Computers and Graphics 23(6), 1999
also presented at EGVE'99, Vienna, Austria, 1999

An Optically Based Direct Manipulation Interface for Human-Computer Interaction in an Augmented World

G. Klinker[1], D. Stricker[2], and D. Reiners[2]

[1]Moos 2, 85614 Kirchseeon, Germany, klinker@in.tum.de

[2]Fraunhofer Institute for Computer Graphics (FhG-IGD), Rundeturmstr. 6, 64283 Darmstadt, Germany, {stricker, reiners}@igd.fhg.de

Abstract. Augmented reality (AR) constitutes a very powerful three-dimensional user interface for many "hands-on" application scenarios in which users cannot sit at a conventional desktop computer. To fully exploit the AR paradigm, the computer must not only augment the real world, it also has to accept feedback from it. Such feedback is typically collected via gesture languages, 3D pointers, or speech input - all tools which expect users to communicate with the computer about their work at a meta-level rather than just letting them pursue their task. When the computer is capable of deducing progress directly from changes in the real world, the need for special abstract communication interfaces can be reduced or even eliminated. In this paper, we present an optical approach for analyzing and tracking users and the objects they work with. In contrast to emerging workbench and metaDESK approaches, our system can be set up in any room after quickly placing a few known optical targets in the scene. We present three demonstration scenarios to illustrate the overall concept and potential of our approach and then discuss the research issues involved.

1 Introduction

Augmented reality (AR) constitutes a very powerful three-dimensional user interface for many "hands-on" application scenarios in which users cannot sit at a conventional desktop computer. Users can continue their daily work involving the manipulation and examination of real objects, while seeing their surroundings augmented with synthetic information from a computer. These concepts have been demonstrated for construction and manufacturing scenarios like the computer-guided repair of copier machines [3], the installation of aluminum struts in diamond shaped spaceframes [12], for electric wire bundle assembly before their installation in airplanes [1], and for the insertion of a lock into a car door [8].

To fully exploit the AR paradigm, the computer must not only augment the real world but also accept feedback from it. Actions or instructions issued by the computer cause the user to perform actions changing the real world – which, in turn, prompt the computer to generate new, different augmentations. Several prototypes of two-way human-computer interaction have been demonstrated. In the ALIVE project [7], users

interact with a virtual dog which follows them and sits down on command. In Feiner et al.'s space frame construction system, selected new struts are recognized via a bar code reader, triggering the computer to update its visualizations. In a mechanical repair demonstration system, Breen et al. use a magnetically tracked pointing device to ask for specific augmentations regarding information on specific components of a motor [9]. Reiners et al. use speech input to control stepping through a sequence of illustrations in a doorlock assembly task [8].

These communication schemes expect users to communicate with the computer at a meta-level about their work rather than just letting them pursue their task. In many real work situations, user actions cause the world to change. When such world changes are captured directly by appropriate computer sensors, the need for abstract communication interfaces can be reduced or even eliminated. The metaDESK system [11] explores such concepts, using graspable objects to manipulate virtual objects like b-splines and digital maps of the MIT campus. It uses an elaborate special arrangement of magnetic trackers, light sources, cameras, and display technology to present the virtual data on a nearly horizontal, planar screen.

In this paper, we present an approach which merges the real-object manipulation paradigm with full three-dimensional AR. Using only optical techniques for analyzing and tracking users or real objects, we do not require elaborate hardware technology. Our demonstrations can be arranged in any room after quickly placing a few known optical targets in the scene, requiring only moderate computing equipment, a miniaturized camera, and a head-mounted display. With such setups, users are then able to control the AR-system simply by manipulating objects or reference symbols in the real world and via simple gestures or spoken commands. With this approach we extend the concepts of the metaDESK towards augmenting more complex realities than planar desktops, arising from a two-dimensional planar desktop reality into a three-dimensional world.

Section 2 presents several demonstrations. Section 3 provides a system overview. Section 4 briefly refers to issues of live camera tracking which are reported in more detail in other papers. Section 5 presents the main focus of this paper, the real-time analysis of real world changes due to user actions. Section 6 proposes schemes for adding 3D GUIs to the real world to support man-machine communication that cannot be automatically deduced from real world changes. Section 7 discusses schemes to automatically account for moving foreground objects, such as users' hands, when merging virtual objects into the scene.

2 Demonstrations

The subsequent three scenarios illustrate the overall concept and potential of optically-based direct manipulation interfaces for AR applications.

2.1 Mixed virtual/real mockups

In many industries (e.g. architecture, automotive design), physical models of a design are built to support the design process and the communication with the customer. Such mockups are time-consuming and expensive to create and thus are typically built only after many of the preliminary decisions have already been made. AR provides the opportunity to build mixed mockups in several stages, using physical models for the already maturing components of the design and inserting virtual models for the currently evolving components. With such mixed prototypes, designers can visualize their progressing models continuously.

Our first demonstration shows a toy house and two virtual buildings. Each virtual house is represented by a special marker in the scene, a black square with an identification label. By moving markers, users can control the position and orientation of individual virtual objects. A similar marker is attached to the toy house. The system can thus track the location of real objects as well. Figure 1a shows an interactively created arrangement of real and virtual houses. Figure 1b shows a VRML-model of St. Paul's Cathedral being manipulated in similar fashion via a piece of cardboard with two markers.

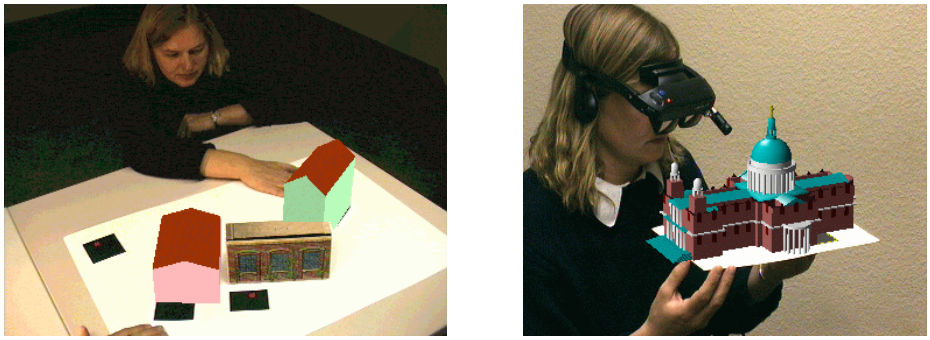

Fig. 1. a) Manipulation of virtual and real objects. b) Manipulation of a model of St. Paul's Cathedral via a piece of card board. (Reprint of Fig. 10b in [10]).

These demonstrations are meant to show that, using our AR-system, preliminary new designs can be combined with existing physical prototypes. Users can analyze a proposed design and move both real and virtual objects about, asking "What if we moved this object overthere?", "What if we exchanged the object with an other one in a different style, color or shape?". The system provides users with intuitive, physical means to manipulate both virtual and real objects without leaving the context of the physical setup. The system keeps track of all virtual and real objects and maintains the occlusion relationships between them.

2.2 Augmented Maps and Cityscapes

Pursuing the interactive city design scenario further, AR can integrate access to many information presentation tools into a reality-based discussion of proposed architectural plans.

In the context of the European CICC project we have used AR to visualize how a proposed millennium footbridge across the Thames will integrate into the historical skyline of London close to St. Paul's cathedral (Figure 2a)[5].

Using a map of London, a simple model of the houses along the river shore, and a CAD model of the proposed bridge and St. Paul's cathedral, we have generated a 3D presentation space on a conventional table which pulls all information together, giving viewers the ability to move about to inspect the scene from all sides while interactively moving the bridge (Figure 2b) and choosing between different bridge styles. A hand wave across a 3D camera icon provides access to another information presentation tool: a movie loop with a pre-recorded augmented video clip showing the virtual bridge embedded in the real scene.[1]

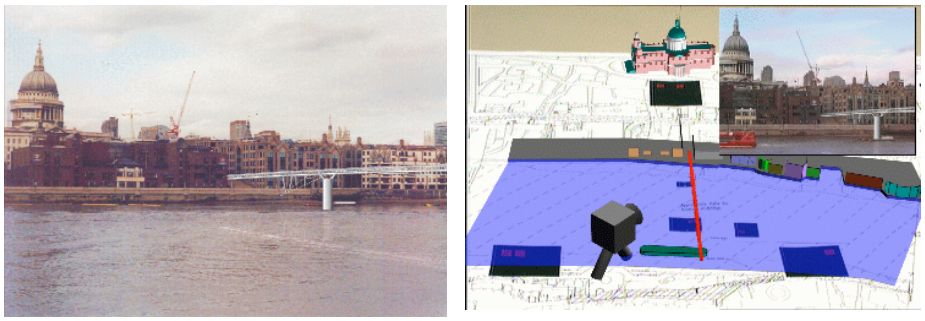

Fig. 2. a) Proposed millennium footbridge across the river Thames. b) An augmented map of London including a movie loop with an AR video clip.

2.3 Augmented Tic Tac Toe

We show more elaborate interaction schemes in the context of a Tic Tac Toe game (Figure 3). The user sits in front of a Tic Tac Toe board and some play chips. A camera on his head-mounted display records the scene, allowing the AR-system to track head motions while also maintaining an understanding of the current state of the game, as discussed in sections 4 and 5.

The user and the computer alternate placing real and virtual stones on the board (Figure 3a). When the user has finished a move, he waves his hand past a 3D "Go" button (Figure 3b) or utters a "Go" command into a microphone to inform the computer that he is done. It is important for the computer to wait for such an explicit

[1] Due to the complexity of the graphical models, this demonstration runs on a high-end graphical workstation (SGI Reality Engine).

"Go" command rather than taking its turn as soon as the user has moved a stone. After all, the user might not have finalized his decision. When told to continue, the computer scans the image area containing the board. If it finds a new stone, it plans its own move. It places a virtual cross on the board where the new stone should go and writes a comment on the virtual message panel behind the game. If it could not find a new stone or if it found more than one, it asks the user to correct his placement of stones.

The technology presented in these demonstrations forms the basis for many maintenance, construction and repair tasks where users do not have access to conventional computer interfaces, such as a keyboard or mouse.

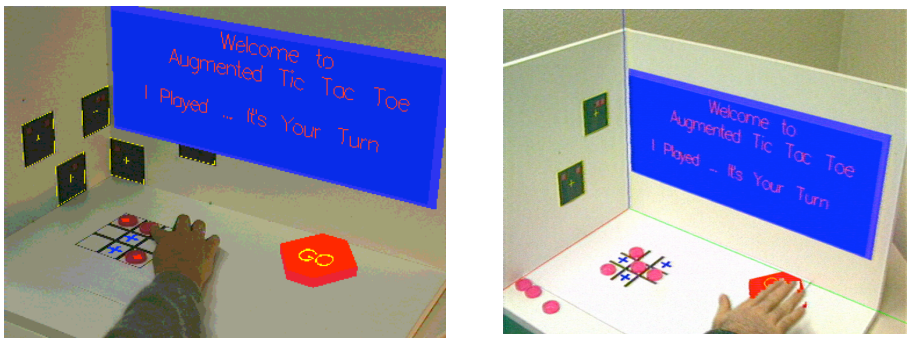

Fig. 3. Augmented Tic Tac Toe. a) Placement of a new stone. b) Signalling the end of user action.

3 The System

Our AR-system works both in a monitor-based and a HMD-see-through setup. It runs on a low-end graphics workstation (SGI O2). It receives images at video rate either from a minicamera that is attached to a head-mounted display (Virtual IO Glasses) (see Figure 1b) or from a user-independent camera installed on a tripod. The system has been run successfully with a range of cameras including high quality Sony 3CCD Color Video Cameras, color and black-and-white mini cameras and low-end cameras that are typically used for video conferencing applications (e.g., an SGI IndyCam). The resulting augmentations are shown on a workstation monitor, embedded in the video image and/or on a head mounted display. In the HMD, the graphical augmentations can be seen in stereo without the inclusion of the video signal ("see through mode").

At interactive rates, our system receives images and submits them to several processing steps. Beginning with a camera calibration and tracking step, the system determines the current camera position from special targets and other features in the scene. Next, the image is scanned for moving or new objects which are recognized according to predefined object models or special markers. Third, the system checks whether virtual 3D buttons have been activated, initiating the appropriate callbacks to

modify the representation or display of virtual information. Finally, visualizations and potential animations of the virtual objects are generated and integrated into the scene as relevant to the current interactive context of the application. (For details, see [8].)

4 Live Optical Tracking of User Motions

The optical tracker operates on live monocular video input. To achieve robust real-time performance, we use simplified, "engineered" scenes, placing black rectangular markers with a white boarder at precisely measured 3D locations (see Figures 1a, 2b, and 3). In order to uniquely identify each square, the squares contain a labeling region with a binary code. Any subset of two targets typically suffices for the system to find and track the squares in order to calibrate the moving camera at approximately 25 Hz. (For details see [5,6,10].)

5 Detection of Scene Changes

Next we search the image for mobile objects using two different approaches. We either search for objects with special markers or we use model-based object recognition principles.

5.1 Detection and tracking of objects with special markers

When unique black squares are attached to all mobile real and virtual objects and if we assume that the markers are manipulated on a set of known surfaces, we can automatically identify the marks and determine their 3D position and orientation by intersecting the rays defined by the positions of the squares in the image with the three-dimensional surfaces on which they lie.

If the markers are manipulated in mid-air rather than on a known surface, more sophisticated approaches are needed, such as stereo vision or the computation of the 3D target location from its projected size and shape.

5.2 Detection of objects using object models

In the Tic Tac Toe game, we use model-based object recognition principles to find new pieces on the board. Due to the image calibration, we know where the game board is located in the image, as well as all nine valid positions for pieces to be placed. Furthermore, the system has maintained a history of the game. It thus knows which positions have already been filled by the user or by its own virtual pieces. It also knows that the game is played on a white board and that the user's stones are red. It thus can check very quickly and robustly which tiles of the board are covered with a stone, i.e. which tiles have a significant number of pixels that are red rather than white. Error handling can consider cases in which users have placed no new stone or

more than one new stone – or whether they have placed their stones on top of one of the computer's virtual stones.

This concept extends to other applications requiring more sophisticated object models, such as the modification of a Lego construction or the disassembly of a motor. In an envisioned scenario, the computer will instruct the technician to remove one particular piece at a time, probably even highlighting the image area showing the piece. Thus, it can henceforth check whether the object has been removed, uncovering objects in the back that had been occluded before. This approach is straightforward, if the camera is kept steady during the process. But even for mobile, head-mounted cameras, their calibrated use dynamically provides the correct search area and the appropriate photometric decision criteria for every image due to the fact that a known object model exists.

5.3 Comparison

Both approaches have their merits and problems. Attaching markers to a few real objects is an elegant way of keeping track of objects even when both the camera and the objects move. The objects can have arbitrary textures that don't even have to contrast well against the background – as long as the markers can be easily detected. Yet, the markers take up space in the scene; they must not be occluded by other objects unless the attached object becomes invisible as well. Furthermore, this approach requires a planned modification of the scene which generally cannot be arranged for arbitrarily many objects. Thus it works best when only a few, well-defined objects are expected to move. In a sense the approach is in an equivalence class with other tracking modalities for mobile objects which require special modifications, such as magnetic trackers or barcode readers.

Using a model-based object recognition approach is a more general approach since it does not require scene modifications. Yet, the detection of sophisticated objects with complex shape and texture has been a long standing research problem in computer vision, consuming significant amounts of processing power. Real-time solutions for arbitrarily complex scenes still need to be developed.

Thus, the appropriate choice of algorithm depends on the requirements of an application scenario. In many cases, hybrid approaches including further information sources such as stationary overhead surveillance cameras that track mobile objects are most likely to succeed. Kanade's Z-keying system is a promising start in this direction [4].

6 Virtual GUIs in the Real World

In many application contexts, GUIs such as buttons and menus have proven to be very useful for computers to guide users flexibly through a communication process. Such interfaces should also be available in AR applications. Feiner et al. have demonstrated concepts of superimposing "windows on the world" for head mounted displays [2].

Rather than replicating a 2D interface on a wearable monitor, we suggest embedding GUI widgets into the three-dimensional world. Such approach has a tremendous amount of virtual screen space at its disposal: by turning their head, users can shift their attention to different sets of menus. Furthermore, the interface can be provided in the three-dimensional context of tasks to be performed. Users may thus remember their location more easily than by pulling down several levels of 2D menus.

As a first step, we demonstrate the use of 3D buttons and message boards. For example, we use such buttons to indicate the vantage points of pre-recorded camera clips of the river Thames. The virtual GO-button and a message board for the Tic Tac Toe game use similar means. Virtual buttons are part of the scene augmentations when users turn their head in the right direction.

When virtual 3D buttons become visible in an image, the associated image area becomes sensitive to user interaction. By comparison with a reference image, the system determines whether major pixel changes in the area have occurred due to a user waving a hand across the sensitive image area. Such an approach works best for stationary cameras or small amounts of camera motion, if the button is displayed in a relatively homogenous image area.

3D GUIs are complementary to the use of other input modalities, such as spoken commands and gestures. Sophisticated user interfaces will offer combinations of all user input schemes.

7 Scene Augmentation Accounting for Occlusions due to Dynamic User Hand Motions

After successful user and object tracking, the scene is augmented with virtual objects according to the current interaction state. To integrate the virtual objects correctly into the scene, occlusions between real and virtual objects must be considered. To this end, we use a 3D model of the real objects in the scene, rendering them invisibly at their current location to initialize the z-buffer of the renderer.

During user interactions, the hands and arms of a user are often visible in the images, covering up part of the scene. Such foreground objects must be recognized because some virtual objects could be located behind them and are thus occluded by them. It is extremely disconcerting to users, if this occlusion relationship is ignored in the augmentations, making it very difficult to position objects precisely. We currently use a simple change detection approach to determine foreground objects. While the camera doesn't move, we compare the current image to a reference image determining which pixels have changed significantly. Looking for compact, large areas of change, we discount singular pixel changes as well as thin linear streaks of pixel changes which can be the result of camera noise and jitter. Z-buffer entries of foreground pixels are then set to a fixed foreground value. In the Tic Tac Toe game, this algorithm allows users to occlude the virtual "Go"-button during a hand waving gesture (Figure 3b).

The change detection approach currently is still rather basic. It has problems dealing with some aspects of live interaction in a real scene, such as proper handling

of the shadows cast by the users' hands. Furthermore, the current scheme only works for monitor-based AR setups using a stationary camera. Another current limitation is the lack of a full three-dimensional analysis of the position of foreground objects. Such analysis requires a depth recovery vision approach, such as stereo vision [4] performing in real-time. Nevertheless, even the use of basic change detection heuristics yields significant improvements to the immersive impression during user.

8 Discussion

How will AR actually be used in real applications once the most basic technological issues regarding high-precision tracking, fast rendering and mobile computing have been solved? In this paper, we have presented a set of demonstrations illustrating the need for a new set of three-dimensional user interface concepts which require that computers be able to track changes in the real world and react to them appropriately.

We have presented approaches addressing problems such as tracking mobile real objects in the scene, providing three-dimensional means for users to manipulate virtual objects, and also to present three-dimensional sets of GUIs. Furthermore, we have discussed the need to detect foreground objects such as a user's hands.

We have focused on optical scene analysis approaches. We expect computer vision techniques to play a strong role in many AR solutions due to their close relationship to the visual world we are trying to augment and due to people's strongly developed visual and spatial skills to communicate in the real world. Furthermore, the technical effort to set up a scene for an optically-based AR system is minimal and relatively unintrusive since little special equipment and cabling has to be installed in the environment. Thus, the system is easily portable to new places.

Our current technical solutions are just the beginning of a new direction of research activities further addressing the problem of tracking changes in the real world. We have developed and tested several algorithmic variations to address the issues, resulting in a toolbox of approaches geared towards the fast analysis of video data for the current set of demonstrations. In most cases, different approaches each come with their own set of advantages and drawbacks. More sophisticated approaches will become available as the processor performance increases, including more sophisticated multi-camera approaches that currently run only on special-purpose hardware at reasonable speeds.

Yet, even though the current algorithms need to be generalized, they have been able to illustrate the overall 3D human-computer interaction issues that need to be addressed. Building upon these approaches towards more complete solutions will generate the basis for exciting AR applications.

ACKNOWLEDGMENTS

The research is financially supported by the European CICC project (ACTS-017) in the framework of the ACTS programme. The laboratory space and the equipment is

provided by the European Computer-industry Research Center (ECRC). The CAD Model of the bridge (Figure 2a) was provided by Sir Norman Foster and OveArup and Partners. We retrieved the virtual model of St. Paul's cathedral from Platinum Pictures' 3D Cafe (http://www.3dcafe.com).

REFERENCES

1. D. Curtis, D. Mizell, P. Gruenbaum, and A. Janin. Several Devils in the Details: Making an AR App Work in the Airplane Factory. 1rst International Workshop on Augmented Reality (IWAR'98), San Francisco, 1998.

2. S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world: 2D windows for 3D augmented reality. UIST'93, pages 145-155, Atlanta, GA, 1993.

3. S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based augmented reality. Communications of the ACM (CACM), 30(7):53-62, July 1993.

4. T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. 15th IEEE Computer Vision and Pattern Recognition Conference (CVPR), 1996.

5. G. Klinker, D. Stricker, and D. Reiners. Augmented reality for exterior construction applications. In W. Barfield and T. Caudell, eds., Augmented Reality and Wearable Computers. Lawrence Erlbaum Press, 1999.

6. G. Klinker, D. Stricker, and D. Reiners. Augmented Reality: A Balance Act between High Quality and Real-Time Constraints. 1. International Symposium on Mixed Reality (ISMR'99), Y. Ohta and H. Tamura, eds., "Mixed Reality – Merging Real and Virtual Worlds", March 9-11, 1999.

7. P. Maes, T. Darrell, B. Blumberg, and A. Pentland. The ALIVE system: Full-body interaction with autonomous agents. Computer Animation '95, 1995.

8. D. Reiners, D. Stricker, G. Klinker, and S. Müller. Augmented Reality for Construction Tasks: Doorlock Assembly. 1rst International Workshop on Augmented Reality (IWAR'98), San Francisco, 1998.

9. E. Rose, D. Breen, K.H. Ahlers, C. Crampton, M. Tuceryan, R. Whitaker, and D. Greer. Annotating real-world objects using augmented reality. Computer Graphics: Developments in Virtual Environments. Academic Press, 1995.

10. D. Stricker, G. Klinker, and D. Reiners. A fast and robust line-based optical tracker for augmented reality applications. 1rst International Workshop on Augmented Reality (IWAR'98), San Francisco, 1998.

11. B. Ullmer and H. Ishii. The metaDESK: Models and prototypes for tangible user interfaces. UIST'97, pages 223-232, Banff, Alberta, Canada, 1997.

12. A. Webster, S. Feiner, B. MacIntyre, W. Massie, and T. Krueger. Augmented reality in architectural construction, inspection, and renovation. ASCE 3. Congress on Computing in Civil Engineering, pp. 913-919, Anaheim, CA, 1996.