Augmented Reality: A Balance Act between High Quality and Real-Time Constraints

Gudrun Klinker (1), Didier Stricker (2), Dirk Reiners (2)

Technical University of Munich, Germany
Fraunhofer Project Group for Augmented Reality at ZGDV, Germany

Abstract

Augmented Reality (AR) constitutes a very powerful three-dimensional user interface paradigm for many "hands-on" application scenarios in which users cannot sit at a conventional desktop computer. Users' views of the real world are augmented with synthetic information from a computer. Current AR research fans out into several different activities, all of which are essential to generating a system which eventually will be able to sustain a truly immersive AR-experience in extended practical applications rather than short laboratory demonstrations. But the current state of technology cannot yet provide simultaneous support for an optimal solution to all aspects of AR. Today's AR systems have to balance a wealth of trade-offs between striving for high quality, physically correct presentations and user modelling on the one hand, and making short cuts and simplifications on the other hand in order to achieve a real-time response. In our work, we have selected two different positions among many possible trade-offs, demonstrating the real-time immersive impression that can be generated with today's technology in one approach, and presenting a glimpse of the future in the other approach, forecasting what quality might be achievable with continuously increasing processing power and data bandwidth. We discuss the trade-offs we made and we present applications in object design, construction, assembly, and maintainance, as well as augmented board games.

1. Introduction

Augmented Reality (AR) constitutes a very powerful three-dimensional user interface paradigm for many "hands-on" application scenarios in which users cannot sit at a conventional desktop computer. Users' views of the real world are augmented with synthetic information from a computer. Users can thus continue their daily work involving the manipulation and examination of real objects. At the same time, they receive additional information about those objects and the task at hand, such as up-to-date instructions how to perform the next step of a task. These concepts have been demonstrated for construction and manufacturing scenarios like the computer-guided repair of copier machines [10], the installation of aluminium struts in diamond shaped spaceframes [35], for electric wire bundle assembly before their installation in airplanes [1,7], and the assembly or repair of machines [23,19].

Current AR research fans out into several different activities, all of which are essential to generating a system which eventually will be able to sustain a truly immersive AR-experience in extended practical applications rather than short laboratory demonstrations: Virtual objects need to be presented as realistically as possible, integrated physically correctly into the real world. This means that occlusion and light reflection properties between virtual and real objects must be established and maintained, as well as physical laws such as non-penetration, gravity and friction [13,1]. Furthermore, users must be free to roam an extended area, without being tethered to a stationary system [11,28,27]. Thus, AR-systems must be wearable and mobile, either by carrying all information "on-board" or by being wirelessly connected to distributed sources of information. At the same time, the system should facilitate collaboration with other AR users, allowing them to work together [1,4]. Finally, in order to make all augmentations worth their while, AR systems must be able to correctly track user motions and even predict future motions ahead of time [2,16] such that virtual objects are rendered according to the user's changing perspective.

The current state of technology cannot yet provide simultaneous support for an optimal solution to all aspects of AR. Most critical in this respect is the real-time performance of the overall demonstration system. Today's AR systems have to balance a wealth of trade-offs between striving for high quality, physically correct presentations and user modelling on the one hand, and making short cuts and simplifications on the other hand in order to achieve a real-time response. Such trade-offs occasionally involve rather perplexing alternatives: not always is the physically most precise approach the best one since a much coarser approach may be so much easier to compute that it can run in real-time - just fast enough to keep pace with the user's actions whereas the former one never even begins to get a grasp at integrating with the quickly changing real world environment. A beautiful picture is useless, if it's rendered too late. Similarly, a precise model of user motion, involving

translational and rotational speeds as well as accelerations is useless, if it cannot adapt in real-time to the erratic head motions of a user assembling a car door.

In our work, we have selected two different positions among many possible trade-offs. In one approach, we emphasize the real-time immersive impression that can be generated with today's technology [30,22]. Our on-line presentations give the user immediate feedback to his actions and thus generate a very tight, immediate human-computer interaction scheme. Our second approach is intended to present a glimpse of the future, forecasting what quality might be achievable with continuously increasing processing power and data bandwidth. In these demonstrations, we currently use pre-recorded video clips which we analyze and augment semi-automatically off-line [14,15].

Due to the current need for trade-offs, the optimal configuration of an AR-system depends on the needs and acceptable simplifications of a particular application, as well as on the selected hardware base. Design decisions may change towards including more sophisticated approaches when new applications are chosen or better hardware becomes available. We thus begin by presenting some of the applications and demonstrators we have built in the recent past (section 2), as well as the underlying hardware and software architecture (section 3). Further sections discuss the trade-offs we made towards developing user tracking algorithms (sections 4, 5), and scene augmentations (section 6).

2. Applications / Demonstrations

Our two approaches to AR have been the basis of a number of demonstrations. We are using the off-line AR system to show how AR can be used to augment outdoor scenes with proposed new buildings, as part of the acquisition and bidding phase of large exterior construction projects (2.3). We use our real-time system in all phases of the life cycle of an object, such as object design (2.1, 2.2), object assembly or construction (2.4, 2.5), and object maintainance (2.5). Finally, we explore the interactive, world-changing nature of AR applications in a board game scenario, augmented Tic Tac Toe (2.6).

2.1 Model presentation and physical manipulation

In many industry sectors (e.g., architecture, automotive design, etc.), digital three-dimensional prototypes of a designed object are becoming common place. Viewing such models is a typical application of Virtual Reality, as well as more mundane 3D viewing kits. Augmented Reality provides a new, very intuitive approach towards viewing and manipulating virtual objects [30,31]. As shown in Fig. 1, objects, such as a VRML model of St. Paul's Cathedral in London, can be attached to a physical placeholder - a cardboard with a few markers. Users can then manipulate the virtual object simply by moving or rotating the attached physical place holder - without having to deal with complex popup menus full of sliders or dials.

Figure 1: St. Paul's Cathedral

2.2 Mixed virtual/real mockups

Going one step further, the virtual prototypes can be mixed with partially existing physical prototypes, thus forming mixed virtual/real mockups [15]. Physical prototypes are still essential to evaluating the design of many products, such as cars and buildings. Such physical mockups are time-consuming and expensive to create. They are thus built only at very critical stages of the design process - typically after many of the preliminary decisions have already been made. AR provides the opportunity to build mockups more gradually, using physical prototypes for the already maturing components of the design and inserting virtual models for the currently evolving components. Fig. 2 shows a real toy house in combination with two virtual buildings. Both the virtual and real objects can be manipulated in the scene by moving associated markers.

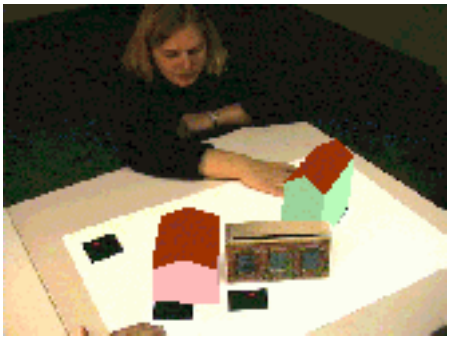

Figure 2: Manipulation of virtual and real objects

Fig 3 shows the relationship between mixed mockups and the already well-known concept of an enhanced desk [34] or Responsive Workbench[17]. In this case, the physical scene is laid out on a planar desktop covered by a (real) map of the city of London. The paper is augmented with a VRML model of St. Paul's cathedral, as well as with a CAD model of a new footbridge across the Thames that is being designed. The new bridge can be moved about interactively until it is placed in the correct spot on the map. The enhanced map is a special case of the more general capabilities of AR. The real world doesn't have to be a flat desk. It can assume arbitrary 3D shapes and become a 3D terrain

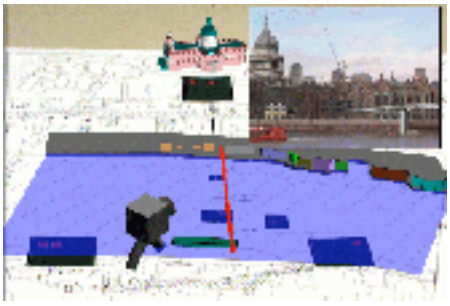

Figure 3: Augmented map of London

By waving a hand across the virtual camera icon at the river shore, users can request to see a video clip showing this area of London augmented with the proposed new footbridge (see 2.3). This illustrates that AR provides users access to all kinds of synthetic information, be it three-dimensionally integrated into the scene or presented like a movie or a graph, in a flat, 2D window.

2.3 Augmented landscapes and cityscapes for architectural design

As part of the project design and acquisition process, architects need to convince their prospective customers that the proposed building will fit well into the existing environment. This concern is of particular importance in the context of large objects that will have an impact on the landscape or the city skyline of a town, such as new bridges, towers, or major exposition areas.

AR technology can help visualize such new buildings in their eventual environmental setting. In the context of the European CICC project, we have augmented video sequences of several potential construction sites with new buildings to facilitate communications between architects and community leaders [15]. Fig. 4a shows the shore line of the river Wear in the Sunderland area, Newcastle, UK, augmented with a proposed Millennium bridge. Similarly, Fig. 4b shows the London Thames area with a millennium footbridge connecting the area near St. Paul's cathedral with the Tate museum. Both objects have been designed by Sir Norman Foster and Ove Arup. In Fig. 4c, the Expo'98 construction site in Lisbon has been augmented with the model of a pavilion being built by Europroject Ingeniera (Spain).

Figure 4: Augmented landscapes and cityscapes
a) Sunderland bridge, b) London bridge, c) Expo'98 pavilion in Lisbon

Exterior construction applications impose very demanding challenges on the robustness and usability of evolving AR technologies. Real construction sites are huge. Information has to be integrated into many views, both at close range and from long distances, requiring a significant range of tracking skills. Furthermore, construction environments are not well structured. Information has to be mixed plausibly with existing natural objects such as bushes and trees and heaps of earth, that are likely to change over time.

Since video sequences of such environments are very complex, we currently pre-record the sequences and employ off-line, interactive calibration techniques to determine camera positions for every frame, as well as a scene description. Given all calibrations, the augmentation of the images with virtual objects can be performed live on a high-end graphics computer. These demonstrations thus offer a glimpse of the future, indicating what kind of complexity AR technology needs to be able to deal with automatically in order to become usable in outdoor exterior construction applications.

2.4 Augmented car door assembly

The phase following object design involves object construction and assembly. During this phase in the life cycle of an object, AR finds many obvious applications. Hands-on work, such as the assembly of a car door, currently is generally performed without the benefit of much computer assistance since such work has to take place far away from desktop computers. AR provides the means for bringing a wealth of information into the workplace in the form of up-to-date 3D illustrations of work steps to be performed or objects parts to be manipulated.

We have demonstrated such concepts during the Hanover Industry Fair '98 at the example of an AR-assisted assembly of a door lock for a car [22]. The task of assembling a doorlock is quite challenging, requiring significant planning and dexterity. It is very spatial and three-dimensional in nature. The movement of the hand holding the lock in the small space inside the door requires precise preparation and motion. Since the space inside the door is just big enough for the lock, it has to be held in a very special way for the hand not to get stuck halfway through. Several screws then have to be inserted and fixed in the right order.

In our augmented car door assembly demonstration, the real-time AR-system instructs the user step-by-step how to hold the door lock, where to insert it with what kind of hand motion, what levers to push and what screws to fix. All illustrations are shown as 3D augmentations to the real car door. In his heads-up display, the user sees in stereo how the virtual objects coexist with the real door, being partially hidden inside the door as part of the process. The user controls stepping through the assembly routines via voice input, requesting to proceed to the next part of the illustration whenever he is ready. Fig. 5 shows a snapshot from the demonstration during the lock insertion stage.

Figure 5: Augmented car door assembly: Doorlock insertion

2.5 Building construction and maintainance

AR is also useful in both the construction and the maintainance phase of buildings [15]. As shown in Fig. 6, virtual walls can be shown before they are built, thus helping construction workers determine their exact location. Using up-to-date online information, workers are thus guaranteed to work with the latest version of an ever-changing construction plan. Furthermore, the plan can be sequentialized into small construction steps suited to the current schedule of immanent activities. Fig. 7 shows a set-up for a small bathroom under construction. The water pipes have already been installed, and a dry wall is scheduled to be installed in front of the pipes the next day. Fig. 7 shows a virtual dry wall in its place.

Figure 6: Outdoor construction site, augmented with a virtual wall about to be built

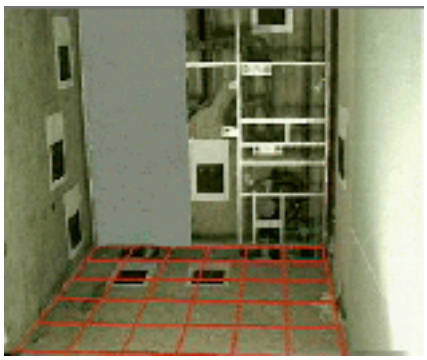

Figure 7: Augmented exterior construction
Bathroom partially augmented with a virtual dry wall

Once the construction has been completed and the building has been taken into commission, the next phase in the life cycle of a building begins: its maintainance, repair and modification. To this end, AR can help maintainance crews access all available information about the building in a suitable manner. For example, many walls are photographed before electric and sanitary installations are covered by the final layers of plaster. AR can overlay such visual data on the walls, thus providing people with x-ray vision skill to find (or avoid) electric wires or pipes in the wall (Fig. 8) when drilling holes.

Figure 8: Same room after completion,
showing the real dry wall with a semi-transparent augmentation of the piping in the wall.

2.6 Augmented Tic Tac Toe

At the example of board games like Tic Tac Toe (Fig. 9) we are exploring various concepts of mouseless 3D user interaction [15,30]. To fully exploit the AR paradigm, the computer must not only augment the real world, it also has to accept feedback from it. In truly 3D human-computer interaction, 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. Gesture languages, 3D pointers or speech input are all tools with which users can communicate with the computer about their work at an abstract level. If the computer is capable of automatically detecting and correctly interpreting scene changes caused by user actions, much such meta-level communication becomes superfluous.

Figure 9: Augmented Tic Tac Toe

In the Tic Tac Toe demonstration, the user sits in front of a real game board wearing a head-mounted display with an attached mini-camera. The user and the computer alternate placing real and virtual stones on the board. During the user's turn, he can try out various moves, playing them out on the board. When he eventually settles on one, he indicates his choice to the computer by waving his hand across a virtual 3D "GO" button or by speaking a command into an automatic speech recognition system. The computer then scans the image, looking for a new stone. If it finds one, it proceeds by planning its own next move. If it doesn't find a new stone or if it finds more than one, it writes an appropriate comment on the virtual 3D panel placed behind the board game. Note that the user is not requested to indicate by voice, text or other means where he has placed the new stone - the computer interprets and evaluates the user's action automatically.

3. System Architecture

3.1 Real-time AR System

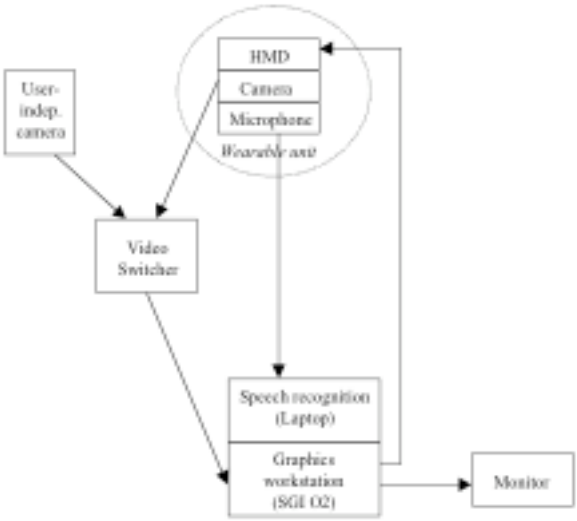

Figure 10: Hardware set-up

Our real-time AR-system (Fig. 10) uses an SGI O2 with a 180 MHz R5k processor and 128 MB memory. The machine has very good video capabilities and reasonably fast rendering, although not comparable to higher end SGI machines used for virtual reality applications.

The strong spatial nature of AR demands a display that can convey spatial information, such as an HMD. We use Virtual IO! i-glasses with an attached Toshiba IK-M48PK camera using a 7.5 mm lens (Figs. 1, 11a). The headset is a standard affordable piece of equipment allowing see-through and feed-through use. The camera is reasonably small and light enough to be worn on the head without undue strain for the user, while still giving very good quality for a single-CCD camera.

Before settling on the minicamera attached directly to the HMD, we have run our live AR-system in a monitor-based set-up with cameras held in the user's hand or positioned on a tripod (Fig. 11b). The tracker worked very reliably across a wide range of cameras including high quality Sony 3CCD Color Video Cameras as well as low-end video-conferencing cameras such as an SGI IndyCam.

Two-handed action in many of our demonstrations (e.g., car door assembly) task requires a hands-free interaction technique. In some applications, we thus use a voice-driven interface. It runs on a separate machine, a Laptop running Windows 95 and an IBM Voice-Type based speech recognition software. It is connected to the O2 via RS-232 which is adequate for the transmissions of short, pre-defined commands.

Figure 11: AR set-ups
a) HMD with attached mini-camera, b) Monitor-based AR

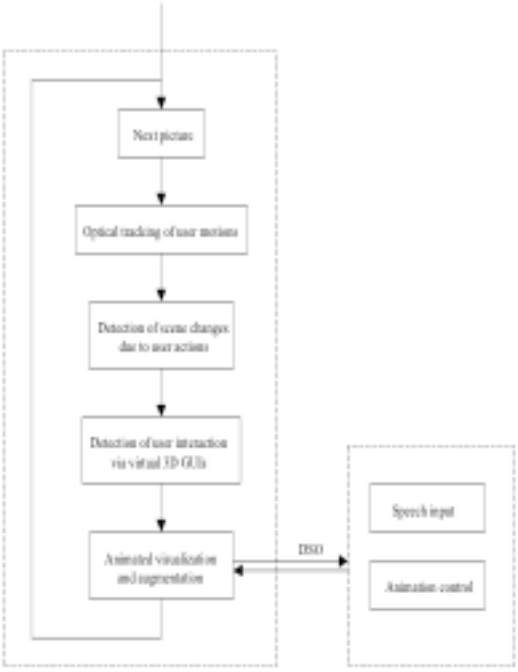

Figure 12: Software architecture

Fig. 12 shows the software architecture of our live AR system. It revolves around the central tracker loop. The tracker deals with reading and analyzing the video image to calculate the camera calibration parameters. A second, coupled component provides the application dependent augmentations and handles user interaction modalities [22].

Experimentally, the system was distributed between two machines, one for tracking (SGI O2) and one for rendering more complex virtual objects (an SGI Onyx RE2). Only the calibration information was sent via Ethernet between the two machines. But the lag resulting from the system and the network proved to be too bad to be useful for head-mounted applications. Thus, we now prefer using a single processor system.

The tracking itself runs at 20-25 Hz. Combined with the additional rendering task the speed drops. In most of our demonstrations, we have observed speeds of about 10-15 Hz.

3.2 Off-line AR system

For the off-line demonstrations, timing is not very critical. We have developed an interactive calibration system (InCal) which runs on any OpenGL-based machine, such as a low-end SGI workstation. It generates a calibration file for every frame of an image sequence [15].

The subsequent live augmentations of the video sequence require a high-end graphics workstation (e.g., an SGI Onyx RE2) in order to render the very complex virtual buildings at sufficient speed. Our AR-Viewer is a highly optimized SGI-Performer-based video augmentation system, using InCal's calibration files to insert virtual objects into each frame of the video sequence while simultaneously allowing the user to interact with them.

4. Live Optical Tracking of User Motions

Tracking user motions is currently the most intensely investigated aspect of AR - due to its critical importance to the overall performance of an AR system. In the past, quite a few technical approaches have been tried (magnetic, inertial, sound, optical). None works perfectly. Currently, the most successful approaches track optical markers in indoor laboratory demonstrations [3,16,20,28,29,33], sometimes combined with information from other tracking modalities in hybrid approaches.

4.1 Use of optical markers

In order to achieve real-time optical tracking performance, simplifying assumptions have to be made. It is currently customary for AR labs to place special, easily recognizable markers at carefully measured locations in the scene and make such information available to the tracker for its operation. The placement of such markers certainly is quite a severe restriction to the overall applicability of the system. It is tedious to install a significant number of markers necessary for robust operation of the system. In some applications, such an approach is completely impractical. Yet, more general solutions are only beginning to approach real-time speed. The current demonstrators thus provide a good starting point to begin experimenting with more general concepts [18,21]

In our approach, we use black rectangles (typically squares) on a white sheet of paper (Fig. 13) [16,15,30,22]. The exact size and shape of each rectangle is specified in a scene configuration file. We can place targets of different sizes at more or less confined positions of the scene (e.g., a car door). Large targets can be identified from quite a distance. Small targets in many intricate positions help the tracker when the user comes close. The system doesn't depend much on the exact illumination conditions and the camera sensing parameters since black rectangles on a bright background can be identified very easily from high-contrast edges. We have been able to run demonstrations in many different settings with a variety of cameras (various demonstrations at fairs, conference exhibits, and other people's laboratories or conference rooms).

To uniquely identify each rectangle independently of the current field of view, the rectangles contain a labeling region consisting of 2 rows with 4 positions (bits) each of small red squares. Using a binary encoding scheme, we can define up to 2^8=256 different targets. In any particular image during a demonstration, any subset of two targets suffices for the system to succeed in determining the current user position. Since targets may rotate by more than 45 degrees when they are attached to mobile objects (Figs. 1, 2), we restrict the labelling scheme slightly: label 1 ("0001") can be confused with label 8 ("1000") when rotated right by 90 degrees. We thus discard such potentially misinterpretable labels. In principle, this problem could be alleviated by adding one further unsymmetrically placed symbol to the target. We could also exploit the non-square aspect ratio of rectangles. Yet, any change in the target design and its detection algorithm can have unexpected negative implications on the robustness and speed of the system and thus needs to be thoroughly investigated - which we haven't found time to do yet.

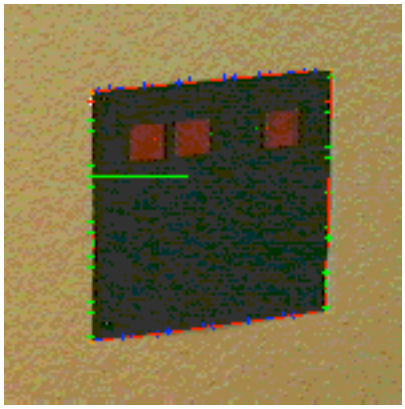

Figure 13: Picture of a black rectangular (square) target

A similar target design based on pentagons rather than rectangles has recently been developed by Mendelsohn et al. [18]. In their system, the bottom edge is gray rather than black, allowing the system to determine potential target rotations. Furthermore, the identifying labels are placed under the pentagon rather than inside it, playing the trade-off concerning the extent of a target versus the ease of detecting it in a scene differently than our system: the larger the extent of a target, the more likely it is to be partially occluded or out of the field of view. The concept of simplifying the target can be investigated further by making each target even simpler (small dot, or maybe an LED) and determining its identity from its position in unique groupings of several targets. Such approaches are used in commercial tracking systems and also in photogrammetry.

Other optical AR systems currently work with multi-colored, concentric circular targets [20,29]. When using c different colors in r concentric rings, the multi-coloring scheme can distinguish between up to c^r different markers. Typical numbers are $c=\{4,6\}$ primary and secondary colors, and $r=\{2,3,4\}$ concentric rings, thus allowing between 4^2=16 and 6^4=1296 different targets [20,29]. There is a trade-off between using a photometric (coloring) scheme versus using a geometric (positional) encoding scheme. Targets distinguished by their colors can be smaller than those using binary codes that have to be discernible from some distance. Colored targets are also less susceptible to partial occlusions. On the other hand, they are less tolerant to changing illumination and sensing conditions than a geometric approach which essentially operates on binary image data. Approaches using multi-colored concentric rings merge the benefits and problems of both approaches, requiring a certain spatial extent for the rings to be distinguishable, as well as being dependent on the illumination.

4.2 Initialization of the optical tracking system

Our system uses fast image processing techniques to initially find the rectangles in an image [15,30]. The robustness, simplicity and speed of not only the on-line tracker but also its initialization routine is absolutely essential to the overall acceptability of the AR system. Even the fastest optical tracking systems cannot always keep pace with erratic user head motions. Thus, trackers tend to fail every once a while, requiring a re-initialization. A fast recovery from tracking failure is absolutely essential since users won't stop moving just for the system to re-initialize. Our initialization routine currently runs at about 3 Hz.

To find the black rectangles in an image, we scan every n^{th} line for strong white-to-black image gradients followed by similar black-to-white gradients. A pair of such gradients is hypothesized to indicate the left and right edges of a rectangle. We use the center in-between them as a starting point to vertically seek for a third black-to-white gradient defining one of the horizontal edges. The horizontally and vertically scanned pixels are evaluated with respect to the overall homogeneity and blackness of the potential target. Inhomogeneous or very bright target candidates are discarded.

We next follow the border around each dark blob, starting from one of the three edge points. To fit rectangles to each blob, the edge pixels are classified into four clusters according to their gradient directions, using a standard ISO-data algorithm. After a statistical homogeneity test, we fit straight lines to each of the four edge clusters. The intersection of neighboring lines defines the corner points of the rectangle.

To determine the ID of a target, we scan the image along two lines known to pass through the two rows of red squares. The position of each row is defined by an offset from the top left corner of the rectangle, scaled relative to the length to the top edge. The size of the red squares and their position in each row is also defined relative to the length of the edge. We sample each line and correlate it with templates representing encodings of labels 0000 through 1111, selecting the label yielding the highest score. This method has proven to be very robust, working well even with low quality cameras and under bad illumination conditions.

When a set of markers has been identified in an image, we use one of a suite of calibration routines to determine the current camera position from a mapping between scene co-ordinates and image co-ordinates of those corner points of the targets that have been reliably detected. Using the pin-hole model, we compute the external camera parameters (camera pose), as well as the internal camera parameters (focal length and center, aspect ratio) by the algorithms described in [36,32]. Whenever possible, we use simplified versions of the algorithms, computing only the external camera parameters, using user-provided data for the internal parameters and thus gaining much more robust calibration estimates.

4.3 Camera tracking

To redetect targets in subsequent images, we predict their position in the next frame. Due to the randomness of user head motion, it is hard to predict future motion from history. Sophisticated motion prediction models that we have experimented with, such as Kalman filters [16], have had serious problems tracking a user-held or user-worn camera. Such prediction of 3D user motion, which includes schemes to model the current translational and rotational speed as well as acceleration of a camera, is too slow to quickly react to users' head rotations (e.g., during a quick glance to one side, or a head shaking motion), generating an effect of "swimming" off track. If the head motion is very abrupt, the system never recovers from its "detour".

Instead, we have now returned to a much simpler, two-dimensional tracking approach. We linearly predict (locally in 2D) the position of a target's corners from their position in the previous two images - ignoring all influence of 3D rotations, acceleration or perspective foreshortening which can result in different 2D motion vectors in different parts of the image. This approach is very fast and gives us a good starting point to search for the precise position of the targets. Due to its speed (close to the real frame rate), we only have to deal with very little head motion between images. We can thus limit the search radius for redetecting targets - thereby further improving tracking performance. Thus, linearizing 2D target motion has had a very positive effect on system performance and robustness - even though it is a very inadequate motion model for describing user head motions from a physical point of view.

When the target locations have been predicted for the next image, we locally search the image for the best match. For every edge of a rectangle, the algorithm scans the new image perpendicularly to the edge along several scan lines for maximal image gradients. The new edge is determined by fitting a line through the set of maximal gradients [30].

The new edge positions are then used to determine the new camera position. We need to account - in principle - for potential changes of all internal and external parameters of the camera. Yet, for the sake of real-time response, we assume that the internal camera parameters are fixed. For the external parameters (camera position and orientation), we use the information from the previous image as a starting point, applying a fast gradient descent technique to settle on a new set of parameters that fit the target locations in the new image [30].

Since the target tracker only requires edge information and not full target descriptions, our tracker can operate in areas of the scene where no rectangular target is visible. To this end, we have extended the scene model to also include descriptions of naturally occurring linear features, such as window sills and edges of walls or tables. The special markers are only required during the initialization phase.

4.4 See-through HMD calibration

For see-through applications, the position of the user's eyes must be established relative to the computed camera (head tracker) position. Unfortunately, no sensor can record exactly what the user sees. We thus cannot determine the eye position without user involvement. To this end, we have developed an interactive augmented computer vision method. The AR-system displays 3D outlines of the rectangular targets on the HMD, using the calibration parameters of the camera that is attached to the HMD. Due to the offset between the camera and each of the user's eyes, the user observes a misregistration between the target outlines and the rectangles on the wall. We then ask the user to indicate the misregistration with the mouse for one eye at a time by drawing lines from the outlined target corners to their true positions, as seen by the user. A specially constructed head rest helps the user keep the head still during this procedure. Using the mappings between projected and actually observed target positions, the offset of each eye can be computed.

Even though this is not a difficult procedure in principle, it depends very much on just how still users can hold their heads during the interactions. During extended mouse motions, users tend to change head position somewhat, resulting in a gradual drift in the mappings of the corners. We are considering using mouseless interfaces, such as cursor control by voice or by pushing the arrow keys on the keyboard. So far, the use of additional constraints regarding the relative position and orientation of two eyes has sufficed to compute camera-

to-eye calibrations that were good enough to generate a believable immersive stereo effect for users when wearing the HMD. Yet, the procedure is not sophisticated enough to deal with a level of precision requiring us to recalibrate the system when different people use the HMD. Thus, we use the same calibration parameters for all users - as long as the attachment of the camera to the HMD is not accidentally moved or twisted.

4.5 Real-time lag reduction via motion prediction

When a scene is augmented with virtual objects, lag cannot be avoided entirely since it takes time to draw the virtual objects. This is not a problem for fast monitor-based solutions since the video image used for calibration can be the one also being used for augmentation - with users barely noticing that the video data they see is slightly outdated. The see-through-mode is much more critical since users keep sight of the real world. Time for capturing the video data, processing it, and generating the augmentations all sums up to a misregistration that depends on the speed of the user's head motion. During a continuous head rotation, virtual objects "lag behind" by a constant distance.

This is the classical tracking and motion prediction problem in AR [2]. Kalman filters are one approach for predicting user motion in three dimensions and thus allowing the system to render objects according to expectations where the user's head will be by the time the drawing is done. Yet, discussions in section 4.3 have outlined why Kalman filters currently have not proven suitable in our applications for real-time AR: head motions are just to erratic to be modelled predictably in 3D to date.

In our current system, we use the 2D feature prediction technique of section 4.3 to also predict where the features will be one time step further in the future. A futuristic calibration then provides the rendering parameters to draw virtual objects ahead of time. In order not to loose time, such calibration only computes the rotational components of predicted head motions since those are affecting the visible misregistration most strongly. When the user spontaneously changes the motion direction, only few images are affected since no complex motion model has to learn about the change of course. After two frames only the new direction is relevant to the motion predictor. This approach works well as long as objects can be rendered at frame rate. Predictions into more distant futures will suffer greatly from the linearized 2D feature prediction approach.

5. Off-line calibration of video sequences

We are currently not able to augment scenes of landscapes or cityscapes automatically and in real-time with complex virtual objects, as shown in section 2.3. This section describes our semi-automatic, off-line techniques for augmenting pre-recorded video sequences of such scenes. Such approaches need to become more automatic and faster for high-quality AR presentations.

5.1 Interactive Calibration

Our interactive calibration system, InCal, provides a user interface to calibrate and track camera positions semi-automatically in image sequences. It superimposes a rough 3D model of the real scene on a start image. The user can interactively manipulate the model to make it fit the scene approximately. Furthermore, the user can interactively indicate correspondences between 3D model features and 2D image features. Such correspondences are then used to automatically compute the current camera position. When the virtual camera is set to the same position, the 3D scene model "snaps" into alignment with the image [15].

When calibrating image sequences, InCal exploits inter-frame coherence to automatically propose feature locations in new images from their locations in previous images, using a normalized cross-correlation technique. Users thus barely have to interact with the system once it has been initialized. Most features are nicely tracked across many images. Occasional mismatches can be corrected interactively.

We have been able to successfully calibrate many live and pre-recorded video sequence this way. Even for complex landscapes, we have been able to interactively calibrate sequences of hundreds of images nearly automatically within a few hours. Yet, calibrations are very sensitive to noise and to the shape of the 3D scene model, requiring suitable heuristics in order to achieve good augmentations of images. Tracking stability is another key issue. Virtual objects must be precisely positioned in the picture and keep their position over time despite camera motion and noise. In video sequences, apparent stability within the scene over time is visually more important than the precise calibration of individual images by themselves. Thus, it is important to make stabilizing assumptions. In particular, assuming that the internal camera parameters remain constant throughout the video recording provides significant overall improvements. Using schemes that avoid computing all six external parameters together for every image further stabilizes the system.

5.2 Acquisition of 3D Scene Models ("Reality Models")

High quality AR-applications require a very accurate model of the environment (a reality model) to augment the current view seamlessly with synthetic information (the virtual model) such that the virtual objects behave in physically plausible manners: They occlude or are occluded by real objects, they are not able to move through other objects, and they cast shadows or light reflections on other objects. The automatic construction of scene models is a long-standing issue in computer vision research. Recent approaches have suggested using it for the reconstruction of buildings [8,9]. In our work, we explore a very applied, pragmatic approach which is closely related to the requirements of rather realistic applications in the exterior construction industry [14].

In our interactive system, InCal, we begin with a very sparse model of a scene, constructed from externally provided information such as the known position and height of a few buildings, power poles or bridge pillars. We measure more information, such as the course of rivers and streets, from 2D maps and insert it a zero height into the model. From this model, we generate an initial camera calibration for a few site photos, interactively indicating how features in the image relate to the scene model. Once an image has been successfully calibrated, the model is overlaid on the image, showing good alignment of the image features with the model features. Models of new structures in the landscape are then added to the model, using their two-dimensional position in the city map and estimating their height from their alignment in several images [14,15].

6. Presentation of Virtual Information

Once appropriate scene models and calibrations have been obtained, they form the basis for mixing real and virtual worlds. This section describes the steps necessary to achieve realistic and fast inclusion of virtual information into the scene.

AR thrives on fast, real-time augmentations of the real world. All virtual information thus has to be rendered very quickly. To this end, we carefully tune and prune geometric models to achieve maximal rendering performance while maintaining an acceptable level of realism. For this purpose, our home-built models are very simple, consisting merely of a few polygons and maybe a texture map which can be rendered very quickly even on low-end graphics machines. When collaborating with industry partners, typical geometric models of virtual car prototypes or buildings are much more complex - too large even to be rendered by high-powered graphics supercomputers at an acceptable frame rate. They thus have to be simplified. We employ an interactive in-house tool [25] building on standard algorithms [24,26]. Except for close-ups, the resulting models are virtually indistinguishable from the originals, albeit at a fraction of the cost.

Mixing virtual objects into a real scene requires that the object obey the basic physical laws - first of all occlusion. Occlusions between real and virtual objects can be computed quite efficiently by the geometric rendering hardware of graphics workstations when provided with a list of the geometric descriptions of all real and virtual models. By first drawing transparently the scene models of the real objects, we initialize the z-buffer to subsequently draw only those virtual objects that are located in front of the real objects. The user thus sees the real scene through the transparent HMD where the real objects are closest to him, and the virtual objects where they are closest.

Real objects cast shadows and reflect light. So should virtual objects. When 3D scene descriptions are available, standard computer graphics algorithms [12] can be used to compute the geometry of shadows cast by virtual objects onto real ones [29]. Given the right hardware, shadows can be blended in real-time with the image of the underlying object, thereby accounting for ambient light. Reflections are more difficult and can only be solved for special cases, such a mirror reflections of virtual objects on planar surfaces of real objects (windows, water). We are using some of the concepts for the off-line augmentations of pre-recorded video sequences of prospective construction sites. We don't handle shadowing and light reflections in the on-line AR-system in order to leave as much time as possible to the optical tracker.

In the final step of actually merging virtual objects into an image we need to consider whether the virtual objects should completely occlude what is behind them or whether a blended presentation of both reality and virtuality is more appropriate. When we want to create the illusion of a physically changed world with as much realism as possible, the opaque inclusion of virtual objects is appropriate. Yet, in many cases, the augmentations are intended to add supporting information to the world rather than change it. For example, users may want to choose a suitable blending ratio to include virtual text panels with data records or instructions (Figure 9) or to gain semi-transparent x-ray viewing capabilities to see what's inside a wall (Figure 8). Thus, our system provides users with the option to interactively select a percentage level at which virtual objects are blended into the real world.

7. Discussion

In this paper, we have emphasized the current need for AR-Systems to trade off perfection for real-time efficiency. Currently, the most critical issue for the system is to be fast enough to keep pace with a user's spontaneous and erratic head motions when being involved in real, hands-on tasks. Our demonstrations show that this is becoming feasible for "reasonable" motions in pre-arranged scenes. Thus, a starting point has been set to begin embarquing on more ambitious goals. Among those, the following seem the most important. First, the tracking approach has to become more general and flexible, getting by with naturally occurring scene features rather than requiring special targets. The system also has to begin learning about the environment while the user works in it, thus incrementally building a knowledge base describing the scene. Second, the system has to become more interactive, responding to real-world changes autonomously without being prompted by the user. We have indicated a first glimpse of such concepts at the example of the Tic Tac Toe game. Yet, much more has to be done to really understand and track the changing real world in which the AR-user is working. Third, the system needs to develop more sophisticated visualization and rendering schemes. Regarding all these issues it is critical to explore from the beginning with real users which approaches are acceptable in real applications and which ones are making the wrong (impractical) simplifying assumptions. Considering the current rate of progress, we expect many of these aspects to be addressed in the foreseeable future. AR has the potential to become a "killer application" for mobile 3D sensing and visualization technology.

Acknowledgements

This work was conducted while the authors formed the Fraunhofer Project Group for AR at ZGDV, in close collaboration with the department for visualization and virtual reality of Fraunhofer-IGD in Darmstadt. Office space was provided by ECRC in Munich. The work was partially funded by the European projects CICC (ACTS) and Cumuli (Esprit).

Bibliography

K.H. Ahlers, A. Kramer, D.E. Breen, P.-Y. Chevalier, C. Crampton, E. Rose, M. Tuceryan, R.T. Whitaker, and D. Greer. Distributed augmented reality for collaborative design applications. *Proc. Eurographics'95*, 1995.

R. Azuma and G. Bishop. Improving Static and Dynamic Registration in an Optical See-through HMD. *Proc. Siggraph'94*, Orlando, July 1994, pp. 197-204.

M. Bajura and U. Neumann. Dynamic registration correction in video-based augmented reality systems. *IEEE Computer Graphics and Applications*, 15(5):52-60, 1995.

M. Billinghurst, S. Weghosrt, T. Furness III. Wearable Computers for Three Dimensional CSCW. Proc. First International Symposium on Wearable Computers, Cambridge, MA, Oct. 1997, pp. 108-115.

D.E. Breen, E. Rose, and R.T. Whitaker. Interactive occlusion and collision of real and virtual objects in augmented reality. Technical Report ECRC-95-02, ECRC, Arabellastr. 17, D-81925 Munich, 1995.

T. Caudell and D. Mizell, Augmented Reality: An Application of Heads-up Display Technology to Manual Manufacturing Processes, *Proc. HICCS'92*.

D. Curtis, D. Mizell, P. Gruenbaum, and A. Janin. Several Devils in the Details: Making an AR App Work in the Airplane Factory. *Proc. First International Workshop on Augmented Reality*, Nov. 1, 1998.

P.E. Debevec, C.J. Taylor, and J. Malik. Modelling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Proc. SIGGRAPH*, pp. 11-20, New Orleans, Aug. 4-9, 1996.

O. Faugeras, S. Laveau, L. Robert, G. Csurka, and C. Zeller. 3D reconstruction of urban scenes from sequences of images. In A. Gruen, O. Kuebler, and P. Agouris (eds.), *Automatic Extraction of Man-Made Objects from Aerial and Space Images*. Birkhauser, 1995.

S. Feiner, B. Macintyre, and D. Seligmann. Knowledge-based Augmented Reality. *Communications of the ACM*, 36(7):53-61, 1993.

S. Feiner, B. MacIntyre, T. Hoellerer and A. Webster. A Touring: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. Proc. First International Symposium on Wearable Computers, Cambridge, MA, Oct. 1997, pp. 74-81.

J.D. Foley, A. Van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics, Principles and Practice*, 2. Edition. Addison Wesley, 1989.

A. Fournier. Illumination problems in computer augmented reality. *Journee Analyse/Synthese d'Images (JASI),* pp. 1-21, January 1994.

G. Klinker, D. Stricker, and D. Reiners. The Use of Reality Models in Augmented Reality Applications. European Workshop on 3D Structure from Multiple Images of Large-scale Environments (SMILE), Freiburg, Germany, June 6-7, 1998.

G. Klinker, D. Stricker, and D. Reiners. Augmented reality for exterior construction applications. In W. Barfield and T. Caudell (eds.), *Augmented Reality and Wearable Computing*. Lawrence Erlbaum Press, 1998.

D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan. Real-time vision-based camera

tracking for augmented reality applications. *Proc. ACM Symposium on Virtual Reality Software and Technology* (VRST'97), Lausanne, Switzerland, Sept. 15-17, 1997. ACM Press.

W. Krueger, C.-A. Bohn, B. Froehlich, H. Schueth, W. Strauss, and G. Wesche. The responsive workbench: A virtual work environment. IEEE Computer, pp. 42-48, 1995.

J. Mendelsohn, K. Daniilidis, and R. Bajcsy. Constrained Self-Calibration for Augmented Reality Registration. *Proc. First International Workshop on Augmented Reality*, Nov. 1, 1998.

J. Molineros, V. Raghavan, and R. Sharma. AREAS: Augmented Reality for Evaluating Assembly Sequences. First International Workshop on Augmented Reality, Nov. 1, 1998.

J. Park, S. You, and U. Neumann. Fast color fiducial detection and dynamic workspace extension in video see-through self-tracking augmented reality. *Proc. Fifth Pacific Conference on Computer Graphics and Applications*, 1997.

J. Park, S. You, and U. Neumann. Natural Feature Tracking for Extendible Robust Augmented Realities. *Proc. First International Workshop on Augmented Reality*, Nov. 1, 1998.

D. Reiners, D. Stricker, G. Klinker and S. Müller. Augmented Reality for Construction Tasks: Doorlock Assembly. *Proc. First International Workshop on Augmented Reality*, Nov. 1, 1998.

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

J. Rossignac and P. Borrel. Multi-resolution 3D approximation for rendering complex scenes. Proc. Second Conference on Geometric Modelling in Computer Graphics, pp. 453-465, June 1993, Genova, Italy.

J. Schiefele. Methoden der automatischen Komplexitätsreduktion zur effizienten Darstellung von CAD-Modellen. Diplomarbeit, TU Darmstadt, 1996.

W.J. Schroeder, J.A. Zarge, and W.E. Lorensen. Decimation of triangle meshes, volume 26, pp. 65-70, July 1992.

A. Smailagic and R. Martin. Metronaut: A Wearable Computer with Sensing and Global Communications Capabilities. Proc. First International Symposium on Wearable Computers, Cambridge, MA, Oct. 1997, pp. 116-122.

T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R.W. Picard, and A. Pentland. Augmented reality through wearable computing. *Presence, Special Issue on Augmented Reality*, 6(4): 386-398, August 1997.

A. State, G. Hirota, D.T. Cheng, W.F. Garrett, and M.A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. *Proc. SIGGRAPH*, pp. 429-438, New Orleans, Aug 4-9, 1996. ACM Press.

D. Stricker, G. Klinker, and D. Reiners. A Fast and Robust Line-based Optical Tracker for Augmented Reality Applications. *Proc. First International Workshop on Augmented Reality*, Nov. 1, 1998.

Z. Szalavari and M. Gervautz. Using the Personal Interaction Panel for 3D Interaction. *Proc. Conference on Latest Results in Information Technology*, Budapest, Hungry, May 1997, pp. 3-6.

R.Y. Tsai. An efficient and accurate Camera calibration technique for 3D machine vision. Proc. CVPR, pp. 364-374, 1986. See also http: //www.cs.cmu.edu/rgw/TsaiCode.html.

M. Uenohara and T. Kanade. Vision-based object registration for real-time image overlay. Proc. Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95), pp. 13-22, Nice, France, April 1995.

B. Ullmer and H. Ishii. The metaDESK: Models and prototypes for tangible user interfaces. *Proc. UIST'97*, pp. 223-232, Banff, Alberta, Canada, Oct. 14-17, 1997.

A. Webster, S. Feiner, B. MacIntyre, W. Massie, and T. Krueger. Augmented reality in architectural construction, inspection, and renovation. *Proc. ASCE Third Congress on Computing in Civil Engineering*, pp. 913-919, Anaheim, CA, June 17-19, 1996.

J. Weng, P. Cohen, and M Herniou. Camera calibration with distortion models and accuracy evaluation. IEEE Trans. On Pattern Analysis and Machine Intelligence, PAMI, 14(10): 965-980, 1992.