

Annotating Real-World Objects Using Augmented Reality

**Eric Rose
David Breen
Klaus H. Ahlers
Chris Crampton
Mihran Tuceryan
Ross Whitaker
Douglas Greer**

ECRC-94-41

Technical report ECRC-94-41
(also published in CG International '95 Proceedings, Leeds, UK,
June 1995, pp. 357-370)

Annotating Real-World Objects Using Augmented Reality

Eric Rose
David Breen
Klaus H. Ahlers
Chris Crampton
Mihran Tuceryan
Ross Whitaker
Douglas Greer



**European Computer-Industry
Research Centre GmbH
(Forschungszentrum)**

Arabellastrasse 17
D-81925 Munich
Germany
Tel. +49 89 9 26 99-0
Fax. +49 89 9 26 99-170
Tlx. 52 69 10

Although every effort has been taken to ensure the accuracy of this report, neither the authors nor the European Computer-Industry Research Centre GmbH make any warranty, express or implied, or assume any legal liability for either the contents or use to which the contents may be put, including any derived works. Permission to copy this report in whole or in part is freely given for non-profit educational and research purposes on condition that such copies include the following:

1. a statement that the contents are the intellectual property of the European Computer-Industry Research Centre GmbH
2. this notice
3. an acknowledgement of the authors and individual contributors to this work

Copying, reproducing or republishing this report by any means, whether electronic or mechanical, for any other purposes requires the express written permission of the European Computer-Industry Research Centre GmbH. Any registered trademarks used in this work are the property of their respective owners.

**For more
information
please**

contact : David E. Breen
david@ecrc.de

Abstract

We describe a system for annotating real-world objects using augmented reality. A virtual form (model) of a real-world object is matched to the real object, allowing one to visually annotate the real components with information from the corresponding model. Augmented reality provides a natural method for presenting the “enhancing” computer-based information by merging graphics with a view of the real object. User queries on the real object can be translated into queries on the model, producing feedback that can augment the user’s view of the real world.

1 Introduction

Today, an enormous amount of information is available via computers and the networks that connect them. One of the challenges associated with this vast store of data involves determining how to effectively access and present it without overwhelming the user with the complexity of the querying procedure or the data itself. For information pertaining to complicated 3D objects, augmented reality (AR) is an effective means for utilizing and exploiting the potential of computer-based information and databases. In augmented reality the computer provides additional information that enhances or augments the real world, rather than replacing it with a completely virtual environment. One of the objectives of AR is to bring the computer out of the desk-top environment and into the world of the user working in a 3D application. In contrast to virtual reality, where the user is immersed in the world of the computer, AR incorporates the computer into the reality of the user.

In AR, the computer contains models of the significant aspects of the user's environment. AR provides a natural interface for processing data requests about the environment and presenting the results of the requests. With 3D tracking devices a user may conduct normal tasks in his/her environment and the computer is capable of acquiring and interpreting his/her movements, actions and gestures. The user can then interact with objects in the real world, generate queries, and the computer provides information and assistance. This new input paradigm is better suited for many 3D applications, in comparison to traditional ones that involve working with a 2D display and 2D interaction methods. Merging graphical representations of augmenting information with the view of the real object clearly presents the relationship between the data and the object. Using AR, the user can easily perceive and comprehend the spatial component of the queried data.

In the application described here, we explore this new query and display paradigm in a mechanical repair scenario, where AR is used to annotate parts of an automobile engine that are identified by a user. The user interacts with a plastic model engine in its natural setting with a pointing device monitored by the computer. As the user points to a specific part of the engine, an AR system displays computer-generated lines and text (annotations) that describe the visible components, as seen in Figure 1.1. Currently the user may identify 20 different components on the model. Queries with the pointing device on the real-world object may be used to add and delete annotation tags. We track the engine, so the annotations may move with the engine as its orientation changes. The lines attaching the annotation tags to the engine follow the appropriate visible components, allowing the user to easily identify the different parts as the view of the engine changes. As visible components become occluded, their annotations are removed. As a component comes into view, its annotation is displayed.

Our technical approach matches the computer's virtual world to the real world,

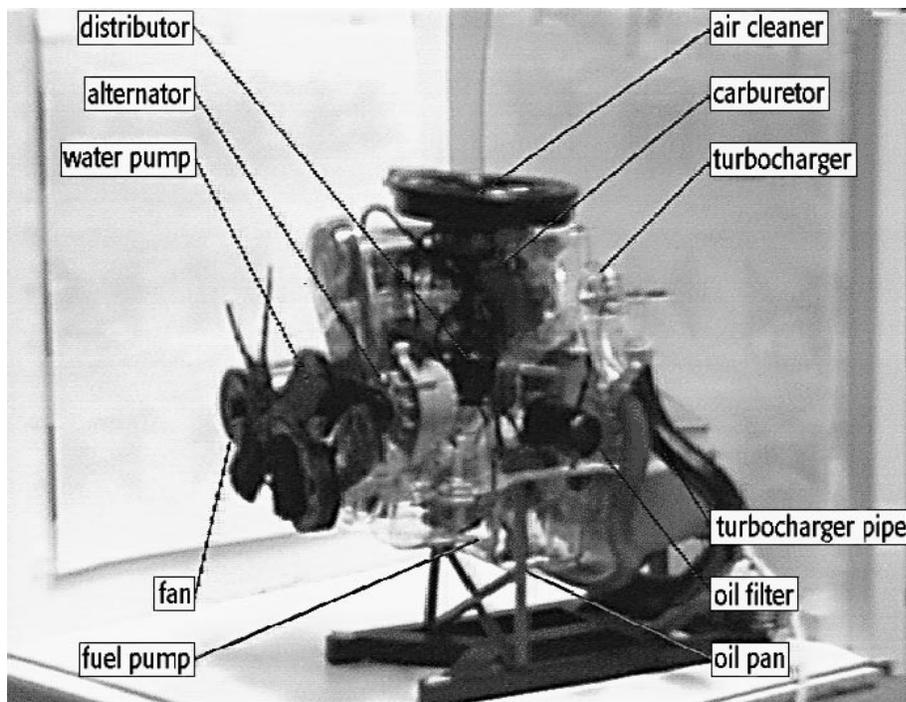


Figure 1.1: Augmented reality view of an engine.

by first registering a virtual camera to a physical camera directed at the object of interest [2, 20]. Currently we use a video monitor to represent the user's view, but eventually the same techniques could be used to display graphics using a see-through head-mounted display showing graphics superimposed on real-world objects. We then take a computer representation of the object and match it spatially to the real one [21]. Once this registration is established, we introduce tracking technology to maintain correspondence as the object or camera moves. To receive input from the user, we provide a tracked pointing device. Because the virtual world corresponds to the real world, the graphics drawn by the computer will appear to the user to be in the real world. Using this approach we can display annotations on the computer that appear in a proper context in the user's view. Queries about the real world are also produced and answered in the user's context. This integrates the computer into the user's working environment.

The remainder of the paper describes the major components of the application. This includes the fundamental features of our augmented reality system, as well as those capabilities developed specifically for the application. Section 3 provides an overview of the software and hardware architecture of our augmented reality system. Section 4 describes the general calibration procedures required for this and most other augmented reality applications. Sections 5 and 7 briefly discuss the issues of tracking and user interaction. Section 6 describes the details of generating and displaying the annotations.

2 Related Work

Several research groups are currently exploring augmented reality for a variety of applications. Feiner et al. [10] have developed a knowledge-based AR system for maintenance and repair instruction. Lorensen et al. [13] have focused more on AR for medical applications. At Boeing [17], AR is being developed to assist in manufacturing processes. A group at the University of North Carolina has also explored medical applications [7], and conducted research in tracking technologies [6] and collision detection [4] for AR. Deering [9] has studied the problems associated with achieving high resolution head-tracked stereo display for AR. Milgram et al. [14] have explored AR for enhancing telerobotic interactions. Grimson et al. [12] present a method for automatically registering clinical data from MRI or CT scans with a patient's head on an operating table. Nakamae et al. [15] propose a technique for accurately overlaying computer-generated images on digital images of real outdoor scenes. Fournier [11] has posed the problems associated with common illumination when combining synthetic images with images of real scenes. While many of the papers here describe applications of AR, our work to date at ECRC has strongly focused on developing general and robust calibration methods for AR applications [2, 18, 21].

3 System Configuration

The User Interaction and Visualization Group at ECRC has developed a general-purpose augmented reality system [1, 2]. The hardware and software components of the system are briefly described below.

3.1 Hardware Components

In the initial stage of our work, we have chosen to side-step the issues involved with utilizing a head-mounted display. We have instead focused on the issues of tracking, calibration, and user interaction, during the development of our system. Our augmented reality system uses a standard video camera and combines the video signal with computer-generated graphics; the result of which is presented on a normal video display. This type of system, where the output is displayed on a standard television monitor rather than special "glasses" worn by the user, is preferable for some applications. For example, a surgeon may not want to wear see-through glasses during surgery, but may still want the assistance of augmented reality. Viewing augmented reality on a video monitor may also be acceptable for certain distributed augmented reality applications [3].

The hardware configuration is illustrated in Figure 3.1. The graphical image is

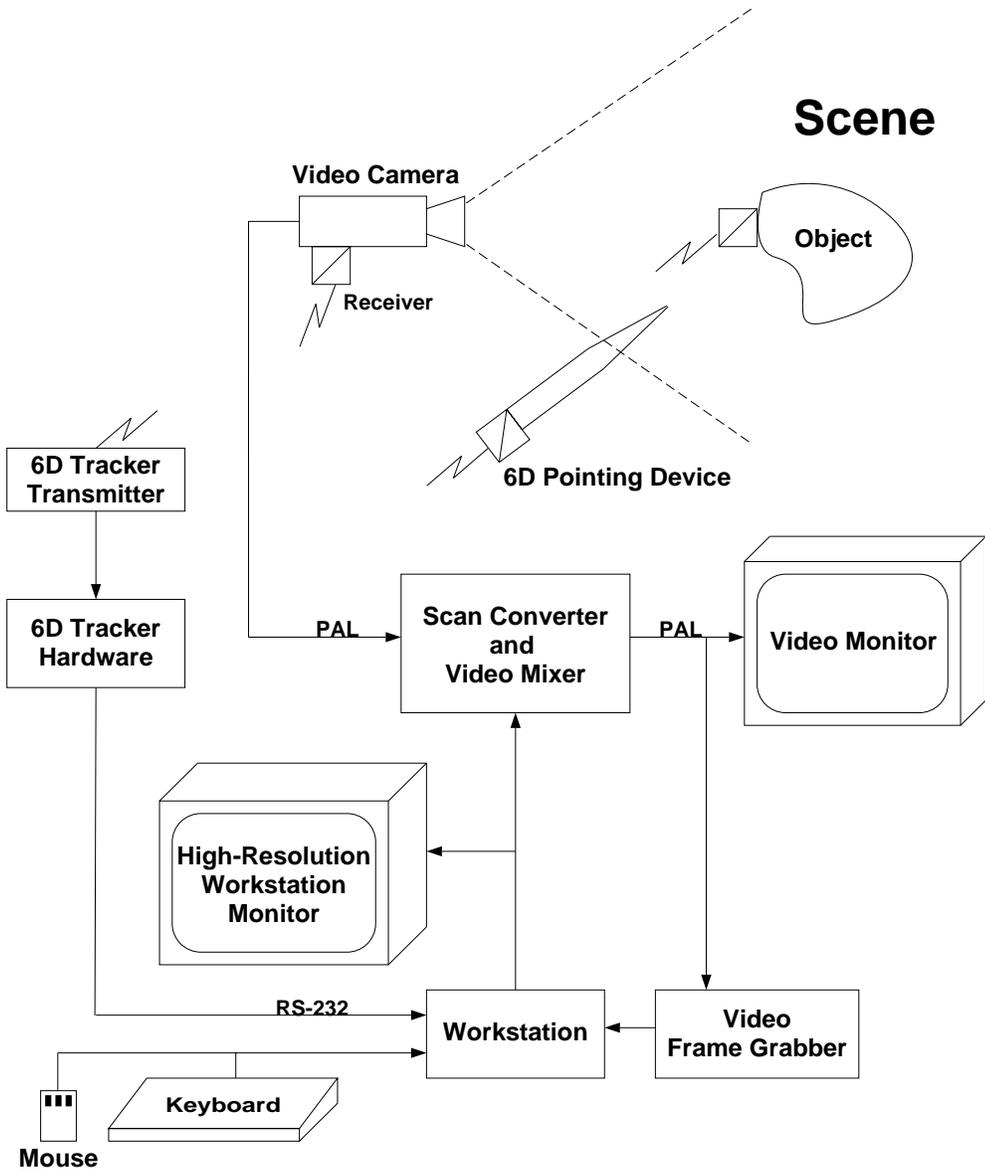


Figure 3.1: System hardware configuration.

generated by the workstation hardware and displayed on the workstation's high resolution monitor along with auxiliary control information. A scan converter takes the relevant portion of the graphical image and converts it to standard video resolution and format. The scan converter also mixes this generated video signal with the video signal input from the camera. A six degrees-of-freedom (6D) magnetic tracker, which is capable of sensing three translational and three rotational degrees of freedom, provides the workstation with continually updated values for the position and orientation of the video camera, a tracked object and the pointing device. A frame grabber is used during the initial calibration procedure as part of the process to determine the optical characteristics of the camera, and the initial transformations of the various objects in the scene.

3.2 Geometric and Scene Modeling

The core of our augmented reality system is an interactive 3D computer graphics system, providing methods for representing and viewing 3D geometric models. The definition of the three dimensional objects and the camera model used to render them are based on the RenderMan system [19]. Numerous geometric primitives may be hierarchically organized to produce complex 3D models. The camera model not only includes its position, but also such camera parameters as the aspect ratio, focal length, and clipping information. Environmental lighting information and shading attributes for the geometric primitives are also supported for image generation.

3.3 Rendering System

Graphics are generated using custom object-oriented software that interfaces to vendor-specific graphics libraries (in this case Sun Microsystems XGL). RenderMan Interface Bytestream (RIB) [16] and AutoCAD DXF [5] file formats are supported, with extensions to allow extra information used for calibration. The rendering system is designed and optimized for repeated interactive display. A Sun Sparcstation 10 with a 24-bit ZX graphics accelerator gives sufficient performance for this application.

3.4 Mixing Video and Graphics

The mixing of the computer graphics and video input is done by a Folsom Research Otto 9500 Scan-Converter. The camera input (PAL) is combined with the graphics and displayed on a monitor. The final merged output is produced with luminance keying, which displays the live video signal in those pixels where the computer-generated graphics is black. We can then control which computer-generated objects/features will be overlaid on the video signal by

rendering them with a non-black color. Small features such as thin lines or text require thickening to show up clearly at PAL resolution. If information from a computer-based model is needed, e.g., to determine occlusion, but it does not need to be displayed, the model may be rendered and drawn in black. This will write the model's depth values into the Z-buffer, but will not draw it on the screen, and therefore the model will not appear in the final video signal.

3.5 Input

We use a Sun Microsystems VideoPix frame-grabber to acquire images from the camera. We use an Ascension Flock-of-Birds (FOB) magnetic tracker for 6D input. Either an extended or normal range transmitter can be used depending on the application requirements. Three receiver units are used: one attached to the tracked object, one attached to the camera, and one attached to a pointing device. The workstation keyboard and mouse are also used as part of the user-interface during scenario setup and calibration procedures.

4 Calibration

Calibration is an essential component of augmented reality. Calibration provides the information that allows for the registration and overlaying of geometric and viewing models onto objects and cameras in the real world [2, 18]. The calibration process involves acquiring information about the real world and determining transformation matrices that map geometric objects from their local coordinate system into a world coordinate system. It may also include calculating intrinsic parameters associated with the various electronic equipment used for AR. For example, the result of calibrating a real camera are the pose and intrinsic parameters for a simulated camera. We currently obtain the needed information with both a camera and a 6D pointing device. The acquisition process produces data with noise and uncertainty, whose effect must be minimized during the calculations. Once calibration is completed, graphics may be generated from the computer models and merged realistically with video of the real world.

Calibration itself does not require the computer; the user could use physical measurements to estimate the parameters that match the virtual world to the real world. Instead we use the computational power of the computer for calibration, providing faster methods and more robust results. Currently, our calibration procedures call for some user-interaction, requiring the user to pick features in images or locate known 3D landmarks with the pointing device. We generally utilize a least-squares method to solve a system of equations that captures the appropriate transformations along with required constraints. A least squares method is employed in order to account for the noise in the input data. When calculating the elements of a transformation matrix, the

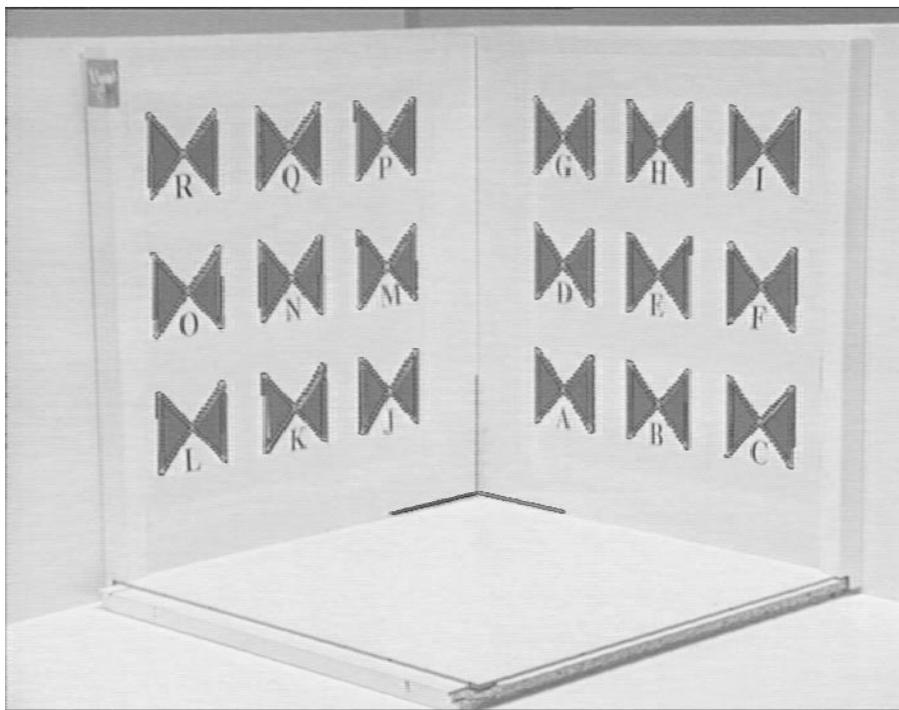


Figure 4.1: An example of camera calibration showing the calibration pattern and the superimposed computer model of the pattern.

optimization process must be constrained in order to ensure that the matrix's affine properties are maintained and a pure rotation is produced.

4.1 Camera Calibration

Calibrating a camera estimates the camera pose and intrinsic parameters (focal length, aspect ratio and optical center) that correspond to a real-world camera. This involves pointing the camera at a calibration pattern whose location in world coordinates is known (usually this defines the world coordinate system). By grabbing an image from the camera and letting the user pick the known points on the image with the mouse, the location of the camera and its intrinsic parameters can be computed [2, 20]. These parameters are then used to set up a virtual camera that corresponds to the real camera (see Figure 4.1). We also perform an extra calibration step to account for offsets and distortions introduced by the scan-converter, which sits in the computer-camera loop.

4.2 Pointer and Tracker Calibration

For user input to the system we use a pointing device that is a wooden dowel attached to an Ascension Flock-of-Birds (FOB) receiver (sometimes called a "mark"). Calibrating this pointing device produces the length of the dowel and the offset of the tip from the mark. By touching the tip of the pointer to the

same point three or more times in different orientations, the length of the pointer and the offset of the mark from the tip can be calculated. Picking three known points in the world coordinate system then provides enough information to compute where the FOB transmitter is located in world coordinates.

4.3 Object Calibration

Object calibration is used to estimate the pose (object-to-world transformation) of an arbitrary object [21]. We currently have two methods for computing an orientation to match a virtual object to a real one (see Figure 6.1). In the first method an image of the object is grabbed with the calibrated camera, then a few points are selected with the mouse on the image whose locations are known in the object's local coordinate system. Computing the object's orientation is then similar to calibrating the camera. With another method, a calibrated pointer can be used to pick known points on the real object, which are then correlated to their world coordinate locations. We have found that the image-based method is sufficient for calibrating objects that do not move. Once a calibrated object is moved, the inherent inaccuracies in the depth information becomes obvious. For moving calibrated objects, the pointer-based method is superior, providing a more accurate transformation. As with the other calibration procedures a least-squares optimization of a constrained system of equations is used here to solve for the transformation that maps the point locations from one coordinate system into another.

4.4 Camera and Object Mark Calibration

Before an object can be tracked, the orientation of the attached tracking receiver (mark) relative to the object's coordinate system is required. This orientation does not change as the object moves, so it can be computed once and used by the tracker to update the object's transformation as it moves. If the positions of the object and the transmitter are known in world coordinates, one can solve for the unknown transformation:

$$T_{object-to-mark} = T_{object-to-world} * T_{world-to-mark}. \quad (1)$$

Calibration of the camera's mark is performed in the same manner.

5 Tracking

Tracking uses input from the real world to maintain correspondence between a virtual object and a real object, in essence, to maintain calibration. Tracking is

the dynamic component of calibration; with tracking one can fix a virtual object to a real one while the real object moves.

When tracking is enabled the system uses the locations of the FOB receivers attached to the pointer, object and camera to update their model transformations. The combination of the orientation of the attached mark (static) with the mark's current orientation in world coordinates (dynamic) yields the object's current local transformation. A real-world object or camera may be moved and their representations within the AR system are updated and remain current and accurate. The following equations are used to calculate the appropriate transformations:

$$T_{\text{mark-to-world}} = T_{\text{mark-to-transmitter}} * T_{\text{transmitter-to-world}}, \quad (2)$$

$$T_{\text{object-to-world}} = T_{\text{object-to-mark}} * T_{\text{mark-to-world}}. \quad (3)$$

6 Model Annotations

The general approach to annotating a real-world object involves first generating a geometric model of the object. For our experiments we interactively created a coarse geometric model of our plastic automobile engine model using AutoCAD. The annotation information is then generated and identifiers that link the annotations with specific object components are specified. These unique identifiers can also be used to reference an audio file or a database entry during the application. Once camera and object calibration have been completed, the geometric model and the real object can be superimposed on the display monitor, as shown in Figure 6.1. In this application, we only wish to see the annotations, and not the model itself. We update the geometric model from the tracking input in order to maintain the calibration. The model is rendered into the Z-buffer, but is drawn in black. This approach provides the system with visibility information, and produces an output that only displays the annotations.

Each component of the annotated object is assigned a short alphanumeric identifier and an attachment (annotation) point in the engine's local coordinate system. A separate annotation file contains the text corresponding to these identifiers, and the point defines where the annotation will be attached to the component. The model can be organized hierarchically to support varying levels of detail. A new set of annotations can be loaded at any time, to change the level-of-detail or target language.

The visibility of the components is determined dynamically during rendering. As the object model is rendered, the world coordinate location of each annotation point is cached. After rendering, every annotation point is mapped into image space and is checked for visibility using a query to the hardware Z-buffer. The annotations with visible attachment points are flagged for display.

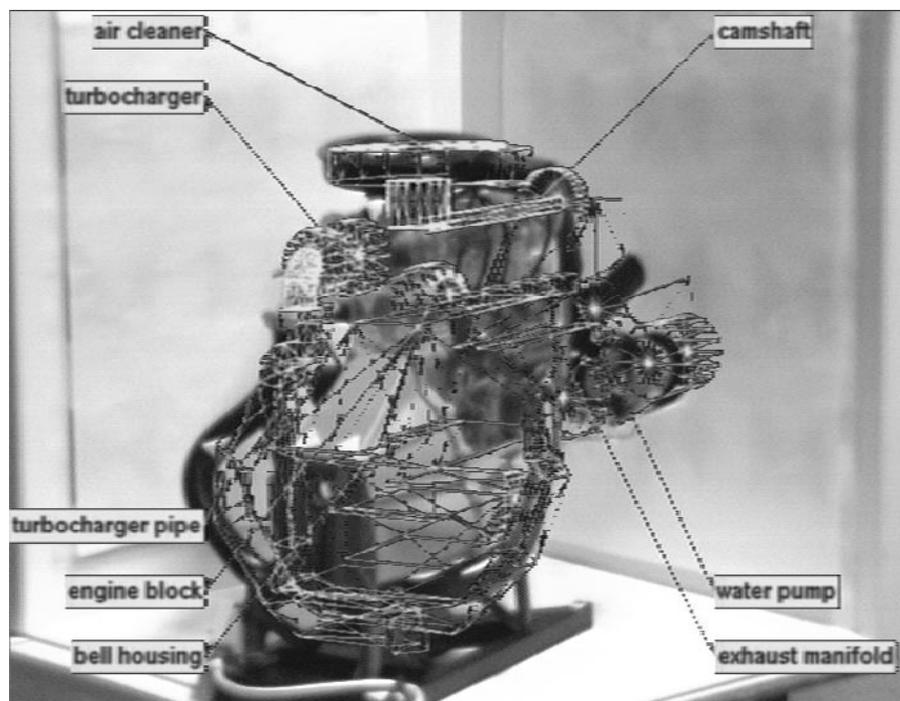


Figure 6.1: View of an object with the computer model superimposed, showing the correspondence between the computer model and real object.

It is difficult to determine if any part of an arbitrary component of the model is visible. We chose one point on the component (annotation point) that we use for visibility queries. Using one point to decide if the entire component is visible is not an unreasonable assumption given that the model and component are not extremely complex, i.e., do not contain many intricate, interlocking parts. In fact it is meaningful to display an annotation only when a significant portion of the component is visible, to avoid confusion with multiple annotations when components are near each other. One problem with using single-point queries occurs due to Z-buffer "aliasing". When the annotation point lies near the model's surface, the component itself may obscure the point, giving the indication that the point is not visible. We considered introducing a tolerance window into the Z-buffer query to solve the problem, but decided instead to lift each annotation point off the surface of the component by a small amount. This gave us better control over each individual point, and avoided problems with assigning spatial meaning to hardware Z-buffer values. Usually 1-2mm was enough distance to eliminate the false negatives without introducing new false positives. 1-2mm is within the error bounds of the camera and object tracking, so the accuracy of the annotations was not affected.

Once the set of visible annotations has been determined, a layout subsystem arranges them to provide the most readable display. The annotations currently consist of text with a line drawn to the annotated component. The text itself is rendered in 2D and is drawn in colored windows using raster fonts, for readability on the video monitor. The layout algorithm divides the screen into quadrants based on the centroid of the projected annotation points. The

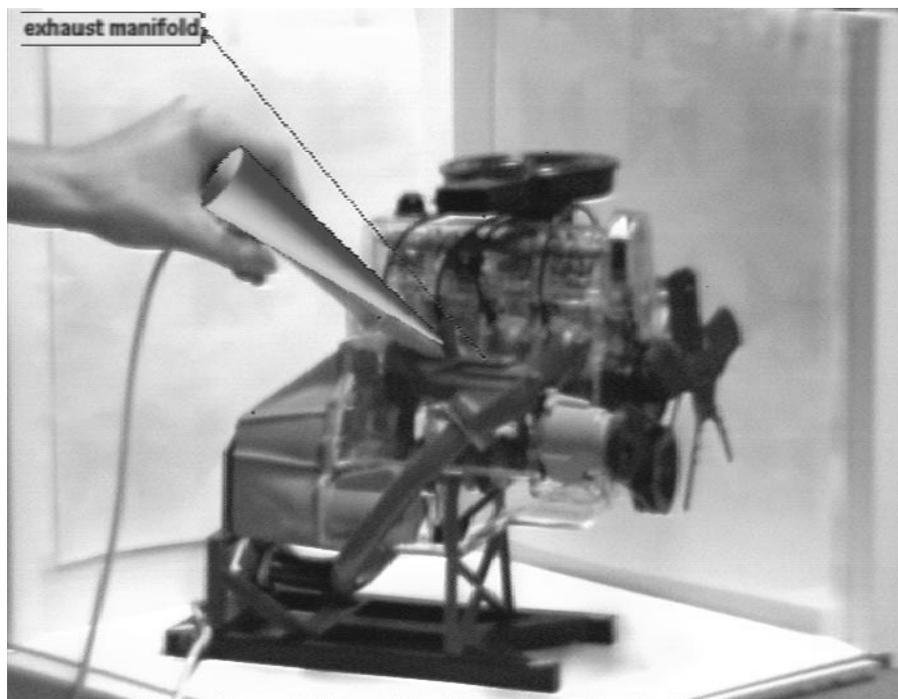


Figure 7.1: An example of object examination. The pointing device picks a component on the model to be examined. The cone is the computer's representation of the physical pointer.

annotation associated with every point in a particular quadrant is displayed in the far corner of the quadrant. The annotations in each quadrant are sorted in order to prevent the crossing of the annotation attachment lines. A radial sort of the annotation points is successively performed around the locations of the annotation text to achieve an uncluttered and comprehensible layout.

7 Object Examination

Examination of the real object can be achieved with queries on the virtual object (model). The user may point to the real object using the pointer and query its components. The 3D query returns the identifier of the component being pointed to by the pointing device. The identifier may now be used to retrieve and display specific information about the component. For example, a component's name or function may be determined. To support these types of queries we imported the concept of "picking" from the 2D world (using a mouse on the workstation's display screen) into the 3D world, using the pointer in space. A query with the 3D pointer is translated into a 2D pick after rendering the geometric model from the point-of-view of the pointer along its line-of-sight. By temporarily freezing the double-buffering update we can perform the render and query without interrupting any of the displayed annotations. The reaction to a successful pick can be linked to an appropriate response: the annotation can be displayed (as in Figure 7.1), the component

highlighted or drawn in wireframe (as graphics to augment the real component), or even recorded spoken information can be played. Linking the query to a database could give the user access to the component's maintenance history or schematics. The concept of 3D picking now allows the user to manipulate a real object, and simultaneously access computer-based information about the object. This aspect of augmented reality supports a natural paradigm for exploring a variety of data associated with 3D geometric objects, where a user performs tasks in the real world that translate into database queries.

8 Application Scenario and Results

In the target scenario for this particular application, a mechanic attempts to repair a complex mechanical engine, possibly for the first time. The mechanic utilizes augmented reality, ultimately with a see-through head-mounted display, to gain information about the engine and the tasks needed to repair it. Without having to look away from the engine itself, he/she would like to have the computer-based assistance provided. The AR system can supply a wide variety of support for the mechanic. Annotations may identify the name of the part, describe its function, or present other important information like maintenance or manufacturing records. In our first demonstration, twenty different annotations in seven languages were defined for identifying individual engine components. AR may lead the mechanic through a specific task by highlighting parts that must be sequentially removed and by showing the path of extraction. The system may also provide safety information. Parts that are hot or electrified can be highlighted to constantly remind the mechanic of the danger of touching them. The mechanic may also be assisted by a remote expert who can control the information being displayed on the mechanic's AR system. Pointing to a part with the pointing device will trigger feedback that produces the relevant information (examination). Of course, as the engine itself or the user's viewpoint changes, the computer-generated data will change appropriately. For example, only those parts that are visible to the user should be annotated, as seen in Figures 8.1 and 8.2.

9 Future Research

A number of issues still require attention before the mechanical repair scenario can be fully implemented. Improving the camera calibration would result in better correspondence between the real and virtual worlds. When the camera is not correctly calibrated, the overlaid graphics will deviate from the real object. This is especially noticeable as the real object approaches the camera. Calibration methods that require less user-interaction are preferable and would automate the augmented reality setup process. This would relieve the user of

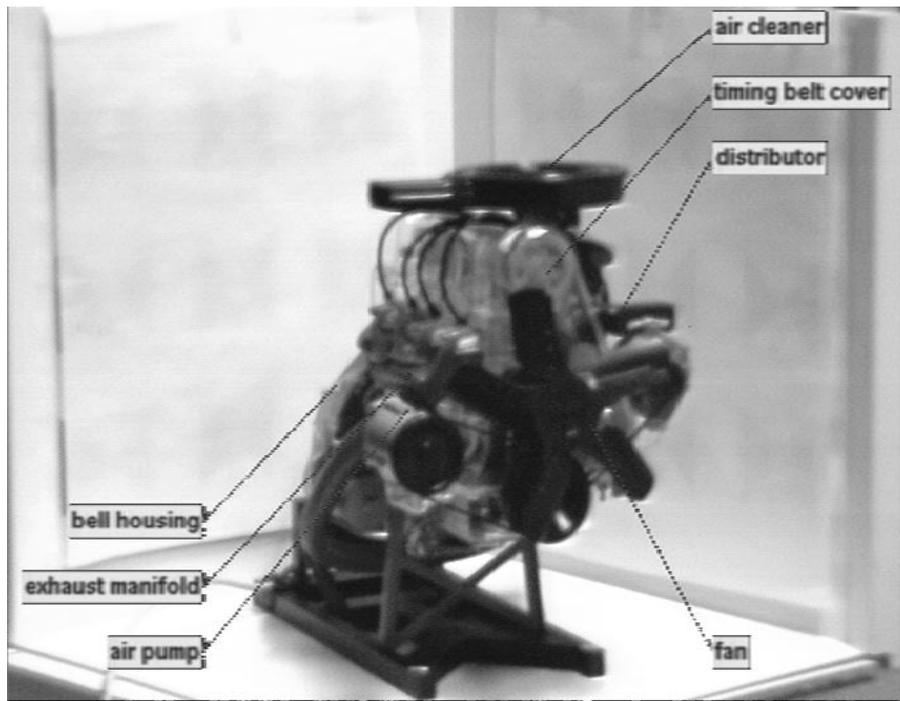


Figure 8.1: Another augmented reality view of the engine.

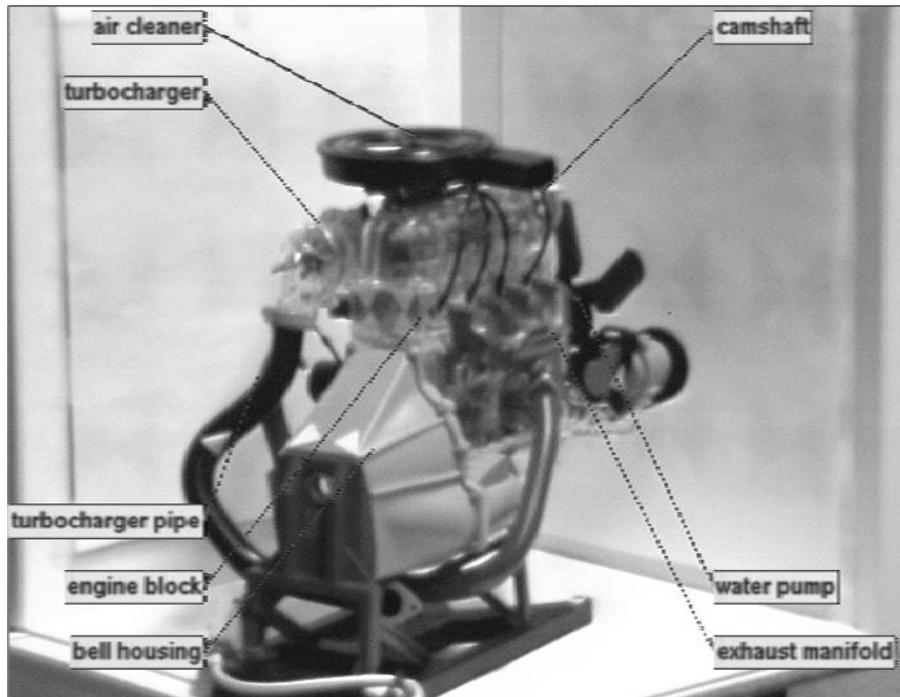


Figure 8.2: As the engine is rotated the annotations are automatically updated.

the time-consuming burden of identifying known calibration points. We have begun to explore computer vision techniques to address this problem. Magnetic tracking has problems with accuracy and lag, yet accuracy is critical in augmented reality applications. Any inaccuracies are immediately apparent to the user, even when small spatial errors occur. The accuracy can be improved using vision-based object recognition and tracking for correction of the magnetic tracker information.

We would like to improve the visibility detection by introducing multiple annotation points for components. This would make component visibility detection more robust. With some model components there is a problem when the annotation query point is located on one surface of the part. In this case the visibility heuristic is not satisfied when the model is turned or the viewpoint is moved to the other side. Having multiple query points for each component would also improve annotation readability. If multiple query points were found to be visible, the one that most clearly indicates the component can be selected.

10 Conclusion

We have described a system for annotating real-world objects using augmented reality. A virtual form (model) of a real-world object is matched to the real object, allowing one to visually annotate the real components with information from the corresponding model. Augmented reality provides a natural method for presenting the “enhancing” computer-based information by merging graphics with a view of the real object. User queries on the real object can be translated into queries on the model, producing feedback that can augment the user’s view of real model. The techniques of camera calibration, object calibration, and tracking are applicable to many AR applications. When the computer can use spatial information about the real world, a more realistic and enhanced view is possible, with constrained interactions and object occlusions [8].

The trend in design and manufacturing is towards utilizing more computer-aided design (CAD). As the amount of computerized information about real-world objects increases, methods of using this information in virtual and augmented reality become more feasible. We have shown one method in which this information can be used directly to annotate real-world objects, giving the user a natural setting for interaction. The ability to see visible components annotated and to examine objects has applications in education, training, medicine, embedded design, and any situation requiring skilled user interaction in a real-world setting. The details or explicit procedures (usually relegated to printed manuals or written notes) can be left to the computer, letting the user or expert work directly on the task at hand.

11 Acknowledgments

The AutoCAD model of the automobile engine was initially produced by Anne Bachy. We would like to thank Pierre-Yves Chevalier and Gudrun Klinker for their helpful comments on the manuscript. The plastic automobile engine model was generously provided by Revell, Inc. This work is financially supported by Bull SA, ICL Plc, and Siemens AG.

Bibliography

- [1] K. Ahlers, D. Breen, C. Crampton, E. Rose, M. Tuceryan, R. Whitaker, and D. Greer. An augmented vision system for industrial applications. In *SPIE Photonics for Industrial Applications Conference Proceedings*, October 1994.
- [2] K. Ahlers, C. Crampton, D. Greer, E. Rose, and M. Tuceryan. Augmented vision: A technical introduction to the grasp 1.2 system. Technical Report ECRC-94-14, ECRC, Munich, Germany, 1994.
- [3] K. Ahlers, A. Kramer, D. Breen, P.-Y. Chevalier, C. Crampton, E. Rose, M. Tuceryan, R. Whitaker, and D. Greer. Distributed augmented reality for collaborative design applications. Technical Report ECRC-95-03, ECRC, Munich, Germany, 1995.
- [4] D. Aliaga. Virtual and real object collisions in a merged environment. In G. Singh, S. Feiner, and D. Thalmann, editors, *Virtual Reality Software & Technology (Proc. VRST '94)*, pages 287–298, Singapore, 1994. World Scientific Publishing Co.
- [5] Autodesk, Inc. *AutoCAD Customization Manual*, 1992.
- [6] R. Azuma and G. Bishop. Improving static and dynamic registration in an optical see-through display. In *Computer Graphics (Proc. SIGGRAPH)*, pages 194–204, July 1994.
- [7] M. Bajura, H. Fuchs, and R. Ohbuchi. Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. *Computer Graphics (Proc. SIGGRAPH)*, 26(2):203–210, July 1992.
- [8] D. Breen, E. Rose, and R. Whitaker. Interactive occlusion and collision of real and virtual objects in augmented reality. Technical Report ECRC-95-02, ECRC, Munich, Germany, 1995.
- [9] M. Deering. High resolution virtual reality. *Computer Graphics (Proc. SIGGRAPH)*, 26(2):195–202, July 1992.
- [10] S. Feiner, B. Macintyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53–62, July 1993.
- [11] A. Fournier. Illumination problems in computer augmented reality. In *Journée INRIA, Analyse/Synthèse D'Images*, pages 1–21, January 1994.
- [12] W. Grimson, T. Lozano-Perez, W. Wells, G. Ettinger, S. White, and R. Kikinis. An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality. In *IEEE Conference on*

- [13] W. Lorensen, H. Cline, C. Nafis, R. Kikinis, D. Altobelli, and L. Gleason. Enhancing reality in the operating room. In *Visualization '93 Conference Proceedings*, pages 410–415, Los Alamitos, CA, October 1993. IEEE Computer Society Press.
- [14] P. Milgram, S. Shumin, D. Drascic, and J. Grodski. Applications of augmented reality for human-robot communication. In *International Conference on Intelligent Robots and Systems Proceedings*, pages 1467–1472, Yokohama, Japan, July 1993.
- [15] E. Nakamae, K. Harada, and T. Ishizaki. A montage method: The overlaying of the computer generated images onto a background photograph. *Computer Graphics (Proc. SIGGRAPH)*, 20(4):207–214, August 1986.
- [16] Pixar. *The RenderMan Interface*, 1989.
- [17] D. Sims. New realities in aircraft design and manufacture. *IEEE Computer Graphics and Applications*, 14(2):91, March 1994.
- [18] M. Tuceryan, D. Greer, R. Whitaker, D. Breen, C. Crampton, E. Rose, and K. Ahlers. Calibration requirements and procedures for augmented reality. Technical report, in preparation.
- [19] S. Upstill. *The RenderMan Companion*. Addison-Wesley, Reading, MA, 1990.
- [20] 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.
- [21] R. Whitaker, C. Crampton, D. Breen, M. Tuceryan, and E. Rose. Object calibration for augmented reality. Technical Report ECRC-95-04, ECRC, Munich, Germany, 1995.