

## **INFORMATION TO USERS**

**This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.**

**The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.**

**In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.**

**Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.**

**ProQuest Information and Learning  
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA  
800-521-0600**

**UMI<sup>®</sup>**





**Université d'Ottawa • University of Ottawa**



# **Mobile Collaborative Augmented Reality: A Prototype for Industrial Training**

**By**

**Xiaowei Zhong, B.A.Sc.**

**A Master's thesis submitted to  
the Faculty of Graduate and Postdoctoral Studies  
in partial fulfillment of the requirements of the degree of**

**Master of Computer Science**

**Under the auspices of the  
Ottawa-Carleton Institute for Computer Science**

**University of Ottawa**

**Ottawa, Ontario, Canada**

**August, 2002**

**Copyright © Xiaowei Zhong, Ottawa, Canada, 2002**



**National Library  
of Canada**

**Acquisitions and  
Bibliographic Services**

**395 Wellington Street  
Ottawa ON K1A 0N4  
Canada**

**Bibliothèque nationale  
du Canada**

**Acquisitions et  
services bibliographiques**

**395, rue Wellington  
Ottawa ON K1A 0N4  
Canada**

*Your file Votre référence*

*Our file Notre référence*

**The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.**

**The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.**

**L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.**

**L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.**

0-612-76659-4

**Canada**

## **Abstract**

**Augmented Reality (AR) is a powerful user interface paradigm which enhances a user's perception by incorporating computer-generated information into the real world. Mobile computing, typically realized by a wearable computer, allows users to access and manipulate information anytime, regardless of location. Computer supported collaborative work allows the computer to be used as a medium for human communication. The combination of these three technologies, called mobile collaborative augmented reality, promises exciting new applications. Training costs are a considerable part of development expenses in many industries, especially when the customers or employees need to learn to use devices that are expensive and costly to repair. The problem is compounded when the trainers are far from the would-be trainees. This thesis describes a prototype developed for collaborative industrial teletraining, based on mobile collaborative augmented reality. The prototype enables trainees to engage in a real environment and do a training exercise on task performance.**

## **Acknowledgement**

The evolution of this thesis was not a solitary effort. Many people assisted me in both the design of the system and in the written portion of the thesis.

First, I would like to express my deepest gratitude to my supervisors, Professors Nicolas D. Georganas and Pierre Boulanger, the latter currently at the University of Alberta. They have not only provided me with direction and focus, but have also given me very helpful advice on my research.

I would also like to thank Dominic Laberge and Winston Wang, who have assisted me with the completion of this thesis. It was a good experience to work with them at the National Research Council for this project, under the instruction of Dr. Pierre Boulanger. In addition, Dominic did me a great favor in helping me understand the field of Augmented Reality at the beginning of the project.

I would also like to thank all members of the Multimedia Communications Research Lab at the University of Ottawa for their cooperation and friendship. Special appreciation goes to Peiran Liu for her cooperation in this project, and Francois Malric for the setup of the system. Peiran and I worked as a team in this research project, but focusing on different topics. I use some of her results in the designing of the prototype. Finally, my family has also been a constant inspiration to me and has given me the necessary support to accomplish my objective.

# Table of Contents

<b>ABSTRACT</b> .....	<b>2</b>
<b>ACKNOWLEDGEMENT</b> .....	<b>3</b>
<b>TABLE OF CONTENTS</b> .....	<b>4</b>
<b>ABBREVIATIONS</b> .....	<b>9</b>
<b>LIST OF FIGURES</b> .....	<b>10</b>
<b>LIST OF TABLES</b> .....	<b>13</b>
<b>CHAPTER 1 INTRODUCTION</b> .....	<b>14</b>
1.1 THESIS ORGANIZATION.....	14
1.2 THESIS CONTRIBUTIONS AND PUBLICATION.....	15
<b>CHAPTER 2 BACKGROUND AND MOTIVATION</b> .....	<b>17</b>
2.1 TYPICAL REQUIREMENTS OF TRAINING APPLICATIONS.....	17
2.2 VIRTUAL ENVIRONMENT FOR TRAINING APPLICATIONS.....	18
2.2.1 <i>Distributed Virtual Environment</i> .....	18
2.2.2 <i>A Prototype for Industrial Training with DVE</i> .....	18
2.2.3 <i>The Advantages of the DVE Solution</i> .....	19
2.2.4 <i>Problems with the DVE Solution</i> .....	19
2.3 AUGMENTED REALITY FOR TRAINING APPLICATIONS.....	20
2.3.1 <i>On-the-job Training</i> .....	20
2.3.2 <i>Augmented Reality</i> .....	20
2.3.3 <i>Augmented Reality for Training</i> .....	20
2.3.4 <i>Related Work on AR Applications</i> .....	21
2.4 MOBILE COMPUTING.....	24
2.4.1 <i>Wearable Computer</i> .....	24
2.4.2 <i>Mobility and Augmented Reality</i> .....	25
2.4.3 <i>Wearable Computer for Training</i> .....	26
2.4.4 <i>Collaborative Wearable System</i> .....	26
2.5 COMPUTER SUPPORTED COLLABORATIVE WORK.....	27

2.6 MOBILE COLLABORATIVE AUGMENTED REALITY .....	28
2.6.1 Remote Collaboration in One Augmented Reality .....	28
2.6.2 Remote Collaboration in Augmented Realities .....	29
2.6.3 Local Collaboration in One Augmented Reality .....	29
2.7 THESIS OBJECTIVE .....	31
2.7.1 Background .....	31
2.7.2 Research Tasks .....	31
2.8 CHAPTER SUMMARY .....	34
<b>CHAPTER 3 RELATED TECHNOLOGIES .....</b>	<b>35</b>
3.1 DISPLAY TECHNOLOGIES IN AUGMENTED REALITY .....	35
3.1.1 Monitor-based AR system .....	36
3.1.2 Optical-based AR System .....	36
3.1.3 Video-based AR System .....	37
3.1.4 Comparison of the Display Technologies .....	38
3.2 TYPICAL REQUIREMENTS FOR AN AR SYSTEM .....	39
3.3 TRACKING AND REGISTRATION .....	39
3.3.1 Tracking .....	40
3.3.1.1 Magnetic Tracking .....	40
3.3.1.2 Mechanical Tracking or Optical Tracking .....	40
3.3.1.3 Vision-based Camera Tracking .....	40
3.3.2 Registration .....	41
3.4 VISION-BASED AR SYSTEM .....	42
3.4.1 Co-ordinate Systems for Augmented Reality .....	42
3.4.2 Camera Viewing Geometry .....	43
3.4.3 Camera Calibration .....	44
3.5 CHAPTER SUMMARY .....	45
<b>CHAPTER 4 DESIGN OF A TRAINING SYSTEM .....</b>	<b>46</b>
4.1 DESIGN GOALS .....	46
4.1.1 The Task: ATM Switch Maintenance .....	46
4.1.2 Problems with the Current Practice .....	46
4.2 MCAR: AN EFFECTIVE SOLUTION .....	47
4.2.1 Virtual Objects .....	48

4.2.2 Collaboration Functions .....	48
4.2.2.1 Audio Conference .....	48
4.2.2.2 Remote Presence .....	48
4.2.2.3 Remote Manipulation .....	48
4.2.2.4 Remote Pointing .....	49
4.2.2.5 Image Freezing .....	49
4.3 HARDWARE CONFIGURATION .....	49
4.3.1 Wearable Computer .....	49
4.3.2 Workstations .....	51
4.4 DESIGN CHALLENGES .....	51
4.4.1 Limitations of the Wearable Computer .....	52
4.4.1.1 Limited Computing Resources .....	52
4.4.1.2 Missing Infrastructure for Building Mobile Applications .....	52
4.4.1.3 Lack of 3D Support in Hardware .....	52
4.4.2 Real-time Performance .....	53
4.4.3 Sensitivity to Errors .....	53
4.4.4 2D Device Limitations .....	53
4.5 DESIGN PRINCIPLES AND DECISIONS .....	53
4.5.1 Display Technique .....	54
4.5.1.1 Principles .....	54
4.5.1.2 Practice and Decision .....	54
4.5.2 Tracking Mechanism .....	55
4.5.2.1 Requirements .....	55
4.5.2.2 Decision .....	55
4.5.3 Distributed Visualization Solution .....	56
4.5.3.1 Centralized Architectures .....	56
4.5.3.2 Replicated Architectures .....	57
4.5.3.3 Centralized Architecture vs. Replicated Architecture .....	57
4.5.3.4 Decision .....	58
4.5.4 Media Communication and Presentation .....	59
4.5.4.1 Quality of Media Stream Presentation .....	59
4.5.4.2 Audio Transmission .....	60
4.5.4.3 Video Compression Standards .....	60
4.5.4.4 Decision .....	61

4.5.5 <i>Data Sharing Mechanism</i> .....	61
4.6 OVERALL ARCHITECTURE .....	61
4.7 CHAPTER SUMMARY .....	64
<b>CHAPTER 5. DETAILED DESIGN AND IMPLEMENTATION .....</b>	<b>65</b>
5.1 DEVELOPMENT ENVIRONMENT .....	65
5.1.1 <i>C/C++</i> .....	65
5.1.2 <i>Microsoft Vision SDK</i> .....	66
5.1.3 <i>ARToolKit</i> .....	66
5.1.4 <i>OpenGL &amp; GLUT</i> .....	66
5.1.5 <i>VRML and OpenVRML</i> .....	66
5.1.6 <i>MFC</i> .....	67
5.1.7 <i>CavernSoftG2</i> .....	67
5.1.8 <i>Intel's JPEG Library</i> .....	67
5.2 DETAILED DESIGN.....	68
5.2.1 <i>Video Capturing</i> .....	68
5.2.1.1 Initialization .....	68
5.2.1.2 Video Image Access.....	68
5.2.2 <i>Video Encode/Decode</i> .....	69
5.2.3 <i>Vision-Based Tracking</i> .....	69
5.2.3.1 Camera Tracking.....	69
5.2.3.2 Recognizing Different Patterns .....	70
5.2.4 <i>Data Transmission</i> .....	71
5.2.4.1 Communication of Collaboration Data .....	72
5.2.4.2 Latecomer Support .....	73
5.2.4.3 Audio Conference .....	73
5.2.5 <i>Graphic Images Rendering &amp; User Interface</i> .....	73
5.2.5.1 User Interaction for Collaboration .....	74
5.2.5.1.1 Virtual Object Manipulation with a Keyboard.....	74
5.2.5.1.2 Virtual Object Manipulation with a Mouse.....	75
5.2.5.1.3 Remote Pointing.....	76
5.2.5.1.4 Image Freezing.....	76
5.2.5.2 User Manager Dialog .....	77
5.2.5.3 Local Viewer & Remote Viewer.....	78

5.2.5.4 Pop-up Menu .....	79
5.3 APPLICATION .....	79
5.4 CHAPTER SUMMARY .....	82
<b>CHAPTER 6. EXPERIMENTAL RESULTS &amp; EVALUATION.....</b>	<b>83</b>
6.1 3D CAPABILITY OF THE WEARABLE COMPUTER.....	83
6.1.1 Evaluation .....	83
6.1.2 Result.....	86
6.1.3 Solution.....	86
6.2 TEST BEDS FOR THE PROTOTYPE.....	86
6.2.1 Test Bed A.....	86
6.2.2 Test Bed B.....	87
6.3 SUBJECTIVE USER EVALUATION .....	89
6.3.1 Function Effectiveness.....	90
6.3.2 System Performance.....	91
6.3.3 Overview to the System .....	92
6.4 CHAPTER SUMMARY .....	93
<b>CHAPTER 7. CONCLUSIONS AND FUTURE WORK.....</b>	<b>94</b>
7.1 CONCLUSIONS .....	94
7.2 FUTURE WORK.....	95
<b>REFERENCES.....</b>	<b>96</b>
<b>APPENDIX A .....</b>	<b>99</b>
CAMERA COMMUNICATION.....	99
1. Sending Video to the Server .....	99
2. Reply Tracking Result to Sender .....	99
3. Sending Data to the Other Clients .....	100

## **Abbreviations**

<b>AR</b>	<b>Augmented Reality</b>
<b>CSCW</b>	<b>Computer Supported Collaborative Work</b>
<b>MCAR</b>	<b>Mobile Collaborative Augmented Reality</b>
<b>HMD</b>	<b>Head-mounted Display</b>
<b>VE</b>	<b>Virtual Environment</b>
<b>VR</b>	<b>Virtual Reality</b>
<b>DVE</b>	<b>Distributed Virtual Environment</b>
<b>DIVERSIONS</b>	<b>Distributed Virtual Environments with Training Applications</b>
<b>NCIT</b>	<b>National Capital Institute of TeleCommunications</b>
<b>VRML</b>	<b>Virtual Reality Modeling Language</b>
<b>MFC</b>	<b>Microsoft Foundation Classes</b>
<b>LAN</b>	<b>Local Area Network</b>
<b>GLUT</b>	<b>OpenGL Utility Toolkit</b>
<b>TCP</b>	<b>Transmission Control Protocol</b>
<b>UDP</b>	<b>User Datagram Protocol</b>
<b>IRB</b>	<b>Information Request Broker</b>
<b>SDK</b>	<b>Software Development Kit</b>
<b>ATM</b>	<b>Asynchronous Transfer Mode</b>
<b>IJL</b>	<b>Intel's JPEG Library</b>
<b>CAR</b>	<b>Collaborative Augmented Reality</b>

## List of Figures

FIGURE 2-1 REPLACING A FAULTY BOARD [1].....	19
FIGURE 2-2 (A) TRIANGULAR TRACKERS ATTACHED TO VARIOUS PARTS; (B) SHOWING HOW TO REMOVE PAPER TRAY [2].....	21
FIGURE 2-3 (A) THREE DISTINCTIVE MARKS ARE ATTACHED AT KNOWN DISTANCES FROM EACH OTHER; (B) SYSTEM LOCATES THE 3D POSITION OF THE OBJECT WITH THE PROBLEM AFTER DIAGNOSTICS [3].....	22
FIGURE 2-4 ENGINE MODEL PART LABELS APPEAR AS USER POINTS AT THEM [19]. .....	23
FIGURE 2-5 (A) PERSON WEARING AN AR SYSTEM; (B) VIEW THROUGH HEAD-MOUNTED DISPLAY, SHOWING OVERLAYS INDICATING DIRECTION TO REMOVE THE COVER OF PC [4]. .....	23
FIGURE 2-6 TWO FIDUCIAL CODES ARE CONNECTED BY A RED ARROW INDICATING THAT THE TWO PARTS ARE RELATED [5]. .....	24
FIGURE 2-7 MARS - MOBILE AUGMENTED REALITY SYSTEM [6].....	26
FIGURE 2-8 (A) WORKER WEARING COLLABORATIVE SYSTEM; (B) ON-LINE MANUAL IS OVELAID ON THE DISPLAY [7]. .....	27
FIGURE 2-9 THREE CATEGORIES OF COLLABORATIVE AR SYSTEMS: (A) REMOTE COLLABORATION IN ONE AUGMENTED REALITY; (B) REMOTE COLLABORATION IN AUGMENTED REALITIES AND (C) LOCAL COLLABORATION IN ONE AUGMENTED REALITY. ....	28
FIGURE 2-10 A COLLABORATIVE INTERIOR DESIGN [13] .....	29
FIGURE 2-11 PLAYING SCENE OF AR2 HOCKEY [9] .....	30
(A) PLAYING SCENE.....	30
(B) PLAYER'S VIEW .....	30
FIGURE 2-12 A GAME OF AR CHESS AS SEEN BY THE "DOCUMENTATION CAMERA" USER [10] .....	31
FIGURE 2-13 PROTOTYPES .....	33
FIGURE 3-1 MONITOR-BASED AR CONCEPTUAL DIAGRAM [12].....	36
FIGURE 3-2 OPTICAL SEE-THROUGH HMD CONCEPTUAL DIAGRAM [12] .....	36
FIGURE 3-3 VIDEO SEE-THROUGH HMD CONCEPTUAL DIAGRAM [12].....	37
FIGURE 3-4 FLOCK OF BIRD .....	40
FIGURE 3-5 FIDUCIALS WITH A WHITE CENTER SURROUNDED BY A BLACK RING.....	41
FIGURE 3-6 COORDINATE SYSTEMS FOR AUGMENTED REALITY [16] .....	42

<b>FIGURE 3-7 COORDINATE SYSTEMS IN CAMERA CALIBRATION</b> .....	<b>44</b>
<b>FIGURE 4-1 A WEARABLE COMPUTER WITH SOME ACCESSORIES</b> .....	<b>50</b>
<b>FIGURE 4-2 WEARABLE COMPUTER PERIPHERALS</b> .....	<b>50</b>
<b>FIGURE 4-3 CENTRALIZED ARCHITECTURES</b> .....	<b>56</b>
<b>FIGURE 4-4 REPLICATED ARCHITECTURES</b> .....	<b>57</b>
<b>FIGURE 4-5 OVERALL ARCHITECTURE OF A COLLABORATIVE AUGMENTED REALITY</b> <b>SYSTEM</b> .....	<b>62</b>
<b>FIGURE 5-1 PHYSICAL VIEW OF THE SYSTEM</b> .....	<b>65</b>
<b>FIGURE 5-2 FLOW CHART FOR VIDEO CAPTURING INITIALIZATION</b> .....	<b>68</b>
<b>FIGURE 5-3 SAMPLE PATTERNS</b> .....	<b>69</b>
<b>FIGURE 5-4 SYSTEM FLOW FOR VISION-BASED TRACKING</b> .....	<b>70</b>
<b>FIGURE 5-5 COMMUNICATION OF COLLABORATION DATA</b> .....	<b>72</b>
<b>FIGURE 5-6 DATA FLOW BETWEEN PERIPHERAL DEVICES AND THE SYSTEM</b> .....	<b>74</b>
<b>FIGURE 5-7 USER MANAGER DIALOG</b> .....	<b>77</b>
<b>FIGURE 5-8 CAMERA VIEWER (LOCAL)</b> .....	<b>78</b>
<b>FIGURE 5-9 REMOTE VIEWER</b> .....	<b>79</b>
<b>FIGURE 5-10 A TRAINEE WORKS WITH A WEARABLE COMPUTER</b> .....	<b>80</b>
<b>FIGURE 5-11 SCREENSHOTS FOR THE INDUSTRIAL TRAINING PROTOTYPE</b> .....	<b>82</b>
<b>(A) CAMERA IS POINTING AT AN ATM SWITCH; (B) DIRECTING THE TRAINEES TO PULL</b> <b>OUT THE SWITCHBOARD;</b> .....	<b>81</b>
<b>(C) TRAINEE A IS PULLING OUT THE SWITCHBOARD; (D) SWITCHBOARD WITH TWO</b> <b>VIRTUAL CHIPS BESIDE IT;</b> .....	<b>81</b>
<b>(E) TRAINEE A PICKS UP ONE VIRTUAL CHIP TO INSTALL IT; (F) THE VIRTUAL CHIP IS</b> <b>LAI D ON THE SWITCHBOARD;</b> .....	<b>82</b>
<b>(G) THE VIRTUAL CHIP IS ROTATED BY TRAINEE B; (H) TRAINER T POINT OUT THAT IS</b> <b>INCORRECT.</b> .....	<b>82</b>
<b>FIGURE 6-1 RENDERING THE VIDEO IMAGES (RV)</b> .....	<b>83</b>
<b>FIGURE 6-2 RENDERING VIDEO IMAGES AND TRACKING VISIBLE VIRTUAL OBJECTS</b> <b>(RV+TR)</b> .....	<b>84</b>
<b>FIGURE 6-3 RENDERING VIDEO IMAGES, TRACKING VISIBLE VIRTUAL OBJECTS AND</b> <b>RENDERING THEM (RV+TR+RG)</b> .....	<b>84</b>
<b>FIGURE 6-4 TEST BED A</b> .....	<b>87</b>
<b>FIGURE 6-5 TEST BED B</b> .....	<b>87</b>

**FIGURE 6-6 (A) VIEWER WITH VIDEO IMAGE IN SIZE (160x120 PIXELS); (B) VIEWER  
WITH VIDEO IMAGE IN SIZE (320x240 PIXELS).....89**

## **List of Tables**

<b>TABLE 4-1 CENTRALIZED ARCHITECTURE VS. REPLICATED ARCHITECTURE.....</b>	<b>57</b>
<b>TABLE 5-1 THE DEFINITION OF OPERATION KEYS FOR REMOTE MANIPULATION .....</b>	<b>75</b>
<b>TABLE 5-2 MOUSE OPERATION SPECIFICATIONS FOR VIRTUAL OBJECT MANIPULATION</b>	<b>76</b>
<b>TABLE 6-1 RUNNING AN AR APPLICATION WITH DIFFERENT CONFIGURATIONS .....</b>	<b>85</b>
<b>TABLE 6-2 RUNNING A CAR APPLICATION WITH DIFFERENT CONFIGURATIONS .....</b>	<b>88</b>
<b>TABLE 6-3 EVALUATION OF THE FUNCTION EFFECTIVENESS .....</b>	<b>90</b>
<b>TABLE 6-4 PERFORMANCE EVALUATION OF THE COMBINATION OF VARIOUS FUNCTIONS</b> <b>.....</b>	<b>91</b>
<b>TABLE 6-5 SYSTEM PERFORMANCE EVALUATION .....</b>	<b>92</b>
<b>TABLE 6-6 OVERVIEW TO THE SYSTEM .....</b>	<b>92</b>

# **Chapter 1 Introduction**

Training costs are a considerable part of the development expenses in many industries, especially when the customers or employees need to learn to use equipment or devices that are expensive and costly to repair. The problem is compounded when the trainers are far from the would-be trainees. Traditionally the manufacturers distributed books, cassettes or tapes to the customers to train them to use and maintain their products. When customers need help, the manufacturers can provide support through telephone, fax or mail, and sometimes even send a technical support person to the site to solve the problem for them.

A training system such as distance learning, on-line help through the Internet and a computer training application system based on virtual environment technology provide instruction or support in the form of text, audio, video and animation. As the emerging training system not only enhances understanding but also saves time and travel costs, it has been considered as very successful.

However, the training systems described above provide training but do not consider the performance of the task in detail. If the real world task is complicated, it is difficult for the trainees to memorize every step learned in the training system. Furthermore, they may not be able to obtain support if they have questions when they are working in the real world.

More effort is required to address the above problems. The trend is to provide on-the-job training. In this thesis, we will introduce a new method that integrates Augmented Reality (AR), mobile computing and computer supported collaborative work (CSCW) to address the above problems. We will present a prototype for an industrial training application.

This opening chapter will briefly introduce the background knowledge of a training system. In section 1.1, an overall structure of this thesis will be given. In section 1.2, the thesis contributions and publication will be presented.

## **1.1 Thesis Organization**

The thesis contains seven chapters, arranged as follows:

- Chapter 1 briefly introduces the motivation of the thesis and its organization. Then it presents the thesis contributions and the publications produced based on the thesis. Chapter 2 serves as the background and objective of the thesis. It presents the typical requirements of a training system and the problems which exist in the current training systems. It introduces the concepts of AR, mobile computing and CSCW and discusses related research in these areas. At the end of this chapter, we present an effective solution for industrial training with mobile collaborative augmented reality (MCAR) and the thesis objective.
- Chapter 3 is a short tutorial of the required technologies used in developing an AR system. It discusses the typical requirements of an augmented reality system, and introduces the display technologies, tracking and registration techniques for an augmented reality system. With this knowledge, one may better understand the following chapters.
- Chapter 4 discusses the design of a MCAR system. It presents design goals, challenges, principles and decisions. The chapter ends with a presentation of the overall architecture of a MCAR prototype for industrial training applications.
- Chapter 5 elaborates the design and implementation of the prototype.
- Chapter 6 presents the experimental results and the evaluation of the prototype.
- Chapter 7 presents the conclusions and future work of this thesis project.

## **1.2 Thesis Contributions and Publication**

The major contributions of this thesis are:

To address the problems in training applications, give a solution with MCAR and develop a prototype for industrial training based on this solution, with the following characteristic:

- The prototype has a client/server architecture which supports multiple geographically distributed trainees and trainers.
- The trainee, equipped with a wearable computer, is able to do a training exercise in the real world with augmented information, such as annotations and 3D models

superimposed on the real world environment. The trainee is also able to organize the 3D models spatially and interact with them directly in a natural way.

- Remote trainees and trainers are able to share the augmented workspace of the trainee with a wearable computer, see what s/he sees, watch his/her action and interact with the 3D models as if the remote users are present in the physical world.
- The trainers and trainees can engage in a real-time audio conversation, and are therefore able to discuss the real world task, and give advice or instructions.

To our knowledge, this is the first successful prototype for an industrial training application based on MCAR technology. This prototype has been created for industrial training applications. However, it can be applied to other applications.

Two pieces of paper based on the thesis are published:

X.Zhong, P.Liu, N.Georganas and P.Boulanger, "Designing a Vision based Collaborative Augmented Reality Application for Industrial Training", Informationstechnik und Technische Informatik, Oldenburg Verlag, Jan. 2003

X. Zhong, P. Boulanger and N.D. Georganas, "Collaborative Augmented Reality: A Prototype for Industrial Training", Proc. 21st Biennial Symposium on Communications, Kingston, Canada, June 2002.

## **Chapter 2 Background and Motivation**

Traditional training or support methods give the learner the best training possible before deployment. A distributed virtual environment (DVE) is known to be an attractive solution for industrial training. However, this solution comes with some problems. Recent advances in computer and computer vision technologies have sparked an interest in migrating training applications to augmented reality and mobile computing technologies.

In this chapter, we will describe the motivation of this thesis. In section 2.1, the typical requirements of training applications are introduced. In section 2.2, the problems which exist in the current industrial training systems using virtual environment technology are discussed. Sections 2.3 and 2.4 introduce augmented reality and mobile computing technologies respectively. Section 2.5 provides a definition of CSCW. Section 2.6 discusses the technologies of mobile collaborative augmented reality (MCAR) - the combination of augmented reality, mobile computing and CSCW. Section 2.7 proposes using MCAR technology for industrial training application and presents the thesis objective. The chapter ends with a short summary.

### **2.1 Typical Requirements of Training Applications**

In general, training is most effective when the following conditions are met [20]:

- Training occurs at the actual work site.
- Supporting information is available.
- The material learned is immediately applied.
- Follow-on expert support/guidance is available.
- Training occurs when the learner needs it most.
- A combination of task-appropriate media (e.g. text, audio, drawings, video) is employed.

## **2.2 Virtual Environment for Training Applications**

In the last few years, Virtual Reality (VR) or Virtual Environment (VE) has become a very important new “medium”. Virtual Reality (VR) can be used as a powerful, three-dimensional method to interface with computers. By wearing a head-mounted audio-visual display, position and orientation sensors, and tactile interface devices, one can actively inhabit an inclusive computer-generated environment.

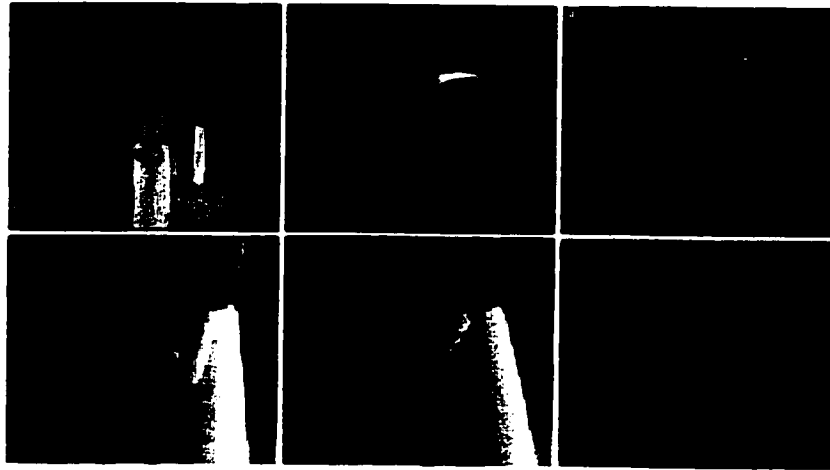
### **2.2.1 Distributed Virtual Environment**

One of the hottest topics in Virtual Reality research is that of “Distributed” Virtual Environments (DVE). In DVE the computers are connected over a network. People using those computers are able to interact and collaborate in real-time, sharing the same virtual world. Tele-immersive DVEs recognize the presence and movements of individuals and objects, and then project them in realistic, multiple, geographically distributed virtual environments where individuals can interact with each other as well as computer-generated models.

### **2.2.2 A Prototype for Industrial Training with DVE**

Industrial training in the field of equipment operation, maintenance and service is one of the many applications that can benefit and grow with the development of the distributed virtual reality technology.

A good example of a DVE application for industrial training is a prototype demonstrated in the Multimedia Communications Research Lab, University of Ottawa [1], in which DVE technology is used to construct a virtual world where users (both trainers and trainees) can interact with one another and the environment in which they reside in order to perform a training exercise. In this DVE, an actual “trainer”, represented by an avatar, removes and replaces faulty parts from an ATM switch, while a “trainee” (avatar) watches and repeats the instructions. Screenshots for this system are shown in figure 2-1. All DVE object manipulations are multicast to all participants via a live session. The trainer or trainee controls his/her avatar movements through either mouse manipulation or voice commands or command menus or tracked body movements.



**Figure 2-1 Replacing a faulty board [1]**

### **2.2.3 The Advantages of the DVE Solution**

A realistic training application can benefit from some tele-presence capabilities which allow the human operator to experience the feeling of being virtually immersed in the working environment. Since collaboration is achieved, it is no longer necessary for trainers and trainees to be in the same geographical location. Instead of working with real objects, the users work with virtual objects placed in a virtual environment, which are accessible to many users. This significantly solves the problem in some applications where the working environment is dangerous or the equipment is expensive and its misuse may lead to serious damage. From an industry perspective, a DVE is an attractive solution for reducing training expenses.

### **2.2.4 Problems with the DVE Solution**

In a DVE, all of the physical objects are represented as virtual, and the users of a DVE are completely immersed in the virtual environment and separated from the real world.

It is time consuming to simulate a real world. And it is also difficult for the users to identify a specific object from the context when there are many objects in the virtual world and these objects are similar to each other. For instance, this is the case where a trainee is trained to repair a specific chip in a switchboard on which there are dozens of similar chips.

Furthermore, the trainees have to remember every step in the virtual training exercise. When trainees work in the real world, they have to review the steps learned in the training exercise, just like replaying a movie. It is hard for them to remember every step correctly and do the task properly, if the task is complicated.

Finally, there is no instant help from the experts (trainers) when the trainees have problems, for instance, which faulty chip should be replaced and how to do it.

## **2.3 Augmented Reality for Training Applications**

### **2.3.1 On-the-job Training**

Traditional training systems train the learners before they do the real job. A DVE training system is a kind of pre-training system in which a real work task is simulated. It cannot give effective training or support when the learners need it most – carrying out the task in the real world. One of the best solutions is “on-the-job” training, giving learners performance support by integrating computer-based guidance and information into the normal working environment.

### **2.3.2 Augmented Reality**

From the perspective of human senses, objects may have behaviors that are not visible in the real world, but may be represented in a virtual environment. For example, electrical circuits operating at high frequencies emit electromagnetic radiation that can be rendered visible in a virtual environment. In this case, the virtual environment is not a mutually exclusive alternative to the real environment, but consists of additional information that can be overlaid on the real world. In this way, both the virtual and real environments are visible simultaneously, which is called augmented reality (AR). Therefore, Augmented Reality (AR) is a variation of Virtual Reality. It supplements reality, rather than completely replacing it.

### **2.3.3 Augmented Reality for Training**

Rendering some behavior of objects visible would provide real-time computer-enhanced assistance for plant-personnel training, product design, maintenance and equipment repair. Computer-generated graphics or text prompts would be developed to train and assist plant personnel during complex manipulation or equipment

maintenance and repair tasks. Virtual environments would allow trainers to conduct interactive what-if multi-domain virtual workbenches with “augmented reality” capacities. For example, when a repairman needs to fix a broken pipeline in a wall, a virtual map of the pipeline, overlaid on the real scene of the wall, will save the repairman time in finding the broken pipeline and breaking the wall to get to it.

In an industrial training application, instructions might be easier to understand if they were presented, not as manuals with text and pictures, but rather as 3-D drawings superimposed upon the actual equipment, showing what to do and the next step to take. These superimposed 3-D drawings can be animated, making the directions even more explicit. Therefore, Augmented Reality is an attractive solution for “on-the-job” training.

### **2.3.4 Related Work on AR Applications**

AR systems have been proposed as solutions in many domains, including medical, entertainment, military training, engineering design, robotics and tele-operation, and so on. One category is the assembly, maintenance, and repair of complex machinery. As the objective of this thesis is to investigate the application in the industrial training domain, which is related to this category, we will highlight some of the proposed applications in this category.

- **KARMA** (Knowledge-based Augmented Reality for Maintenance Assistance)



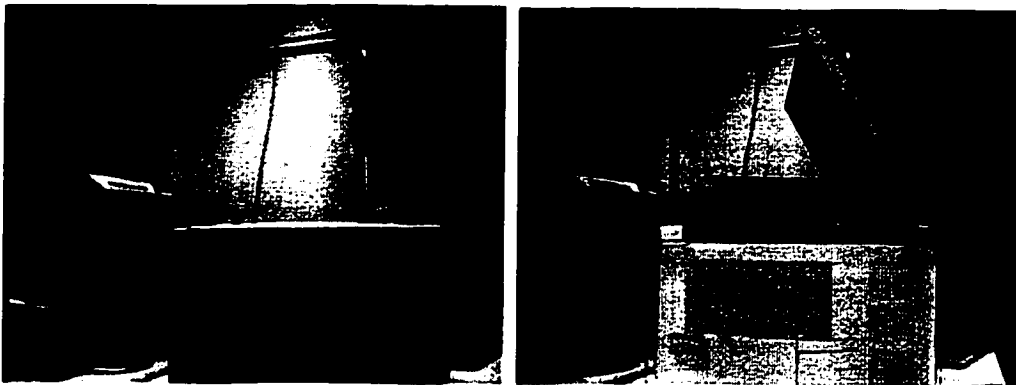
**Figure 2-2 (a) Triangular trackers attached to various parts; (b) Showing how to remove paper tray [2]**

KARMA is a prototype system built by Steve Feiner's group at Columbia. KARMA uses a see-through head-mounted display to explain simple end-user maintenance for

a laser printer. Several Logitech 3D trackers (the small triangles in figure 2-2(a)) were attached to key components of the printer, allowing the system to monitor their position and orientation.

- **Repair Instruction**

The prototype is currently underway in the MIT Media Laboratory [3]. It allows a wearable camera system to track the object in the user's visual field. In this prototype, three distinctive marks at known distances from each other are attached to the product to track the object in the user's visual field. A wearable camera with known focal length can recover the 3D location of the plane defined by these three marks. Thus, when a repair technician walks up to a faulty device, the machine can transmit its diagnostics to the technician's wearable. The wearable automatically determines the problem, locates the 3D position of the object, and overlays specific 3D real-time step-by-step guidelines on the object for the technician to follow.



**Figure2-3 (a) Three distinctive marks are attached at known distances from each other; (b) System locates the 3D position of the object with the problem after diagnostics [3].**

- **Annotation AR System**

At the European Computer-Industry Research Center (ECRC), a user is able to point to the parts of an engine model and the AR system displays the name of the part that is being pointed at [19]. This is illustrated in figure 2-4, when the user points to the exhaust manifold on an engine model and the label "exhaust manifold" appears.



**Figure 2-4 Engine model part labels appear as user points at them [19].**

### ● PC Maintenance

This prototype system [4] could provide informative graphical overlays to give directions to a person to remove parts within the PC. A set of fiducial targets (small white and black circles) is affixed to the cover of a PC, and also to the interior frame of the chassis. A different five-point target pattern was used on the cover and on the interior, thus enabling the system to distinguish between the two.



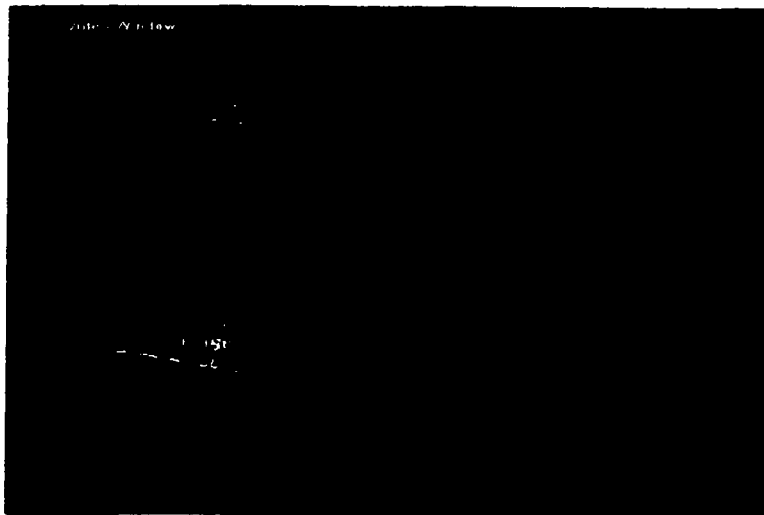
**Figure 2-5 (a) Person wearing an AR system; (b) View through head-mounted display, showing overlays indicating direction to remove the cover of PC [4].**

Figure 2-5 (a) shows a person wearing the AR helmet, looking at the case of the PC. Figure 2-5 (b) shows what he sees in the Head-mounted Display (HMD). To obtain this picture, they placed a video camera at the position of the person's left eyes, and grabbed a frame from the video output. Visible in the picture is the outside frame of the eyepiece, and the overlays through the center of the eyepiece. The overlays in this

particular example are a set of arrows, which show the person how to pull off the cover of the PC.

- **ART (Augmented Reality for Training)**

The ART system [5] uses a monitor-based display to convey the information to the user. This project presents an assembly sequence, real-time video with augmentation, parts graph, and assembly step descriptive video. All parts are tracked using fiducials, special codes that are used to uniquely mark each part. Each fiducial is an asymmetric dot code. The example below shows the wheel assembly step for a Take-Apart Dump Truck. The wheel and base are identified, and labeled appropriately based on the current assembly step.



**Figure 2-6 Two fiducial codes are connected by a red arrow indicating that the two parts are related [5].**

## **2.4 Mobile Computing**

Mobile computing allows users to access and manipulate information anytime, independent of location. With computing devices decreasing in size and with options like wireless networking, a user is no longer limited to his/her physical desktop.

### **2.4.1 Wearable Computer**

Wearable computers are the next generation of mobile computing devices. A typical wearable computer may be composed of an insight processor and a battery mounted

on a belt or backpack, a head mounted display (HMD), wireless communications hardware and an input device such as a touchpad or chording keyboard or voice input utilities.

A wearable computer has many advantages, such as:

- See-through display allows the user to read the text and watch video while working
- Earphones allow the user to hear explanatory audio narration while looking at information on the display
- Miniature microphone, voice recognition software, and applications allow the user to control the computer via voice so his or her hands are free for other tasks
- A computer allows the information designer to select the most effective medium (e.g., text, graphics, video, sound) to communicate information to the user, or illustrate a dynamic procedure in a way that is impossible with a paper-based text or manual
- Batteries, belt-mounted computer, wireless network, and head-mounted display allow the user to be mobile, getting information when and where he or she needs it

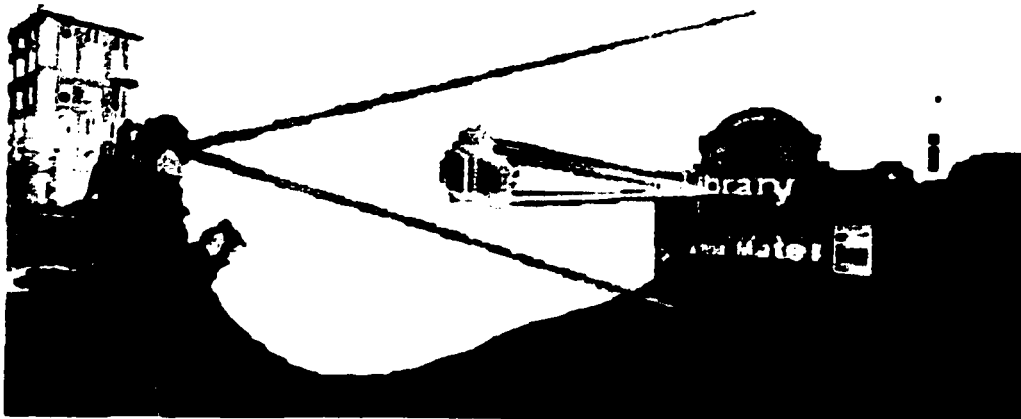
### **2.4.2 Mobility and Augmented Reality**

Augmented Reality is a technology for visual improvement or enrichment of the surrounding environment, by overlaying spatially aligned computer-generated information onto a human's view. Early AR systems are designed for a specific usage in a fixed environment, such as the digital desk.

Progress in wireless networks and the occurrence of wearable computers has made it possible to build a mobile AR system.

A mobile augmented reality system for exploring the urban environment has been developed by the Computer Graphics and User Interfaces Lab, Columbia University [6]. This prototype acts as a campus information system, assisting a user in finding places and allowing users to query information about items of interest, like buildings, statues, and so on. As figure 2-7 shows, a user carries a backpack computer with a wireless network and wears a head-mounted display. The position of the user is

tracked by differential GPS while orientation data are provided by the head-mounted display itself. When the user looks around the campus, the see-through HMD overlays textual labels on campus buildings. The user can interact with the system to bring up related information about any building.



**Figure 2-7 MARS - Mobile Augmented Reality System [6]**

### **2.4.3 Wearable Computer for Training**

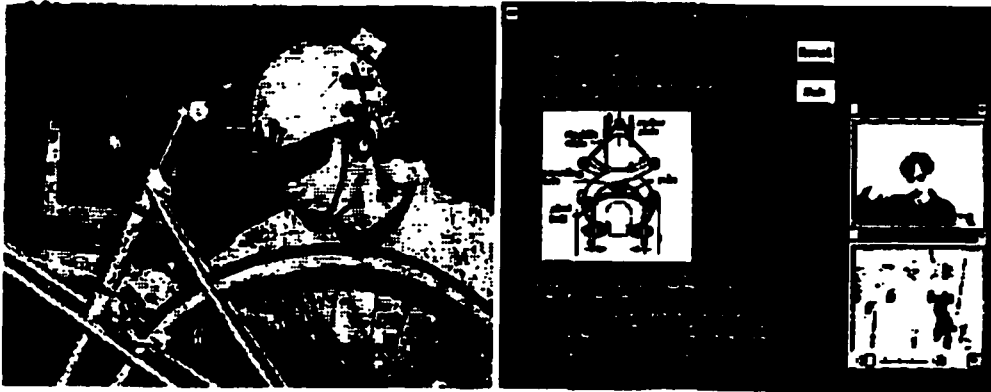
Compared to desktops, wearable computers enable training while the learner performs the task. Compared to traditional training systems, wearable training systems provide relevant, just-in-time information when the user needs it and when the user asks for it. This information can include:

- Just-in-time, task-specific training
- Expert advice about a job task
- Reference information about a job task or closely related set of tasks
- Communication links for remote collaboration with experts

### **2.4.4 Collaborative Wearable System**

All of the systems described above are stand-alone systems that provide users with automatic, context-sensitive access to information, but do not support inter-personal communication and collaboration. There is an obvious need for effective communication and collaboration in the domains of maintenance, repair, construction

and manufacturing. One of the examples is a mobile collaborative system to support maintenance tasks on a bicycle.



**Figure 2-8 (a) Worker wearing collaborative system; (b) On-line manual is overlaid on the display [7].**

In this system, a worker wearing an HMD mounted with a camera is repairing a bicycle (figure 2-8 (a)), while a remote expert watches his action through the video captured by the camera, gives him instruction by live audio, and accesses and overlays an on-line manual onto the HMD of the worker (figure 2-8 (b)).

This is a good example of a collaborative training application. However, the supporting information to the field worker and the collaboration functions between the field worker and expert in the current collaborative wearable system are very limited.

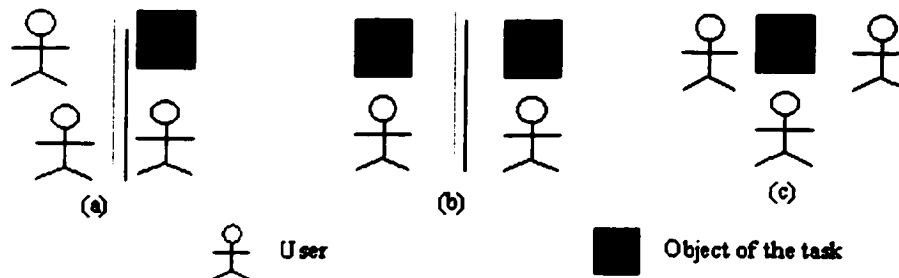
## **2.5 Computer Supported Collaborative Work**

Computer supported collaborative work (CSCW) allows a computer to be used as a medium for human communication. Virtual Reality appears a natural medium for three-dimensional CSCW. However, the current trend in CSCW is toward the Open Shared Workspace, in which the computer is adapted to work with the user's traditional tools, rather than separating the user from them, as immersive VR does. One of the attractive solutions is through Augmented Reality, the overlaying of virtual objects on the real world. Shared Space is a concept which can be described as an Open Shared Workspace featuring Augmented Reality, in which multiple users are allowed to work in both the real and virtual worlds simultaneously.

## 2.6 Mobile Collaborative Augmented Reality

The combination of Augmented Reality, mobile computing and CSCW produces a new technology, called Mobile Collaborative Augmented Reality (MCAR).

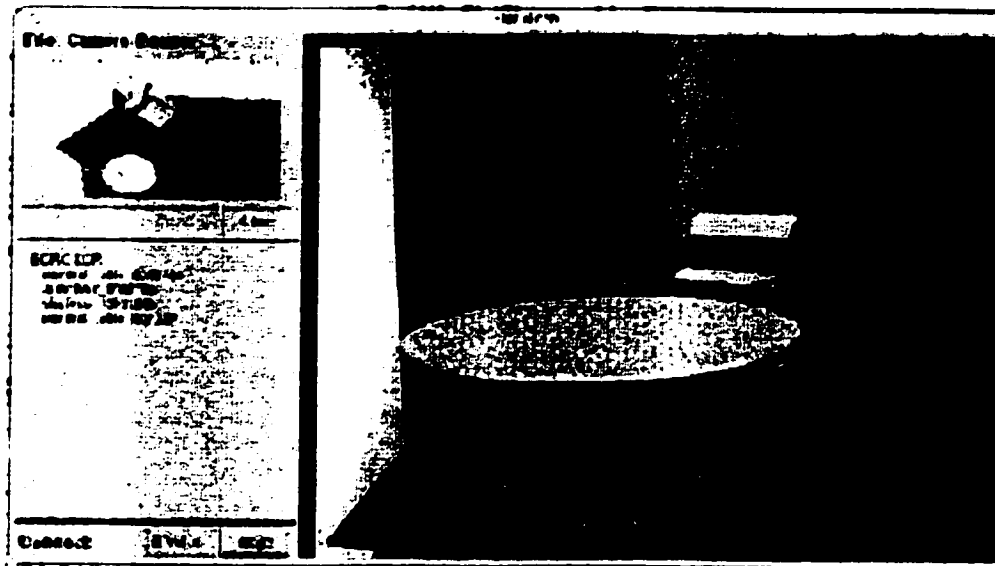
In a collaborative augmented reality system, some virtual objects which augment real objects are overlaid on the real world. Considering the distance between the users and also taking into account the distance between one or several users and the objects of the task, we can classify the collaborative AR systems into three categories, as schematically shown in figure 2-9. Because the object and/or its environment is augmented, at least one user must be in the workspace and next to the object of the task or else the system is no longer an augmented reality one, and falls into the collaborative tele-operating class.



**Figure 2-9 Three categories of collaborative AR systems: (a) remote collaboration in one augmented reality; (b) remote collaboration in augmented realities and (c) local collaboration in one augmented reality.**

### 2.6.1 Remote Collaboration in One Augmented Reality

This category includes systems in which at least one user is at the same location of the objects of the task and some users are distant. For example, in the collaborative interior design system developed at ECRC [13], an office manager works together with an interior designer on the layout of a room. On a monitor they can see a picture of the room from the viewpoint of the camera. By interacting with various manufacturers over the network, they select furniture by querying databases using a graphical paradigm. As they move the camera, they can see the furnished room from different points of view (Figure 2-10).



**Figure 2-10 a Collaborative interior design [13]**

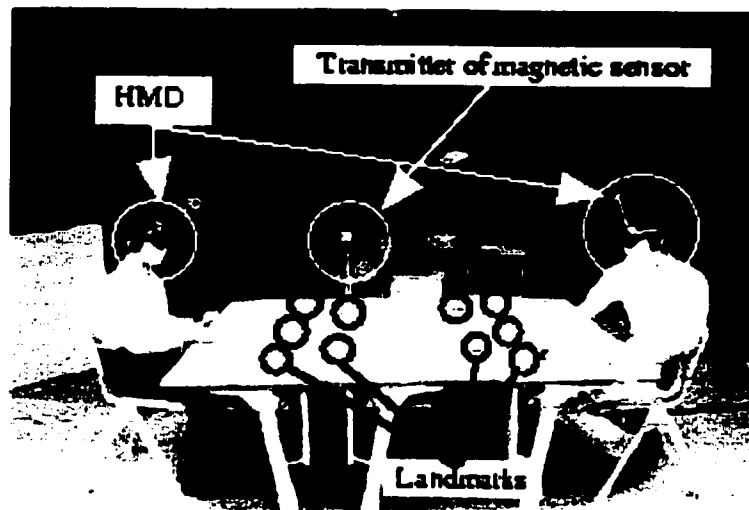
### **2.6.2 Remote Collaboration in Augmented Realities**

This category encompasses systems where there are several objects which are remotely linked together and physically present in different sites. Each user performs a part of the task on his physical object. An example is a collaborative augmented whiteboard [8].

### **2.6.3 Local Collaboration in One Augmented Reality**

This category is also called co-located augmented reality. It represents a system where all the users are positioned together with the objects of the task and the shared physical environment is augmented by information and action from the users. This kind of system allows users to see each other, as well as the 3D virtual objects in the space between them. Although they are all together (same physical environment), the key point is that their views of the augmented environments are different.

Currently most of the research on collaborative AR is focusing on this category. A good example is the AR2 Hockey system of Ohshima [9], which allows two users to play virtual air hockey against each other. Air hockey is a game in which two players hit a puck with mallets on a table and shoot it into goals. In this system, each player holds a mallet as a physical device to hit a virtual puck.



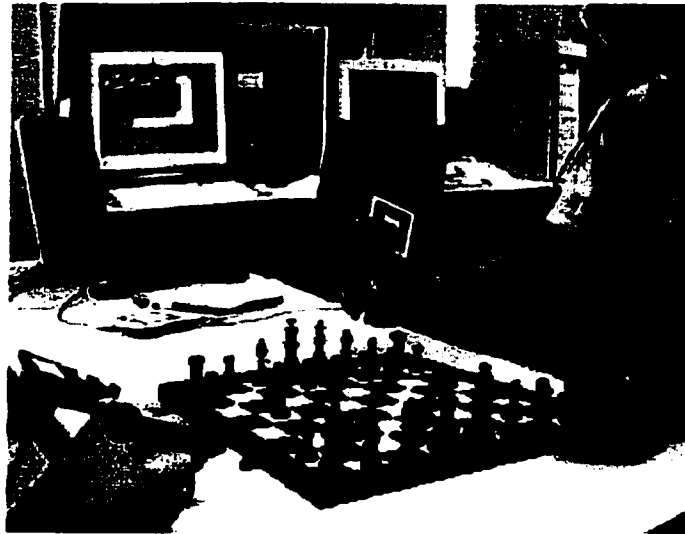
(a) Playing scene



(b) Player's view

**Figure 2-11 Playing scene of AR2 Hockey [9]**

Another example: a game of Augmented Reality Chess demonstrates a collaboration of the stationary user (left) with the mobile user (right) in Reitmayr's system [10], as shown in figure 2-12.



**Figure 2-12 a game of AR Chess as seen by the “documentation camera” user [10]**

## **2.7 Thesis Objective**

From the above discussion, MCAR holds exciting potential for on-the-job industrial training. Though some research has been done in augmented reality, and some stand-alone AR training systems and collaborative wearable systems have been developed, research on applying MCAR to industrial training application is still rare.

The objective of this thesis is to develop a collaborative augmented reality prototype to be applied to industrial training. A crucial question for the research is how to make the best use of the wearable computer which is available to create an intuitive interface for the distributed virtual environment (augmented reality) user.

### **2.7.1 Background**

The proposed research is actually a part of “Augmented Reality Interface” in the 2-year project, “Distributed Virtual Environments with Training Applications (DIVERSIONS)”, which is sponsored by National Capital Institute of Tele-Communications (NCIT).

### **2.7.2 Research Tasks**

Ms. Peiran Liu and I worked as a team in this research project, but focusing on different topics.

As designing an Augmented Reality prototype for an industrial training application is the emphasis of our research, I propose a solution with a Mobile Collaborative Augmented Reality for industrial training application and designed a prototype based on this solution. This prototype allows a group of trainers and trainees, each on geographically disparate locations, to do a training task collaboratively by sharing the workspace of a trainee who is equipped with a wearable computer. In this prototype, an intuitive interface for the distributed virtual environment (augmented reality) user and some collaboration functions, such as remote manipulation of the virtual objects and remote pointing, are provided. I also investigated the choice between “see-through” and “see-around” viewing area, based on users’ acceptance. I will give more details on this in the following chapter.

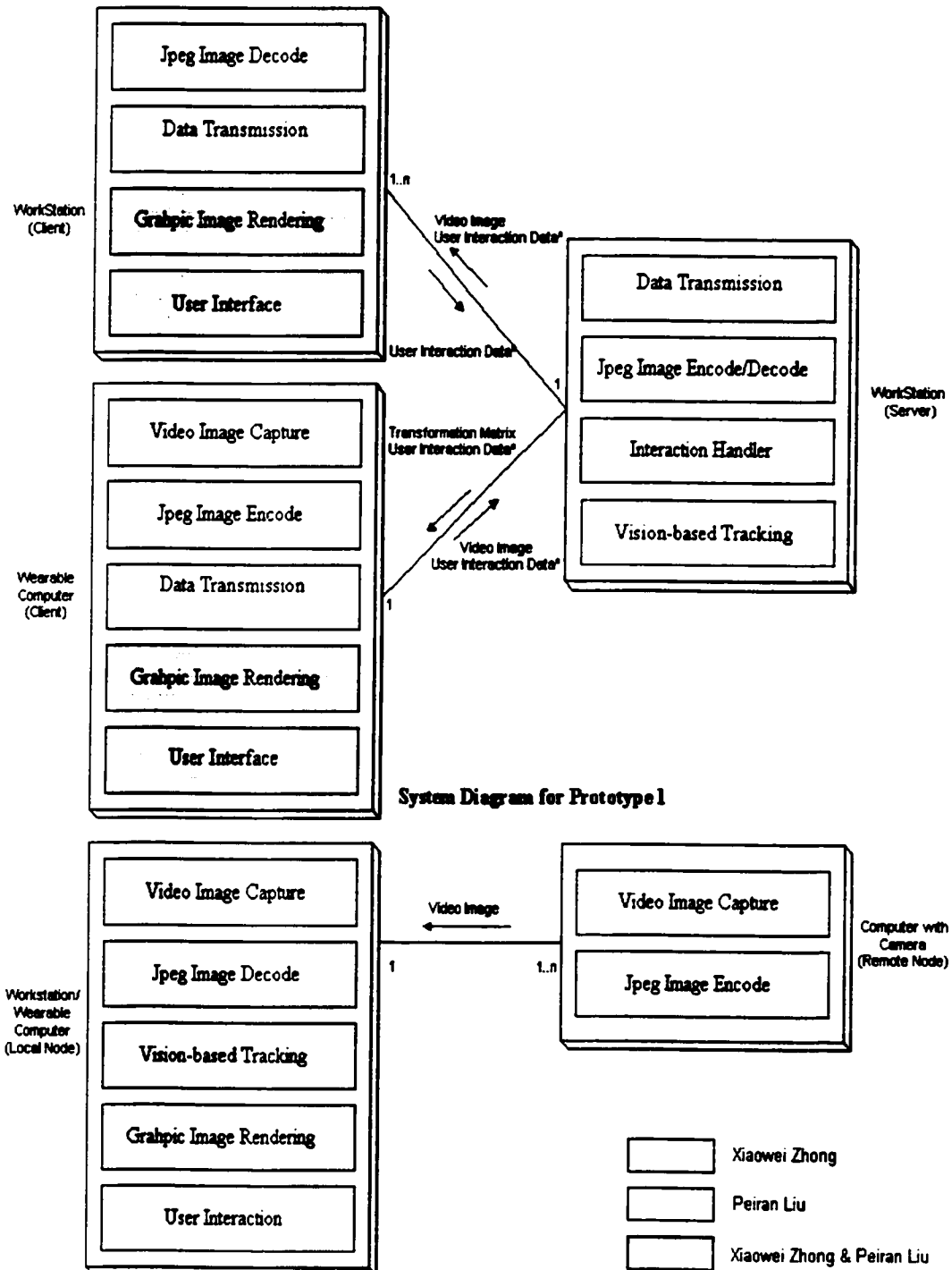
The head position and orientation is needed in the virtual environment to continually project the virtual environment into the visual field of the users, as they walk through the real world. Which virtual object should be superimposed on a specific location of the real world has to be decided when more than one virtual object has to be projected to the real world. Peiran focuses on how to recognize virtual objects and has created a Binary Square Marker Algorithm.

Since we started the project at the same time, most of the time we found that we could not carry out work based on one another’s results directly. Therefore, in order to reach our research target, two separate prototype systems were developed, but we continued to share some work.

As shown in figure 2-13, the system diagram for prototype 1 describes the system of my prototype. The system diagram for prototype 2 describes the system of Peiran’s prototype. Peiran successfully makes use of the OpenVRML library to browse the VRML model in the OpenGL browser in this project. I use her result in the designing of the *Graphic Image Rendering* module. In addition, she specifies some keys to operate a virtual object by keyboard in the local view. To retain consistency, I use her specification in my *User interface* module.

I used the head pose tracking and virtual object recognition techniques proposed by kato-billinghurst-Weghorst-Furness[6] and the Artoolkit library to design the *Vision-based Tracking* module. Peiran designed a new object identification algorithm for the

*Vision-based Tracking.* However, I didn't use her work in the module described in this thesis since she had not finished her work when I was designing this section.



\* User Interaction Data consist of command data, virtual object manipulation data and remote pointing coordinates

**Figure 2-13 Prototypes**

## **2.8 Chapter Summary**

**This chapter presented the background and motivation for this thesis. It described the typical requirements of training applications and problems with the current training systems. The concepts of augmented reality, mobile computing and CSCW were introduced and some applications which make use of these technologies were studied. MCAR, the combination of the above three technologies, was then introduced. And we studied the classifications of collaborative augmented reality systems. The chapter concluded with a presentation of the thesis objective, designing a prototype for industrial training with MCAR technology and intuitive interface.**

## **Chapter 3 Related Technologies**

An augmented reality immerses the user in a combination of the real world and the virtual world. For this immersion to be effective, the egocentric frame of reference maintained by the user's body and head must be registered with the virtual world reference. This requires that motions or changes made by the user will result in the appropriate changes in the perceived virtual world. Therefore, we use (head) tracking to get the position and orientation of the user. Since the user is looking at a virtual world, there is no natural connection between these two reference frames, and a connection must be created. The task is now to register the virtual frame of reference with what the user is seeing.

In the following section, I will introduce the display technologies of AR systems. I will then discuss the typical requirements of an AR system, and the tracking and registration techniques used in an AR system. This discussion will highlight the ways in which augmented reality is an area where multiple technologies blend together into a single system. The fields of computer vision, computer graphics and user interfaces are actively contributing to advances in augmented reality systems.

### **3.1 Display Technologies in Augmented Reality**

There are two display configurations for augmented reality systems to accomplish the combining of real and virtual. One makes use of a monitor and another one employs a see-through head-mounted display (HMD). A see-through HMD is one device used to combine the real and virtual world. HMDs have been widely used in virtual reality systems. The standard closed-view HMDs used in virtual reality applications give the user complete visual isolation from the surrounding environment. However, a see-through HMD lets the user see the real world, with virtual objects superimposed by optical or video technologies.

### 3.1.1 Monitor-based AR system

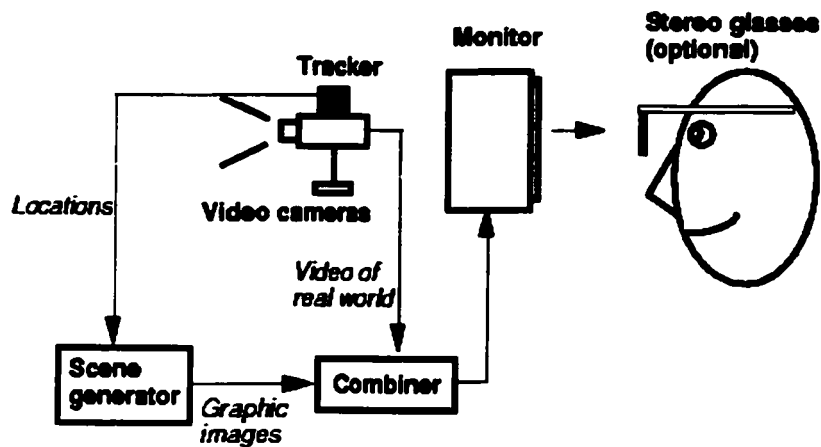


Figure 3-1 Monitor-based AR conceptual diagram [12]

Figure 3-1 shows how a monitor-based system might be built. In this case, one or two video cameras view the environment. The camera may be static or mobile. In the mobile case, the cameras might move around a tracked location via robotics or another user. The video of the real world and the graphic images generated by a scene generator are combined, just as in the video see-through HMD case, and displayed on a monitor in front of the user. The user does not wear the display device. Optionally, the images may be displayed in stereo on the monitor, which then requires the user to wear a pair of stereo glasses. However, the user of a monitor-based system has little feeling of being immersed in the environment created by the display.

### 3.1.2 Optical-based AR System

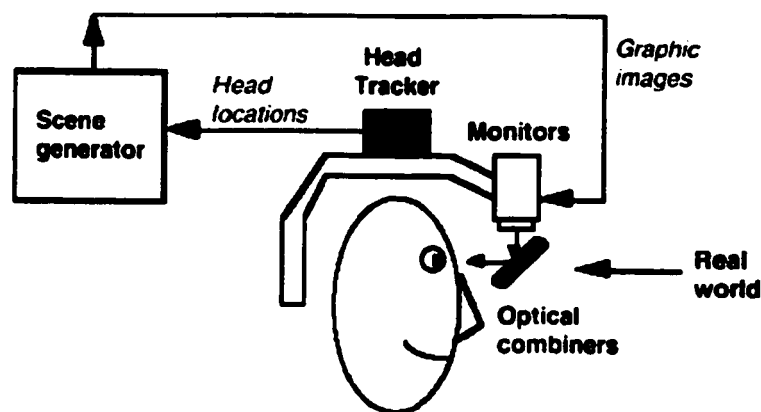
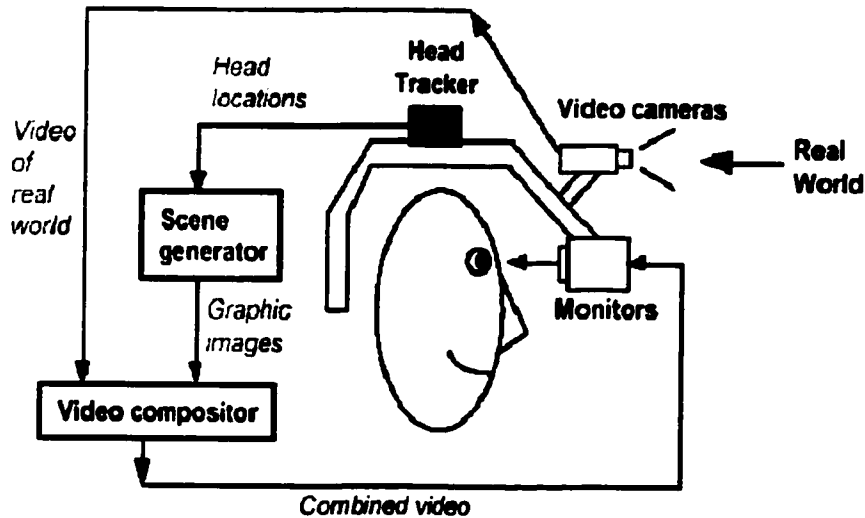


Figure 3-2 Optical see-through HMD conceptual diagram [12]

As shown in figure 3-2, an optical see-through HMD works by placing an optical combiner in front of the user's eyes. The combiner is partially transmissive and partially reflective, so that the user can look directly through it to see the real world, and see the virtual images bounced off the combiner from the head-mounted monitor as well. The optical combiner acts like a half-silvered mirror and only lets in some of the light from the real world so that it can reflect some of the light from the monitor into the user's eyes. Most existing optical see-through HMDs reduce the amount of light that the user sees from the real world so they act like a pair of sunglasses when the power is cut off.

### 3.1.3 Video-based AR System



**Figure 3-3 Video see-through HMD conceptual diagram [12]**

In contrast, a video see-through HMD works with one or two head-mounted video cameras that provide the user's view of the real world. It blends the real and virtual world by combining the video from these cameras with the graphic images created by the scene generator. The combined video is sent to the monitor in front of the user's eyes in the closed-view HMD. Figure 3-3 shows a conceptual diagram of a video see-through HMD. However, if the power is cut off, the user of a video see-through HMD is effectively blind. Therefore, an HMD in a typical wearable computer uses a mechanism similar to that of the optical see-through HMD to solve the problem. The monitor is mounted on the head and a non-transparent mirror is placed in front of the user's eyes. The mirror reflects the combined video from the monitor to the user's

eyes. Therefore the view of the user is not totally blocked. He can see the real world directly no matter whether the power is cut off or not.

### **3.1.4 Comparison of the Display Technologies**

Each of the display technologies described above has its advantages and disadvantages. They are discussed in detail by Azuma [12]. With both the monitor-based display and the video-based HMD display, which use a video camera to view the real world, there is a forced delay of up to one frame during which combined video rendering is performed. Everything the user sees is under the system's control; therefore compensation for this delay could be made by delaying the video of the real scene. With an optical see-through display the view of the real world is instantaneous, so it is not possible to compensate for system delays in other areas. On the other hand, with monitor-based and video see-through displays, a video camera is viewing the real scene. An advantage of this is that the image generated by the camera is available to the system to provide tracking information. The optical see-through display does not have this additional information. The only position information available with the display is the head tracker on the head mounted display. The head tracker can be a camera, a magnetic tracker, and so on.

Each technology has its role, and the choice of technology depends on the requirements of the application. Some monitor-based AR systems were demonstrated on tele-operation, such as the ARGOS (Augmented Reality through Graphic Overlays on Stereo video) system, which is a tool for enhancing human tele-robot interaction [15]. The collaborative design system developed at ECRC [13] is a monitor-based demonstration for the purpose of interior design.

See-through HMDs are more widely used in augmented reality applications in which a see-through HMD combines a view of the real world with computer enhancements, or augmentations, so as to make it easier to perform the task at hand. Many of the prototypes for industrial applications, such as mechanical assembly, maintenance and repair, use optical approaches, possibly because of the cost and safety issues: for example, if the power is cut off the user won't be left unable to see. In contrast, most of the prototypes for medical applications use video approaches, probably for the

flexibility in blending real and virtual and for the additional registration strategies offered.

### **3.2 Typical Requirements for an AR System**

The registration between synthetic and real objects is critical in an augmented reality system because we are very sensitive to visual misalignments. Augmented reality systems are expected to run in real-time so that a user will be able to move around freely within the scene and see a proper rendered augmented image. Therefore, one of the most important requirements for an AR system is accurate registration of the real and virtual.

Another important requirement is a reasonable update rate for generating the augmenting image. According to research by Durlach [14], delays greater than 60ms between head motion and visual feedback impair adaptation and the illusion of presence. Therefore an update rate of 10 frames per second is needed for acceptable real-time performance. A reasonable image generation rate and stereopsis are important for good depth perception. The lack of stereoscopic depth cues greatly reduces the believability of an augmented environment.

Furthermore, in most applications, the user wants to move without restriction and requires minimal constraint on his or her motion.

### **3.3 Tracking and Registration**

Accurate dynamic registration is a key issue for an AR system. In the current AR systems, most of the registration errors are due to the limitations of the tracking systems. No conventional tracker satisfies all of the typical requirements. To register the synthetic and real object, we use tracking to get the position and orientation of the viewer, which is called 6 degrees of freedom (3 degrees for position, and another 3 for orientation).

### **3.3.1 Tracking**

#### **3.3.1.1 Magnetic Tracking**

Today's magnetic trackers are subject to large amounts of error and jitter. Despite their lack of accuracy, magnetic trackers are popular because they are robust and place minimal constraints on user motion. Flock of Bird is a typical kind of magnetic tracker.



**Figure 3-4 Flock of Bird**

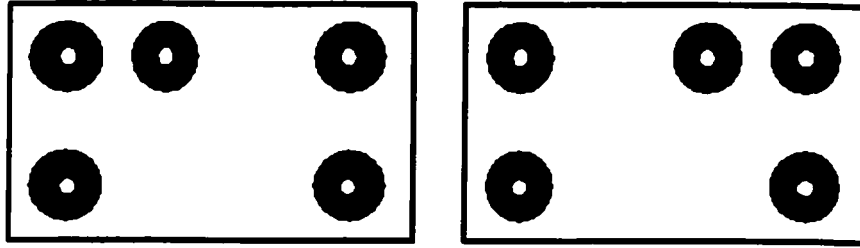
#### **3.3.1.2 Mechanical Tracking or Optical Tracking**

Some systems use mechanical or optical tracking systems. Both of these systems generally have better accuracy than magnetic trackers, but are burdensome. Mechanical systems tether the user and have a limited working volume, and the optical tracker requires four dedicated tracking cameras mounted on the user's HMD.

#### **3.3.1.3 Vision-based Camera Tracking**

Vision-based camera tracking utilizes computer vision technology. It can track the position and orientation of the viewer more accurately than the above techniques. Vision-based tracking methods are based on detecting and tracking certain features in images. These features can be lines, corners or squares, which are easily and reliably detected in the images and can be uniquely associated with features of the 3D world. For instance, in the system [4] introduced in last chapter, fiducials with a white center surrounded by a black ring are applied (figure 3-5). By offsetting the position of the

middle of the fiducials, two distinct patterns are created. The measured position of the middle feature point determines which of the patterns is recognized.



**Figure 3-5 Fiducials with a white center surrounded by a black ring**

Magnetic and ultrasonic trackers used in VR are constrained by their inaccuracy and limited volume. Therefore, they do not provide accurate and portable solutions to overlaying a virtual image on a video image in AR. More practical are vision-based AR systems that capture images by means of a video camera to track the camera's position and orientation relative to the objects in the real world. Camera tracking has been extensively investigated in the field of computer vision and has demonstrated potential to provide the accurate registration data needed by AR systems. Therefore, we are going to use the vision-based camera tracking technique in our prototype. We will look at this in detail in the following section.

### **3.3.2 Registration**

The objects in the real and virtual world must be properly aligned with respect to each other, or the illusion that the worlds co-exist will be compromised. Registration errors are difficult to adequately control because of the high accuracy requirements and the numerous sources of error. The sources of errors can be divided into two types:

One is noise in the system. The position and pose of the camera with respect to the real scene must be sensed. Any noise in this measurement has the potential to be exhibited as an error in the registration of the virtual image with the image of the real scene. Fluctuations of values while the system is running will cause jittering in the viewed image. As mentioned previously, an AR system is very sensitive to visual errors, which in this case would be the perception that the virtual object is not stationary in the real scene or is incorrectly positioned.

Another type of source of errors is system delay, which is the time difference between the moment that the tracking system measures the position and the orientation of the viewpoint to the moment when the generated images corresponding to that position and orientation appear in the displays. The delays could take the form of calculation of the camera position, communication delays, the time it takes the scene generator to draw the appropriate images in the frame buffers, and the scan-out time from the frame buffer to the displays. If there are delays then the augmented objects will tend to lag behind motion in the real scene. The system design should minimize the delays to keep overall system delay within the requirements for real-time performance.

### 3.4 Vision-based AR System

In a video see-through AR system, video images of the user's view are always available. Using those images to track the camera's position and orientation should be a reasonable approach, and camera tracking has been extensively investigated in the field of computer vision. However, recovering 3D information from 2D images is not an easy task. This also addresses the problem of accurately tracking the 3D motion of the camera in a known 3D environment and dynamically estimating the 3D camera location.

#### 3.4.1 Co-ordinate Systems for Augmented Reality

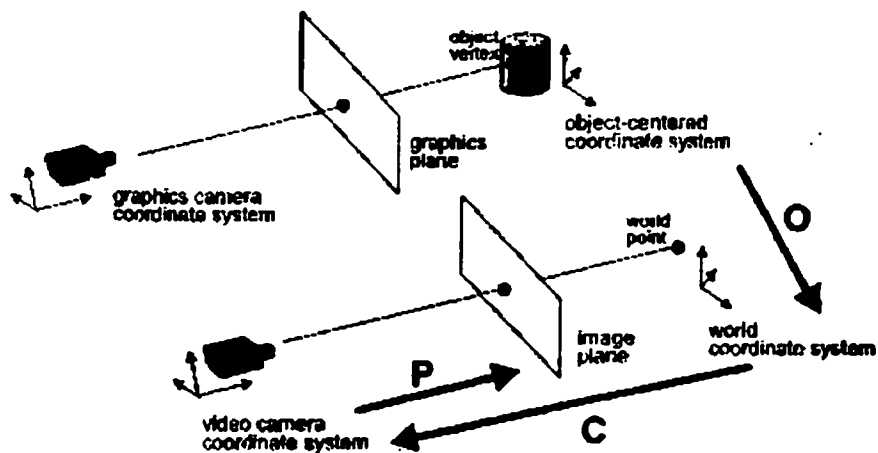


Figure 3-6 Coordinate systems for augmented reality [16]

Figure 3-6 shows the coordinate system of a typical AR system. The scene is viewed via an imaging device, which in this case is depicted as a video camera. Therefore, an AR system essentially has two cameras: a real one, which generates video of the real environment; and a virtual one, which generates the 3D graphics to be merged with the live video stream. The video camera performs a perspective projection of the 3D world onto a 2D image plane. The intrinsic (focal length and lens distortion) and extrinsic (position and pose) parameters of the device determine exactly what is projected onto its image plane. The generation of the virtual image is done with a standard computer graphics system. The virtual objects are modeled in an object reference frame. The graphics system requires information about the imaging of the real scene so that it can correctly render these objects. This data will control the synthetic that is used to generate the image of the virtual objects. This image is then merged with the image of the real scene to form the augmented reality image.

### 3.4.2 Camera Viewing Geometry

Accurate projection of a virtual object requires knowing precisely the combined effect of the object-to-world, world-to-camera, and camera-to-image transformations. In homogeneous coordinates, this projection is described by the equation

$$\begin{bmatrix} u \\ v \\ h \end{bmatrix} = P_{3 \times 4} C_{4 \times 4} O_{4 \times 4} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

where  $[x \ y \ z]$  is a point on the virtual object,  $[u \ v \ h]^T$  is its projection in the graphics image plane,  $O_{4 \times 4}$  is the homogeneous transformation from virtual object coordinates to world coordinates,  $C_{4 \times 4}$  is the transformation from world coordinates to graphics camera coordinates, and  $P_{3 \times 4}$  describes the projection operation of the synthetic graphics camera onto the graphics image plane.

The video imaging and graphic rendering described above are relatively straightforward. The research activities in augmented reality center around two aspects of the problem. One is to develop methods to register the two distinct sets of images and keep them registered in real time. Some new work in this area has started

to make use of computer vision techniques. The second direction of research is in display technology for merging the two images.

### 3.4.3 Camera Calibration

Camera calibration is the process of calculating intrinsic and extrinsic camera parameters. Camera parameters are the numbers describing a particular camera configuration. The intrinsic camera parameters are those that specify the properties of the camera characteristics; these parameters include the focal length, that is, the distance between the camera lens and the image plane; the location of the image center in pixel coordinates; the effective pixel size; and the radial distortion coefficient of the lens. The extrinsic camera parameters describe the spatial relationship between the camera and the world; they are the rotation matrix and translation vectors specifying the transformation between the camera and world reference frames.

A vision-based AR system also has two cameras, a real one and a virtual one. The real camera generates a video stream of the user's environment; the virtual one generates the 3D graphics to be merged with the live video stream. In order to align the virtual objects to the real ones properly, these two cameras must have the same internal and external parameters. To achieve this, an initial calibration of the real camera and a dynamic update of its external parameters are required.

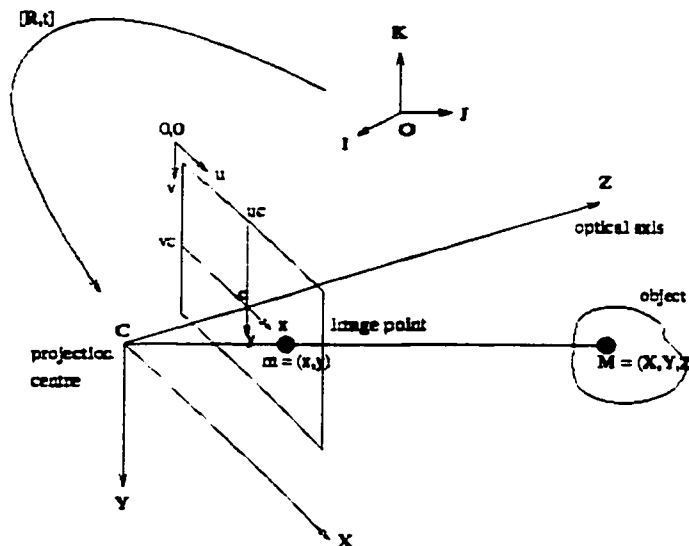


Figure 3-7 Coordinate systems in Camera Calibration

Figure 3-7 shows the coordinate systems involved in camera calibration. A point  $M$  on an object will be imaged at some point  $m$  in the image plane. In a *standard coordinate system* involving the camera, the Z-axis is along the optical axis. The transformation from  $M$  to  $m$  is implemented by a perspective projection matrix. The elements of the matrix are internal parameters. But in general, the 3D world coordinates are not *standard coordinate systems*, therefore we have to include a change from the general frame to the standard coordinate system. The elements of the transformation matrix are external parameters.

As discussed above, the extrinsic and intrinsic parameters are the result of calibration. If a continuous update of extrinsic parameters is regarded as tracking, registration is the process of acquiring the object's pose through known intrinsic parameters and computed extrinsic parameters.

### **3.5 Chapter Summary**

In order to better understand the following chapters, this chapter introduced the related technologies used in developing an augmented reality system. The display choices of an AR system, the tracking and registration technologies were discussed.

## **Chapter 4 Design of a Training System**

This chapter discusses the design of a training system using MCAR technology. Section 4.1 defines the design goals of the system. Section 4.2 describes the hardware configuration. Section 4.3 presents the challenges met during the designing period. Section 4.4 discusses the design principles and the decisions made for this system. In section 4.5, the overall architecture of the system is described and the distributed mechanism is also introduced. In section 4.6, the chapter summary is presented.

### **4.1 Design Goals**

#### **4.1.1 The Task: ATM Switch Maintenance**

The purpose of the thesis is to design and develop a prototype for an industrial training application based on mobile collaborative augmented reality. Imagine some trainees in different locations are trained to repair an ATM switch. The task consists of removing a switchboard from an ATM switch and installing a chip on the switchboard. As an ATM switch is an expensive piece of equipment, only one trainee has access to it. The other trainees would like to participate in the training exercise to get some experience. In addition to trainees, there are some trainers in remote locations who are knowledgeable experts in the domain.

During the training exercise, the trainee who is equipped with the ATM switch may need to ask a trainer for advice. For example, the trainee may contact the trainer to ask for general advice (“How do I...?”) or for particular pieces of information (“which piece of the switchboard should I work with?”; “Should I install the chip this way or that way?”). On the other hand, the trainer may give the trainees some directions or point out mistakes.

#### **4.1.2 Problems with the Current Practice**

Below is a list of some of the problems with the current practice of training in ATM switch maintenance:

**Costly:** the ATM switch is an expensive device. Not all of the trainees have had access to this device. Furthermore, errors in installation could lead to damage to the

chip or the pins of the switchboard. Therefore, it could be costly to do a real training exercise.

**Inefficiencies:** Since the remote trainer is not always available, the trainees often have to interrupt the current training exercise until they have an opportunity to talk to a trainer in person.

**Insufficient Voice-only conversations:** When a trainee needs to ask the remote trainer for advice, phone conversations can be helpful in asking for a specific type of information, but they are not effective in situations where an expert has to provide a trainee with step-by-step directions, depending on the situation. Sometimes video is indispensable in interactive conversation. Furthermore, it is difficult for a trainer to instruct a trainee to operate a specific device, especially in a case where the trainer has to tell the trainee to work on a specific piece of chip among dozens of similar chips on a switch card. Therefore, effective means are needed to establish a common understanding during computer-mediated conversation.

**Informative Supporting is expected:** A problem arises when trainees would like to be shown how to do the job but the trainers are far away or not available. It is not informative for a remote trainer to tell a trainee via voice conversation to “pull the switchboard in this direction in this way”.

**Less hands-on experience:** The trainees who are not equipped with ATM switches and other facilities are not able to carry out an effective training exercise.

## **4.2 MCAR: an Effective Solution**

MCAR is an effective method of remedying the shortcomings with the current practices, described above. Wearable computing enables the local trainee to get information while she/he is walking around in the working environment; while augmented reality can be used to overlay supporting information on his/her workspace, for instance, 3D graphics showing how to remove the switchboard, etc. CSCW enables remote trainers and trainees to cooperate in the training exercise by sharing the view of the local trainee through real-time video.

### **4.2.1 Virtual Objects**

As the devices which the trainees learn to use or repair are expensive and misuse can lead to serious and costly damage, some virtual objects are used as alternatives to the real objects. With the AR technique, 3D models of the devices are superimposed on the real world scene, just like the real devices in the physical space. The users are then able to manipulate the virtual devices in the training exercise. This helps to reduce the cost of training.

### **4.2.2 Collaboration Functions**

To support and encourage cooperation, cooperative applications must allow users to be aware of the activities of others. The purpose of a cooperative multi-user interface is to establish and maintain a common context, allowing the activities of one user to be reflected on other users' screens. This common context is achieved by sharing application information. The real-time presentation and manipulation of shared information is the main function of cooperative multi-user interfaces.

From this observation I came up with some possible collaboration functions with which multiple trainees and trainers are able to carry out the collaboration work effectively through a wearable system:

#### **4.2.2.1 Audio Conference**

This allows users to engage in real-time audio conversation during the training session. It is also a good method of ensuring awareness with the remote users.

#### **4.2.2.2 Remote Presence**

If the trainer is able to see what a remote trainee sees, as if she/he were physically present, she/he can give the trainee step-by-step instructions. This is called "remote presence" and it can be achieved by transmitting real-time video of the trainee's view to the remote trainer, over a wireless network.

#### **4.2.2.3 Remote Manipulation**

Remote manipulation refers to a user's ability to manipulate objects in another user's physical environment. Remote manipulation creates a heightened sense of co-

presence. The remote participant can use voice conversation to request the local user to manipulate the physical objects in the physical world, but they cannot operate the objects by themselves. The use of virtual objects in the shared space allows the remote trainees to manipulate the virtual objects by themselves, thereby gaining some value training experience.

#### **4.2.2.4 Remote Pointing**

The ability to control a remote cursor enables users to point at objects in other users' view. Such objects can either be virtual objects (a wire in a wiring diagram) or real objects captured by the camera of a wearable computer. Remote pointing can increase the effectiveness of verbal communications by directing the participants' attention.

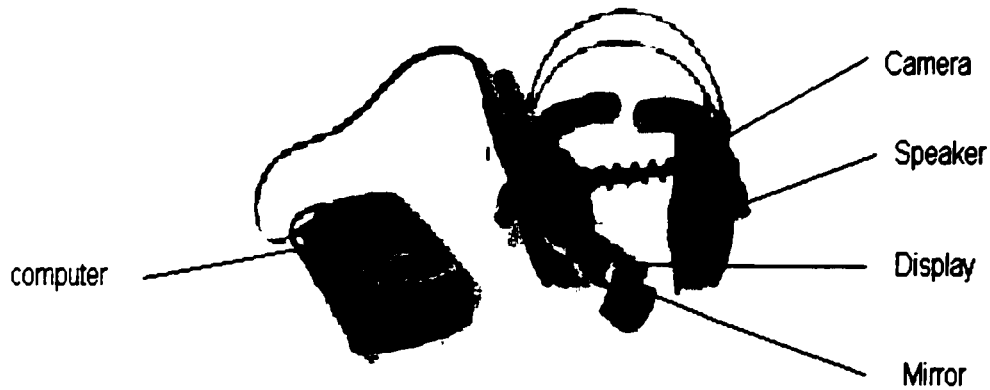
#### **4.2.2.5 Image Freezing**

As discussed in the last chapter, to ensure real-time performance for an augmented reality application, the view update rate should be no less than 10 frames/second. When one remote user wants to direct the attention of the other users to a specific area in the shared view, using a remote pointer, s/he will have difficulty in capturing the area when the wearable computer user is moving her/his head. In order to compensate for the wearable user's head movements, the remote trainer is provided with an image-freezing feature.

### **4.3 Hardware Configuration**

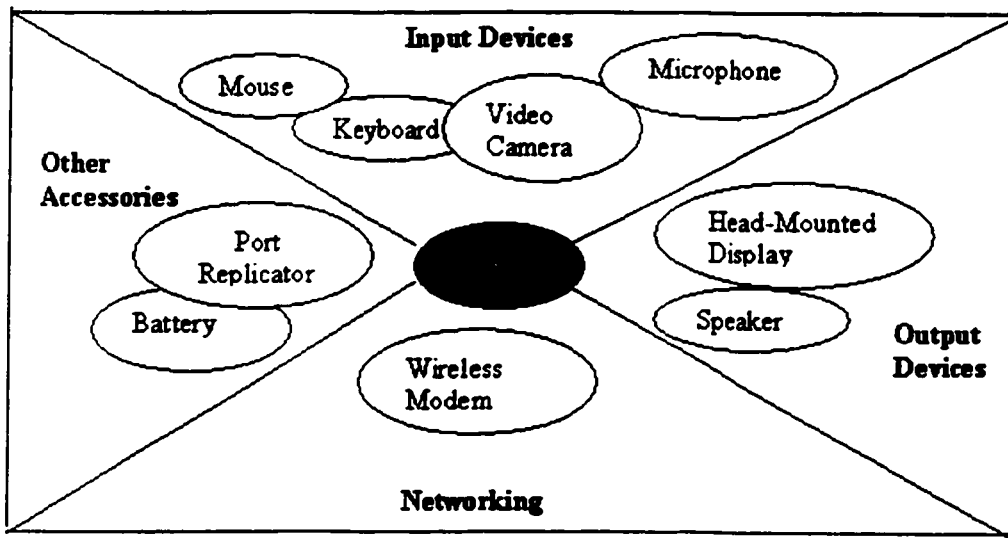
#### **4.3.1 Wearable Computer**

Currently our research lab is equipped with a commercial off-the-shelf product, the Xybernaut Mobile Assistant MA IV wearable computer. Figure 4-1 shows this wearable computer.



**Figure 4-1 A Wearable Computer with some accessories**

The wearable computer consists of Intel Pentium MMX CPU 233, 160 MB RAM and a hard drive with 4.3 GB capacities. Figure 4-2 shows the categories of accessories of the wearable computer.



**Figure 4-2 Wearable computer peripherals**

**(1) Input Devices**

A mouse and keyboard allow the user to input data, and a microphone and another output device speaker can enable the user to engage in voice conversation. A 16 million colors, 640 by 480 pixels video camera is attached to the head-mounted display to enable real-time or still video.

**(2) Output Devices**

The size of the head-worn display (640x480) is 18.7cmx6.3cmx11.7cm. It has two interchangeable mirrors: one is completely opaque for blocking the display and part of the viewing area, while the other is semitransparent for AR overlay and does not block any viewing area. However, the display mirror is very fragile and not suitable for real workplace environments. The display is not well suited for use in bright outdoor situations, as the contrast is too low for application in normal sunlight. The display has no binocular stereo capability.

### **(3) Networking Devices**

The wearable computer is connected to a Local Area Network (LAN) infrastructure that includes Lucent Technologies WavePOINT-II access points through a waveLAN/IEEE Turbo 11Mb PC card. This PC card provides a theoretical bandwidth of 11 Mbit/s. The realistic throughput is less than that, however.

### **(4) Other Accessories**

The full-function port replicator is flexible so that the user can connect to external devices easily. It has some ports, including a serial port, parallel port and others that enable the wearable computer to connect to the external 3.5 inch floppy drive, an SVGA interface for external display, a PS/2 keyboard and PS/2 mouse, and so on.

To power the computer, an easy-to-replace, rechargeable, nickel metal hydride battery pack is developed and carried in a backpack or vest in a real-work task.

From the above introduction we can understand that the model of wearable computer has all the functionality and connectivity of a fully-function networked desktop PC. The computer currently runs the Windows 98 operating system.

## **4.3.2 Workstations**

Some Pentium III workstations equipped with GetForce2Go video chip are used by remote trainees and trainers. They are running on MS Windows 2000.

## **4.4 Design Challenges**

There are some challenges in the design and implementation of the system.

## **4.4.1 Limitations of the Wearable Computer**

There are some limitations to the wearable computer. Even though the upgrade to the wearable computer is underway, the new hardware equipment is not available in the design and experiment process. Therefore, the design of the system using this solution will be based on the equipment that we have at present.

### **4.4.1.1 Limited Computing Resources**

Due to limitations in size and weight, and subsequent limitations in battery consumption, a typical wearable computer has limited CPU, less storage and memory than an ordinary PC. However, augmented reality applications require higher computing ability for head tracking and graphic image rendering. The local computing resources of the wearable computer are not enough.

A wearable computer can overcome resource limitations by accessing remote resources over a wireless network. For example, a wearable computer can access data stored on a remote file or database server or it can run computer intensive jobs on computer servers.

### **4.4.1.2 Missing Infrastructure for Building Mobile Applications**

The mobility of the wearable host and the special characteristics of the wireless link (limited bandwidth, latency, low reliability) increase the complexity of the software. Yet, there is still a lack of high-level support for building distributed mobile applications.

### **4.4.1.3 Lack of 3D Support in Hardware**

With regard to the CPU power and 3D capability problems with the wearable computer, I spoke to Mr. Torsten Bergander, an engineer at Xybernaut Corporation, who was involved in the 3Dwm/Xybernaut joint research project. To improve the CPU power of our wearable computer, he recommended that we upgrade it from the MA IV series to the 400MHz MA TC series. To improve the 3D capability he suggested that we add a good 3D-accelerator card with a 1-slot CardBus-to-PCI Expansion System to the wearable computer. A demonstration has already been done with 19 frames/seconds on the TC series and 10 frames/second on the IV series with the accelerated 3D hardware.

#### **4.4.2 Real-time Performance**

The computational tasks that operate on head tracking, virtual object recognition and graphic image rendering can be time consuming and computationally intensive, especially if the data being explored are complex and huge. This invariably leads to very low frame rates. As we discussed in the former section, delays greater than 60ms between head motion and visual feedback impair adaptation and the illusion of presence. A collaborative augmented reality system needs to provide the minimum frame rate (10 frame/sec), while collaborating with other users in parallel.

#### **4.4.3 Sensitivity to Errors**

Since the information has to overlap with the real world, the smallest errors in tracking information are detected by the eye. Any mismatch between augmented objects and real objects can be uncomfortable and also result in incorrect information being given to the user.

#### **4.4.4 2D Device Limitations**

It is a big challenge to effectively choose a virtual object and manipulate it when multiple virtual objects are in the view. A standard mouse comes with the wearable computer is a 2D device, which is difficult to use to manipulate the virtual object in the 3D world. Typically, the user is walking around and the view is updated 10 frames per second, which makes it difficult for the user to capture the virtual object in the view.

### **4.5 Design Principles and Decisions**

Some issues have to be addressed in designing an effective training system: a suitable display technique for the wearable computer has to be decided upon; an accurate and effective tracking mechanism has to be provided; an effective media communication and presentation mechanism including audio transmission method and video compression technique must be selected; and a mechanism for proper data sharing is necessary when clients and servers reside on heterogeneous platforms. I will discuss the issues and the decisions taken in detail in the rest of the section.

## **4.5.1 Display Technique**

### **4.5.1.1 Principles**

As discussed in Chapter 3, both optical-based and video-based display techniques provide “see-through” characteristics for the wearable computer. An optical see-through HMD employs a mirror beam splitter to reflect images onto small opaque screens. The real world can be seen through the half-silvered mirrors, as the graphics are overlaid and rendered on the display.

In a video-based head-mounted system, cameras on the HMD view the real world and the recorded video stream is merged with the computer graphics. The mixed image is projected onto small monitors inside the helmet.

A benefit of the optical see-through display is that it enables users to view the real scene directly, maintaining the full fidelity of their vision. In addition, these tend to be lower in power than video see-through HMDs, since it is not necessary to render the video image of the real world, therefore they are more portable. Theoretically, an optical-based system is more suitable for fieldwork applications.

On the other hand, the video approach is almost problem-free, since the real world scene comes from the real-time video. With this approach, the real world scene and the virtual scene are seamlessly merged together by the system to provide a composed scene. All the information, both real and synthetic, is digitized, and is therefore more realistic visually and also easier to control.

### **4.5.1.2 Practice and Decision**

Unfortunately, during testing we found that with an optical display of our wearable computer the computer graphic overlays take on a sort of ghostly appearance because they are reflections off a mirror. We could not make heads and tails of it. It is not very effective in cases where both environments are rich or complicated.

However, with the video-based display technique, even though the mirror is completely opaque for blocking the display, the user can still have a direct view of the surrounding environment since the mirror only blocks part of his view. In addition, the quality is much better than with an optical-based display. Furthermore, we implemented the system and tested it with the workstations most of the time. If the

system is developed with a video-based HMD technique, this doesn't affect the research objective of this thesis. Therefore, we chose to implement the system with the video-based display technique.

## **4.5.2 Tracking Mechanism**

For an AR application a good tracking mechanism has to be provided to decide where and which virtual objects should be superimposed on the real world.

### **4.5.2.1 Requirements**

An AR application for industrial training is potentially used to boost fieldwork performance. It has some basic requirements:

- **Minimum range restriction**

The trainee may need to walk around with minimum range restriction, when s/he is doing the fieldwork.

- **Accuracy**

The graphics have to be overlaid accurately on top of the real-world objects.

- **Flexibility**

There could be some virtual objects which need to be overlaid in the view. Providing a mechanism to recognize different virtual objects and setting up the augmented reality environment with flexibility is very important.

### **4.5.2.2 Decision**

As discussed in chapter 3, magnetic and ultrasonic trackers commonly used in VR do not provide the accuracy and portability required in AR. They do not provide a good enough mechanism for recognizing virtual objects. Computer vision, on the other hand, can potentially recognize and locate objects in the environment by putting some features in the view and measuring their locations and tracking them over time, as the user moves his or her head. It is also possible to provide accurate data at long ranges. By using head-mounted cameras and wearable computers, a truly portable system could be developed. Therefore, we decided to make use of the computer vision tracking technique to locate and recognize virtual objects.

### 4.5.3 Distributed Visualization Solution

In this system, the visualization endpoints that allow users to participate in the collaborative augmented reality workspace include:

- A wearable computer worn by a trainee in the real-world task;
- One or more desk-bound workstations used by the trainer and other trainees in the remote location.

All of these visualization end points are connected via a network. They have to share the view while carrying out collaboration work. The collaboration data consist of an object manipulation stream, pointer stream and command stream.

The architecture of the Collaborative Augmented Reality (CAR) system determines many aspects of the implementation. There are two kinds of distributed visualization architectures.

#### 4.5.3.1 Centralized Architectures

In a centralized (or client-server) architecture, a central server program handles all user input and display output events, which are routed via clients [17]. Local workstations act as graphical terminals and window servers. The master-slave architecture is a variant in which one client is merged with the server and all other nodes run as clients.

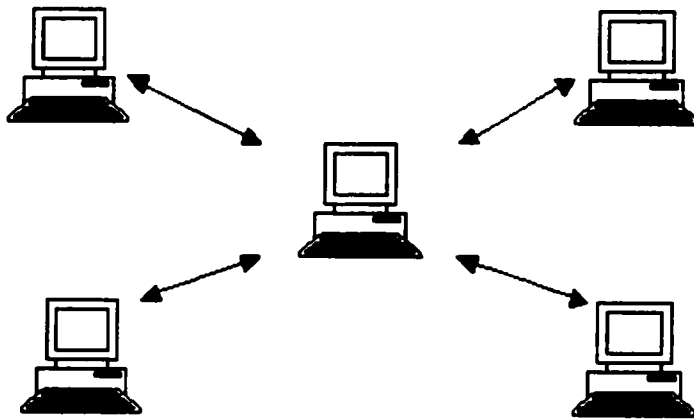
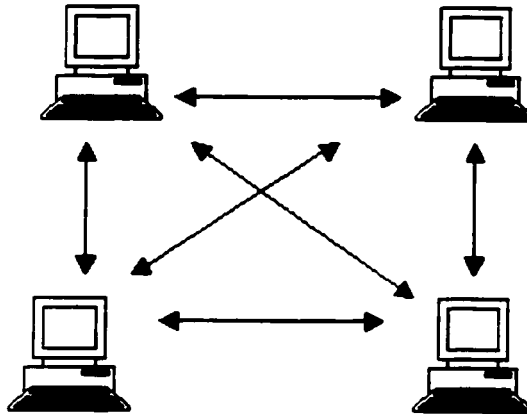


Figure 4-3 Centralized architectures

### 4.5.3.2 Replicated Architectures

Replicated architectures maintain exact copies or replicas of the data at each workstation. To maintain consistency, each replica handles screen management and feedback locally and broadcasts any change to the application data to all other replicas.



**Figure 4-4 Replicated architectures**

### 4.5.3.3 Centralized Architecture vs. Replicated Architecture

Both centralized and replicated architecture offer benefits and limitations. Table 4-1 summarizes the characteristics of different distributed-system architectures.

	Centralized architecture	Replicated architecture
Advantages	<p>Easy to implement</p> <p>Easy to add/remove nodes</p> <p>Easy to keep consistent</p>	<p>Easy to provide different levels of sharing</p> <p>Easy to provide rapid feedback</p>
Disadvantages	<p>Hard to provide different levels of sharing</p> <p>Hard to provide rapid feedback</p>	<p>Hard to implement</p> <p>Hard to add/remove nodes</p> <p>Hard to keep consistent</p>

**Table 4-1 Centralized architecture vs. replicated architecture**

The primary advantage of the client-server approach is simplicity: the application and all data are held centrally, simplifying access management and data consistency. It is relatively easy to support presentation-level sharing, because the server can replicate display directives to all clients. Because the sharing policy is embedded in the central server and is not visible to the clients, it is difficult to provide different levels of sharing with this architecture. The centralized architecture is also vulnerable to delayed feedback and failure of the central node because all events must travel over a network.

The major advantage of the replicated approach is that it supports different views easily. End-user interface tailoring is relatively easy to provide, as each replica can adapt its sharing policy to the user's preferences. With this architecture, the end-user can provide rapid feedback to other users.

The drawback of a replicated architecture is that it makes the management of shared information more complicated. Simultaneous updates that conflict with each other can lead to inconsistencies. In contrast, a centralized architecture is easier to build.

A further problem occurs when users join a group session already in progress. This dynamic registration is straightforward with a centralized approach, since new clients need only contact the central server. The server can then bring the new client up to date. With a replicated approach, however, a new replica must contact all other replicas to tell them that it needs to receive updates. This means that new replicas must know or be able to learn the locations of all other replicas.

#### **4.5.3.4 Decision**

Our objective is to develop an architecture that supports multi-user interfaces. It should be easy to add nodes to the architecture and maintain consistency among displays. Therefore, the infrastructure of the system is built in the form of a client-server architecture, in which the wearable computer and workstation works as a client to handle the user's input event and render the graphic images while the server provides an information sharing mechanism. It distributes the sharing data to clients and maintains consistency between them.

The process of vision-based tracking has high requirements with regard to the computing ability of the machine; therefore we allocated this process to the server for the following reasons:

- Reduce the workload of the wearable computer, which has lower battery durability and CPU speed compared to the PC workstation;
- Centralize the database for vision-based tracking so that each client is not necessary to keep a copy of it.

With this architecture the system maintains good consistency and has good extension capability. The system allows clients to join at different times.

#### **4.5.4 Media Communication and Presentation**

In our system, the video stream and audio stream are required to be transmitted among clients. Since the data coming from the capture devices in our system lack compression, the amount of original data is too large to send across the network where data bandwidth is constrained. Therefore, the captured data have to be compressed before being sent to the network. A software encoder/decoder is used for this purpose.

##### **4.5.4.1 Quality of Media Stream Presentation**

In a multimedia communication application, the quality of a media stream presentation often depends on several factors:

- The compression scheme used
- The processing capability of the playback system
- The bandwidth available for media streams acquired over the network

Higher-quality media streams are often achieved by using more data (e.g. bits per pixel). Higher-quality compression schemes usually result in a better quality media stream but also produce more data as well. Therefore, they require more powerful playback systems and higher bandwidth to process and transmit such media streams. Unfortunately, the overall network bandwidth is limited and often has to be shared. A higher-quality compression scheme used on the source-side may cause more data

packet loss on the receiver-side if the network is congested. As a result, the receiver may not present the expected stream quality, and the audio may be clipped or the video jerky, or other failures may occur.

#### **4.5.4.2 Audio Transmission**

Audio and video streams may be transmitted separately or interleaved and transmitted as one stream. Since media streams have different creation, transmission and playback characteristics, they also have different requirements with regard to quality of service. The audio and video streams will use separate multicast channels. The audio capturing module allows participants to enter into an audio-conferencing session. The module is based on the Microsoft NetMeeting SDK.

#### **4.5.4.3 Video Compression Standards**

Researchers have developed many techniques for video compression. Four important video (and audio) compression standards have been proposed in the past few years:

- *JPEG* for still-image compression
- CCITT (now ITU-TS) *H.261* for videophone and teleconference applications, at a bit rate of multiples of 64 kbps
- *MPEG* for motion video and associated audio compression
- ITU-TS *H.263* for videophone applications at a bit rate below 64 Kbps.

H.261 and H.263 are optimized for video compression of scenes where there is not a lot of action, with which the interframe and intraframe redundancy can be greatly exploited. They are generally used for videoconferencing applications at a low bit rate network.

MPEG is the coding of combined video and audio information. MPEG-1 specifies a decoder and representation for the retrieval of full-motion video from storage. MPEG-2 is designed to address broadcast TV sample rates. MPEG-4 is an audio-visual coding standard for very-low-bit-rate channels.

JPEG is able to compress still-images to a ratio of 10:1 and 50:1 [18], without visibly affecting image quality. JPEG for full-motion video works independently from frame to frame, and therefore does not consider inter-frame redundancies.

#### **4.5.4.4 Decision**

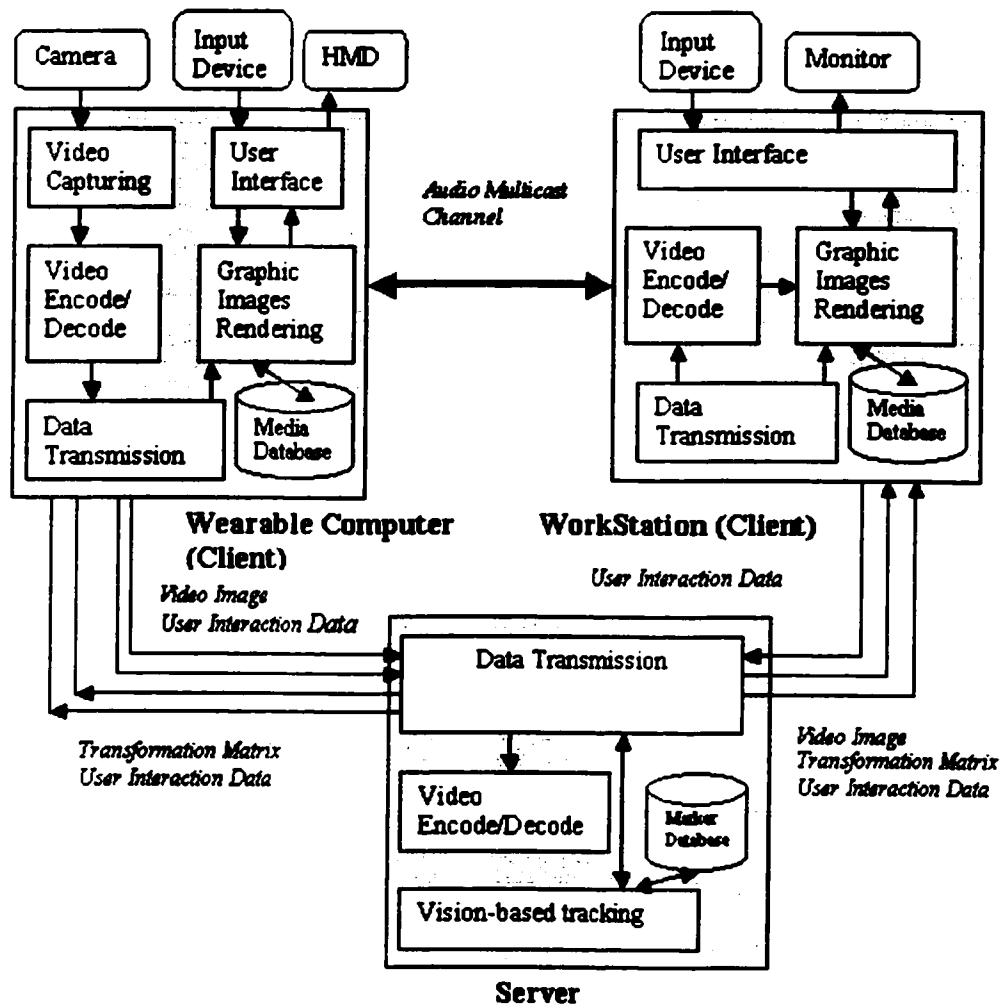
Since high quality video images are needed for accurate tracking and registration, the JPEG compression technique was chosen for video compression/decompression in our system.

#### **4.5.5 Data Sharing Mechanism**

A mechanism for effective data sharing is necessary between clients and server when they reside on heterogeneous computing platforms. Furthermore, the data should be optimized before transmission. Therefore, an effective data transmission mechanism and data packing routine are provided.

### **4.6 Overall Architecture**

As shown in Figure 4-5, a typical architecture includes three major components: a wearable computer, a server and one or more workstations. They are connected via a network.



**Figure 4-5 Overall Architecture of a Collaborative Augmented Reality System**

The client runs on a *wearable computer* or *workstation*. It consists of *video capturing*, *video encode/ decode*, *user interface* and *data transmission*. However, the *workstation* usually does not contain *video capturing*.

As shown in this figure, the camera mounted on the *wearable computer* captures live video through the *video capturing* module. The video image is encoded by the *video encode/decode* module and transmitted to the *server* through the *data transmission* module. The *server* receives the data from the *data transmission* module. The encoded image is decoded by the *video encode/decode* and analyzed by the *vision-based tracking* module to calculate the *transformation matrix* of the virtual objects relative to the video camera. Some patterns in the image are matched against pattern templates

in the marker database to recognize the virtual object's identification. The marker's identification and camera transformation matrix are then sent back to the *wearable computer*. The *wearable computer* extracts the 3D models from the *media database* according to the virtual object's identification, and generates the graphic image and renders it via the *graphic images rendering* module, together with the captured video image.

The *server* also sends the encoded video image, virtual object's identification and camera transformation matrix to the *workstation*. The *workstation* receives the data from the *data transmission* module and uses the *video encode/decode* module to decode the video image. The 3D model is extracted from the media database based on the virtual object's identification and rendered by the *graphic images rendering* module, together with the decoded video image.

This architecture has the following benefits:

- **Distributed Computing**

The system is based on a client/server architecture. It allows each component to share the overall workload. Furthermore, it is good for different-level sharing. As a result, it improved the overall system performance.

- **Problem Decomposition**

The architecture decomposes the complex system into relatively simple modules that are easy to construct and implement.

- **Easy Deployment**

The separate components benefit from easy deployment.

- **Hiding Computing complexity from the client**

Locating the vision-based tracking process in the server hides the computational complexity from the clients so that the clients won't overload.

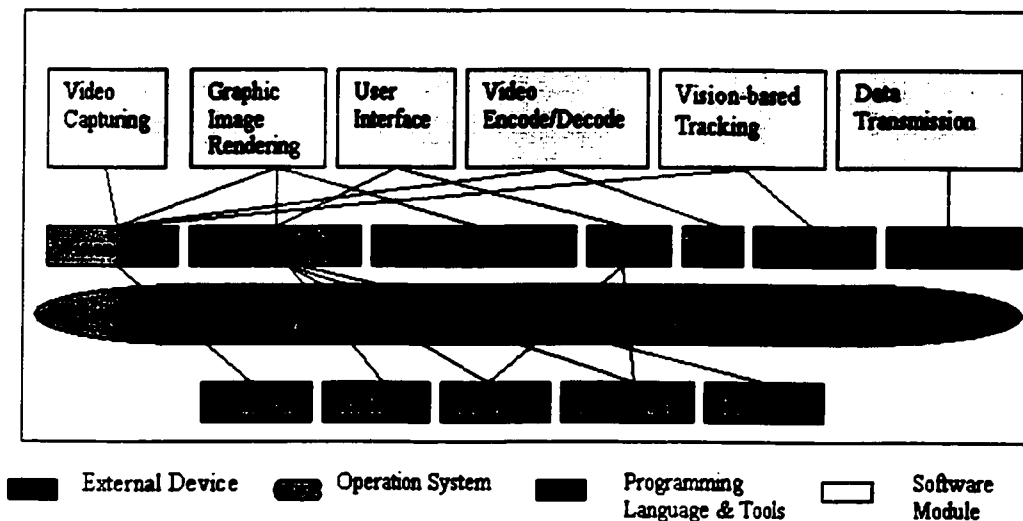
## **4.7 Chapter Summary**

**This chapter defined the design goals of the system for industrial training using mobile collaborative augmented reality. The hardware configuration was then described, and the challenges met during the designing step were set out. After that, the problems with the current practices were discussed and the decisions made for this system were presented. Finally, the overall architecture of the system was presented.**

## Chapter 5. Detailed Design and Implementation

As discussed in chapter 4, the prototype leverages a client/server architecture. It consists of six modules: video capturing, video encode/decode, data transmission, vision-based tracking, graphic images rendering and user interaction.

Figure 5-1 shows the physical view of the external devices, software modules, the programming languages and tools in the system and their relationship.



**Figure 5-1 Physical view of the system**

This chapter will introduce the development environment. The design and implementation of the prototype will then be investigated in detail. Finally, an application based on this prototype will be presented.

### 5.1 Development Environment

#### 5.1.1 C/C++

Compared to Java, C/C++ is more suitable for real-time applications especially when the image must be processed bit by bit for vision-based tracking. Since both the wearable computer and the workstation run Windows, Visual C++ 6.0 will be used to develop the application.

### **5.1.2 Microsoft Vision SDK**

Microsoft Vision SDK is a library for image manipulation and analysis on computers running Microsoft Windows operating systems. It is a low-level library including classes and functions for working with images, but it does not include image-processing functions.

In the implementation, Microsoft Vision SDK is used to access the camera, perform image manipulation and analysis in video capturing, graphic image rendering, video encode/decode and vision-based tracking.

### **5.1.3 ARToolKit**

ARToolKit is an Augmented Reality tracking library that uses computer vision techniques to calculate with precision a camera position and orientation relative to a tracking marker. With these libraries an AR system can draw 3D virtual objects that are exactly aligned with a real object. It is used in the implementation of the vision-based tracking module.

### **5.1.4 OpenGL & GLUT**

OpenGL is a software library which wraps the low-level graphic hardware operation. This library consists of about 250 distinct commands for producing interactive three-dimensional applications. OpenGL is designed as a streamlined, hardware-independent interface. It has been implemented on many different hardware platforms. The prototype uses OpenGL 1.2.

However, OpenGL doesn't handle user input or Windows events. Therefore, the OpenGL Utility Toolkit (GLUT), a window-system-independent toolkit, which hides the complexities of differing window system APIs, is used in the implementation of the graphic images rendering and user interface modules.

### **5.1.5 VRML and OpenVRML**

OpenGL doesn't provide high-level commands for describing models of 3D objects. Such commands might allow you to specify relatively complicated shapes such as computers or human beings with a fair amount of detail. With OpenGL, the

programmer must build up his designed model from a small set of geometric primitives – points, lines, and polygons.

The Virtual Reality Modeling Language (VRML) is a file format for describing interactive 3D objects and worlds. With VRML, some complicated 3D objects can be built, but a tool to browse the VRML file in OpenGL is needed. OpenVRML is a portable, free VRML browser available under the GNU Lesser General Public License. With this tool, some complicated models can be browsed on an OpenGL viewer.

Therefore, the VRML and OpenVRML are used in the development of graphic images rendering. However, in our experience, some characteristics of VRML models, such as colors and light, cannot be rendered in OpenGL correctly because of the limited command in OpenGL.

### **5.1.6 MFC**

The Microsoft Foundation Class (MFC) Library, a C++ Windows™ interface, is used to implement the graphic user interface.

### **5.1.7 CavernSoftG2**

CavernSoftG2 is a C++ hybrid-networking/database library optimized for the rapid construction of collaborative Virtual Reality. It is widely used in the VR community. CavernSoftG2 is selected to support the distribution and collaboration functions in the system.

### **5.1.8 Intel's JPEG Library**

Intel's JPEG Library (IJL) is used for video image compression/decompression. The IJL is a software library for application developers that provides high performance JPEG encoding or decoding of full color, and grayscale, continuous-tone still images. The IJL was designed for Intel processor-based systems and has been tuned for high performance and efficient memory usage.

## 5.2 Detailed Design

### 5.2.1 Video Capturing

This module is responsible for video capturing and is implemented with the Vision SDK. The captured images that we work with are specified as RGBA pixel types in which a pixel is represented as 4 bytes. Each byte represents a red, blue, green or alpha value. A sequence is defined to connect to the image source, which is used as a "pipeline," with images pushed in at the back and popped out at the front.

#### 5.2.1.1 Initialization

This module consists of two functions. One of the functions is the initialization of video capturing. The procedure is as follows:

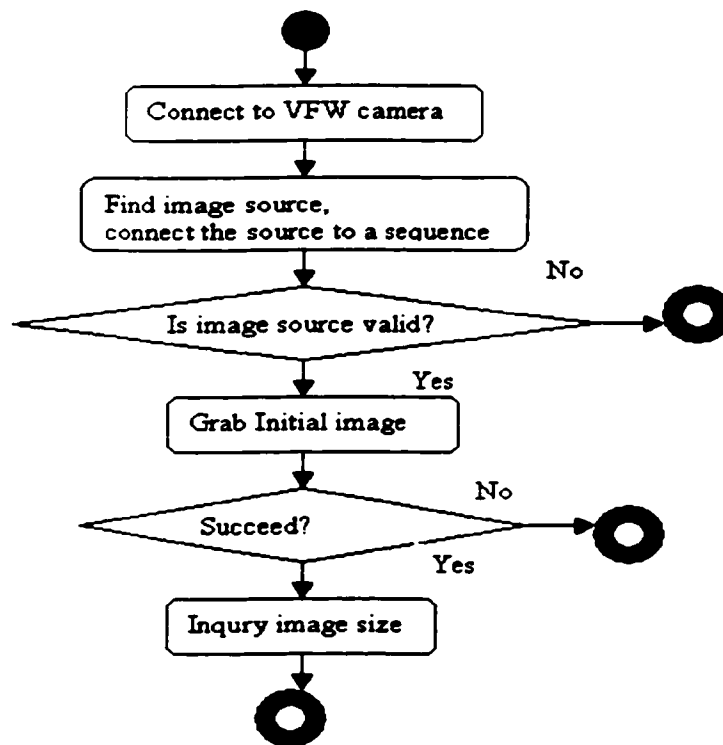


Figure 5-2 Flow chart for video capturing initialization

#### 5.2.1.2 Video Image Access

To get a video image from the image source, we simply pop an image from the front of the sequence as:

```
if (!sequence.Pop(image, 40000)){  
    printf("Couldn't get the image\n");  
    return false;  
}
```

If the image is available within 40000 milliseconds, the image will be removed from the *sequence* and *image* will contain the desired image. If the image is not available after 40000 milliseconds, **false** will be returned and *image* will not be modified

### 5.2.2 Video Encode/Decode

To encode/decode the video images in the system, a *JpegEncode* and a *JpegDecode* class are developed based on the Intel JPEG Library. *JpegEncode* is used to encode the video image to a buffer while *JpegDecode* is used to decode *Jpeg* data from a buffer.

### 5.2.3 Vision-Based Tracking

I used the technique proposed in the ARtoolkit for vision-based tracking. In this proposal, a marker with a square region and a pattern inside of the square region is used, as shown in figure 5-3.

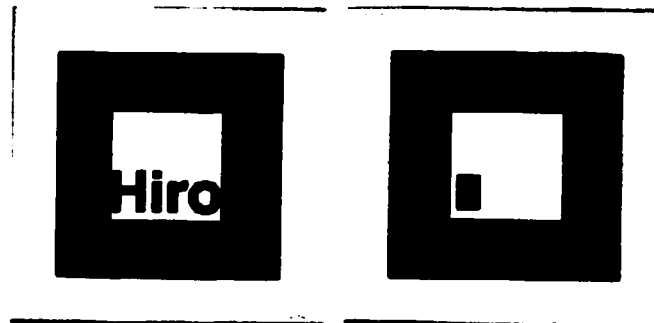


Figure 5-3 Sample patterns

#### 5.2.3.1 Camera Tracking

The known square size and pattern orientation is used to calculate the position of the real video camera relative to the marker. This matrix is then used to set the position of

the virtual camera coordinates. Since the virtual and real camera coordinates are the same, the computer graphics are drawn precisely overlay the real marker.

Figure 5-4 shows the system flow for vision-based tracking. In this figure, the data flow between client and server is simplified.

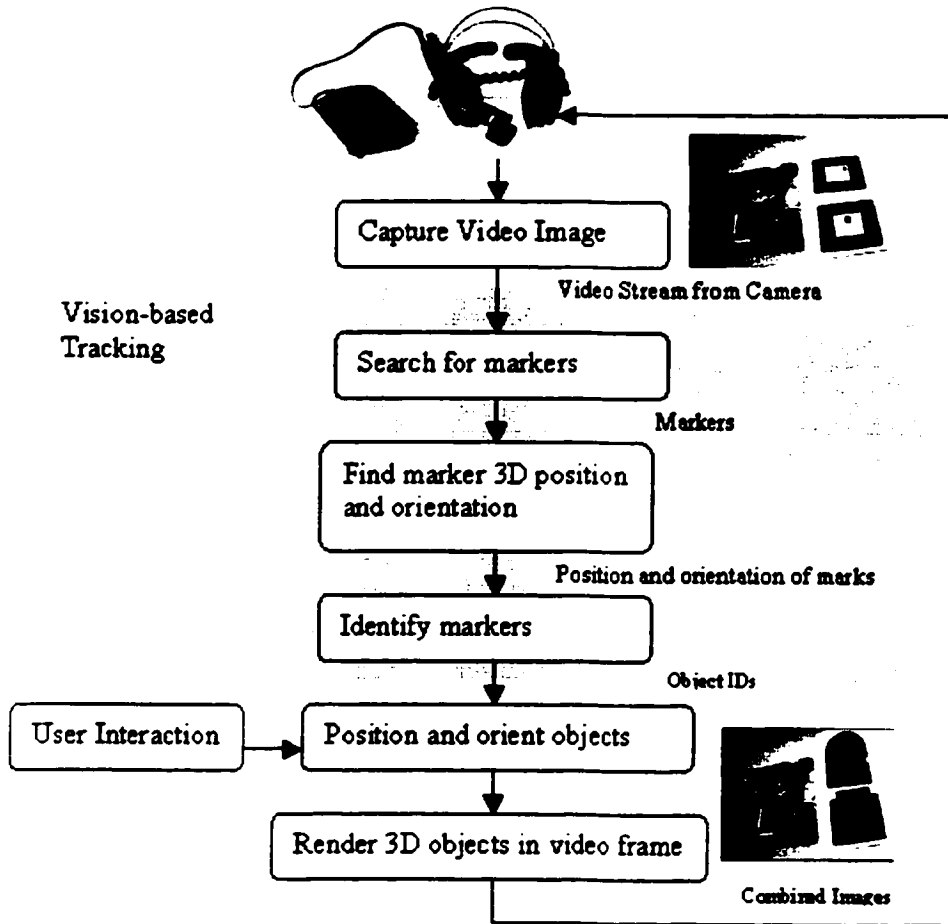


Figure 5-4 System flow for vision-based tracking

### 5.2.3.2 Recognizing Different Patterns

To decide which virtual object should be superimposed on the marker, the patterns inside the marker square in the video input stream are matched against pre-trained pattern templates. These patterns are loaded at run time; the text file *object\_data* specifies which marker objects are to be recognized and the patterns associated with each object. The *object\_data* file begins with the number of objects to be specified as well as the text data structure for each object. Each of the markers in the *object\_data* file are specified by the following structure:

*Virtual Object/Marker Id*

*Virtual Object Name/VRML File Name and Directory*

*Pattern Recognition File Name*

*Width of Tracking Marker*

For example the structure corresponding to the marker with a VRML module “chip” is:

2

Wrl/chip.wrl

Data/patt\_Hiro

80.0

It shows that the identification for the marker or virtual object is 2. The marker pattern file name is patt\_hiro in the data directory. The pattern is associated with a VRML module specified with chip.wrl. The width of the tracking marker is 80.0 mm.

#### **5.2.4 Data Transmission**

The communication layer is responsible for all exchanges of information amongst users. When clients and server reside on heterogeneous computing platforms, data format conversion is necessary for proper data sharing. A data packing routine provides data conversion and serialization for the data with most base types (ints, floats, etc).

In our system, communication is composed of two primary components. First, a TCP reflector in the server transmits reliable TCP packets when the information being transmitted is essential, for instance the video images, camera pose translation matrix and visible object identification. This component is called camera communication (see Appendix A).

Finally, a database component manages collaboration information such as commands, the coordinates for a remote pointer, and the translations of the visualization geometry resulting from object manipulation.

The server's primary function is to serve as a central connection point for each of the clients. The server reflects commands and operations to each of the connected clients to enforce the shared states of the environment.

#### 5.2.4.1 Communication of Collaboration Data

The collaboration data, including collaboration commands, the coordinates of remote pointer and object manipulation are transmitted between participants.

We used a distributed shared memory technique via a database class provided by CavernSoftG2 to disperse the collaboration data between participants. Its basis is the notion of uniquely named "keys" that hold data, and an Information Request Broker (IRB) that manages them. A key in one IRB can be linked to a key in another IRB to share changes in data. The IRB is actually a process thread that provides methods for setting and getting data from a key. On receipt of new data, it calls a user callback function to handle the new data. If new data are set into the key, the IRB disperses them to all linked keys.

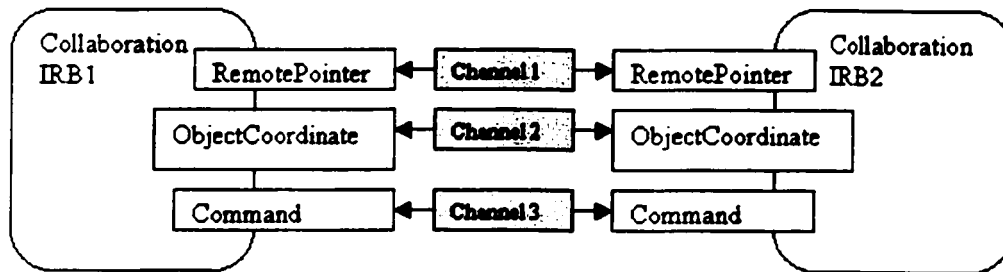


Figure 5-5 Communication of collaboration data

As shown in figure 5-5, there are three keys used to communicate collaboration data: RemotePointer, ObjectCoordinate and Command are defined in collaboration IRB1 and collaboration IRB2 respectively in our system.

When a participant inputs a command or manipulates a virtual object with a keyboard or a mouse, this event will update the database and it will be propagated to all the subscribers of the database via the server. The subscribers will be notified via a callback function. The updates are sent to the server with UDP as the data package is small and its loss won't affect the synchronization between the clients. The server will then forward the data to all of the clients with reliable TCP, as any loss of this information could lead to misunderstandings in the communication between

participants. The database server can arbitrate any conflicts and maintain uniformity, thereby ensuring consistency in the shared environment. To enforce greater consistency requirements, a locking mechanism to lock data on a shared memory is provided so that only a single client may update the shared memory at any one time. Attempts by other clients to update the memory will be ignored.

#### 5.2.4.2 Latecomer Support

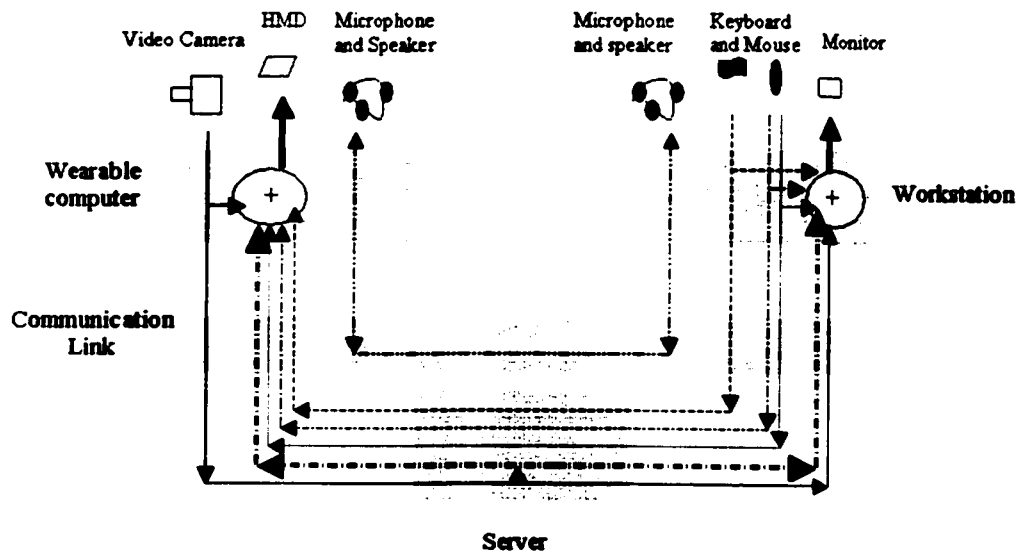
In the collaboration applications, some participants will join the application at a later time. In our application, the server will check the new clients regularly and add them to the client list once they have been checked. As long as the newcomer subscribes to the shared memory of the specific data, it will be sent the data in the shared memory and keep the same state with the other subscribers.

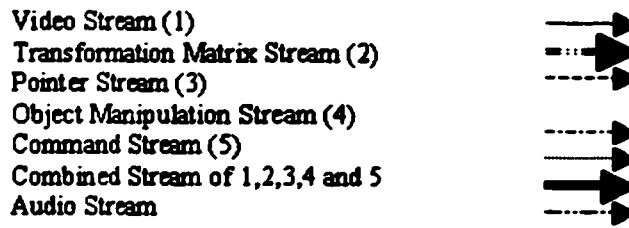
#### 5.2.4.3 Audio Conference

The audio capturing module allows participants to engage in an audio-conferencing session. The module is based on the Microsoft NetMeeting SDK.

#### 5.2.5 Graphic Images Rendering & User Interface

The system supports real-time collaboration in which the users share an augmented workspace with some collaboration functions such as remote pointing, virtual object manipulation and freeze-image action.





**Figure 5-6 Data flow between peripheral devices and the system**

Figure 5-6 shows the data flow of peripheral devices and the system. As we can see, the video images captured by the camera mounted on the wearable computer are transmitted to the server. The server replicates the images to other clients. Furthermore, the server processes the images and produces the transformation matrix of the visible virtual objects and distributes the data to all of the clients.

When the users interact with the system for collaboration, using a mouse or keyboard, the pointer stream, virtual object manipulation stream and command (freeze image, etc) stream are distributed to the other clients by the server.

Finally the video, collaboration command, remote pointer, virtual object transformation matrix and manipulation data streams are combined together to generate graphic images and rendered on the wearable computer and workstations respectively.

#### **5.2.5.1 User Interaction for Collaboration**

The system manages the user interaction events for collaboration. The users are offered to manipulate the virtual objects with a keyboard or a 2D mouse.

##### **5.2.5.1.1 Virtual Object Manipulation with a Keyboard**

With regard to manipulation with a keyboard, some specific keys are defined to manipulate the virtual objects. The numbers “1~9” are defined to represent the virtual object identifications. Some letters are defined to move, rotate and scale the virtual object with regard to axis  $x,y$  and  $z$  on the direction of “+” or “-” respectively, as shown in the table below.

Manipulation	Axis	Key (+)	Key (-)
Translation	x	'Q'	'W'
	y	'A'	'S'
	z	'Z'	'X'
Rotation	x	'O'	'P'
	y	'K'	'L'
	z	'L'	'M'
Scale	/	'C'	'V'

**Table 5-1 The definition of operation keys for remote manipulation**

In the operation, the user can press the number key to choose the virtual object; for instance, he can press "1" then "Q" or "P" to move or rotate a virtual object on the plus or minus direction of the x-axis.

One click to the letter results in the corresponding manipulation with one unit. In the default setting, the unit is one pixel. This is too small to be recognized as a manipulation effect in the rendering. Therefore, a bigger unit for each manipulation is defined. For instance, with every operation on an object, the units for movement and rotation are 60 pixels and 30 degrees respectively.

#### **5.2.5.1.2 Virtual Object Manipulation with a Mouse**

I have used GLUT to demonstrate 2D mouse interactivity for 3D object manipulation in a 3D environment by detecting mouse clicks and motion.

To manipulate a virtual object with a 2D mouse some effective mouse button clicks and corresponding motion coordinates are specified, as follows:

<b>Manipulation</b>	<b>Axis</b>	<b>Mouse Button</b>	<b>Motion Coordinate</b>
<b>Translation</b>	x	Left Button	x
	y	Left Button	y
	z	Right Button	x
<b>Rotation</b>	x	Left & Right Button	x
	y	"Shift"+Left & Right Button	x
	z	"Ctrl"+Left & Right Button	x

**Table 5-2 Mouse operation specifications for virtual object manipulation**

As shown in this table, with the help of the modifier keys "Shift" and "Ctrl", we can use only the left and right button of a 2D mouse to move and rotate a virtual object. When there is more than one virtual object in the view, we use the specified number key which represents the ID of the virtual object to pick it up for manipulation.

#### **5.2.5.1.3 Remote Pointing**

In order to direct the attention of the remote participants, a viewer of a shared image can use a remote pointer. When he/she clicks the mouse button, the (x,y) coordinates of the mouse are sent to the remote participants and rendered as a cursor in addition to the regular cursor.

#### **5.2.5.1.4 Image Freezing**

A remote user is able to press the mouse button to freeze the most current video frame on both the wearable HMD and the desktop computers. The remote users can then move the remote pointer without having to follow the wearable user's head movements. Any trainer /trainee can turn off the image-freezing feature any time so that the real-time video sharing is recovered.

### 5.2.5.2 User Manager Dialog

The user manager dialog is implemented with MFC and used by a client to start the application. As shown in figure 5-7, this dialog allows a user to input the server IP address and port to be connected. A client also has choices with regard to the real-time video: the client can specify whether the video should be shared or not, or the video should just be received. For workstation users who have no real-time video resource, their only choice is to receive video. For a wearable computer user, he/she can have choice to provide video or not. If he/she provides it, then the workstation users who connect to the server later on can receive real-time video, join a training exercise and do the collaboration work. If he/she chooses to not provide it, then he/she just does the training exercise with the system alone, without collaboration with the other users.

In the dialog, the client also has the choice to define whether or not a full screen should be used to view the graphic image window/ viewer. Since the wearable computers have a limited screen space, in most cases wearable computer users prefer to have a full window for graphic image rendering. For this reason I set up a full window flag as an option for the user; when the flag equals 1, then the user's application window covers the entire screen window.

The screenshot shows a dialog box titled "User Manager". It contains the following fields and controls:

- Server IP:** A text input field that has been blacked out.
- Port:** A text input field containing the value "6010".
- Provide video?:** A text input field containing the value "1". To its right is the legend: "{0-No; 1-Yes; 2-Receive video}".
- Full windows?:** A text input field containing the value "0". To its right is the legend: "{0-No; 1-Yes}".
- Buttons:** "OK" and "Cancel" buttons are located at the bottom of the dialog.

Figure 5-7 User manager dialog

### 5.2.5.3 Local Viewer & Remote Viewer

The graphic images, including the video image, virtual 3D models, prompt and remote pointer, are rendered with OpenGL and displayed in the camera viewer of the wearable computer and the remote viewer of the workstations, as shown in figure 5-8 and 5-9 respectively. When the wearable computer connects to the server successfully, a camera viewer is displayed. In the same way, if a workstation connects to the server and successfully joins the current wearable computer session, the same information as the camera viewer is displayed. As shown, the remote pointer is represented with a red wire frame including a red point to direct the user "look at this area".

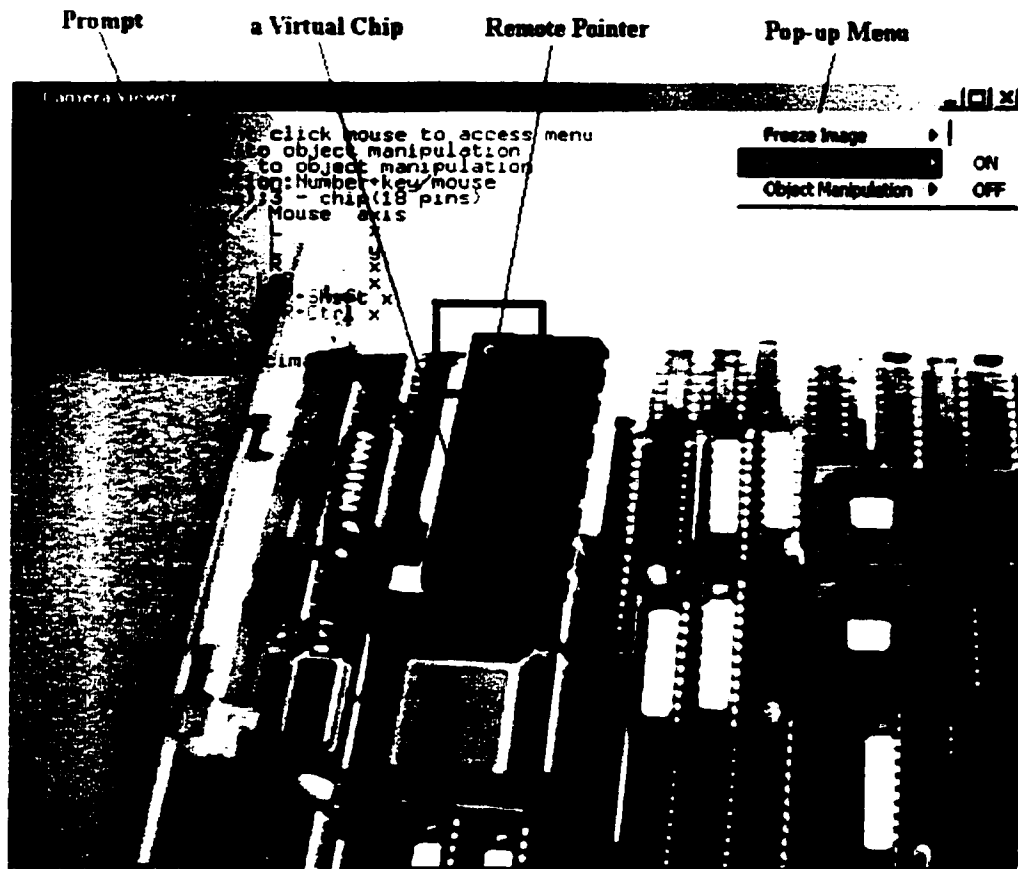


Figure 5-8 Camera viewer (Local)



equipped with a wearable computer and an ATM switch, runs the training system and works with these devices. Figure 5-10 shows a trainee, wearing a wearable computer, working with a switchboard. Let's refer to him as local trainee A. Trainer T and remote trainee B are working on the remote workstations, watching A's actions through the local trainee's shared view. All are equipped with a microphone and are able to interact through live audio.



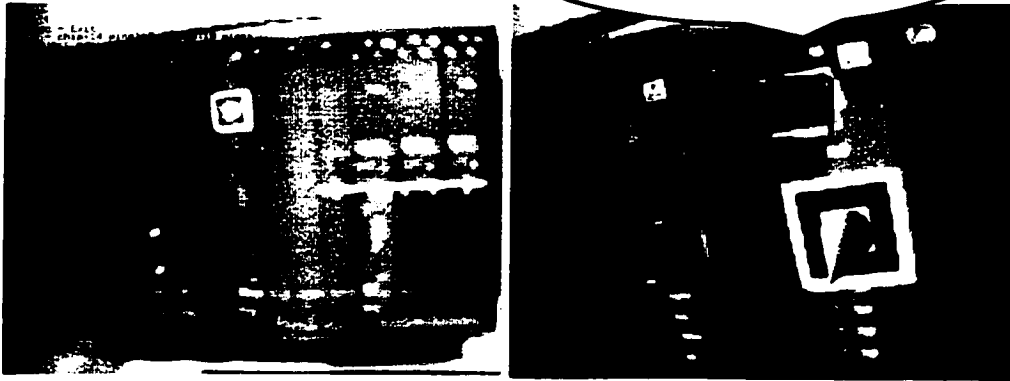
**Figure 5-10 a trainee works with a wearable computer**

Figure 5-11 shows the screen shots of the local viewer and camera viewer of the collaborative training exercise running on our system. The first thing that the trainee learns is how to pull out the switch card. When local trainee A points at the ATM switch with the camera mounted on his wearable computer, he sees an ATM switch with more than ten switchboards, as shown on figure 5-11(a). When he approaches the ATM switch, a 3D model--cone style arrow is superimposed on a switchboard to instruct the user to push out this switchboard (figure 5-11(b)). Using a remote pointer, trainer T directs him to push up the button on the top of the surface of the switchboard, and push down the button on the bottom of the surface of the switchboard at the same time, then pull out the switchboard in the direction of the overlaid cone style arrow. As shown in figure 5-11(c), trainer A pulls out the switchboard as directed.

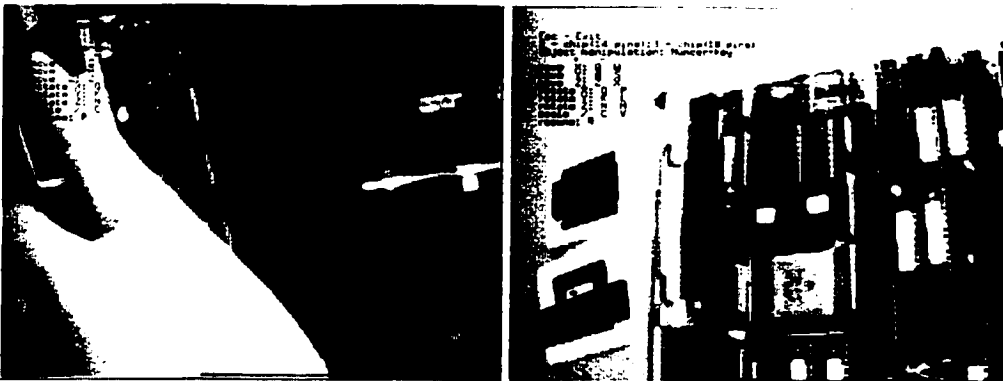
The switchboard is put on the table with two virtual chips next to it (figure 5-11(d)). As shown in figure 5-11(e), trainee A picks up the chip with seven pairs of pins and moves it close to the switchboard. As shown in figure 5-11(f), the chip is laid on the socket with all of the pins lined up. Pin 1, marked with a white dot on the chip surface, is at the top left of the figure. Remote trainee B would like to try installing

the chip in another way, and he shows trainer T and trainee A his idea by orientating the virtual object with the keyboard or mouse. As shown in figure 5-11(g), the virtual chip is rotated by trainee B with the small white dot on the bottom right of the picture. Trainer T points to the small white dot on the virtual chip surface and says that both pin 1 and the socket should be matched up for correct installation, or it will result in serious damage to the chip (figure 5-11(h)). Through this interactive "real" training exercise the trainees can gain valuable experience.

Trainer T: Pull the button up,  
then pull out the switchboard.



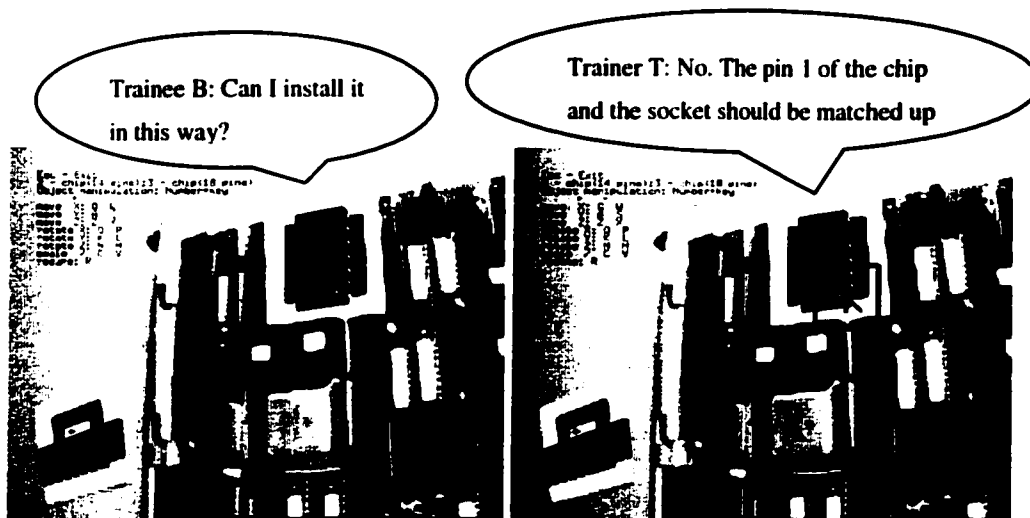
**(a) Camera is pointing at an ATM Switch; (b) Directing the trainees to pull out the switchboard;**



**(c) Trainee A is pulling out the switchboard; (d) Switchboard with two virtual chips beside it;**



**(e) Trainee A picks up one virtual chip to install it; (f) the virtual chip is laid on the switchboard;**



**(g) The virtual chip is rotated by trainee B; (h) Trainer T point out that is incorrect.**

**Figure 5-11 Screenshots for the industrial training prototype**

## **5.4 Chapter Summary**

This chapter presented the detailed design and implementation of the system. Firstly, the programming language and tools used in the development were introduced. Then the implementation of each module of the system was described. The chapter ended with a presentation of the prototype for the training application for ATM switch maintenance.

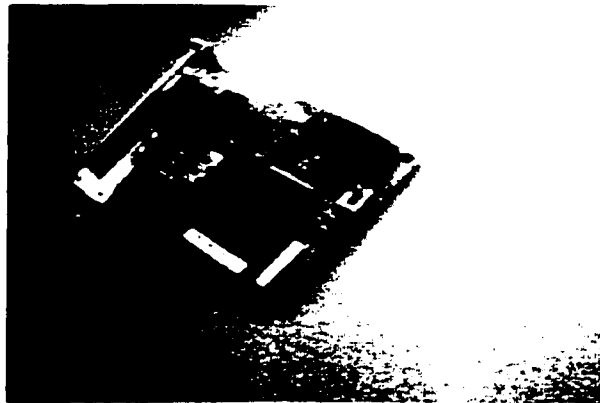
## **Chapter 6. Experimental Results & Evaluation**

This chapter will present the experimental results and evaluation. First the 3D capability of the MA IV Xybernaut wearable computer is discussed and some preliminary evaluation results are presented. Then the test beds of the collaboration system are introduced and some test results based on these test beds are presented. Finally some subjective user evaluation to the system performance and effectiveness of the functions provided by this prototype are presented and discussed.

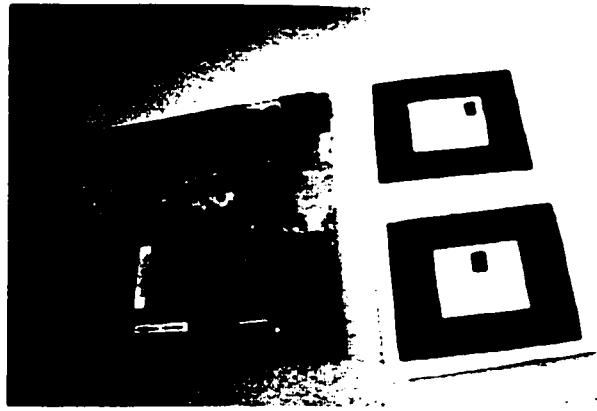
### **6.1 3D Capability of the Wearable Computer**

#### **6.1.1 Evaluation**

The Xybernaut wearable computer that we are using turned out to lack 3D hardware and be low on CPU power. How does this affect the augmented reality application with a wearable computer? Figures 6-1, 6-2 and 6-3 show the screenshots of the experiments.



**Figure 6-1 Rendering the video images (RV)**



**Figure 6-2 Rendering video images and tracking visible virtual objects (RV+TR)**



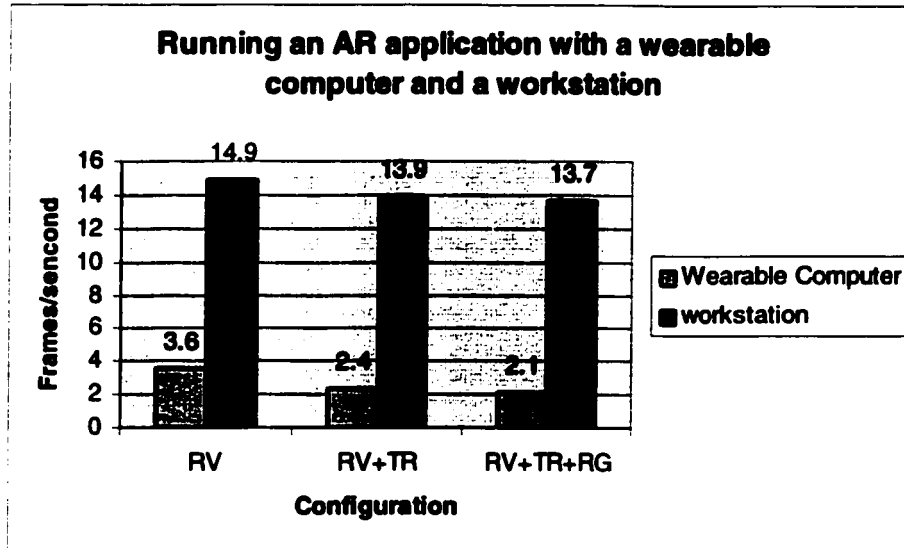
**Figure 6-3 Rendering video images, tracking visible virtual objects and rendering them (RV+TR+RG)**

In the AR application, there are three important processes:

- 1) Rendering the captured video images (RV);
- 2) Detecting the markers, calculating the camera viewpoint relative to the real world and recognizing the patterns in the video input frame to decide which virtual object is visible - this is referred to as vision-based tracking (TR);
- 3) Rendering the graphic images of visible virtual objects (RG).

Figure 6-1 shows the experiment with only the process of video image rendering enabled. This configuration is referred to as RV. Figure 6-2 shows the experiment in which the video image is rendered, the position of the video camera relative to the physical marker is calculated and the visible objects are recognized. This

configuration is referred to as RV+TR. Figure 6-3 shows the experiment wherein not only the RV+TR configuration is enabled, but also some simple virtual objects, such as cube and sphere, are rendered. This configuration is referred to as RV+TR+RG and is the configuration of a typical AR application. Table 6-1 shows the update rate of the view when the application is run on the Xybernaut Mobile IV wearable computer without 3D hardware support and the workstation (Pentium III-300) with GetForce2Go video card respectively with the above configurations.



**Table 6-1 Running an AR application with different configurations**

The video images (320x240) in the application are in RGBA format, in which a pixel is represented with 4 bytes. As the figure shows, the view update rate of the wearable computer is much lower than that of the workstation with the same configuration. The update rate for rendering the video images is 3.6 frames/second. It is time consuming for a wearable computer to draw an image, using OpenGL, pixel by pixel in RGBA format without 3D hardware support. However, a workstation equipped with a GetForce2Go video chip can achieve 14.9 frames/seconds. The time the workstation (Pentium III-300) spends on the TR process is a little bit faster than a wearable computer (Intel 233). Drawing two simple virtual 3D models on the video images just makes the rendering a little bit more complicated. It doesn't add much load to the workstation or the wearable computer either.

### **6.1.2 Result**

From this experiment we can conclude that without 3D support in the hardware, a wearable computer, of the type we had, is not sufficient for augmented reality applications that require a 10 frames/seconds update rate to retain a sense of immersion in the real world.

### **6.1.3 Solution**

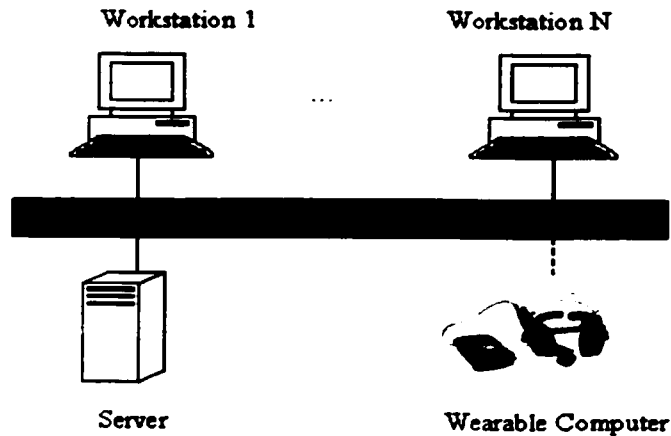
A powerful wearable computer with a good 3D accelerator is recommended. A possible alternative could be a powerful laptop carried by the user in a backpack and equipped with a GeForce2Go video chip and an I-glass see-through stereoscopic color Head Mounted Display mounted with a camera.

## **6.2 Test Beds for the Prototype**

The 3D capacity of the wearable computer model that we own is not satisfactory for the real time AR application, especially the collaborative AR application that requires real-time support for the collaboration and high quality of rendering.

### **6.2.1 Test Bed A**

The first test bed for our prototype consisted of three Pentium III workstations equipped with GeForce2Go video chips running MS Windows 2000, and a Mobile IV Xybernaut wearable computer running MS Windows 98. One workstation works as a server and the other two workstations work as clients. The workstations are connected via a 100 BaseT Ethernet. The wearable computer communicates with the workstations through 11 Mb/s wireless link.

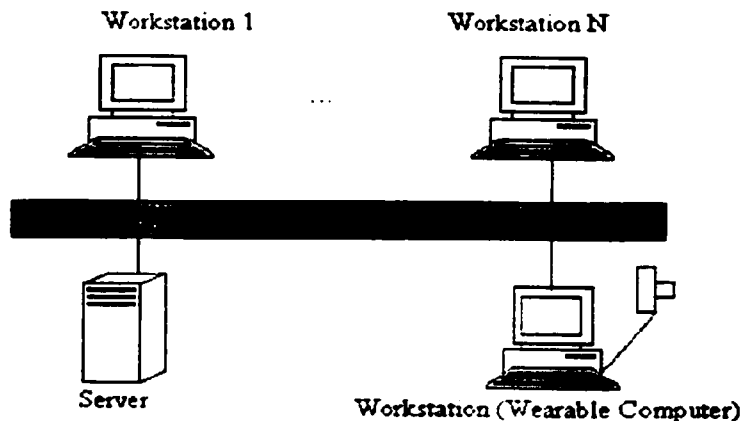


**Figure 6-4 Test Bed A**

Running the prototype with this test bed, the view update rate is about 3 frames/seconds to the video image with 160 pixels by 120 pixels. This is not satisfactory for the real time AR application. Because of the low view update rate, the 3D objects flow in the AR environment when the wearable computer user moves his head. This means that the manipulation of the virtual objects doesn't make sense. Therefore, most of the experiments were tested on test bed B.

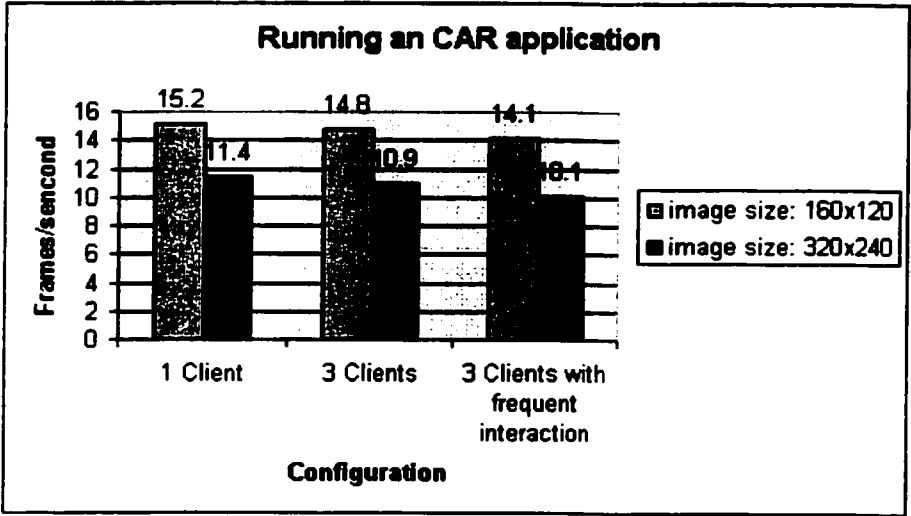
### 6.2.2 Test Bed B

In this test bed the wearable computer was replaced with a workstation with the same settings as the other workstations, with the exception that it was equipped with a video camera (figure 6-5).



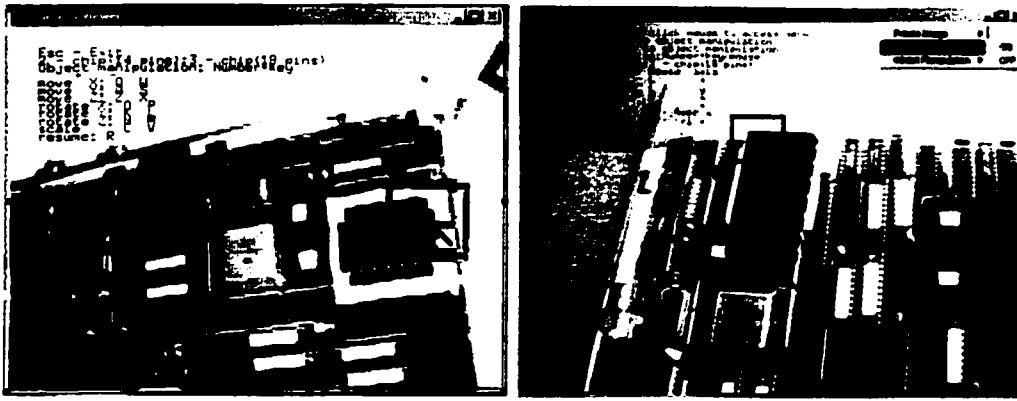
**Figure 6-5 Test Bed B**

Running the CAR prototype for the application presented in chapter 5 with this test bed, the view update rate varies in different configurations. As shown in table 6-2, the view update rate for an application with video images in 160x120 pixels is higher than the view update rate for an application with video images in 320x240 pixels. When three clients participate in a session to do collaboration work, the view update rate is about 14.8 frames/second for the application with the video images in size 160x120 and 10.9 frames/second for the application with the video images in size 320x240. If the frequency of collaboration or interaction is higher, the view update rate is a little bit lower, but it meets the minimum requirement of 10 frames/second for the real-time AR application.



**Table 6-2 Running a CAR application with different configurations**

As shown in Figure 6-6, the view quality of the application with video images in size 320x240 is better than that with video images in size 160x120. However, it produces more data and adds load to the network. Therefore, the bigger image size, the better view quality but the lower view update rate.



**Figure 6-6 (a) Viewer with video image in size (160x120 pixels); (b) Viewer with video image in size (320x240 pixels).**

### **6.3 Subjective User Evaluation**

In order to evaluate the performance of the system and the effectiveness of the functions provided to support training performance, we conducted an experimental study in which 18 students used the training system to learn to remove a switchboard from the ATM switch and install a chip on the switchboard. The training tasks are described in details with some screenshots in section 5.3. We used test bed B for our experiments and rendered the images in 320x240 pixels. During the experiments, I moved the camera mounted on the workstation to simulate the camera mounted on a wearable computer worn by a user who is walking around and moving his/her head.

The participants consisted of 7 females and 11 males ranging in age from 18 to 45. Though 94.4% of the participants have 3 years or more experience in using the computer in their work, most of them had no prior experience in collaborative augmented reality and 72.2% had never used this software before. 22.2% of the participants had some prior understanding of augmented reality by virtue of having used this system previously. One of the participants is a graduate student whose research area is in augmented reality. She used the system occasionally before the evaluation.

Based on the tasks performed, each participant was asked to answer some questions regarding the function effectiveness, system performance and overview of the system.

### 6.3.1 Function Effectiveness

Based on the tasks performed, each participant was asked to rate the effectiveness of the following functions.

- 1). Replacing the real equipment with virtual objects (VO)
- 2). Using a mouse to manipulate the remote virtual objects (MM)
- 3). Using a keyboard to manipulate the remote virtual objects (KM)
- 4). Using a freeze-image feature to help remote pointing and conversation (FI)
- 5). Using a remote pointer to direct the participants' attention (RP)
- 6). Overlaying the supporting information on the real world with Augmented Reality Technology (SAR)

The participants were asked to rate "effective", if they thought the function completed the requirement sufficiently and rate "ineffective" if they thought the function didn't complete the requirement sufficiently. If they could not make up their mind, they were asked to rate "undecided".

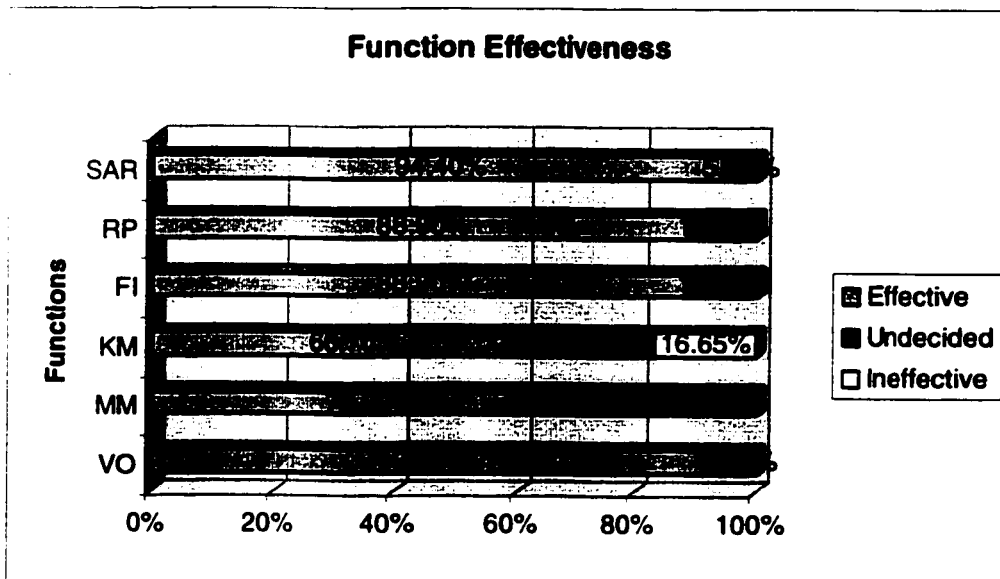
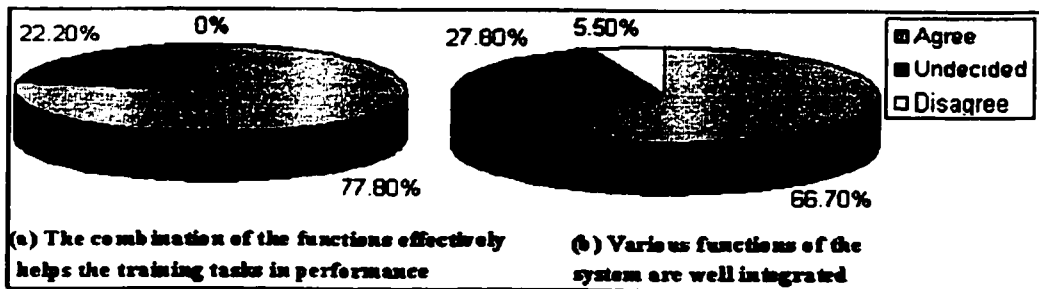


Table 6-3 Evaluation of the function effectiveness

Table 6-3 shows that around 90% of the participants are very satisfied with the functionalities VO, FI, RP and SAR and rate them as effective. Around 70% of the participants think that it is effective to manipulate the remote virtual objects with a keyboard or mouse. However, the KM is rated not so effective as MM. Only 66.7% of the participants thought that it is effective to use a keyboard to manipulate virtual objects and 16.7% of the participants regard it as ineffective. The reason is that when a user manipulates a virtual object with a keyboard he/she has to operate with a specified unit, for instance, rotates 30 degrees, moves in x axis with 30 pixels length in the frame image, which is inflexible.

Overall, 77.8% of the participants said that the combinations of the functions effectively help the training performance. Furthermore, 66.7% of the participants found various functions of the system are well integrated (table 6-4).

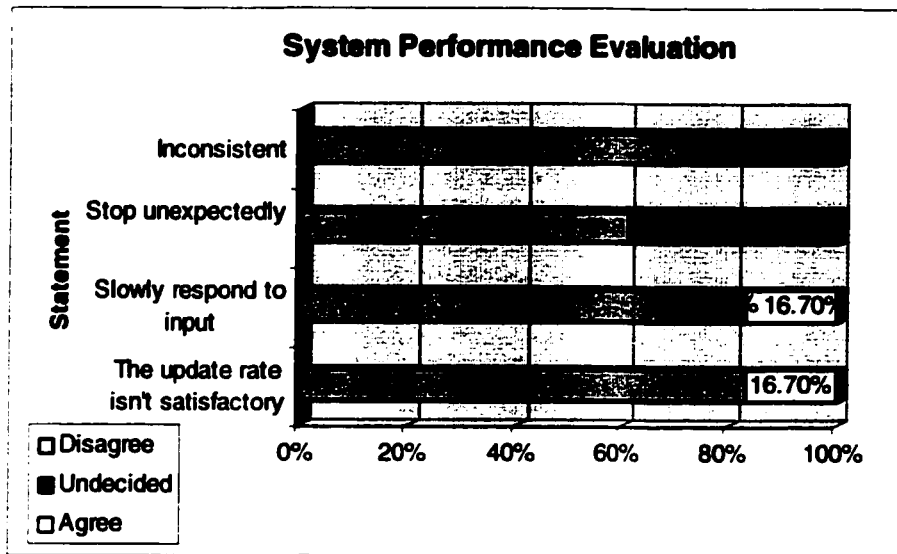


**Table 6-4 Performance evaluation of the combination of various functions**

### 6.3.2 System Performance

To evaluate the system performance, the participants were asked to exclusively select “agree”, “undecided” and “disagree” to the following statement:

- 1). The view update rate isn't satisfactory.
- 2). The software responds too slowly to input.
- 3). The software has at some time stopped unexpectedly.
- 4). The software is inconsistent.



**Table 6-5 System performance evaluation**

As shown in table 6-5, more than 61% of the participants were satisfied with the overall performance of the system. 16.7% of the participants found that the view update rate with 10.9 frames/seconds wasn't satisfactory and the system responded too slowly to inputs since the rendering of the manipulation of the virtual objects is in the same update rate with the view. Even though an update rate of 10 frames per second is needed for acceptable real-time performance, some participants expect a higher update rate.

### 6.3.3 Overview to the System

No	Questions	Agree	Undecided	Disagree
1	The software has a very attractive presentation	83.3%	11.1%	5.6%
2	The instruction and prompts are helpful	66.7%	22.2%	11.1%
3	The way that system information is presented is clear and understandable	88.9%	11.1%	0
4	I need to learn a lot of things before I could get going with this system	16.7%	11.1%	72.2%
5	I thought the system was easy to use	88.9%	11.1%	0
6	Tasks can be performed in a straightforward manner using this software	72.8%	27.8%	0
7	I would recommend this software to my colleagues	66.7%	27.8%	5.5%
8	I would prefer using this training system for the training tasks rather than using the manuals	72.2%	27.8%	0

**Table 6-6 Overview to the system**

Table 6-6 shows 88.9% of the participants thought the way in which the system information is presented is clear and understandable, and they found the system was easy to use. 72.8% of the participants found that the tasks could be performed in a straightforward manner using this software and they would prefer using this training system rather than using the manuals. However, 16.7% of the participants found that they needed to learn a lot of things before they could get going with this system, as the MCAR technology applied was totally new to them. 11.7% of the participants found the instruction and prompts were not helpful enough especially when the remote participants were manipulating a virtual object. They mentioned that it would be nice if some information about who was operating a virtual object could be displayed on the viewer. Nevertheless, 66.7% of the participants would recommend this software to their colleagues.

## **6.4 Chapter Summary**

In this chapter, the evaluation of the system and the results of the experiments were provided. First the 3D capability of the MA IV Xybernaut wearable computer was discussed. The preliminary evaluation results show that the MA IV wearable computer is not sufficient for augmented reality. Then the test beds of the collaboration system were introduced. It was shown that the view update rate of the system based on test bed B is acceptable. Finally some subjective user evaluation to the system performance and effectiveness of the functions provided by this prototype were presented. It seems that the participants were satisfied with the system performance. Most of them found the functions provided were effective.

## **Chapter 7. Conclusions and Future Work**

The proposed existing schemes for training applications which use either Distributed Virtual Environments or distance learning have their drawbacks. This thesis has proposed a novel mobile collaborative augmented reality mechanism for application in training and demonstrated it via the implementation of a prototype for ATM switch maintenance.

### **7.1 Conclusions**

This prototype provides training in task performance. It allows a group of trainers and trainees, each in geographically disparate locations, to carry out a training task collaboratively by sharing the workspace of a trainee who is equipped with a wearable computer. In this prototype, an intuitive interface for the distributed virtual environment (augmented reality) user, and some collaboration functions - such as remote manipulation of the virtual objects and remote pointing - were provided. The choice of display between "see-through" and "see-around" viewing areas, based on users' acceptance, was investigated.

The evaluation of the 3D capacity of a wearable computer demonstrated that this is very important in the AR application. The result shows that a wearable computer without 3D hardware capacity, for instance, Xybernaut Mobile IV wearable computer, is not able to satisfy the requirements of a real-time AR application.

The evaluation of the system's efficiency shows that with the proposed client/server architecture the system maintains good consistency and has good extension capability. The subjective user evaluation shows that the participants were satisfied with the system performance, and most of them found the functions provided by the system, including augmenting reality with supporting information, replacing the real equipment with virtual objects, remote manipulation to the virtual objects, remote pointing and image freezing, really help the training performance.

## **7.2 Future Work**

**Future hardware improvements to the wearable computer, including 3D hardware, CPU power, reduced weight and prolonged battery life, will make the wearable computer more suitable for AR applications.**

**In the current implementation of the prototype, all the clients participate in one collaborative visualization session. A collaborative visualization framework ideally should provide multiple such sessions and allow clients to participate in any such session.**

**In our work, we tested the system in the local area network and did not test it in the wide area network. Internet is popular and can be accessed by most of the people. The ideal training system is based on the Internet but some issues have to be addressed, for instance, the critical requirement of real-time performance of AR applications and the limited bandwidth of Internet. Future research efforts may therefore investigate these issues.**

## References

- [1] J.C.Oliveira, M.Hosseini, S. Shirmohammadi, M.Cordea, E.Petriu, D. Petriu and N.D.Georganas, *VIRTUAL THEATER for Industrial Training: A Collaborative Virtual Environment*, Proc. 4th WORLD MULTICONFERENCE on Circuits, Systems, Communications & Computers (CSCC 2000), Greece, July 2000.
- [2] Computer Graphics and UI Lab, Columbia University, 1993  
<http://www.cs.columbia.edu/graphics/projects/karma/karma.html>
- [3] MIT Media Laboratory, Augmented Reality,  
<http://wex.www.media.mit.edu/projects/wearables/augmented-reality.html>
- [4] W. Hoff, T. Lyon, and K. Nguyen, *Computer Vision-Based Registration Techniques for Augmented Reality*, Proc. of Intelligent Robots and Computer Vision XV, Vol. 2904, in Intelligent Systems & Advanced Manufacturing, SPIE, Boston, Massachusetts, Nov. 19-21, pp. 538-548, 1996.
- [5] J. Kustaborder, R. Sharma, *Experimental Evaluation of Augmented Reality for Assembly Training*, IWAR'99, San Francisco, October 20-21, 1999.
- [6] T. Hollerer, S. Feiner and J. Pavlik, *Situated Documentaries: Embedding Multimedia Presentations in the Real World*, In: Proc. ISWC '99 (Third Int. Symp. On Wearable Computers), San Francisco, CA, October 18-19, 1999, pp. 79-86.
- [7] R. Kraut, M. Miller and J. Siegel, *Collaboration in Performance of Physical Tasks: Effects on Outcomes and Communication*, Proceedings, Computer Supported Cooperative Work Conference, CSCW' 96.
- [8] N. Streitz, J. Geibler, J. Haake, and J. Hol, *Dolphin: Integrated Meeting Support Across Local and Remote Desktop Environments and Liveboards*, Conference Proc. of Computer Supported Cooperative Work, CSCW'94, ACM (1994) 345-358.

- [9] T. Ohshima, K. Satoh, H. Yamamoto and H. Tamura, *AR<sup>2</sup> Hockey: A Case Study of Collaborative Augmented Reality*, Virtual Reality Annual International Symposium, March 14-18, 1998.
- [10] E. Reitmayr and D. Schmalstieg, *Mobile Collaborative Augmented Reality*, the Second IEEE and ACM International Symposium on Augmented Reality, Oct. 2001.
- [11] G. Kortuem, M. Bauer and Z. Segall, *NETMAN: The Design of a Collaborative Wearable Computer System*, ACM/Baltzer Journal of Mobile Networks and Applications, Volum 4, Number 1, 1999.
- [12] R. Azuma, *A Survey of Augmented Reality*, in Presence: Teleoperators and Virtual Environments 6, 4 (August 1997), 355 - 385. Earlier version appeared in course Notes #9: Developing Advanced Virtual Reality Applications, ACM SIGGRAPH (Los Angeles, CA, 6-11 August 1995), 20-1 to 20-38.
- [13] K. Ahlers, A. Kramer, D. Breen, P. Chevalier, C. Crampton, E. Rose, M. Tuceryan, R. Whitaker and D. Greer, *Distributed Augmented Reality for Collaborative Design Applications*, published in Eurographics '95 Proceedings, Maastricht, NL, 1995.
- [14] N. I. Durlach, *Psychophysical Considerations in the Design of Human-Machine Interfaces for Teleoperator and Virtual-Environment Systems*, In Proceedings of Medicine Meets Virtual Reality II: Interactive Technology and Healthcare: Visionary Applications for Simulation Visualization Robotics, (pp.45-47), 1994, Aligned Management Associates.
- [15] P. Milgram, S. Zhai, D. Drascic and J. Gradski, *Applications of Augmented Reality for Human-Robot Communications*, Proceedings of 1993, IEEE/RSJ International Conference on Intelligent Robots and Systems, 1993.
- [16] K. Kutulakos, J. Vallino, *Calibration-Free Augmented Reality*, IEEE Transaction on Visualization and Computer Graphics, 1998.
- [17] R. Bentley, T. Rodden, P. Sawyer, and I. Sommerville, *Architectural Support for Cooperative Multiuser Interfaces*, IEEE Computer, vol. 7 num. 4, May 1994.

- [18] N. D. Georganas, Class notes “ **Multimedia Communications**”, 2000.
- [19] E. Rose, D. Breen, K. Ahlers, C. Crampton, M. Tuceryan, R. Whitaker, and D. Greer, *Annotating Real- World Objects Using Augmented Reality*, Proceedings of Computer Graphics International '95 (Leeds, UK, 25-30 June 995), 357-370.
- [20] C. Thompson, L. Najjar, and J. Ockerman, *Wearable Computer Based Training and Performance Systems*, 19<sup>th</sup> Interservice/Industrial Training, Simulation and Education Conference Proc., Arlington, VA, 1997.

# Appendix A

## Camera Communication

Shared AR applications must have the ability to transmit actual images or video to other remote interfaces. To address the challenge of transmitting real-time video to all participants, we make use of TCP, because of its reliability.

### 1. Sending Video to the Server

In this system a video image from the camera is encoded as a JPEG image. It is packed with the related information and converted to a byte stream by the packing routine and sent to the server in the following format:

0                    1                    2                    2 +ImgSize

ShareVideoFlag	ImgSize	JpegImage
----------------	---------	-----------

*ShareVideoFlag* is a flag to identify that the video images captured by the camera of this user are shared with other participants in this session or not; 1 means sharing with others, 0 means not. *ImgSize* is the actual size of the compressed image. *JpegImage* is the compressed JPEG image data.

### 2. Reply Tracking Result to Sender

The server gets the data and unpacks it. The compressed image is decompressed and used to track the transformation matrices of the virtual objects to the camera and identify the visible objects. It produces the visible object IDs and transformation matrices. Finally the following data are packed and converted to byte streams with the collaboration packing routine, and sent to the sender:

0                    1                    2                    16                    17                    31...

ViObjNumber	ObjId1	Matrix1	ObjId2	Matrix2	...	ObjIdN	MatrixN
-------------	--------	---------	--------	---------	-----	--------	---------

The length of this message is variable. It consists of the number of the visible virtual objects, followed by the object IDs and the translation matrices of virtual objects to the camera. As we can see, the data type for *viObjNumber* and *ObjId* is int; therefore,

the length for each data is 1. The virtual object transformation Matrix is an array of float, and the length of the array is 7. Therefore, a transformation matrix is converted to 14 bytes.

### 3. Sending Data to the Other Clients

Since the other participants need the video image for rendering, the data packet to be sent to the other participant in the sessions is the combination of the video image and the virtual object transformation matrices. The data are packed and converted to a byte stream in the following format:

0            1            1+ImgSize    2+ImgSize 3+ImgSize ...

ImgSize	JpegImage	ViObjNumber	ObjId1	Matrix1	...	ObjIdN	MatrixN
---------	-----------	-------------	--------	---------	-----	--------	---------

As we can see, the length of the message is variable.