PRESENCE: Teleoperations and Virtual Environments
Special Issue on Augmented Reality
MIT Press, Spring 1997

Confluence of Computer Vision and Interactive Graphics for Augmented Reality

Gudrun J. Klinker, Klaus H. Ahlers, David E. Breen, Pierre-Yves Chevalier, Chris Crampton, Douglas S. Greer, Dieter Koller, Andre Kramer, Eric Rose, Mihran Tuceryan, Ross T. Whitaker

European Computer-Industry Research Centre (ECRC)
Arabellastraße 17, 81925 Munich, Germany

Confluence of Computer Vision and Interactive Graphics for Augmented Reality

Gudrun J. Klinker, Klaus H. Ahlers, David E. Breen, Pierre-Yves Chevalier, Chris Crampton, Douglas S. Greer, Dieter Koller, Andre Kramer, Eric Rose, Mihran Tuceryan, Ross T. Whitaker

European Computer-Industry Research Centre (ECRC)[1]
Arabellastraße 17, 81925 Munich, Germany

Abstract

Augmented reality (AR) is a technology in which a user's view of the real world is enhanced or augmented with additional information generated from a computer model. Using AR technology, users can interact with a combination of real and virtual objects in a natural way. This paradigm constitutes the core of a very promising new technology for many applications. However, before it can be applied successfully, AR has to fulfill very strong requirements including precise calibration, registration and tracking of sensors and objects in the scene, as well as a detailed overall understanding of the scene.

We see computer vision and image processing technology play an increasing role in acquiring appropriate sensor and scene models. To balance robustness with automation, we integrate automatic image analysis with both interactive user assistance and input from magnetic trackers and CAD-models. Also, in order to meet the requirements of the emerging global information society, future human-computer interaction will be highly collaborative and distributed. We thus conduct research pertaining to distributed and collaborative use of AR technology. We have demonstrated our work in several prototype applications, such as collaborative interior design, and collaborative mechanical repair. This paper describes our approach to AR with examples from applications, as well as the underlying technology.

1. Introduction

Augmented reality (AR) is a technology in which a user's view of the *real* world is enhanced or augmented with additional information generated from a computer model. The enhancement may consist of virtual artifacts to be fitted into the environment, or a display of non-geometric information about existing real objects. AR allows a user to work with and examine real 3D objects, while receiving additional information about those objects or the task at hand. By exploiting people's visual and spatial skills, AR brings information into the user's *real* world rather than pulling the user into the computer's *virtual* world. Using AR technology, users can thus interact with a mixed virtual and real world in a natural way. This paradigm for user interaction and information visualization constitutes the core of a very promising new technology for many applications. However, real applications impose very strong demands on AR technology that cannot yet be met. Some of such demands are listed below.

- In order to combine real and virtual worlds seamlessly so that the virtual objects align well with the real ones, we need very precise models of the user's environment and how it is sensed. It is essential to

---

[1] As of March 31, 1996, ECRC has closed its research branch. Some authors (Kramer, Tuceryan, Greer) left before this decision, others as a result of it. Authors' current affiliations:
Klinker,Rose: Fraunhofer Proj.Group for AR at ZGDV, Arabellastr. 17 (at ECRC), D-81925 Munich; gudrun@ecrc.de, erose@ecrc.de
Ahlers, Crampton: ECRC, Arabellastr. 17, D-81925 Munich; colas@ecrc.de, cmc@ecrc.de
Breen: Computer Graphics Lab, MS 348-74, California Inst. of Tech., Pasadena, CA 91125, USA; david@gg.caltech.edu
Chevalier: Rank Xerox Research Ctr., 6 Ch. de Maupertuis, 38240 Meylan, France; Pierre-Yves.Chevalier@grenoble.rxcx.xerox.com
Greer: San Diego Supercomputer Center, Box 85608, San Diego, CA 92186-9784, USA; dsg@sdsc.edu
Koller: Dept. of Electrical Eng. 136-93, California Inst. of Tech., Pasadena, CA 91125, USA; koller@vision.caltech.edu
Kramer: APM Ltd, Poseidon House, Castle Park, Cambridge CB3 ORD, UK; ak@ansa.co.uk
Tuceryan: Texas Instruments, P.O Box 655474, M/S 238, Dallas, TX 75265, USA; tuceryan@csc.ti.com
Whitaker: Electrical Eng. Dept., 330 Ferris Hall, U. of Tennessee, Knoxville, TN 37996-2100, USA; whitaker@vision1.engr.utk.edu

determine the location and the optical properties of the viewer (or camera) and the display, i.e.: we need to *calibrate* all devices, *register* them and all objects in a global coordinate system, and *track* them over time when the user moves and interacts with the scene.

- Realistic merging of virtual objects with a real scene requires that objects behave in physically plausible manners when they are manipulated, i.e.: they occlude or are occluded by real objects, they are not able to move through other objects, and they are shadowed or indirectly illuminated by other objects while also casting shadows themselves. To enforce such *physical interaction constraints* between real and virtual objects, the AR-system needs to have a very detailed description of the physical scene.

- In order to create the illusion of an AR interface it is required to present the virtual objects with a high degree of realism, and to build user interfaces with a high degree of immersion. Convincing interaction and information visualization techniques are still very much a research issue. On top of that, for multi-user applications in the context of AR it is necessary to address the distribution and sharing of virtual environments, the support for user collaboration and awareness, and the connection between local and remote AR installations.

We see computer vision and image processing technology — although still relatively brittle and slow — play an increasing role in acquiring appropriate sensor and scene models. Rather than using the video signal merely as a backdrop on which virtual objects are shown, we explore the use of image understanding techniques to calibrate, register and track cameras and objects and to extract the three-dimensional structure of the scene. To balance robustness with automation, we integrate automatic image analysis with interactive user assistance and with input from magnetic trackers and CAD-models.

In our approach to AR we combine computer-generated graphics with a live video signal from a camera to produce an enhanced view of a real scene, which is then displayed on a standard video monitor.[2] We track user motion and provide basic pointing capabilities in form of a 3D pointing device with an attached magnetic tracker, as shown in Figure 6. This suffices in our application scenarios to demonstrate how AR can be used to query information about objects in the real world. For the manipulation of virtual objects, we use mouse-based interaction in several related 2D views of the scene on the screen.

We conduct research pertaining to distributed and collaborative use of AR technology. Considering the growing global information society, we expect an increasing demand for collaborative use of highly interactive computer technology over networks. Our emphasis lies on providing interaction concepts and distribution technology for users who collaboratively explore augmented realities, both locally immersed and remotely in the form of a telepresence.

We have demonstrated our work in several prototype applications, such as collaborative interior design, and collaborative mechanical repair. This paper describes our approach to AR with examples from applications, as well as the underlying technology.

[2] We are considering to use a head-mounted display rather than a monitor-based setup. HMDs do provide a much more immersive experience. Yet, it remains to be determined how video input, equivalent to what the user sees could then be provided for our computer vision based algorithms. One suitable setup might use feed-through technology which does not use semi-transparent glasses but rather displays a video signal inside the HMD.

2. Previous Work

Research in augmented reality is a recent but expanding area of research. We briefly summarize the research conducted to date. Baudel and Beaudouin-Lafon have looked at the problem of controlling certain objects (e.g., cursors on a presentation screen) through the use of free hand gestures (Baudel & Beaudouin-Lafon, 1993). Feiner et al. have used augmented reality in a laser printer maintenance task. In this example, the augmented reality system aids the user in the steps required to open the printer and replace various parts (Feiner, MacIntre & Seligmann, 1993). Wellner has demonstrated an augmented reality system for office work in the form of a virtual desktop on a physical desk (Wellner, 1993). He interacts on this physical desk both with real and virtual documents. Bajura et al. have used augmented reality in medical applications in which the ultrasound imagery of a patient is superimposed on the patient's video image (Bajura, Fuchs & Ohbuchi, 1992). Lorensen et al. use an augmented reality system in surgical planning applications (Lorensen, Cline, Nafis, Kikinis, Altobelli & Gleason, 1993). Milgram and Drascic et al. use augmented reality with computer generated stereo graphics to perform telerobotics tasks (Milgram, Zhai, Drascic & Grodski, 1993; Drascic, Grodski, Milgram, Ruffo, Wong & Zhai, 1993). Caudell and Mizell describe the application of augmented reality to manual manufacturing processes (Caudell & Mizell, 1992). Fournier has posed the problems associated with illumination in combining synthetic images with images of real scenes (Fournier, 1994).

The utilization of computer vision in AR has depended upon the requirements of particular applications. Deering has explored the methods required to produce accurate high resolution head-tracked stereo display in order to achieve sub-centimeter virtual to physical registration (Deering, 1992). Azuma and Bishop, and Janin et al. describe techniques for calibrating a see-through head-mounted display (Azuma & Bishop, 1994; Janin, Mizell & Claudell, 1993). Gottschalk and Hughes present a method for auto-calibrating tracking equipment used in AR and VR (Gottschalk & Hughes, 1993). Gleicher and Witkin state that their through-the-lens controls may be used to register 3D models with objects in images (Gleicher & Witkin, 1992). More recently, Bajura and Neumann have addressed the issue of dynamic calibration and registration in augmented reality systems (Bajura & Neumann, 1995). They use a closed-loop system which measures the registration error in the combined images and tries to correct the 3D pose errors. Grimson et al. have explored vision techniques to automate the process of registering medical data to a patient's head using segmented CT or MRI data and range data (Grimson, Lozano-Perez, Wells, Ettinger, White & Kikinis, 1994; Grimson, Ettinger, White, Gleason, Lozano-Perez, Wells & Kikinis, 1995). In a related project, Mellor recently developed a real-time object and camera calibration algorithm that calculates the relationship between the coordinate systems of an object, a geometric model, and the image plane of a camera (Mellor, 1995). Uenohara and Kanade have developed techniques for tracking 2D image features, such as fiducial marks on a patient's leg, in real time using special hardware to correlate affine projections of small image areas between images (Uenohara & Kanade, 1995). Peria et al. use specialized optical tracking devices (calibrated plates with LEDs attached to medical equipment) to track an ultrasound probe and register it with SPECT data (Peria, Chevalier, François-Joubert, Caravel, Dalsoglio, Lavallee & Cinquin, 1995). Betting et al. as well as Henri et al. use stereo data to align a patient's head with MRI or CT data (Betting, Feldmar, Ayache & Devernay, 1995; Henri, Colchester, Zhao, Hawkes, Hill & Evans, 1995).

Some researchers have studied the calibration issues relevant to head mounted displays (Bajura, Fuchs & Ohbuchi, 1992; Caudell & Mizell, 1992; Azuma & Bishop, 1994; Holloway, 1994; Kancherla, Rolland, Wright & Burdea, 1995). Others have focused on monitor based approaches (Tuceryan, Greer, Whitaker, Breen, Crampton, Rose & Ahlers, 1995; Betting, Feldmar, Ayache & Devernay, 1995; Grimson, Ettinger, White, Gleason, Lozano-Perez, Wells & Kikinis, 1995; Henri, Colchester, Zhao, Hawkes, Hill & Evans, 1995; Mellor, 1995; Peria, Chevalier, François-Joubert, Caravel, Dalsoglio, Lavallee & Cinquin, 1995; Uenohara & Kanade, 1995). Both approaches can be suitable depending on the demands of the particular application.

3. Application Scenarios

We have developed a comprehensive system, GRASP, which we have used as the basis for our application demonstrations. This section discusses two examples. The next sections describe in detail the GRASP system and the research issues that we focus on.

3.1 Collaborative Interior Design

Figure 1. Augmented room showing a real table with a real telephone and a virtual lamp, surrounded by two virtual chairs. Note that the chairs are partially occluded by the real table while the virtual lamp occludes the table.

The scenario for the interior design application assumes an office manager who is working with an interior designer on the layout of a room (Ahlers, Kramer, Breen, Chevalier, Crampton, Rose, Tuceryan, Whitaker & Greer, 1995). The office manager intends to order furniture for the room. On a computer monitor they both see a picture of the real room from the viewpoint of the camera. By interacting with various manufacturers over a network, they select furniture by querying databases using a graphical paradigm. The system provides descriptions and pictures of furniture that is available from the various manufactures who have made models available in their databases. Pieces or groups of furniture that meet certain requirements such as color, manufacturer, or price may be requested. The users choose pieces from this "electronic catalogue" and 3D renderings of this furniture appear on the monitor along with the view of the room. The furniture is positioned using a 3D mouse. Furniture can be deleted, added, and rearranged until the users are satisfied with the result; they view these pieces on the monitor as they would appear in the actual room. As they move the camera they can see the furnished room from different points of view.

The users can consult with colleagues at remote sites who are running the same system. Users at remote sites manipulate the same set of furniture using a static picture of the room that is being designed. Changes by one user are seen instantaneously by all of the others, and a distributed locking mechanism ensures that a piece of furniture is moved by only one user at a time. In this way groups of users at different sites can

work together on the layout of the room (see Figure 1). The group can record a list of furniture and the layout of that furniture in the room for future reference.

3.2 Collaborative Mechanical Repair

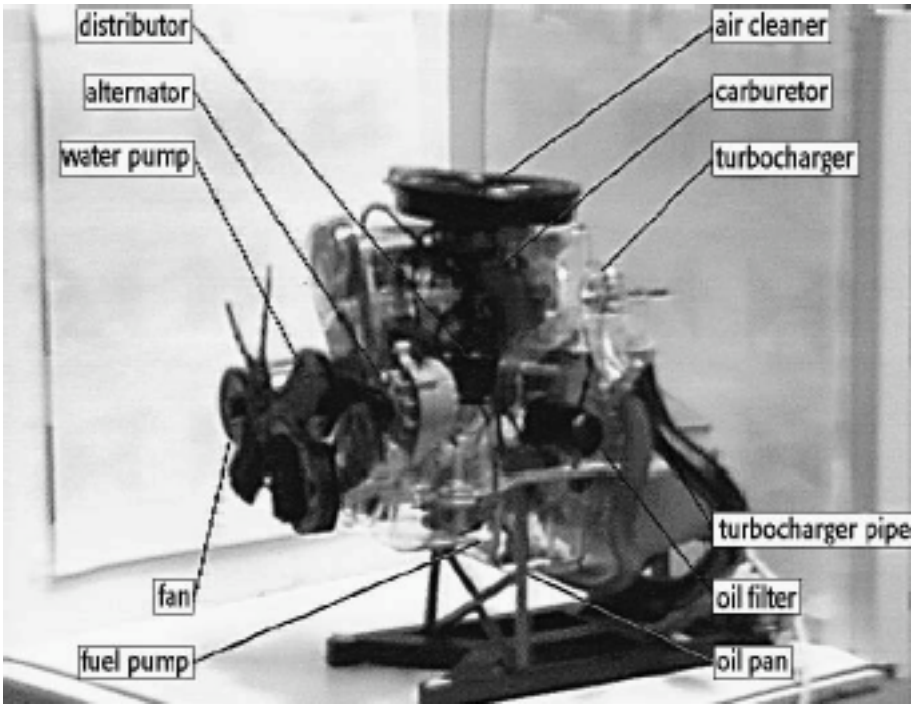

Figure 2. Augmented engine.

In the mechanical maintenance and repair scenario, a mechanic is assisted by an AR system while examining and repairing a complex engine (Kramer & Chevalier, 1996). The system presents a variety of information to the mechanic, as shown in Figure 2. Annotations identify the name of parts, describe their function, or present other important information like maintenance or manufacturing records. The user interacts with the real object in its natural setting with a pointing device monitored by the computer. As the mechanic points to a specific part of the engine, the AR system displays computer-generated lines and text (annotations) that describe the visible components or give the user hints about the object. Queries with the pointing device on the real-world object may be used to add and delete annotation tags. Since we also track the engine, the annotations move with the engine as its orientation changes. The lines attaching the annotation tags with the engine follow the appropriate visible components, allowing the user to easily identify the different parts as the view of the engine changes. The mechanic can also benefit from the assistance of a remote expert who can control what information is displayed on the mechanic's AR system.

4. System Infrastructure

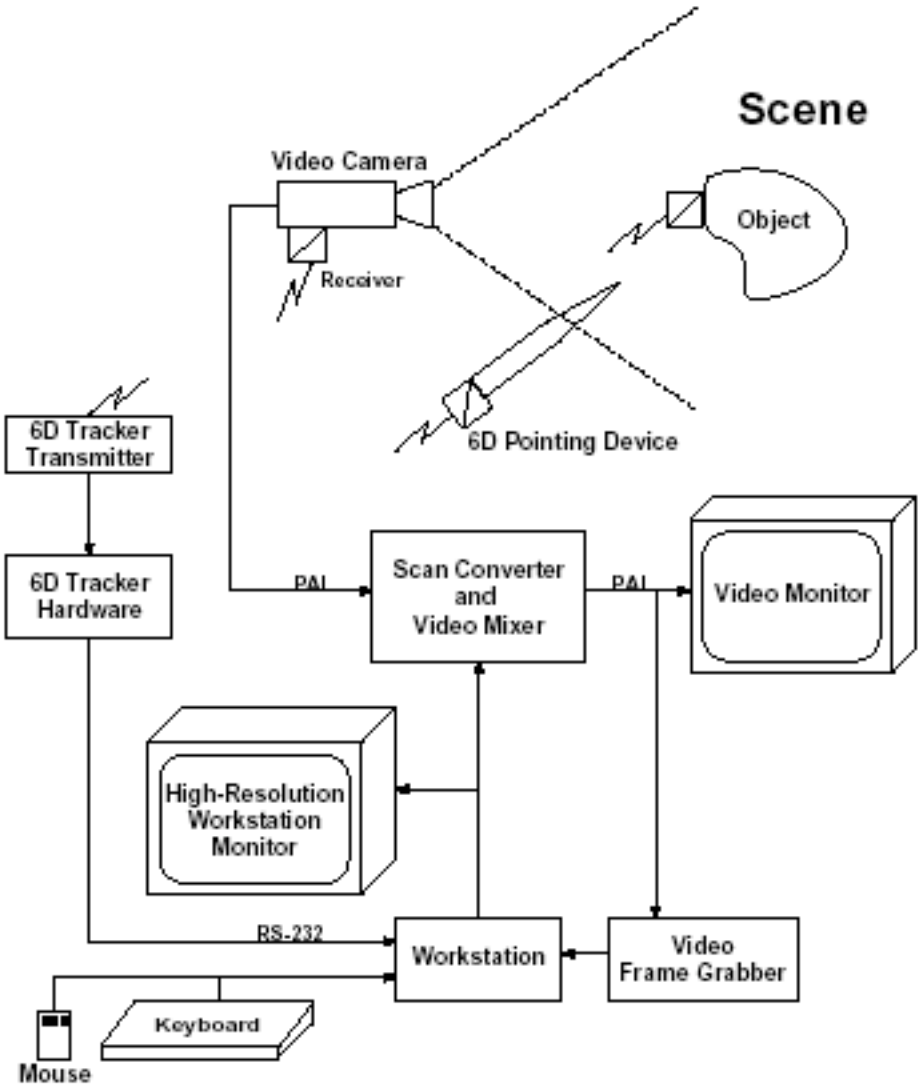

Figure 3. The GRASP system hardware configuration.

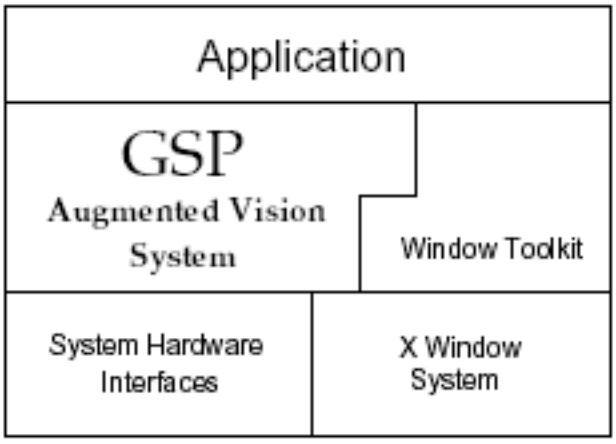

Figure 4. The GRASP system software configuration.

The GRASP system forms the central core of our efforts to keep the graphics and visual scene in alignment and to provide an interactive three-dimensional interface (Ahlers, Crampton, Greer, Rose & Tuceryan, 1994). Figure 3 shows a schematic of the GRASP hardware configuration. The workstation hardware generates the graphical image and displays it on a high resolution monitor. A scan converter transforms the graphics displayed on the monitor into a standard video resolution and format. The scan converter also mixes this generated video signal with the video signal input from the camera via luminance keying. A 6-DOF magnetic tracker, which is capable of sensing the three translational and the three rotational degrees of freedom, provides the workstation with continually updated values for the position and orientation of the tracked objects, including the video camera and the pointing device. A frame grabber digitizes video images for processing within the computer during certain operations. The software has been implemented using the C++ programming language. A schematic diagram of the software architecture is shown in Figure 4.

5. Specification and Alignment of Coordinate Spaces

In order to align the virtual and real objects seamlessly, we need very precise models of the user's environment and how it is sensed. It is essential to *calibrate* sensors and display devices (i.e., to determine their locations and optical properties), to *register* all objects and interaction devices in a global coordinate system, and to *track* them while the user operates in the scene.

5.1 Calibration of Sensors and Video Equipment

During the initial setup, the camera characteristics, the location of the 6D tracker and the effects of scan conversion and video mixing must be determined. These procedures are referred to as the *image*, *camera*, and *tracking calibration* (Tuceryan, Greer, Whitaker, Breen, Crampton, Rose & Ahlers, 1995). We now describe several such techniques that mix computer vision algorithms with varying amounts of model-based information and interactive input from the user.

5.1.1 Image Calibration

One of the essential steps of our AR system is the mixing of live video input with synthetically generated geometric data. While the live input is captured as an analog video signal by the camera system, the synthetic data is rendered digitally and then scan converted into a video signal. In order to align the two signals, we need to determine the horizontal and vertical positioning of the rendered, scan converted image with respect to the camera image, as well as the relationship between the two aspect ratios.

We use a synthetic test image that has two markers in known positions to compute four distortion parameters (2D translation and scaling). The test image is scan converted into a video signal. For image calibration purposes, we redigitize it and determine the location of the markers in the grabbed image. The discrepancy between the original location of the markers and their position in the grabbed image determines the translational and scaling distortions induced by the scan converter.[3] This interactive image calibration method asks the user to identify the two markers in the grabbed image.

The GRASP system also provides an alternative, automatic routine to compute the distortion parameters. Algorithmically, it is easier to find a large, homogeneously colored area in an image than the thin lines of a marker. Accordingly, the automatic algorithm uses a different test image which contains one black

[3] Distortion due to frame grabbing can be ignored since both the live signal and the synthetic signal pass through this device (Tuceryan, Greer, Whitaker, Breen, Crampton, Rose & Ahlers, 1995).

square. It finds the dark area, fits four lines to its boundaries and thus determines the corners of the square. Two of the corners suffice to determine the distortion parameters of the scan converter.

The comparison of the two approaches illustrates an important distinction between interactive and automatic algorithms: while humans work best with sharp line patterns to provide precise interactive input, automatic algorithms need to accommodate imprecision due to noise and digitization effects and thus work better on thicker patterns. On the other hand, automatic algorithms can determine geometric properties of extended areas more precisely than humans, such as the center, an edge or a corner of an area. In conclusion, it is essential to the design of a system and to its use in an application that visual calibration aides be chosen according to their intended use. This is a recurring theme in our work.

5.1.2 Camera Calibration

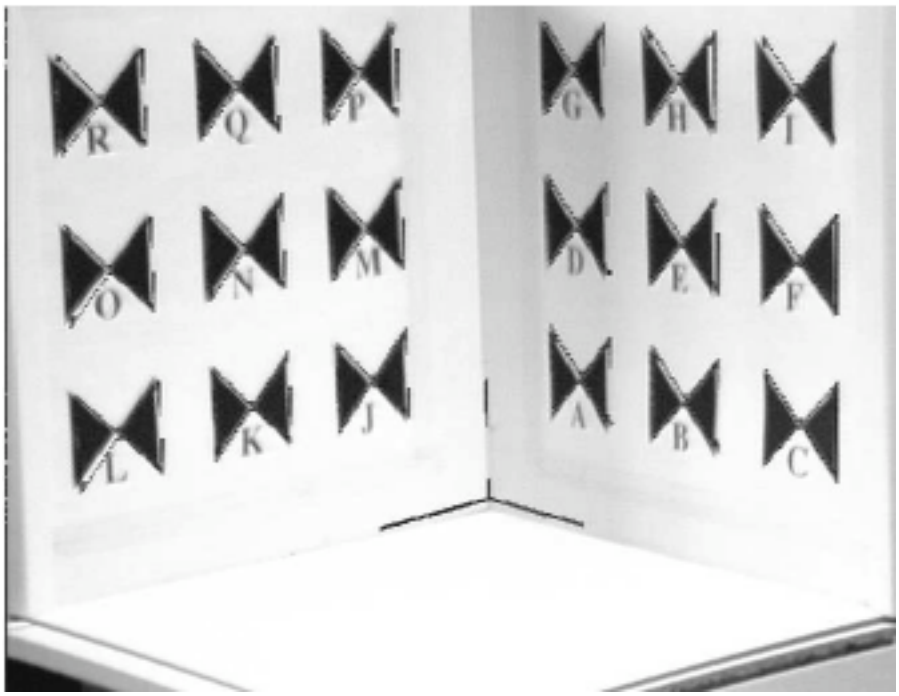

Figure 5. The camera calibration grid.

Camera calibration is the process which calculates the extrinsic (position and orientation) and intrinsic parameters (focal length, image center, and pixel size) of the camera. We assume that the intrinsic parameters of the camera remain fixed during the augmented reality session. The camera's extrinsic parameters may be tracked and updated.

To compute the camera's intrinsic and extrinsic parameters, we point the camera at a known object in the scene, the calibration grid shown in Figure 5. The position of the grid and, in particular, the position of the centers of the butterfly markers on the grid are known within the 3D world coordinate system. We use the mapping from these 3D object features to 2D image features to calculate the current vantage point of the camera and its intrinsic image distortion properties. In principle, each mapping from a 3D point to 2D image coordinates determines a ray in the scene that aligns the object point with the focal point of the camera. According to the pinhole camera model, several such rays from different object points intersect at the focal point and thus uniquely determine the pose of the camera, as well as its imaging properties. Accordingly, we can define a system of equations to compute the intrinsic and extrinsic camera parameters using a mapping of object points to image points and minimizing measurement errors. The details are described in (Tuceryan, Greer, Whitaker, Breen, Crampton, Rose & Ahlers, 1995).

The GRASP system provides an interactive camera calibration routine: A user indicates the center of all butterfly patterns with a mouse and labels them by typing the appropriate code name on the keyboard.

We also use an automatic, computer vision-based camera calibration algorithm. In this approach, we use a calibration board that shows an arrangement of 4×2 black squares on a white background. Processing the image at a coarse scale, we quickly determine the positions and extents of black blobs in the image. By fitting rectangles to the blob outlines at finer scales and matching them left to right and top to bottom to the squares of the calibration board, we determine the calibration parameters of the camera.

5.1.3 Magnetic Tracker Calibration

Although we emphasize in this paper the use of computer vision techniques for AR, we do not rely exclusively on optical information. Complementarily, we also exploit magnetic tracking technology, as well as other interactive or model-based input. The tracking system consists of a transmitter and several receivers (trackers) that can be attached to objects, cameras and pointers in the scene. The tracking system automatically relates the 3D position and orientation of each tracker to a tracking coordinate system in the transmitter box. It is the task of the tracker calibration procedure to determine where the tracking coordinate system resides with respect to the world coordinate system of the AR application. This is a critical issue that usually does not arise in VR applications since such systems only need to track relative motion. Yet, the absolute positioning and tracking of objects and devices within a real world coordinate frame is of greatest importance in AR scenarios where reality is augmented with virtual information.

At the beginning of each session, we calibrate the magnetic tracking system, relating its local coordinate system to the world coordinate system. This process is currently performed interactively, using the same calibration grid as for camera calibration. We do this by determining the location of at least three points on the calibration grid with magnetic trackers[4]. Since these points are also known in the world coordinate system, we can establish a system of linear equations, relating the tracked coordinates to the world coordinates and thus determining the unknown position and orientation parameters of the tracker (Tuceryan, Greer, Whitaker, Breen, Crampton, Rose & Ahlers, 1995).

5.2 Registration of Interaction Devices and Real Objects

In addition to the sensing devices that were calibrated in the previous section, scenes also contain physical objects that the user wants to interact with using 3D interaction devices. Such objects and gadgets need to be registered with respect to the world coordinate system.

5.2.1 Pointer Registration

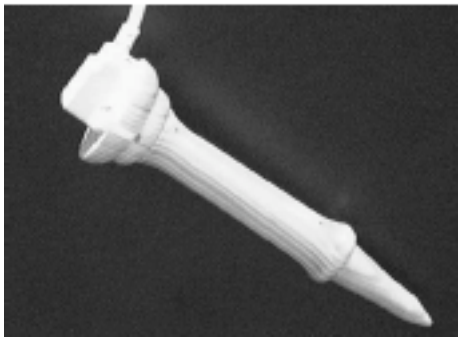

Figure 6. 3D pointing device.

[4] Using a calibrated pointer that is attached to one of the trackers — see section 5.1.1.

Currently, we use the magnetic tracking system to register and track the position of a 3D pointer in our system (see Figure 6).

For the pointer registration, we need to determine the position (offset) of the tip of a pointer in relationship to an attached magnetic tracker. Our procedure requires the user to point to the same point in 3D space several times, using a different orientation each time for a pointer that has been attached to one of the trackers. For each pick, the position and the orientation of the tracker mark within the tracker coordinate system are recorded. The result of this procedure is a set of points and directions with the common property that the points are all the same distance from the single, picked point in 3D space and all of the directions associated with the points are oriented toward the picked point. From this information, we can compute six parameters defining the position and orientation of the pointing device, using a least-squares approach to solve an overdetermined system of linear equations.

5.2.2 Object Registration

Object registration is the process of finding the six parameters that define the 3D position and orientation, i.e. pose, of an object relative to some known coordinate system. This step is necessary, even when tracking objects magnetically, in order to establish the 3D relationship between a magnetic receiver and the object to which it is fastened.

We have studied two strategies for determining the 3D pose of an object (Whitaker, Crampton, Breen, Tuceryan & Rose, 1995). The first is a camera based approach, which relies on a calibrated camera to match 3D landmarks ("calibration points") on the object to their to projection in the image plane. The second method uses the 3D coordinates of the calibration points, as indicated manually using the 3D pointer with magnetic tracking, in order to infer the 3D pose of the object.

There has been extensive research in pose determination in the computer vision (Lowe, 1985; Grimson, 1990), but most of these techniques apply to only limited classes of models and scenes. The focus of the computer vision research is typically automation and recognition, features that are interesting, but not essential to augmented vision. In our work, the locations of landmark points in the image are found manually by a user with a mouse. We assume that the points are mapped from known locations in 3-space to the image via a rigid 3D transformation and a projection.

We represent the orientation of the object as 3×3 rotation matrix, which creates a linear system with 12 unknowns. Each point gives 2 equations, and 6 points are necessary for a unique solution. In practice we assume noise in the input data and use an overdetermined system with a least squared solution in order to get reliable results. However, because we use a 3×3 rotation matrix, $\mathbf{R}$, and treat each element as an independent parameter, this linear system does not guarantee an orthonormal solutions for this matrix, and it can produce "non-rigid" rotation matrices. Such non-rigidities can produce undesirable artifacts when these transformations are combined with others in the graphics system.

Orthonormality is enforced adding an additional penalty to the least-squared solution, $(\mathbf{R}\mathbf{R}^T - \mathbf{I})^2$. This creates a nonlinear optimization problem which we solve through gradient descent. The gradient descent is initialized with the unconstrained (linear) solution, and constrained solutions are typical found in 10-15 iterations.

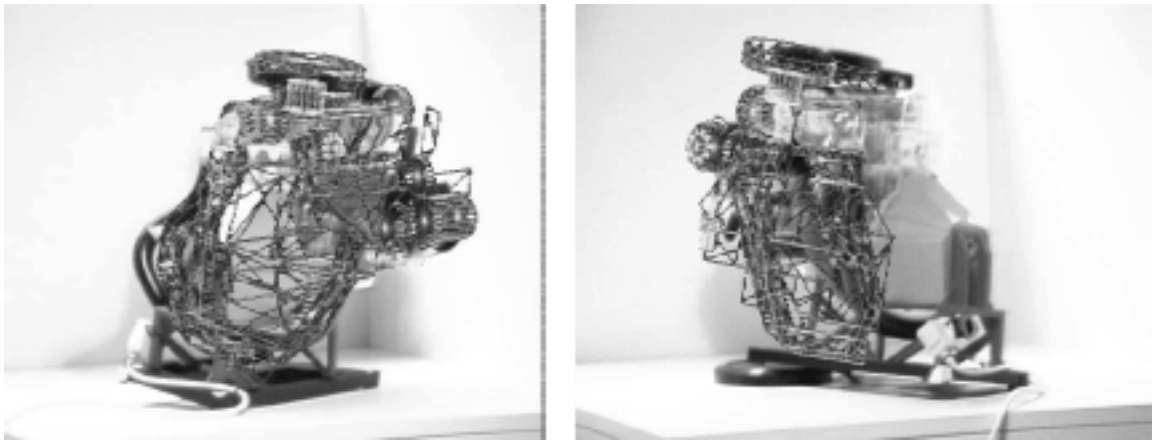

Figure 7. Calibration and tracking an engine model: A wireframe engine model registered to a real model engine using an image-based calibration (a), but when the model is turned and its movements tracked (b), the graphics show the misalignment in the camera's *z*-direction.

Despite good pointwise alignment in the image plane, the image-based calibration can produce significant error in the depth term which is not seen in the reprojected solutions. For instance, in the case of the engine model shown in Figure 7(a), the image-based approach can produce a rigid transformation which matches landmark points in the image to within about 2 pixels. Yet the error in the *z*-direction (distance from the camera) can be as much as 2-3 centimeters. This error becomes evident as the object is turned as in Figure 7(b). We attribute this error primarily to error in the camera calibration, and better camera models and calibration procedures are a topic of ongoing research. Because of such error we have developed the procedure described in the next section for calibrating objects with a 3D pointing device.

The problem here is to compute the rigid transformation between a set of 3D point pairs. Using the 3D pointer and several keystrokes the user indicates the world coordinates (or some other *known* coordinate system) of landmark points on the object. also gives rise to a linear system of 12 unknowns. For a unique solution 4 points are needed, but in most cases we use more than 4 points and solve for the least-squares error. As with the image-based object calibration, error in the measurements can produce solutions that represent non-rigid transformations. Thus, the same nonlinear penalty term can be introduced in order produce constrained solutions.

5.3 Tracking of Objects and Sensors

Calibration and registration refer to stationary aspects of a scene. In a general AR scenario, however, we have to deal with dynamic scene changes. With tracking we denote the ability of our system to cope with those dynamic scene changes. Thus, while the calculation of the external camera parameters and of the pose of an object are the results of calibration and registration, tracking can be regarded as a continuos update of those parameters. We are currently exploring and using two approaches to tracking, magnetic tracking, and optical tracking.

5.3.1 Magnetic Tracking

As a magnetic tracking device we use the 6D tracker “Flock of Birds” from Ascension Technology Corporation. Receivers are attached to the camera and each potential moving object. These receivers sense the six degrees of freedom (three translational and three rotational) with respect to a transmitter, whose location is being kept fixed in world coordinates.

Initially, we have relied exclusively on this magnetic technology since the trackers provide positional and orientational updates at nearly real-time speeds and operate well in a laboratory setup. However, magnetic

tracking is not practicable in large scale, realistic setups, because the tracking data can easily be corrupted by ferro-magnetic materials in the vicinity of the receiver and because the trackers operate only in a limited range. Another drawback is the limited accuracy of the sensor readings.

5.3.2 Optical Tracking

Optical tracking methods are based on detecting and tracking certain features in the image. These can be lines, corners or any other salient features, which are easy and reliable to detect in the image and can uniquely be associated with features of the 3D world. Our tracking approach currently uses the corners of squares attached to objects or walls (seeFigure 8) to track a moving camera. Once the camera parameters are recovered, the scene can be augmented with virtual objects, such as shelves and chairs (see Figure 9).

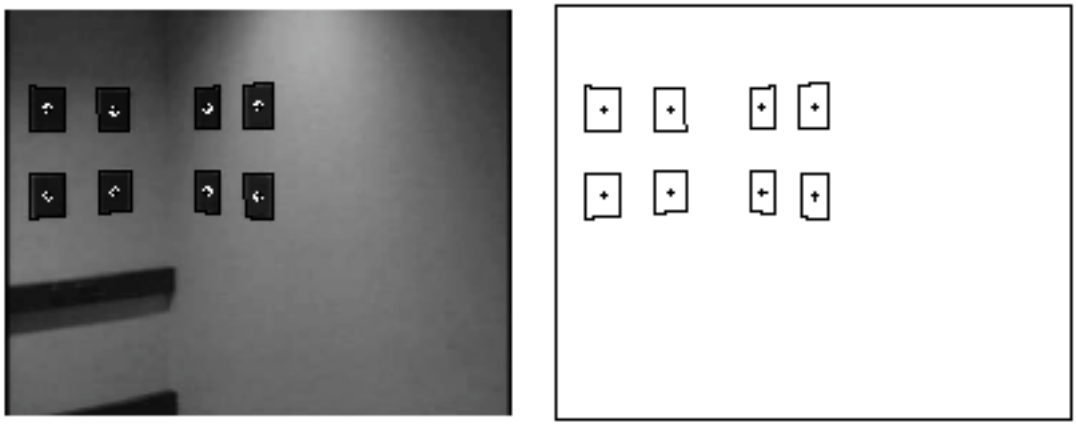

Figure 8. Our optical tracking approach currently tracks the corners of squares. The left figure shows a corner of a room with eight squares. The right figure shows the detected squares only.

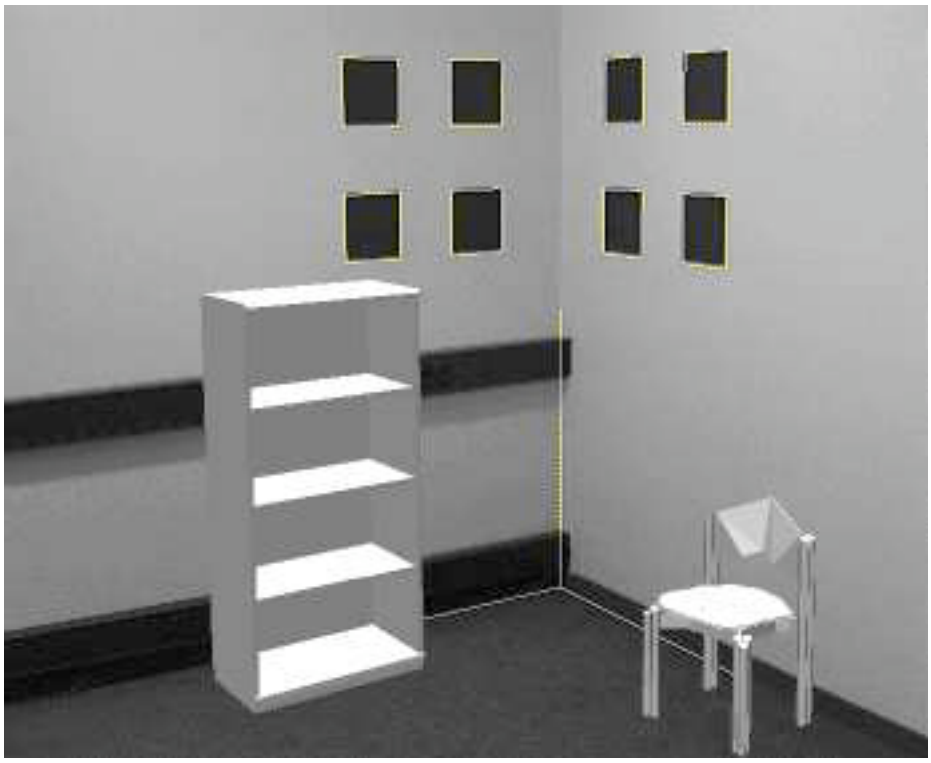

Figure 9. Augmented scene with a virtual chair and shelf that were rendered using the automatically tracked camera parameters.

This scenario is relevant to many AR applications where a user moves in the scene and thus continuously changes his (the camera's) viewpoint. We use a fixed world coordinate system, thus recomputing the camera parameters relative to the world frame in each step. Conversely, we could also recompute the position of the world system relative to the camera frame, thus using an egocentric frame of refererence. The advantage of the former approach is that we can thus exploit certain motion invariants which make the tracking problem much simpler.

We assume that a model of the scene exists and that we are able to add "fiducial marks", such as black squares, to the scene to aid the tracking process. The squares are registered in the 3D scene model. Thus, in principle, the same camera calibration techniques described in section 5.1.2. can be used to determine, at any point in time, the position of the camera in scene. Yet, during the tracking phase, we need to pay particular attention to speed and robustness of the algorithms. To our advantage, we can exploit time coherence of user actions: users move in continuous motions. We can benefit from processing results of previous images and from an adaptive model of the user motion to predict where the tracked features will appear in the next frame. We thus do not need to perform the full camera calibration procedure on every new incoming image.

It is well known that reasoning about three dimensional information from two dimensional images is error prone and sensitive to noise, a fact which has to be taken into account in any image processing method using real video data. In order to cope with this noise sensitivity we exploit physical constraints of moving objects. Since we do not have any a priori knowledge about forces changing the motion of the camera or the objects, we assume no forces (accelerations) and hence a constant velocity. In this case a general motion can be decomposed in a constant translational velocity of the center of mass of the object, and a rotation with constant angular velocity around an axis through the center of mass (e.g. Goldstein, 1980). This constitutes our so-called *motion model* (see Figure 10). So we do not only measure (estimate) the position and orientation of the camera and moving objects — as in the case of magnetic tracking — but also their change in time with respect to a stationary world frame, i.e. their translational and angular velocity. This is also referred to as *motion estimation*.

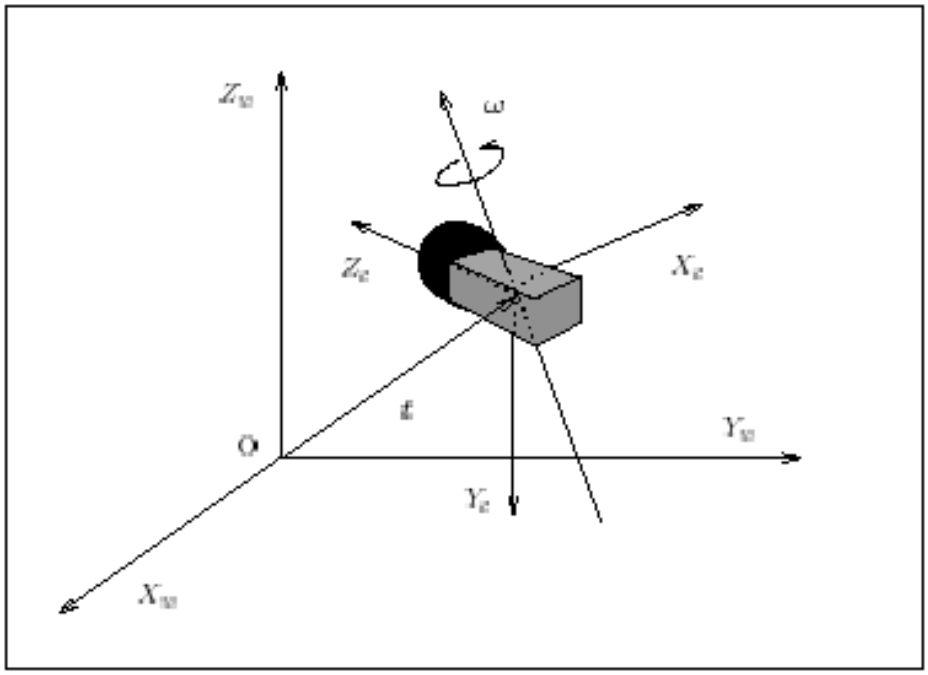

Figure 10. Each 3D motion can be decomposed in a translation *t* and a rotation Ω. We choose a rotation about an axes through the center of mass of the objects, which is constant in the absence of any forces. (X_w, Y_w, Z_w) denotes the world coordinate frame, and (X_c, Y_c, Z_c) denotes the camera coordinate frame.

The motion parameters (translational and angular velocity according to the motion model) are estimated using time-recursive filtering based on Kalman Filter techniques (e.g. Bar-Shalom & Fortmann, 1988; Gelb, 1974), where the unknown accelerations are successfully modeled as so-called *process noise*, in order to allow for changes of the velocities. The time-recursive filtering process enables smooth motions even in the presence of noisy image measurements, and enables a prediction-measurement-update step for each video frame. The prediction allows a reduction of the search space for features in the next video image and hence speeds up the process.

A typical drawback of optical methods is based on the fact that we want to reason about three dimensional information from two dimensional image measurements, which can lead to numerical instabilities if not performed carefully. On the other hand there is the advantage of the image of real objects being almost perfectly aligned with the rendered counterpart since the alignment error can be minimized in the image. Optical tracking approaches can hence be very accurate. Another advantage of optical tracking is that it is a nonintrusive approach, since it operates just on visual information, and it is basically not limited to any spatial range. It is furthermore somehow natural since it is the way most humans track objects[5] and navigate within an environment.

6. Object Interaction

Realistic immersion of virtual objects into a real scene requires that the virtual objects behave in physically plausible manners when they are manipulated, i.e.: they occlude or are occluded by real objects, they are not able to move through other objects, and they are shadowed or indirectly illuminated by other objects while also casting shadows themselves. To enforce such physical interaction constraints between real and virtual objects, the Augmented Reality system needs to have a very detailed description of the physical scene.

6.1 Acquisition of 3D Scene Descriptions

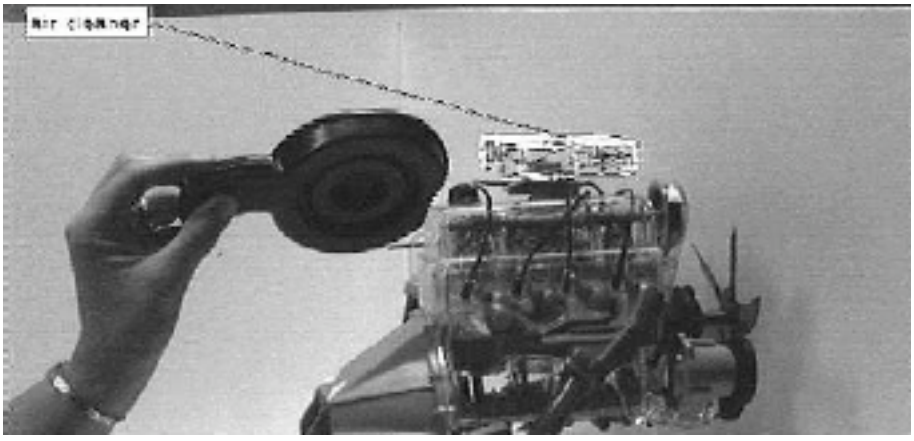

Figure 11. Modified Engine. The fact that the user has removed the air cleaner is not yet detected by the AR system. The virtual model thus does not align with its real position.

The most straightforward approach to acquiring scene descriptions would suggest the use of geometric models, e.g., CAD-data. Given such models, the AR system needs to align them with their physical counterparts in the real scene, as described in section 5.2.2. The advantage of using such models is that

[5] Tracking objects in the context of *active vision* refers more to the goal of keeping an object in the fovea instead of updating its pose in each frame.

they can easily serve as starting points for accessing high-level, semantic information about the objects, as is demonstrated in the mechanical repair application.

However, there are some problems with this approach. First, geometric models are not available in all cases. For example, interior restoration of old buildings typically needs to operate without CAD-data. Second, available models are not complete. Since models are abstractions of reality, real physical objects typically show more detail than is represented in the models. In particular, generic scene models cannot fully anticipate the occurrence of new objects, such as coffee mugs on tables, cars or cranes on construction sites, users' hands, or human collaborators. Furthermore, the system needs to account for the changing appearances of existing objects, such as buildings under construction or engines that are partially disassembled (see Figure 11). When users see such new or changed objects in the scene, they expect the virtual objects to interact with these as they do with the rest of the (modeled) scene.

Computer vision techniques can be used to acquire additional information from the particular scene under inspection. Although such information generally lacks semantic descriptions about the scene and thus cannot be used directly to augment reality with higher-level information, such as the electric wiring within a wall, it provides the essential environmental context for the realistic immersion of virtual objects into the scene. Thus, we expect future AR systems to use hybrid solutions, using model data to provide the necessary high-level understanding of the objects that are most relevant to the tasks performed, and enriching the models with automatically acquired further information about the scene.

We are investigating how state-of-the-art image understanding techniques can be used in AR applications. One particular paradigm in computer vision, *shape extraction*, determines depth information as so-called *2_-D sketches* from images. These are not full 3D descriptions of the scene but rather provide distance (depth) estimates, with respect to the camera, for some or all pixels in an image. Ongoing research develops techniques to determine object shape from stereo images, from motion sequences, from object shading, from shadow casting, from highlights and gloss, and more. It is important to consider whether and how such algorithms can be used continuously, i.e., while the user is working in the scene. Alternatively, the algorithms could be used during the initial setup phase, gathering 3D scene information once and compiling a rough sketch of the scene that then needs to be updated with other techniques during the AR session. Yet other options involve the use of other sensing modalities besides cameras, such as laser range scanners or sonar sensors.

This section discusses two approaches we are investigating.

6.1.1 Dense Shape Estimates from Stereo Data

Stereo is a classical method of building three-dimensional shape from visual cues. It uses two calibrated cameras with two images of the scene from different vantage points. Using *stereo triangulation*, the 3D location of dominant object features that are seen in both images can be determined: if the same point on an object is seen in both images, rays cast from the focal points of both cameras through the feature positions in the images intersect in 3D space, determining the distance of the object point from the cameras.

Shape from stereo has been studied extensively in the computer vision literature. The choice of image feature detection algorithms and of feature matching algorithms between images is of critical importance. Depending on the type of methods and algorithms one uses, shape from stereo may result in sparse depth maps or dense depth maps. For our research, the goal is to use the computed 3D shape information in the AR applications. In most if not all such scenarios, the availability of *dense* maps are needed. Therefore, we have taken an existing algorithm (Weng, Huang & Ahuja, 1989) to compute a dense depth map which is used in the AR context. The camera geometry is obtained by calibrating both cameras independently using one of the camera calibration methods described in section 5.1.

The details of the stereo algorithm are given in the paper (Weng, Huang & Ahuja, 1989). In summary, the heart of the algorithm lies in the computation of the *disparity map (du, dv)* which describes the distance between matched points in both images. This is accomplished by computing matches between four kinds of image features derived from the original images: smoothed intensity images, edge magnitudes, positive

corners, and negative corners. The positive and negative corners separate the contrast direction at a corner. Distinguishing between these four feature types improves the matching results by preventing that incompatible image features are matched between the images, such as positive and negative corners.

The overall algorithm iteratively determines the (locally) best match between the image features that have been computed in both images. Starting with an initial hypothetical match, the matches are iteratively changed and improved, minimizing an energy function which integrates — over the entire image — the influence of several error terms related to the quality of the edge matches between the left and right image, as well as a smoothness term which ensures that the recovered surface is not exceedingly rough and noisy.

Figure 12 shows a pair of stereo images. The disparity maps computed from these images are shown in Figure 13 and the depth map is shown in Figure 14(a). Finally, Figure 14(b) shows how the computed depth map is used to occlude three virtual floating cubes.

Figure 12. An example pair of stereo images: (a) Left image and (b) Right image.

Figure 13. The disparities computed on the stereo pair in Figure 12(a) disparities in rows (*du*) and (b) disparities in columns (*dv*). The brighter points have larger disparities.

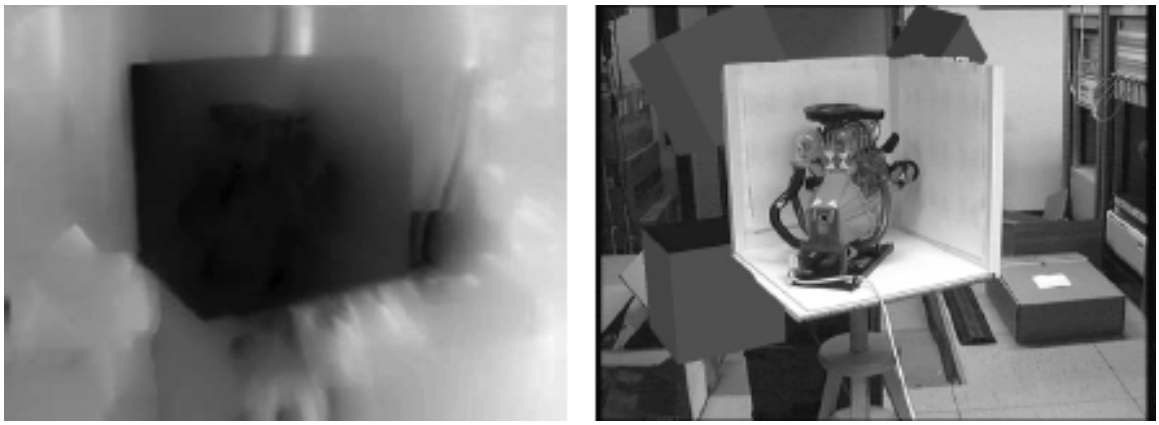

Figure 14. (a) The computed depth map from the pair of images in Figure 12. The brighter points are farther away from the camera. (b) The computed depth map in (a) is used to occlude the virtual object (in this case a cube) which has been added in the scene.

6.1.2 Shape from Shading

Complementary to geometric shape extraction methods, some approaches exploit the photometric reflection properties of objects. An image of a smooth object with uniform surface reflectance properties exhibits smooth variations in the intensity of the reflected light referred to as *shading*. This information is used by human and other natural vision systems to determine the shape of the object. The goal in shape from shading is to replicate this process to the point of being able to design an algorithm that will automatically determine the shape of a smooth object from its image (Horn & Brooks, 1989).

This shape information can be used in a number of application areas where knowledge of the spatial characteristics in a scene is important. In particular, shape from shading information can fill the gaps in sparse depth maps that are left by geometry-based shape extraction methods. Geometric extraction works best on highly textured objects where many features can be matched between images. Shape from shading, on the other hand, can propagate shape information into homogeneous areas.

We are investigating how the second derivative, or *hessian*, of a smooth object surface can be determined directly from shading information. The method of *characteristic strips* which is often used for calculating shape from shading (Horn, 1986), is set in the framework of modern differential geometry. We extend this method to compute the second derivative of the objects surface, independently from the standard surface orientation calculation. This independently derived information can be used to help classify critical points, verify assumptions about the reflectance function and identify effectively impossible images (Greer & Tuceryan, 1995).

6.2 Mixing of Real and Virtual Worlds

Once appropriate scene descriptions have been obtained interactively or automatically, they form the basis for mixing real and virtual worlds. Since the mixing must be performed at interactive rates, great emphasis has to be placed on efficiency. Depending on the representation of the scene descriptions, different options can be pursued.

If the scene description is available as a geometric model, we can hand the combined list of real and virtual models to the geometric renderer which will then compute the interactions between real and virtual objects for us. By rendering models of real objects in black, we can use the *luminance keying* feature[6] of the video

[6] When a mixer combines two input signals into one output signal, it uses — independently for every pixel position — the luminance value of the pixel from the first input as a decision criterion: if the pixel

mixer to substitute the respective area with live video data. As a result, the user sees a picture on the monitor that blends virtual objects with live video, while respecting 3D occlusion relationships between real and virtual objects.

This is a straightforward approach in applications where geometric, polygonal scene descriptions are available. If the descriptions are computed as depth maps, as described in section 6.1, the depth maps still need to be converted into a geometric representation, by tessellating and decimating the data (Schroeder, Zarge & Lorensen, 1992; Turk, 1992).

Alternatively, we can side-step the tessellation and rerendering phases for real objects by initializing the Z-buffer of the graphics hardware with the depth map (Wloka & Anderson, 1995). Occlusion of the virtual objects is then performed automatically. When the virtual object is rendered, pixels that are further away from the camera than the Z values in the depth map are not drawn. By setting the background color to black, the real objects present in the original video are displayed in these unmodified pixels. Figure 14(a) presents three virtual cubes occluded by a wooden stand with an engine and occluding the other objects in a real room, using the depth-based approach.

These approaches have advantages and disadvantages, depending on the application. Full 3D geometric models are best for real-time movement of cameras. Polygonal approximations to depth maps can be used over a certain range of camera positions[7] since the synthesized scene model is rerendered when the camera moves. Copying the depth maps directly into the Z-buffer is the hardest approach: the map needs to be recomputed after each camera motion because the new projective transformation "shifts" all depth values in the depth map. Thus, this approach only works with stationary cameras or with shape extraction algorithms that perform at interactive speeds.

On the other hand, the geometric modeling approach suffers from an inherent dependence on scene complexity. If the scene needs to be represented by a very large polygonal model, the rendering technology may not be able to process it in real time. In contrast, the size of a depth map does not depend on scene complexity. Which approach to use in an application depends on the overall requirements and the system design.

7. Collaborative Use of AR

So far we were discussing techniques and solutions that make AR "work" for the single user. Object modeling, object interaction, realistic display and immersive interfaces all serve to present the user with a consistent and coherent world of real and virtual objects.

When we consider the application scenarios described above we are reminded of the fact that in any virtual or real environment it appears natural to encounter other persons and to interact with them. Virtual environments are a promising platform for research in the CSCW area, and distributed multi-user interfaces are a challenge for many VE systems (e.g. the efforts related to the VRML proposal (Bell, Parisi & Pesce, 1995)). In the context of the GRASP system, we are interested in the problem and the paradigms of distributed AR. We are investigating solutions in the area of distributed computing and experiment with system architectures for collaborative interfaces to shared virtual worlds.

7.1 Architecture for Shared AR

---

has a specified luminance value (black), the corresponding pixel from the second video stream is chosen. Otherwise, the first video stream has preference.

[7] Depth maps, by their very nature, do not provide full 3D information since large parts of the scene — e.g., the back faces of objects — are not seen by a single view. Derived polygonal models suffer from the same incompleteness. Large camera motions uncover previously unseen areas of the scene and thus need new descriptions.

Each system supporting multi-user virtual environments can be characterized by the degree or type of concurrency, distribution, and replication in the system architecture (Dewan, 1995). Sharing between users has to be based on separability in the user interface: we call the database of shared logical objects the "model", and create "views" as a specific interpretation of the model in each interface. The need for rapid feedback in the user interface makes a replicated architecture attractive for AR. This in turn leads to object-level sharing where each user can view and manipulate objects independently. It is necessary to manage the shared information so that simultaneous and conflicting updates do not lead to inconsistent interfaces. This is guaranteed by the distribution component in our applications.

The model replication and distribution support allow the user interfaces of one application to execute as different processes on different host computers. GRASP interfaces are not multi-threaded, so the degree of distribution corresponds to the degree of concurrency in the system. The resulting architecture was implemented and successfully used in the interior design demonstration.

7.2 Providing Distribution

The replicated architecture is directly supported by the Forecast library of the GRASP system. Based on a message bus abstraction, Forecast provides an easy, reliable, and dynamic approach to constructing distributed AR applications.

Central to this support is a one-to-many reliable communication facility which can be described as a distributed extension of a hardware system bus. Components, situated on different machines, can dynamically connect to the same distributed bus and send and receive messages over it. This analogy has been used before for group communication or broadcast systems and its messaging and selection capability are common to systems such as Linda and Sun's ToolTalk (Sunsoft, 1991).

The Forecast message bus implements a one-to-many FIFO (first in first out) multi-cast transport protocol. A special sequencer process is used to impose a unique global ordering on messages. In the simpler form of the protocol, nodes that wish to broadcast send their message to the sequencer which then uses the one-to-many reliable protocol to disseminate the message. A unique global order is imposed on the message streams since all messages pass through the sequencer. Nodes can detect how their messages were scheduled by listening to the global message stream. The protocol is similar to the Amoebae reliable multi-cast protocol (Kaashoek & Tanenbaum, 1992), except that it uses reliable buffered transmission between nodes and the sequencer node at the expense of extra acknowledgments.

We choose the message bus abstraction because it provides location, invocation and replication transparency for applications (Architecture Projects Management, 1989) which makes the programming of these applications easier. GRASP programmers are familiar with the concept of multiple local views and events, both of which we have extended to our distributed setting.

The Forecast message bus is used within our two collaborative AR demonstrators to implement model replication, direct interaction between components (e.g., to send pointer tracking information to remote participants), and also using generic functions like floor control and locking, state transfer, shared manipulators, video transmission (based on the MBONE audio and video library (Macedonia & Brutzman, 1994), and synchronization between video and tracking events (using RTP style time-stamps).

8. Discussion

Using Augmented Reality in realistic applications requires the computer to be very well informed about the 3D world in which users perform their tasks. To this effect, AR systems use various different approaches to obtain, register and track object and scene models. Of particular importance are different sensing devices, such as cameras or magnetic trackers. They provide the essential real-time link between the computer's internal, "virtual" understanding of the world and reality. All such sensors need to be calibrated carefully so that the incoming information is in alignment with the physical world.

Sensor input is not used to its full potential in current AR systems — due to real-time constraints, as well as due to the lack of algorithms that interpret signals or combine information from several sensors. Research fields such as computer vision, signal processing, pattern recognition, speech processing, etc. have investigated such topics for some time. Some algorithms are maturing so that — considering the projected annual increases in computer speed — it should soon become feasible to consider their use in AR applications. In particular, many applications operate under simplified (engineered) conditions so that scene understanding becomes an easier task than the general *Computer Vision Problem* (see, for example (Marr, 1980)).

We operate at this borderline between computer vision and AR, injecting as much automation into the process as feasible while using an engineering approach towards simplifying the tasks of the algorithms. In this respect, we emphasize the hybrid use of various different techniques, including interactive user input where convenient, as well as other sensing modalities (magnetic trackers). This paper has shown how we have developed and explored different techniques to address some of the important AR issues. Our pragmatic approach has allowed us to build several realistic demonstrations. Conversely, these applications influence our research focus, indicating clearly the discrepancy between the state of the art and what is needed. Tradeoffs between automation and assistance need to be further explored. User interaction should be reserved as much as possible to the high-level control of the scene and its augmentation with synthetic information from multi-media data bases. More sensing modalities need to be explored which will allow the user to interact with the computer via more channels, such as gesture and sound. Experimentation with head-mounted, see-through displays is crucial as well — especially in regard to the question whether and how the AR system can obtain optical input similar to what the user sees so that computer vision techniques can still be used. The foremost concern, however, remains the provision of fast, real-time interaction capabilities with real and virtual objects integrated seamlessly in an augmented world. To this end, the accurate modeling, tracking and prediction of user or camera motion is essential.

A related research direction leads us to investigate the collaborative use of Augmented Reality. As reported in this paper, we have developed a distributed infrastructure so that all our demonstrations can operate in a collaborative setting. We consider the collaborative use of AR technology to be a key interaction paradigm in the emerging global information society. The highly interactive, visual nature of AR imposes hard requirements on the distributed infrastructure, and demands the development of appropriate collaboration styles.

Augmented Reality, especially in a collaborative setting, has the potential to provide much easier and more efficient use of human and computer skills by merging the best capabilities of both. Considering the rapid research progress in this field, we expect futuristic scenarios like collaborative interior design, or joint maintenance and repair of complex mechanical devices to soon become reality for the professional user.

Acknowledgments

This work was financially supported by Bull SA, ICL PLC, and Siemens AG. We would like to thank the director of ECRC, Alessandro Giacalone, for many stimulating discussions regarding potential application scenarios for distributed, collaborative Augmented Reality. Many colleagues at ECRC, especially Stefane Bressan and Philippe Bonnet, contributed significantly to the successful implementation and presentation of the Interior Design and Mechanical Repair demonstrations, providing other key pieces of technology (data base access) that were not discussed in this paper.

References

Ahlers, K.H., Crampton, C., Greer, D., Rose, E., & Tuceryan, M. (1994). Augmented vision: A technical introduction to the GRASP 1.2 system. Technical Report ECRC-94-14, http://www.ecrc.de.

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

Architecture Projects Management. (1989). *ANSA: An Engineer's Introduction to the Architecture*. APM Limited, Poseidon House, Cambridge CB3 ORD, United Kingdom, Nov.

Azuma, R., & Bishop, G. (1994). Improving static and dynamic registration in an optical see-through display. *Computer Graphics*, July, 194-204.

Bajura, M., Fuchs, H., & Ohbuchi, R. (1992). Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. *Computer Graphics*, July, 203-210.

Bajura, M., & Neumann, U. (1995). Dynamic registration correction in augmented-reality systems. *Proc. of the Virtual Reality Annual International Symposium (VRAIS '95)*, 189-196.

Bar-Shalom, Y., & Fortmann, T.E. (1988). *Tracking and Data Association*. Academic Press, New York.

Baudel, M., & Beaudouin-Lafon, M. (1993). Charade: Remote control of objects using freehand gestures. *Communications of the ACM*, 37(7), 28-35.

Bell, G., Parisi, A., & Pesce, M. (1995). The virtual reality modeling language, version 1.0 specification. http://vrml/wired.com/vrml.tech/

Betting, F., Feldmar, J., Ayache, N., & Devernay, F. (1995). A framework for fusing stereo images with volumetric medical images. *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed '95),* 30-39.

Caudell, T., & Mizell, D. (1992). Augmented reality: An application of heads-up display technology to manual manufacturing processes. *Proc. of the Hawaii International Conference on System Sciences*, 659-669.

Deering, M. (1992). High resolution virtual reality. *Computer Graphics*, 26(2), 195-202.

Dewan, P. (1995). Multi-user architectures. *Proc. EHCI'95*.

Drascic, D., Grodski, J.J., Milgram, P., Ruffo, K., Wong, P., & Zhai, S. (1993). Argos: A display system for augmenting reality. *Formal video program and proc. of the Conference on Human Factors in Computing Systems (INTERCHI'93),* 521.

Feiner, S., MacIntyre, B., & Seligmann, D. (1993). Knowledge-based augmented reality. *Communications of the ACM*, 36(7), 53-62.

Fournier, A. (1994). Illumination problems in computer augmented reality. *Journée INRIA, Analyse/Synthèse D'Images*, Jan, 1-21.

Gelb, A. (ed.) (1974). *Applied Optimal Estimation*. MIT Press, Cambridge, MA.

Gleicher, M., & Witkin, A. (1992). Through-the-lens camera control. *Computer Graphics*, July, 331-340.

Goldstein, H. (1980). *Classical Mechanics*, Addison-Wesley, Reading, MA.

Gottschalk, S., & Hughes, J. (1993). Autocalibration for virtual environments tracking hardware. *Computer Graphics*, Aug., 65-72.

Greer, D.S., & Tuceryan, M. (1995). Computing the hessian of object shape from shading. Technical report ECRC-95-30, http://www.ecrc.de.

Grimson, W.E.L., Ettinger, G.J., White, S.J., Gleason, P.L., Lozano-Perez, T., Wells, W.M. III, & Kikinis, R. (1995). Evaluating and validating an automated registration system for enhanced reality visualization in surgery. *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95),* 3-12.

Grimson, W.E.L., Lozano-Perez, T., Wells, W.M. III, Ettinger, G.J., White, S.J., & Kikinis, R. (1995). An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95),* 430-436.

Grimson, W.E.L. (1990). *Object Recognition by Computer*. MIT Press, Cambridge, MA.

Henri, C.J., Colchester, A.C.F., Zhao, J., Hawkes, D.J., Hill, D.L.G., & Evans, R.L. (1995). Registration of 3D surface data for intra-operative guidance and visualization in frameless stereotactic neurosurgery. *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95),* 47-58.

Holloway, R. (1994). *An Analysis of Registration Errors in a See-Through Head-Mounted Display System for Craniofacial Surgery Planning*. Ph.D. thesis, University of North Carolina at Chapel Hill.

Horn, B.K.P. (1986). *Robot Vision*. MIT Press, Cambridge, MA.

Horn, B.K.P., and Brooks, M.J. (1989). *Shape from Shading*. MIT Press, Cambridge, MA.

Janin, A., Mizell, D., & Caudell, T. (1993). Calibration of head-mounted displays for augmented reality applications. *Proc. of the Virtual Reality Annual International Symposium (VRAIS'93)*, 246-255.

Kaashoek, M.F., & Tanenbaum, A.S. (1992). Fault Tolerance using Group Communication. *Operating Systems Review*.

Kancherla, A.R, Rolland, J.P., Wright, D.L., & Burdea, G. (1995). A Novel Virtual Reality Tool for Teaching Dynamic 3D Anatomy. *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95),* 163-169.

Kramer, A., & Chevalier, P.-Y. (1996). Distributing augmented reality. Submitted to *Virtual Reality Annual International Symposium (VRAIS'96).*

Lorensen, W., Cline, H., Nafis, C., Kikinis, R., Altobelli, D., & Gleason, L. (1993). Enhancing reality in the operating room. *Proc. of the IEEE Conference on Visualization*, 410-415.

Lowe, D. (1985). *Perceptual Organization and Visual Recognition.* Kluwer Academic, Norwell, MA.

Macedonia, M.R., & Brutzman, D.P. (1994). MBONE provides audio and video across the internet. *IEEE Computer*, April.

Marr, D. (1980). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Freeman, San Francisco.

Mellor, J.P. (1995). Real-time camera calibration for enhanced reality visualizations. *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95),* 471-475.

Milgram, P., Zhai, S., Drascic, D., & Grodski, J.J. (1993). Applications of augmented reality for human-robot communication. *Proc. of the International Conference on Intelligent Robots and Systems (IROS'93),* 1467-1472.

Peria, O., Chevalier, L. François-Joubert, A., Caravel, J.-P., Dalsoglio, S., Lavallee, S., & Cinquin, P. (1995). Using a 3D position sensor for registration of SPECT and US images of the kidney. *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95),* 23-29.

Schroeder, W., Zarge, J. & Lorensen, W. (1992). Decimation of triangle meshes. *Computer Graphics*, 26(2), 65-70.

SunSoft (1991). The Tooltalk Service. Technical report, SunSoft, June.

Tuceryan, M., Greer, D., Whitaker, R., Breen, D., Crampton, C., Rose, E., & Ahlers, K. (1995). Calibration requirements and procedures for a monitor-based augmented reality system. *IEEE Transactions on Visualization and Computer Graphics*, 1, 255-273.

Turk, G. (1992). Retiling polygonal surfaces. *Computer Graphics*, 26(2), 55-64.

Uenohara, M. & Kanade, T. (1995). Vision-based object registration for real-time image overlay. *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95),* 13-22.

Wellner, P. (1993). Interacting with paper on the digital desk. *Communications of the ACM*, 36(7), 87-96.

Weng, J., Huang, T.S., & Ahuja, N. (1989). Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5), 451-476.

Whitaker, R., Crampton, C., Breen, D., Tuceryan, M., & Rose, E. (1995). Object calibration for augmented reality. *Proc. Eurographics'95*.

Wloka, M. & Anderson, B. (1995). Resolving occlusion in augmented reality. *Proc. of the ACM Symposium on Interactive 3D Graphics*, 5-12.

Table of Contents

List of Figures