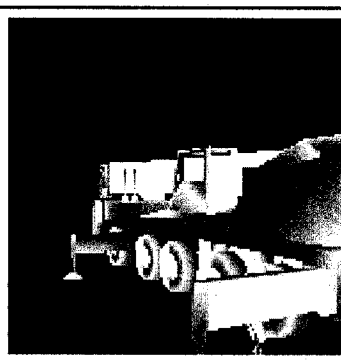
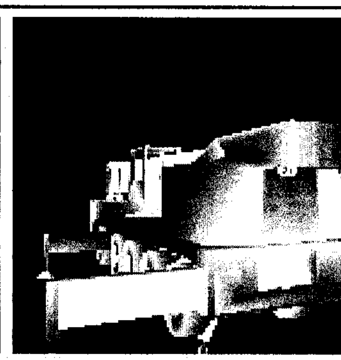
```
{
// parse NMEA sentence
boolean done = false;
boolean gotsentence = false;
boolean sentenceactive = false;
boolean eatchecksum = false;
while (!done)
{
try
{
newData = is.read();
switch (newData)
{
case -1:
done = true;
break;
case '$':
sentenceactive = true;
break;
case '*':
eatchecksum = true;
break;
case 0x0d:
break;
case 0x0a:
sentenceactive = false;
gotsentence = true;
```

```
done = true;
break;
default:
if (!eatchecksum)
{
inputBuffer.append((char)newData);
}
break;
}
{ catch (IOException ex)
{
System.err.println(ex.getMessage());
}
}
if (gotsentence)
{
trace.append("NMEA Sentence\t
[" + new String(inputBuffer) + "]\r\n");
inputBuffer = new StringBuffer();
}
}
break;
// If break event append BREAK RECEIVED message.
case SerialPortEvent.BI:
trace.append("\n--- BREAK RECEIVED ---\n");
break;
}
}
```

In this code, we parse the message and put the data from the message into the inputBuffer. You can use the data as you required. For example, you can use the data to display your current location in GIS system.

APPENDIX F

Tests of Stereo View

Left eye view	Right eye view	Inter-ocular distance (cm)
		4
		6
		8

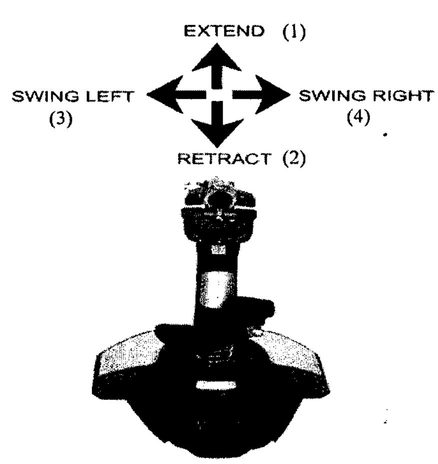
APPENDIX G

User Operation

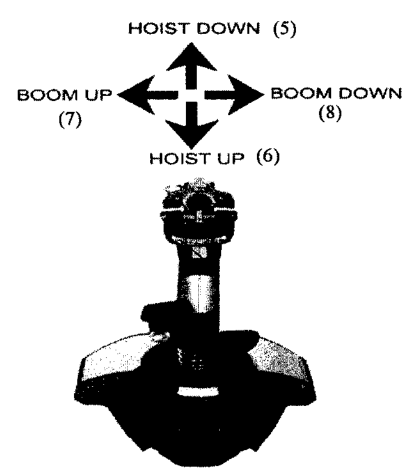
Table G.1 Keyboard operation:

Number	Key	Function
1	n/N	Boom extend
2	m/M	Boom retract
3	e/E	Swing left
4	r/R	Swing right
5	b/B	Hoist down
6	v/V	Hoist up
7	1/L	Boom up
8	k/K	Boom down
	Up/Down	Rotate the crane around the X-axis
	Left/Right	Rotate the crane around the Y-axis
	1/3	Toggle the first and third person view
	s/S	Start/stop 3D orientation tracking
	g/G	Start/stop GPS tracking

Joystick operation:



(a) Left control



(b) Right control

APPENDIX H

Offline Tests

Figure H.1 shows another left view of an offline test. It includes two virtual cranes and correctly aligns the two virtual cranes on the Jacques Cartier Bridge. Figure H.2 shows a virtual crane with two real cranes. The two real cranes are used for comparison.

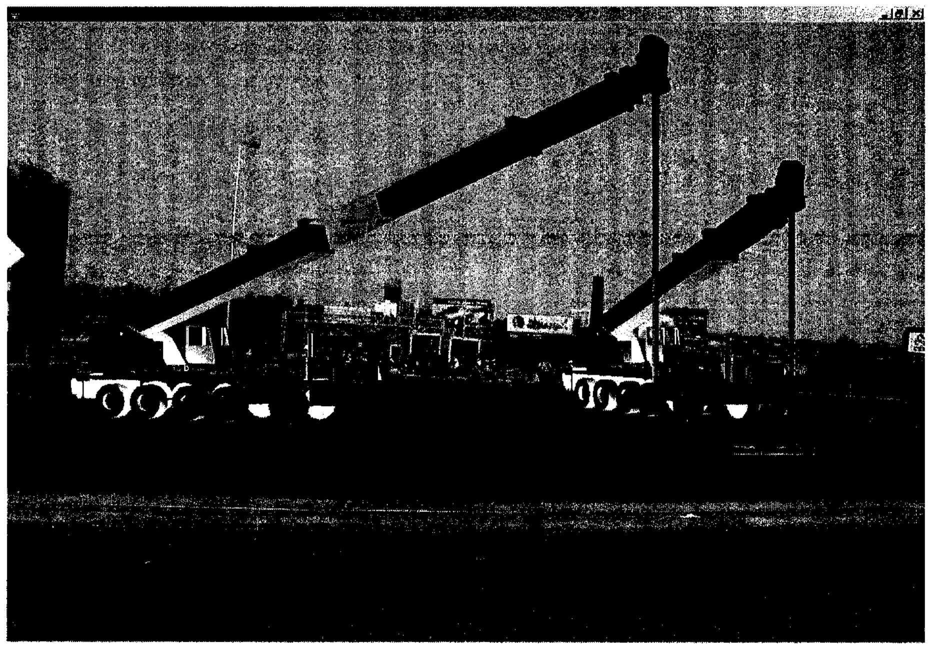


Figure H.1: A left view of two virtual cranes with the construction site

Figure H.2: A virtual crane and two real cranes on Jacques Cartier Bridge

APPENDIX I

HMD Configuration for Stereo View

1. Connect two VDS cables of HMD to the video control unit.
2. Enable the 'Horizontal Spanning' mode using the nView tool under the advanced Setting of the video card.
3. Set the parameters of the advanced timing as below:
   Pixel clock: 108.00 MHz
   Horizontal geometry
   Sync Polarity: +
   Sync Width: 112 pixels
   Front porch: 48 pixels
   Hsync frequency: 63.981 Hz
   Vertical geometry
   Sync polarity: +
   Sync width: 3 lines
   Back porch: 38 lines
   Front porch: 1 line
   Vsync frequency: 60.020 Hz
4. Connect two video cables from two independent video outputs of the video card to the video control unit.
5. Power on the video control unit.

Currently we only found the Nvidia Quadro 4500X2 video card supports the 'Horizontal Spanning' mode and provides two independent video outputs. The test results of stereo view with the video card can be found in Appendix F.

APPENDIX J

List of Publications

Hammad, A., Wang, H. and Mudur, S.P. (2008). Distributed Augmented Reality for Visualizing Collaborative Construction Tasks, Submitted Janauary 2008 to the Journal of Computing in Civil Engineering.

Wang, H., Hammad, A. and Mudur, S.P. (2007). Design of a Distributed Mixed Reality System for Construction Simulation, ASCE International Workshop on Computing in Civil Engineering, Pittsburgh, PA.

Hammad, A., Wang, H., Zhang, C. and Al-Hussein, M. (2006). Visualizing Crane Selection and Operation in Virtual Environment, the 6th International Conference on Construction Applications of Virtual Reality, Florida.