A Real-Time Augmented Reality System for Industrial Tele-Training

Pierre Boulanger[1], Nicholas Geoganas[2], Xiaowei Zhong[2], and Peiran Liu[2]
[1] Department of Computing Science, University of Alberta, Alberta
[2]School of Information Technology and Engineering, University of Ottawa, Ontario

Abstract

Augmented Reality (AR) is a departure from standard virtual reality in a sense that it allows users to see computer generated virtual objects superimposed over the real world through the use of see-through head-mounted display. Users of such system can interact in the real/virtual world using additional information, such as 3D virtual models and instructions on how to perform these tasks in the form of video clips, annotations, speech instructions, and images. In this paper, we describe two prototypes of a collaborative industrial Tele-training system. The distributed aspect of this system will enables users on remote sites to collaborate on training tasks by sharing the view of the local user equipped with a wearable computer. The users can interactively manipulate virtual objects that substitute real objects allowing the trainee to try out and discuss the various tasks that needs to be performed. A new technique for identifying real world objects and estimating their coordinates in 3D space is introduced. The method is based on a computer vision technique capable of identifying and locating Binary Square Markers identifying each information stations. Experimental results are presented.
Keywords: Augmented Reality, Wearable Computing, Collaborative Virtual Environments, Industrial Training, Target Detection.

1. INTRODUCTION

1.1. Augmented Reality

From human's perspective, the real world is composed of physical objects that people can perceive using their five senses. Unfortunately, the information people get from their senses is limited to what is available in the real world and the hidden nature of objects and information associated with them is usually not available. In some applications, extra information augmenting the real world is essential. For example, when a repairman wants to fix a broken pipe in a wall, a virtual map of the pipes, overlaid on the real wall could be displayed helping him to locate rapidly the broken pipe or to open the wall without damage.

Augmented Reality (AR) is a departure from traditional Virtual Reality (VR) in the sense that VR technologies completely immerse a user inside a completely synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with computer-generated information superimposed over the real world. The ability to combine 3D graphics with the real world in a 3D space is useful in that it enhances a user's perception of the real world by showing its hidden structure. The augmented information could be in the form of annotations, speech instructions, images, video clips, and 3D models. AR systems have been used in many domains of application, including medical, entertainment, military training, engineering design, robotics and tele-operation, and many others [1]. For example, doctors can see virtual ultrasound images overlaid on a patient's body, giving them the equivalent of X-ray vision during a needle biopsy, while a car driver can see the infrared imagery of night vision overlaid on the road ahead. Another domains of application are in the field of assembly, maintenance, and repair of complex machinery, in which computer-generated graphics and text prompts is used to train and assist plant personnel during complex manipulations or equipment maintenance and repair tasks [2][3][4].

For example, an instructional system was developed for the Boeing Company [5], and the telecommunications services product for Bell Canada field-service technicians [6]. Various new applications of this technology can be categorized into three categories: **outdoor and mobile AR, collaborative AR, and commercial AR**.

1.2. *Mobile Computing*

The basic idea behind mobile computing is that users can access and manipulate information anytime and anywhere. With computing devices decreasing in size and with options like wireless networking, a user is no longer limited to his/her physical desktop. Wearable computers are the next generation of mobile computing devices. A typical wearable computer may be composed of low power consumption processor and a battery mounted on a belt or backpack, a head mounted display (HMD), wireless communication hardware and input devices such as a touch pad or chording keyboard or voice input utilities. Progress in wireless networks and the occurrence of wearable computers has made it possible to build a mobile AR system. Some of them have already been demonstrated. A prominent example is Columbia's "Touring Machine" [7], which assists a user in locating places and allowing a user to query information about items of interest, like campus buildings, library, and so on. Several attempts have been made to construct more informative and natural collaborative workspaces with a wearable computer. The NETMAN [8] proposes a collaborative wearable computer system that supports computer technicians in their tasks to maintain a campus network with the help of remote experts.

1.3 Mobile Collaborative Augmented Reality

Computer supported collaborative work (CSCW) allows the computer to be used as a medium for human communication. The combination of Augmented Reality, mobile computing and CSCW produces a new technology, called **M**obile **C**ollaborative **A**ugmented **R**eality (MCAR). 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. There are a number of researchers that have developed tabletop AR systems for face-to-face collaboration. The AR2 Hockey system of Ohshima [9] allows two users to play virtual air hockey against each other. A game of Augmented Reality Chess demonstrates collaboration of the stationary user with the mobile user in Reitmayr's system [10]. However, the users in these systems have to be present in the same physical space. The remote users can not join the shared space and do the collaboration work.

2.2 MCAR for Training

Training costs are a considerable part of expenses for many industries, especially when the customers or employees need to learn to use equipment or devices that are expensive to repair. The problem is compounded when the trainers are far from the would-be trainees. 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 texts, audio clips, video clips, and animations. An example is a prototype Collaborative Virtual Environment developed at the Multimedia Communications Research Lab, University of Ottawa [11] in which a real work task of ATM switch maintenance is simulated. As the emerging training system not only enhances understanding but also saves time and travel costs, it has been considered as very successful. However, it cannot give effective training or support when the learners need it the most – carrying out the task in the real world. If the real world task is complicated, it is difficult for the trainees to memorize every steps 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 these problems. The trend is to provide on-the-job training, giving learners performance support by integrating computer-based guidance and information into the normal working environment through augmented reality technology. In this paper, we will introduce a new method with Mobile Collaborative Augmented Reality (MCAR) technology to address these problems. We will present a prototype system for industrial-training applications in which MCAR embeds computer-based tools is used by the users' in a real training application.

2. VISION-BASED TRACKING

2.1. *Introduction*

In general, AR is a system that combines real and virtual images using some form of real-time tracking devices [1]. Vision based tracking systems uses image processing and computer vision techniques to perform this 3D registration. Two basic technologies are available for accomplishing the fusion of real and virtual. One is based on head-mounted display (HMD) equipment with a wearable computer; the other is based on of overlaying computer-generated graphic images on captured video.

The advantage of using a HMD is its mobility. When a user wearing a HMD and a wearable computer is moving around, the pose of the user's head is tracked in real time by the camera of the HMD. The object recognition and the graphic image rendering are performed by the wearable computer. That technology is most useful in outdoor applications where a computer graphic workstation is non-reachable. For AR applications, the image generation rate must be at least 10 frames per second. According to research by Durlach [12], delays greater than 60ms between head motion and virtual feedback impair the adaptation and the illusion of presence. The processing speed and graphic capability of the existing wearable computer products on the market are very limited which make it almost impossible to reach the real-time constraint of 10Hz.

Accurate dynamic registration is key to an AR system. Magnetic and ultrasonic trackers used in VR are constrained by their inaccuracy and their limited volume. Vision-based AR systems use images captured by the video camera to track the camera's position and orientation relative to fixed objects in the real world. Three-dimensional positioning information is recovered from 2D images based on detecting and tracking certain features in the images. Camera tracking has been extensively investigated in the field of computer vision and has the potential of providing the accurate registration information necessary for AR systems. In computer vision, there are several ways to identify objects in scenes:

- Edge based methods, which are applicable to non-real-time applications due to the complexity of the algorithms**;**
- Reconstruction of the 3D coordinates of a number of points in a scene, given several images obtained by cameras of known relative positions and orientations. These systems are applicable for non-mobile applications;
- Reconstruction of 3D coordinates of objects by detecting fiducial points or landmarks in the image obtained by one camera from a single view. This class of algorithms are the most practical for AR systems due to their simplicity and low cost.

2.2 Motivation & Related Work

Before 1998, all of the real-time vision-based AR systems implemented registration based on landmark detection techniques; these include boundary detection, color segmentation, watershed, and template matching techniques. Landmarks could be solid or concentric circular fiducials, ping-pong balls, LED, or the corners of a big rectangle [4][13][14]. Those landmarks might facilitate lightweight fiducial detection and accurate 3D coordinate localization, but they are usually difficult to identify. This is a key element since in real-life applications, we need to determine what kind of information, or more precisely, what 3D model, annotations or texture images should be superimposed over the real world image. In all those systems, the computer knows in advance what the object with reference to the landmarks is and which 3D virtual model or texture image will be registered with the object seamlessly or what annotation will be displayed on the screen. This constraint hindered the vision-based AR technology to be used in more practical and large scale applications, in which the environment around the user changes dynamically and more than one object in the user's view need to be detected and augmented with computer generated information. This raises a question. How to distinguish one real object from another based on the landmark detection techniques? In 1998, Jun Rekimoto proposed a solution: **CyberCode**. In that vision-based tracking system, 2D matrix code markers are used as landmarks [15]. The markers are 2D barcodes, which can identify a large number of objects by unique IDs. The complexity of detecting the 2D barcode and extracting bar code ID depends on how the 2D matrix code marker is defined. Template matching technique was first proposed by H.Kato and M.Billinghurst in 1999 [16]. The algorithm was implemented and embedded in the ARToolKit software library [16]. An object in the real world is recognized by matching the pattern inside the square marker tagged on the object to a predefined template.

2.3 Camera Tracking

Camera calibration is the process of estimating the intrinsic and extrinsic parameters of a camera, which are used to determine the relationship between what appears on the image plane and where it is located in the 3D world. Registration is the process of acquiring the objects' pose. The camera tracking can be regarded as a continuous update of extrinsic parameters. Camera tracking addresses the problem of accurately tracking the 3D motion of the camera in a known 3D environment and dynamically estimating the 3D camera location relative to the object in the real space.

Vision-based tracking approaches are based on detecting and tracking certain features in images. These features could be lines, corners, or circle points, which can be easily and unambiguously detected in the images and can be associated with objects in the 3D world. Our approach uses the corners of the square markers attached to moving objects for tracking. The overall dataflow of the algorithm is illustrated in Figure 1. The implementation consists of four parts: **image processing, pattern recognition, camera pose estimation, and multimedia information rendering**.

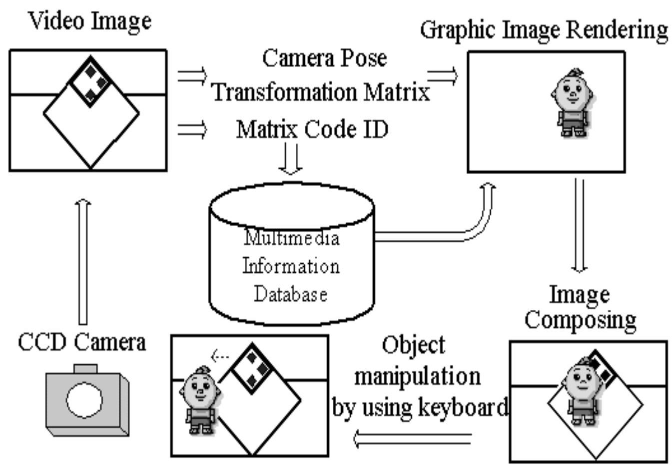


Fig. 1: Augmented reality system workflow.

2.3.1 Image Preprocessing

(1) Image binarization. The program uses an adaptive threshold to binarize the video image. Binary images contain only the important information, and can be processed very rapidly.
(2) Selecting the quadrilateral regions. These regions become candidates for the square marker.
(3) Searching and recording the four corners of the quadrilateral region found in step (2).

2.3.2 Pattern Recognition

(1) The system recognizes the 2D binary code pattern inside the square marker region.
(2) Extracting the binary code

Our algorithm is based on the following observation: Given the four vertices of the marker region, the projection of every point on the marker can be computed as a linear combination of the projections of the four vertices. The reference frame is defined as a non-Euclidean, affine frame. In the system, the marker pattern is defined as a 4x4 matrix. Every grid in the matrix represents one bit of the binary code of the marker. The whole pattern is in black and white color. The grid in black represents 1, and the grid in white represents 0 (Figure 2). According to the theorem in [17], the division ratio is an affine invariant, which means

$$(ABC) = (T(A)T(B)T(C)).$$

If A, B and C are three different arbitrarily chosen collinear points of the plane (or space) and T is an arbitrary affine transformation, (ABC) is the division ratio of the collinear three points A, B and C.
Clearly, since the relative location of every grid in the pattern is predefined, it is easy to calculate the central point coordinates of the grid in the projective plane with reference to the projections of four vertices of the marker. Consequently, the color or value of each grid in the pattern is determined by the color of the corresponding pixels in the binarized image.

(3) Error control

In the implementation of the algorithm, the block sum checking and the Hamming code checking were used for error detection and error correction. If a code error is detected, the system can either drop the current image and do the image preprocessing again on the next image from step (1), or correct the error bits.
The pattern of a marker represents a 16-bits binary code. In the block sum checking (Figure 2.a), 5 bits are used to represent the marker ID. Four bits determine the orientation of the marker in the camera's dynamic view. 6 bits are

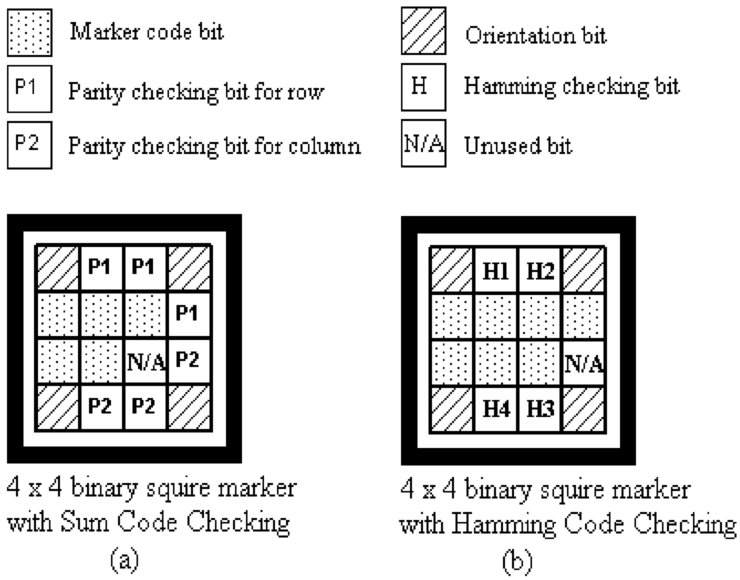


Fig. 2: Binary Square Marker.

for the block sum checking. In the marker pattern, the data block is of 3 rows and 3 columns. The block sum checking can correct any single bit error in the bit block. In the Hamming code checking (Figure 2.b), 7 bits are used to represent the marker ID. Four bits determine the orientation of the marker in the camera's dynamic view. In the rest 5 bits, 4 bits are for the error checking. The Hamming code checking can correct any single bit error.

(4) Determination of marker orientation

Four bits at the corners of the pattern determine the orientation of the marker in the camera's view dynamically. The system can keep tracking the orientation of the square marker, register the virtual model on the marker even if the marker rotate, and read the binary code of the marker in correct order.

2.3.3 Camera Pose Estimation

The recognized marker region is used for estimating the camera position and orientation relative to the marker tagged to the object. From the coordinates of four corners of the marker region on the projective image plane, a matrix representing the translation and rotation of the camera in the real world coordinate system can be calculated. Several algorithms have been developed over the years. Examples are the Hung-Yeh-Harwood pose estimation algorithm [18] and the Rekimoto 3-D position reconstruction algorithm [15].

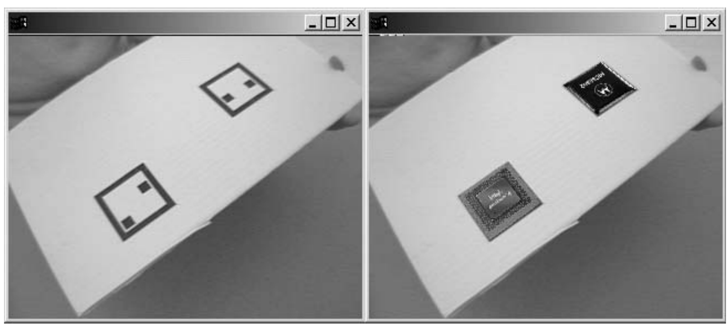
We currently use the algorithm proposed by Kato-Billinghurst-Weghorst-Furness [16], which incorporates lens distortion. Once the transformation matrix is calculated, all kinds of multimedia information corresponding to the marker could be rendered over the real world image through human-machine interfaces.

2.3.4 Multimedia Information Rendering

Currently, our system is implemented to embed the following multimedia information into the real world from the user's view.

(1) Text, such as character description of physical objects and instruction for performing physical tasks in form of annotation.
(2) Audio, such as speech instructions.
(3) Image.
(4) Real-time video stream.
(5) 3D model, such as OpenGL model and VRML model.

The 7 bits code of the pattern can identify 2^7 different objects. However, how to decide what computer generated information should be rendered on which object labeled with a square marker becomes an issue to solve. We implemented a multimedia database using C++. In the database, the key word for each record is the marker code ID, and the property fields are the media information the system will render when the corresponding marker labeled object runs into the camera's view (Figure 3).



(a) (b)

Fig. 3 (a) Image of a white board, captured from a CCD camera. (b) Computer generated graphic image superimposed on the white board.

2.4 Comparison With Other Visual Marking Technologie

Some systems, like the Augmented Reality for Construction (ARC) [19], use a 1D barcode as the marker to handle a large number of real world objects. The drawback is that an extra barcode reader is needed for each user. Another similar visual marking method adds color into the 1D barcode. Each bar in a special color represents a certain value. With the same amount of bars, the color barcode can identify more objects than a traditional black and white color barcode. One constraint is that the marker must be located in daylight or fluorescent environment; otherwise, the color of the bar will not be recognized correctly. Two existing vision-based AR systems use 2D square markers.
One is the CyberCode system. Another is a template matching system. In the template matching system [16], the pattern of the marker region is compared with predefined templates, which were given to the system before identifying a specific marker ID. The pattern template could be user names, characters or an image. A significant disadvantage of the method is that it does not have a specific way to differentiate one pattern from the other. The ability to correctly pick up the template, matching the marker pattern, mostly depends on the resolution of the camera and the algorithm to match the pattern. In the CyberCode system, Jun Rekimoto proposed a method to use a 2D matrix code marker as landmark in his vision-based system [15]. The complexity of detecting the 2D barcode and extracting bar code ID depends on how the 2D matrix code marker is defined. Compared to the marker definition method we proposed, the 2D matrix barcode has some limitations. Every marker has a guide bar aside, which helps to locate the four corners of the marker and determine the orientation of the marker. So, an extra step is needed to find the guide bar. Since the marker code is a binary code, error correction is possible to implement. The problem is that once an error bit is detected, the system will have to skip the current image and extract the marker code from the next image instead of correcting the error bit. In our experimental system, we found that the error bit doesn't occur independently from one image to another in that in most cases the error is caused by the camera capability, lighting in the environment or the material made of the marker. Therefore, dropping the currently processed image when an error occurs will probably not get the correct marker code in the consequent video images.

A good method to solve the problem mentioned above is to add an error control function into the pattern recognition algorithm. Block sum checking and Hamming code checking are two alternatives to detect up to 2 bits errors and correct any single bit error in the 2D binary code block.
The proposed Binary Square Marker recognition algorithm was tested using markers in different patterns. The maximum visible ranges of square markers in different size placing in different orientations relative to the camera (tilted degree) were measured. The object recognition algorithms with and without error correction function were tested individually. From the test results (Figure 4), it is clear to see that the visible ranges of the markers are affected by the tilted degree of the markers. As the markers are more tilted, the patterns are less visible. As a result, the recognition becomes more unreliable. The larger the size of physical markers, the further away the patterns of the markers can be detected. The visible range of the markers with an error correction function is 12.8% larger on the average than that of the markers without an error correction function. In conclusion, by using the marker of the same size, the Binary Square Marker recognition algorithm we proposed can make the marker being recognized in a larger range in comparison with other 2D square marker recognition algorithms without error control. The pattern of markers with larger black regions (black color grids are connected) can be recognized correctly in a larger range in comparison with markers, which has no connected black color grids. Therefore, we can say that the pattern of the marker also affects somewhat the visible range of the object tagged by the marker.

A limitation of this marking system is that the context information of the objects represented by tagged markers can only appear when the markers are in the user's view. This limits the size and the volume of movement of the objects. What's more, if the marker is partially covered by other objects the marker can't be recognized by the system, and the graphic image of the 3D model and annotation will not be displayed to the scene. In the AR system, the square shaped markers are used for recognizing and tracking objects. This raises a question. Why do we use square shaped markers? In a vision-based AR application, the object labeled with a marker could be of any size or shape. In order to make our system recognize not only big but also small objects, we should minimize the marker area and the marker code length as far as possible.
Suppose the marker ID is a N-bit code with X rows and Y columns.

$$X \times Y + X + Y = N + X + N / X$$

If the block sum checking function is used, the number of parity bits is X+Y. The total marker code is X x Y + X + Y bits.

$$X \times Y = N$$

In order to minimize the total marker code length, X must be equal to Y.

$$(N + X + N / X)' = 1 - N / X^2$$

$$1 - N / X^2 = 0$$

$$X = Y = \sqrt{N}$$

If the marker code length is fixed, the square shaped marker has shorter perimeter and smaller area than the markers in rectangle.

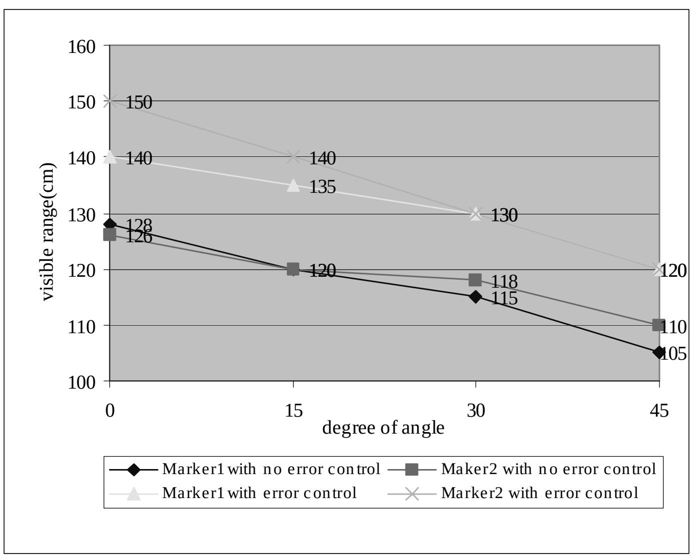


Fig.4 Test results of vision-based tracking using Binary Square Markers

3 AR for ATM Maintenance Training

3.1 The Task: ATM Switch Maintenance

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.

3.2 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.

3.3 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.

3.3.1 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.

3.3.2 Collaboration Functions

From this observation, we 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:

(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.

(2) **Remote Presence**: If the trainer is able to see what a remote trainee sees, as if she/he were physically present,

Fig.5 A trainee working with a wearable computer

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.

(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.

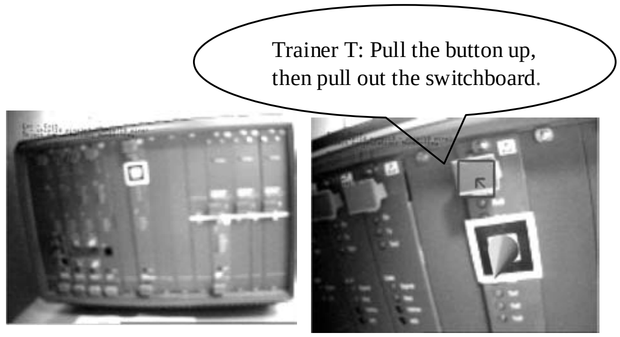
(4) **Remote Pointing**: The ability to control a remote cursor enables users to point at objects in other users' view. Such objects can either be virtual objects (a wire in a wiring diagram) or real objects captured by the camera of a wearable computer. Remote pointing can increase the effectiveness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

(5) **Image Freezing**: 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 PROTOTYPE

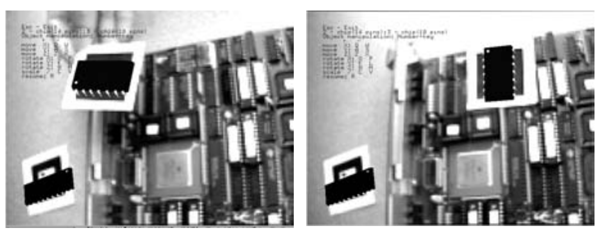
Figure 5 shows a trainee, equipped with a wearable computer, who runs the training system and works with a switchboard. Let’s refer to him as the local trainee A. Trainer T and remote trainee B who are in different locations are working on the remote workstations, watching A’s action through the shared view of the local trainee. All of them are equipped with a microphone and are able to talk through live audio. Figure 6 shows the screenshots of a collaborative training exercise running with our tool. The first thing that the trainee learns is how to pull out the switchboard. 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 6(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 6(b)). Using a remote pointer, trainer T directs him to push up the button on 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 6(c), trainer A pulls out the switchboard as directed. The switchboard is put on the table with two virtual chips next to it (figure 6(d)). As shown in figure 6(e), trainee A picks up the chip with seven pairs of pins and moves it close to the switchboard. As shown in figure 6(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 6(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 6(h)). Through this interactive "real" training exercise the trainees can gain valuable experience

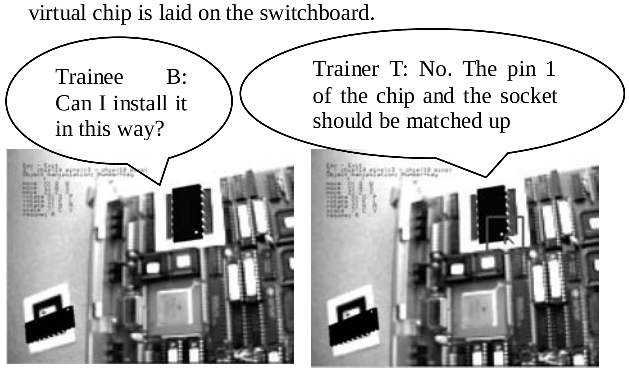


(a) Camera is pointing at an ATM Switch; (b) Directing the trainees to pull out the switchboard

(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.

Fig.6 Screenshots for the training prototype.

5 Design System

5.1 Design Considerations

In this system, the visualization end points that allow the user to participate in the collaborative augmented reality workspace include: Wearable computers worn by a trainee in the real-world task; one or more desk-bound workstations used by the trainers and other trainees in the remote location. All of these visualization end points are connected with 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. To be easy to add nodes to the architecture and maintain consistency among displays, the infrastructure of the system is built in the form of a client-server architecture, in which the wearable computer and workstation work as a client to handle the user's input event and render the graphic images while the server provides an information sharing mechanism. 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 and is able to support a maximum of 64 clients at a time.

5.2 Overall Architecture

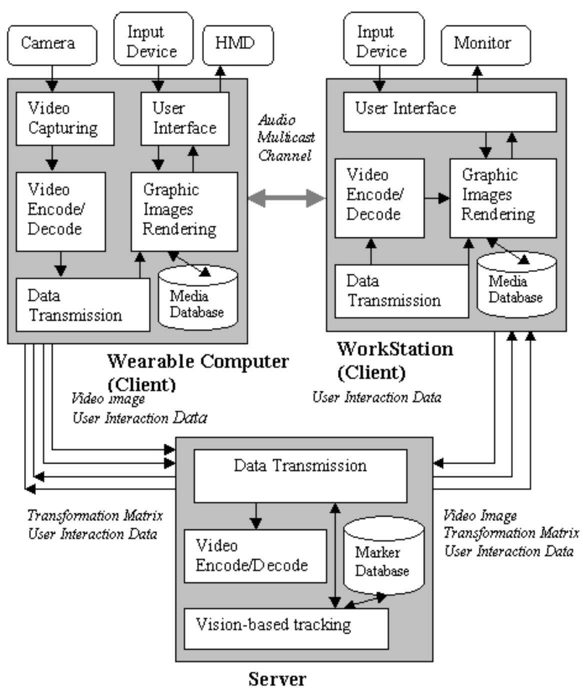


Fig.7 Overall architecture

As shown in Figure 7, a typical architecture includes three major components: a wearable computer, a server and one or more workstations. They are connected via a network. 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 Figure 7, 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 *serve*r 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. The camera transformation matrix and the marker's identification extracted from the video image are then sent back to the *wearable computer*. The *wearable computer* retrieves the 3D models from the *media database* according to the marker's identification, 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, marker'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 retrieved from the media database based on the marker's identification and rendered by the *graphic images rendering* module, together with the decoded video image.

5.3 Hardware Implementation

2.1.1. 5.3.1 First Hardware Configuration

The first test bed for the prototype consisted of three Pentium III workstations equipped with GetForce2Go 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.

The Mobile IV Xybernaut wearable computer that we are using turned out to lack 3D hardware and be low on CPU power. Running the prototype with this test bed, the view update rate is about 3 frames/second to the video image with 320 pixels by 240 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.

2.1.2. 5.3.2 Second Hardware Configuration

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.

Running the Collaborative Augmented Reality (CAR) training prototype with this test bed, the view update rate varies in different configurations. 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. 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.

6. CONCLUSION AND FUTURE WORK

The experimental results show that a wearable computer without 3D hardware capacity, for instance the Xybernaut Mobile IV wearable computer, is not able to satisfy the requirements of a real-time AR application. 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 NVIDIA GetForce4 graphic card and an I-glass see-through stereoscopic color Head Mounted Display mounted with a camera.

In this paper, we addressed the problem of identifying a great number of real world objects with a robust marking approach by using computer vision AR techniques. The binary code error detection and correction functions used in the marker recognition algorithm make the algorithm more robust to lighting conditions. We also presented a prototype developed for industrial Tele-training which features augmented reality and collaboration.

In the near future, we intend to improve the prototype in various ways. The user interface and interaction methods need enhancement to make manipulation to the graphic objects easier, and to convey more awareness to the collaboration.

ACKNOWLEDGMENTS

This work was supported in part by the National Capital Institute of Telecommunications (NCIT). The contributions of staff and resources of the National Research Council of Canada are also gratefully acknowledged.

REFERENCES

[1] *R.T. Azuma*: SIGGRAPH '95 Course Notes: A Survey of Augmented Reality. Los Angeles, Association for Computing Machinery, 1995.

[2] MIT Media Laboratory, Augmented Reality, http://wex.www.media.mit.edu/projects/wearables/augmented- reality.html

[3] *J. Kustaborder and R. Sharma*: Experimental Evaluation of Augmented Reality for Assembly Training, IWAR'99, San Francisco, October 20-21, 1999.

[4] *W.N. Hoff:* A Computer vision based registration techniques for augmented reality. In Proceedings of Intelligent Robots and Computer Vision XV, SPIE Vol.2904, Boston, MA, pp. 538-548 Nov 18-22, 1996.

[5] Instructional Systems - Augmented reality, http://www.boeing.com/defense-space/military/aerosupport/instruct/augmented.htm

[6] Bell Canada, IBM Canada and Xybernaut(R) Launch World's First Large-Scale Market Trial of Wearable Computers. http://www2.software.ibm.com/casestudies/swcs.nsf/customername

[7] *S. Feiner, B. MacIntyre, T. Höllerer, and T. Webster:* A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment, In Proc. ISWC '97 (Int. Symposium on Wearable Computers), Cambridge, MA, October 13-14, 1997.

[8] *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.

[9] *T. Ohshima, K. Satoh, H. Yamamoto, and H. Tamura*: AR^2 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] *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, In Proc. 4th WORLD MULTICONFERENCE on Circuits, Systems, Communications & Computers (CSCC 2000), Greece, July 2000.

[12] *N.I. Durlach*: Psychophysical Considerations in the Design of Human-Machine Interfaces for Teleoperator and Virtual-Environment Systems. In Medicine Meets Virtual Reality II: Interactive Technology & Healthcare: Visionary Applications for Simulation Visualization Robotics (pp. 45-47). San Diego, CA, USA: Aligned Management Associates, 1994.

[13] *Y. Cho, J. Lee, and U. Neumann*: A Multi-ring Color Fiducial System and An Intensity-invariant Detection Method for Scalable Fiducial-Tracking Augmented Reality. In IWAR, 1998.

[14] *A. State, G. Hirota, D.T. Chen, B. Garrett, and M. Livingston*: Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking, SIGGRAPH 1996, New Orleans, LA, pp.429-438, 1996.

[15] *J. Rekimoto and Y. Ayatsuka*: CyberCode: Designing Augmented Reality Environments with Visual Tags, Designing Augmented Reality Environments. In DARE (2000).

[16] *M. Billinghurst and H. Kato*: Collaborative Mixed Reality. In Proceedings of International Symposium on Mixed Reality (ISMR '99). Mixed Reality--Merging Real and Virtual Worlds, pp. 261-284, 1999.

[17] *I. Herman*: The Use of Projective Geometry in Computer Graphics. Lecture Notes in Computer Sciences No. 564, eds. G. Goos and J. Hartmanis, Springer Verlag (1992).

[18] *Y.Hung, P.Yeh, and D.Harwood*: Passive Ranging to Known Planar Point Sets, Proceeding of IEEE International Conference on Robotics and Automation, Vol.1, St.Louis, Missouri, 25-28 March, pp.80-85, 1985.

[19] *A. Webster, S. Feiner, B. MacIntyre, M. Massie, and T. Krueger*: Augmented Reality in Architectural Construction, Inspection, and Renovation. In Proc. ASCE Computing in C.E., 1996.