

# Augmented Reality for Construction Tasks: Doorlock Assembly\*

Dirk Reiners, Didier Stricker, Gudrun Klinker, Stefan Müller

Fraunhofer IGD

Department Visualization and Virtual Reality

Rundeturmstraße 6

64283 Darmstadt Germany

{reiners,stricker,klinker,stefanm}@igd.fhg.de <http://www.igd.fhg.de/www/igd-a4/>

## Abstract

Augmented Reality is a technology that integrates pictures of virtual objects into images of the real world. Besides the technical problems still to be solved, for industry to be interested in this technology the additional amount of work needed to use the technology in relation to the benefit has to be visible. Furthermore the question how this integrates into the information technology infrastructure of the company is important. This paper describes an Augmented Reality demonstrator for the task of doorlock assembly into a car door that was developed trying to create a practical, realistic application that can transport the concepts behind Augmented Reality to a casual observer. To reach that goal a new fast and robust optical tracking algorithm was developed and integrated into a three-dimensional animation and rendering system, creating a real-time fully three-dimensional HMD-based training application showing how to assemble the doorlock into the door. The system was demonstrated to the general public at the Hannover Industrial Fair 1998 and this demonstration of Augmented Reality for one of the first times to a large non-expert audience created a lot of interest into this new area.

## 1 Introduction

Augmented Reality is a technology that integrates pictures of virtual objects into images of the real

world. These images can either be taken from a camera or, with the use of a see-through head-mounted display, the user's direct view on the world can be augmented. There are still some technological problems to solve, but for industrial partners to be interested in investing into this technology its possible benefit and its integration into the whole company has to be visible.

Many companies are trying to move bigger parts of their design and construction processes into the computer, up to the point where physical prototypes are replaced by virtual prototypes for packaging, assembly and security evaluations. This is especially evident in the airplane and car construction companies, where prototypes are costly and time to market is a significant factor. A side-effect is that a large number of three-dimensional models become available at little or no additional cost.

AR can help using this growing amount of digital product information to help the people who have to handle the fast changing products. A consequence of reduced time-to-markets, shorter product cycles and smaller production counts is the increasing demand for training. Assembling a complex machine like a car or a plane is a difficult process that takes time to learn. If that training is only useful for a small number of production runs, in the worst case only once, the efficiency of the worker decreases. The situation is even worse for service personnel, as the faster product cycles don't necessarily go along with shorter life cycles. Thus a service person is confronted with an ever faster growing number of different generation systems and has to be able to handle all of them. Given the far, possibly worldwide, distribution of service personnel for example

---

\*presented at: the 1rst International Workshop on Augmented Reality (IWAR'98), San Francisco, Nov. 1998. AK Peters, pp. 31-46.

for cars makes a centralized training difficult, expensive or impossible.

The availability of three-dimensional models allows to do that in novel ways that are more direct and more intuitive than written instructions or even electronic hypertext manuals. Three-dimensional animated instructions can be integrated into the surrounding environment at the exact place where the action has to be performed, so that no mental transfer is needed and the action can be observed from different viewpoints to understand the spatial relationships involved.

This paper describes an implementation of an augmented reality system that builds this connection between virtual prototyping and real assembly tasks, in this case the task of assembling a doorlock into a car door. After reviewing previous work in using augmented reality for assembly and maintenance work in sec. 2 we introduce the application task in sec. 3. Following that is a description of the used hardware (sec. 4.1) and software (sec. 4.4 and 4.3) systems together with motivations and alternatives. The system has been shown to the general public at the Hannover Industrial Fair 1998 (sec. 5) and as a conclusion experiences and open areas for further work are analyzed in sec. 6.

## 2 Previous Work

Assembly and maintenance tasks have been a target area of AR demonstrations for quite a long time. One of the first demonstrations was using Augmented Reality for a photocopier maintenance task [3]. This was done using wireframe graphics and a monochrome monoscopic HMD. Head- and object tracking were achieved using ultrasonic trackers. The graphics used were rather simple and not taken from any CAD system as the main objective was extending an existing two dimensional automated instruction generation system to an augmented environment. The same scenario was revived recently by [5], this time tracked optically.

Another application in the construction area was [10], where augmented reality technology was used for spaceframe assembly support. Tracking was done using an active optical system using blinking LEDs coupled to a passive inertial system. The graphics used were again rather stylized, as the objective was not to convey a sense of spatial presence

but to bring rather simple instructions across.

Boeing has used an Augmented Reality system for wire bundle assembly support. To the best of our knowledge the tracking was done using an inertial/ultrasonic system. The augmentations used (2D lines showing the path of the cable to be added to the bundle) were simple and could be generated from a database without manual involvement or a CAD system.

## 3 Application

The application targeted in this demonstration is the assembly of the doorlock into a car door (s. fig. 12). This task was examined as an example for virtual prototyping and an assembly feasibility study by the virtual environments group of our department for BMW (s. fig. 1). Thus CAD data taken from the system used to construct the actual objects was available not only for the lock but also for the whole door.

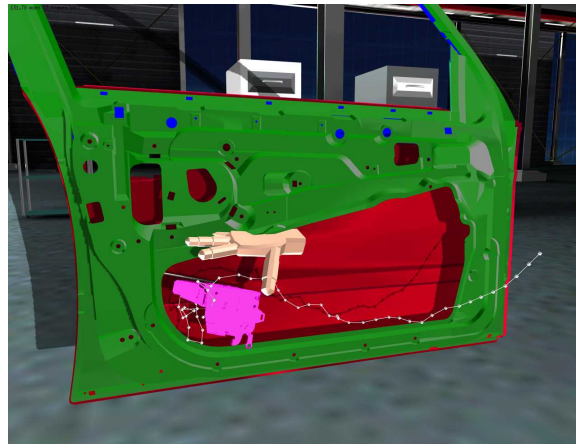


Figure 1: Virtual doorlock assembly

The task itself poses several challenges. It is very spatial and three-dimensional in nature, the movement of the hand holding the lock in the small space inside the door necessitates precise preparation and movement. A little lever that has to engage in the lock deep inside the door, which is not easily visible from the outside (s. fig. 10), has to be pushed to the right position beforehand, otherwise the final engaging of the lock will be impossible. Furthermore the space inside the door is just big enough

for the lock, so it has to be gripped in the right way (s. fig. 11) for the hand not to get stuck halfway through. This tight space gives strong haptic feedback helping with the task, but it demands a good three-dimensional understanding of the space.

Once the lock's antenna rests in its final position three screws have to be tightened to lock it into place. The tasks of inserting the antenna into the door compartment and of tightening the screws demand the use of both hands, so a hands-free interaction method was needed to control the instruction flow.

## 4 System

### 4.1 Hardware

The system was built using common off-the-shelf hardware, a standard SGI O2 with a 180 MHz R5k processor and 128MB memory. The machine has very good video capabilities and reasonably fast rendering, although not comparable to higher end SGI machines used for virtual reality applications like the one mentioned in sec. 3.



Figure 2: Head-mounted display with camera

The strong spatial nature of the task demanded a display that could convey that spatial information, an HMD being the logical choice. The HMD used is a Virtual I/O i-glasses set with an attached Toshiba IK-M48PK camera using a 7.5 mm optics as shown in fig. 2. The headset is a standard affordable piece of equipment allowing see-through and feed-through use, making it interesting for dif-

ferent applications. The camera is reasonably small and light enough to be worn on the head without undue strain for the user. The quality is very good for a single-CCD camera.

The two-handed action needed for the task required a hands-free interaction technique. Thus a voice-command driven interface was used. It runs on a separate machine, a standard Intel-based Laptop running Windows 95 and an IBM Voice-Type based speech recognition software. It is connected to the O2 via RS-232, which is adequate for the transmission of the short recognized commands.

### 4.2 Tracking

Tracking the user's actions has been a major focus of work for the Augmented Reality community as well as the Virtual Reality community. Different technical solutions for the problem of measuring 6 degrees of freedom motion have been developed.

The magnetic trackers which are very often found in VR applications were considered not applicable for the task at hand. This is due to their sensitivity to metallic objects in the surrounding area. The door is real and metallic, so there was no way of removing all metal from the work area, as is often done for VR installations. Furthermore, given that the final deployment location for the application would be a construction hall, where everything is metallic, an alternative solution had to be found.

An optical method was considered superior. There has been previous work on tracking using optical markers in Augmented Reality. Neumann et al are using color-coded dots [6] or rings [2] for a larger number of possible markers. Starner et al [7] use a color-coded bit pattern, but they don't do a full camera reconstruction as their application using 2D text and graphics augmentations is equally possible using only position and orientation coupled with size. Vallino et al [4] don't need to reconstruct a full camera parameter set either from their recognized feature points, as they are using a non-projective rendering scheme.

For the given task full three-dimensional position and orientation was considered mandatory, so our tracker had to be able to do a full camera calibration.

**Targets** It uses square targets (s. fig. 3) made from black cardboard. These targets are attached

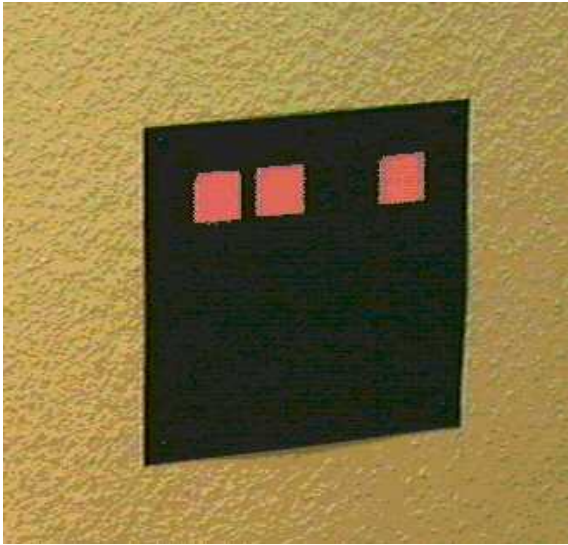


Figure 3: Square Target

to the door (s. fig. 4), as this allows to move the door freely around without needing a recalibration or remeasuring. The relative position of the targets is measured beforehand, e.g. using a tape measure. Their size can vary freely, the ones used on the door were 2.5 x 2.5 cm. They are identified by a code of small red squares attached to them at known positions. 2 rows of 4 ids are possible, allowing 256 different squares to be used. The symmetric nature of the ids makes some of the combinations not unique in case of rotations. For the door application and most others this is not a problem, as users rarely rotate their heads more than 45 degrees relative to the horizon. If it does pose a problem the offending ids can just be left out to make all used ids rotationally unique.

**Initialization** At initialization time the full picture is searched for black squares with valid markers. From the position of their corners in the image, as calculated by intersecting their edges, the camera parameters are calculated using either [11] or [9]. Every identified square adds 4 points to the calibration algorithm, so as a full calibration needs 11 points three squares are needed. The camera used (s. sec. 4.1) has a fixed focus and fixed internal parameters, allowing a variant of [9] to be chosen that only calculates the external parameters (position and orientation), thus giving better pre-

cision and stability and needing only two squares for initialization.

**Operation** After this initial step, which takes ca. 200 ms on our hardware, all further measurements are done incrementally. The 2D position of the squares' edges are extrapolated based on the position and speed in the previous frames, the position and orientation of the real edge is found using a local search and the new camera parameters are calculated using an error minimization technique (details in [8]). As only a small part of the image has to be examined the speed goes up to 25 Hz, which is quite adequate for head-mounted usage.

Just using lines for calibration has two additional benefits. The first is that the red ids don't have to be analyzed anymore, so it is possible to extend the operational range beyond the point where they are no longer discernible. The second is that a more general target form of target can be used: any high-contrast line. Given the 3D position and orientation, any line can be used to add information to the tracking process, making it more stable and robust. For this reason a number of white tape marks have been added to the door (see fig. 4), which are used in conjunction with the white squares surrounding the black squares and are significantly less disturbing visually compared to the rather prominent squares, while still adding useful information.

As the ids are not checked in the incremental mode, theoretically after a fast move a tracking edge could get locked onto another, wrong feature edge in the image. This is detected by measuring the error in the calibration. If it gets too big (because some lines were lost or locked onto wrong image features) a new initialization step is forced, thus getting back to known good data.

**See-through calibration** One of the major problems for see-through applications is the calibration between the head-tracker (in our case the head-mounted camera) and the user's eyes. The solution decided upon was to reuse the know-how and algorithms gained for the camera calibration problem by abusing the user as a feature finder. For calibration purposes the user is asked to manually (using the mouse) click on the corners of the calibration targets as he sees them through his HMD. This effectively gives enough information to calculate a

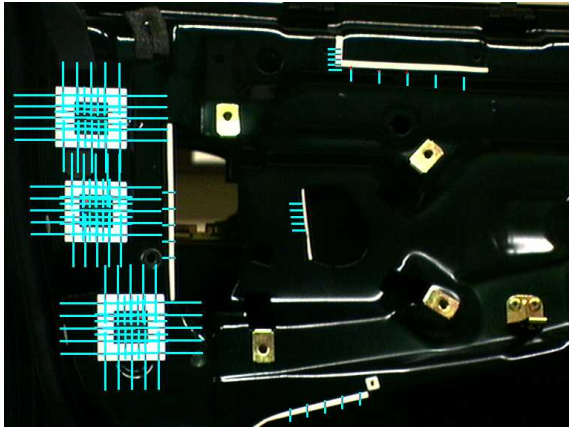


Figure 4: Door with tracked features

calibration for the user's eye. Given additional constraints valid for the two eyes (that they have the same orientation except for a rotation around the up-axis) it is possible to calculate the relative position and orientation in relation to the camera with an appropriate precision. This method turned out to work quite well, but it is dependant on the users keeping their heads very still during the calibration procedure. To ease that a simple headrest was built, which improved calibration quality considerably.

The variation in parameter sets for different users was small enough for another person's parameters to be usable. For best results individual calibration is still needed, but for a quick impression just exchanging the HMD is acceptable.

### 4.3 Data Aquisition

One of the goals of the project was to see how much data for an Augmented Reality application could be taken from other sources.

All the model data could be taken from the virtual application that had been developed (s. sec. 3). Had it been possible to do the augmented application on a high-end graphics machine like the Reality Engine 2 used for the virtual application the data could have been used directly. The O2 used however has very good video capabilities, but its' rendering power is a lot less than the Reality Engine's. Thus the models had to be reduced in complexity to prevent an unacceptable slowdown of the application. The models were half manually, half

automatically reduced to a point where the graphical complexity was acceptable which compromising rendering quality (s. fig. 5).

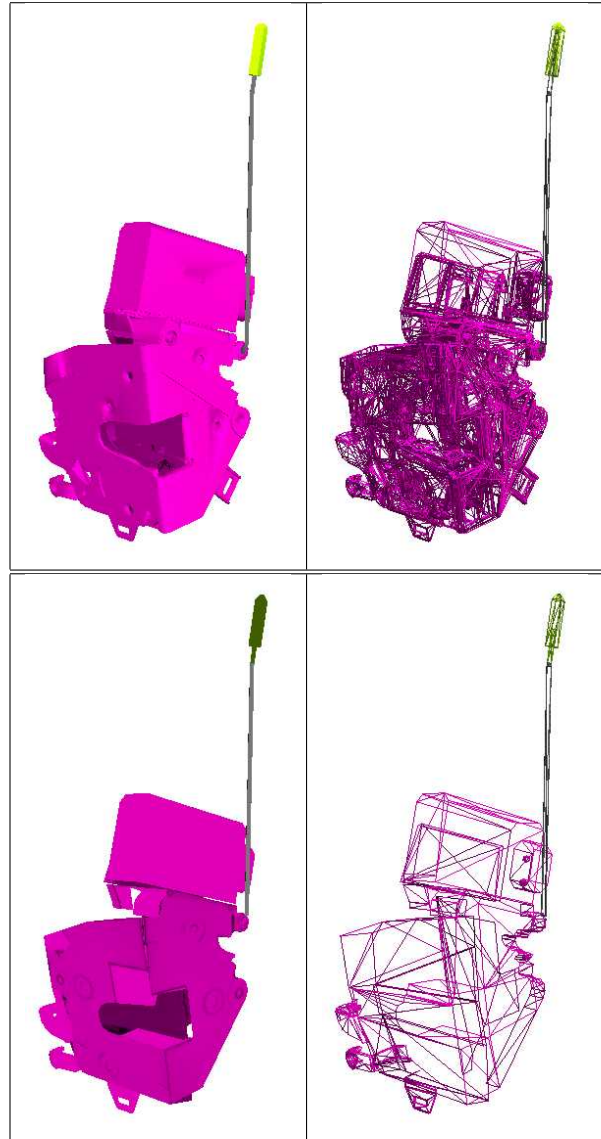


Figure 5: Original (12156 polygons) and reduced lock models (656 polygons)

The door model needed for occlusion (see. fig. 13) was created mostly by just deleting unnecessary parts and some remodeling.

The animation pathes used for the animated instructions were also taken from a performance of the task in the virtual world. Due to inaccuracies

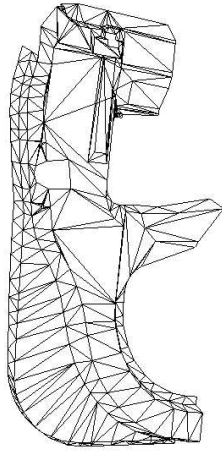


Figure 6: Occlusion model for the door (427 polygons)

in the tracking system used the virtual data was not of high enough quality to be used directly as a demoable instruction, instead some frames were chosen manually and used as keyframes for the animated instructions. This gave a very smooth and continuous animation with very little effort.

#### 4.4 Software

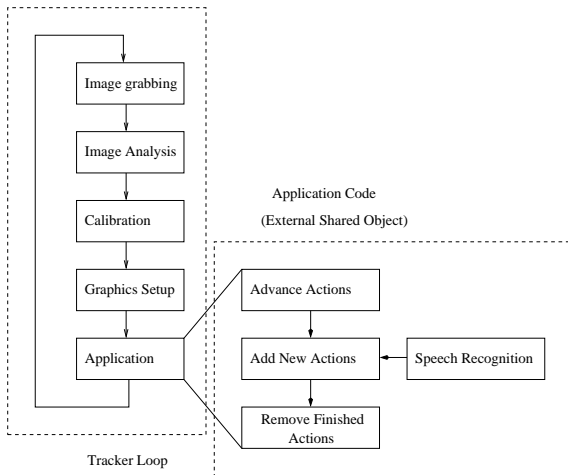


Figure 7: Software Architecture

The software architecture as depicted in figure 7 revolves around the central tracker loop. The

tracker cares about reading and analyzing the video image to calculate the camera calibration parameters, which are translated into OpenGL setup matrices.

**Distribution** Experimentally the system was distributed between two machines, one for tracking (the mentioned SGI O2) and one for rendering (an SGI Onyx RE2). Only the calibration information was sent via Ethernet between the two machines. But the lag resulting from the system and the network proved to be too bad to be useful for head-mounted applications, so for the final application a single system was used. The tracking itself runs between 20 and 25 Hz for the given setup. Combined with the additional burden of rendering the augmentations the speed dropped to 13-15 Hz, which together with a frame of display latency adds up to ca. 100 ms end-to-end delay. This is right at the border commonly accepted for head-mounted applications, and the system would benefit from a faster update rate, but for demonstration purposes the performance was acceptable.

**See-through vs. Feed-through** Another advantage of the closely connected system is the possibility to let the tracker system draw the video frame used for calibration as a background before passing control to the application, thus allowing a switch between see-through and feed-through within the same system. This was used for the Hannover Fair exhibition as described in sec. 5. It also allowed to move the decision whether to use feed-through or see-through to a late stage in the development cycle so it could be tested in the real situation. The technically unavoidable distance between the head-mounted camera and the user's eye in connection with the very close working range for the task made using feed-through feel very awkward, the difference between the kinesthetic information coming from the arms and hands and what could be seen in the display proved very distracting. Another major disadvantage of the feed-through approach was the availability of only one camera, which forced a monoscopic feed-through mode. For the given task this lack of stereoscopic depth made performing the task very difficult, thus see-through was chosen to be the preferred mode of operation.

**Application integration** The actual application is attached to the tracker by means of a simple procedural interface. Three functions (init,loop,exit) are needed to attach a program to the Augmented Reality system. When the loop function is called the program is free to call any kind of OpenGL commands to generate an appropriate image and react to user input. For our application OpenInventor was used for rendering. It has a clean and simple interface for rendering to an arbitrary OpenGL window and for the simple models employed its performance deficiencies were acceptable.

This division between the tracker and the application eased application development considerably due to the possibility to use a simple simulator program that adhered to the same procedural interface using prerecorded calibration data. Thus actions were easily reproducible and development could be done on any OpenGL-capable machine independent of any special video hardware.

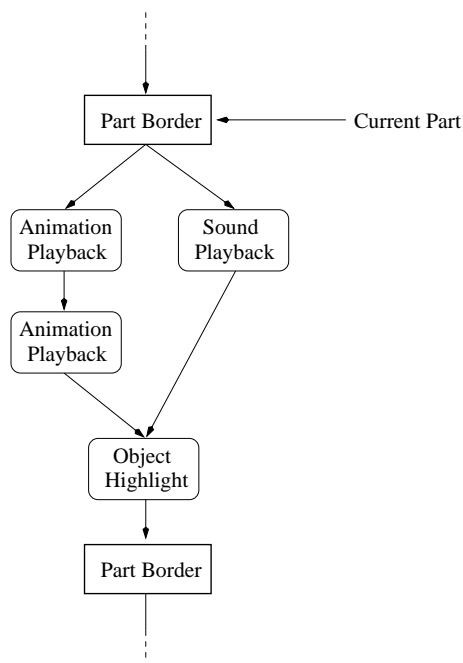


Figure 8: Action graph

**Application Code** The application code as implemented for this scenario uses an action graph as depicted in fig. 8. The whole task was divided

into separate parts that are short enough to be executed by the user without the need for breaks. The instructions to be given for the part are structured as an acyclic directed graph, where every structural node (ovals) has an associated type and duration. Branches in the graph are executed in parallel, joins are synchronization points where all actions leading into the join have to finish before the next action is executed.

Possible actions include sound and animations playback as well as object highlighting and pausing. New actions can easily be added due to the object-oriented structure of the system. Text output was implemented, but the low resolution of the HMD did not allow including a useful amount of text without hiding too much of the visible area, thus digitized speech was used as a means of explanation.

At any given time one of these parts is current. When first advancing into a new part the associated actions are executed. If the user wasn't able to follow the actions on the first try (or rather wants to watch first before doing something) she can request a replay of the part's actions via a voice command. If she gets lost a context can be given by replaying the previous and the current part or if gotten completely lost stepping back is also possible.

**Rendering** The rendering is done using the three-dimensional models described in sec. 4.1. The availability of data for the virtual door allowed to incorporate the real door as an occlusion object in the rendering of the virtual objects, so that the virtual objects can be hidden by the real door (s. fig. 13). This is done by first rendering the virtual door only into the Z-buffer of the graphics hardware, thus occluding objects rendered later without interfering with the contents of the framebuffer (which might be the background video image as used in feed-through mode). By using scan-line interleaved rendering and a straight-through video output a separate image can be sent to each eye, resulting in a stereoscopic image to accurately convey the depth involved.

**Code reuse** The biggest part of this application software was directly taken from the virtual reality system, which uses it for the same task (playing back prerecorded sequences of actions). It had to be

ported to use OpenInventor as a rendering engine, but the amount of code reuse allowed a fast and fluid realization of the system and an easy transfer of animation data.

## 5 Presentation

The system was demonstrated to the general public at the Hannover Industrial Fair 1998 as a part of the Fraunhofer Vision booth. To our knowledge this was the first presentation of an Augmented Reality application to a wide industrial audience in Germany. The presentation was conducted using a live demonstration on stage and at the same time showing an augmented feed-through view from a tripod-mounted movable camera or from the head-mounted camera running on a second independent machine on a large projection screen (see fig. 9).



Figure 9: Setup at the Hannover Industrial Fair '98

The system ran stable for the whole week without problems. After an initial trial phase the speech recognition was switched off in favor of an automatically advancing demo, the ambient noise from talking robots and video screens all around proved to be too much for the used microphone and speech recognition.

Interested visitors were allowed to wear the HMD and take their own peek at the application, but as no individual calibration was done due to time constraints the results were not as overwhelming as some people expected.

Another unappreciated problem was the initial delay of the optical tracker. Right after donning

the HMD a number of visitors started to move their head around wildly looking for the augmentations, thus the tracker never got a stable picture for the .2 seconds needed. A quick switch to feed-through and asking them to look at the squares helped for the initialization, after which the switch back to see-through conveyed the full three-dimensional impression.

The general feedback of the audience was very positive. The physical presence of the door eased the transfer from seeing things on the projection that were just not there in reality, allowing a good lead-in for the concept of integrating computer-generated imagery into views of the real world. This was apparent after the demonstration was shown on german TV. The days after that people came explicitly looking for our booth, some with very specific ideas and questions. A number of companies were interested enough to come up with concrete project concepts and applications and some of them are in the definition phase right now.

## 6 Results & Future Work

An Augmented Reality system for training the doorlock assembly task was created. It uses CAD data taken directly from the construction and production database, as far as possible using today's hardware, and animation and instruction data that was prepared as part of a Virtual Prototyping planning session. This allows the system to be integrated into the existing infrastructure and feeding off data generated for other purposes, thus giving an added value to that data. An optical tracking system was designed and implemented using low cost passive markers that is fast enough for careful HMD use. Showing the system at the Hannover Industrial Fair allowed a number of unrelated visitors to try the system, showing that the tracker is not quite able to handle naive users yet. We are looking into integrating an inertial tracking hardware into the system, which should solve the problematic cases of fast head motion generating motion-blurred, thus unusable, images for the optical tracker. At the same time the optical tracker will be able to alleviate the weaknesses of the pure inertial system, namely its drift. The two approaches should complement each other nicely. The fast acceleration data provided by the iner-



tial system will also be used for better motion prediction to better compensate the system-immanent lag, giving better registration between real and virtual objects.

Ambient sound turned out to be a bigger problem than expected, so a more robust speech recognition software/hardware is being searched. As the interface is extremely simple (ASCII strings of the recognized commands sent via RS-232) nearly any commercial solution should be easily attachable to the system.

One goal is to move the system onto a wearable platform like the Xybernaut machines [1]. For that a Linux port is being worked on.

Industrial partners have been interested enough to start concrete project planning negotiations, so there seems to be a bright light at the horizon for industrial applications of Augmented Reality.

## 7 Acknowledgments

Thanks go to our colleagues Mike Weintke and Claudia Pilo for their hard work on the models right after a tough deadline. We also thank the BMW Virtual Reality Lab, especially Mr. Gomes de Sa and Mr. Baake for their support by allowing us to use the CAD data and by supplying a real door and doorlock for our demonstration. Parts of this work were supported by the european ACTS project CICC (AC017). Laboratory space and equipment were provided by ECRC.

## References

- [1] Xybernaut Corporation. Xybernaut website. <http://www.xybernaut.com/>.
- [2] Neumann e.a. Personal communication. Aug. 97 visit.
- [3] S. Feiner, B. Macintyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53–61, 1993.
- [4] K.N. Kutulakos and J. Vallino. Affine object representations for calibration-free augmented reality. In *Proc. VRAIS'96*, Hong Kong, 1996. IEEE.
- [5] Steve Mann. Repair instruction. <http://lcs.www.media.mit.edu/projects/wearables/augmented-reality.html>, 1998.
- [6] U. Neumann and Y. Cho. A self-tracking augmented reality system. In *Proc. of Virtual Reality Software and Technology (VRAIS96)*, pages 109–115, Hong Kong, 1996.
- [7] T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R.W. Picard, and A. Pentland. Augmented reality through wearable computing. *Presence, Special Issue on Augmented Reality*, 6(4):386–398, August 1997.
- [8] D. Stricker, G. Klinker, and D. Reiners. A fast and robust line-based optical tracker for augmented reality applications. In *1. International Workshop on Augmented Reality (IWAR'98)*, pages 129–145, San Francisco, November 1998. AK Peters.
- [9] R.Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. CVPR*, pages 364–374. IEEE, 1986. see also <http://www.cs.cmu.edu/~rgw/TsaiCode.html>.
- [10] A. Webster, S. Feiner, B. MacIntyre, W. Massie, and T. Krueger. Augmented reality in architectural construction, inspection, and renovation. In *Proc. ASCE Third Congress on Computing in Civil Engineering*, pages 913–919, Anaheim, CA, June 17-19 1996.
- [11] 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.

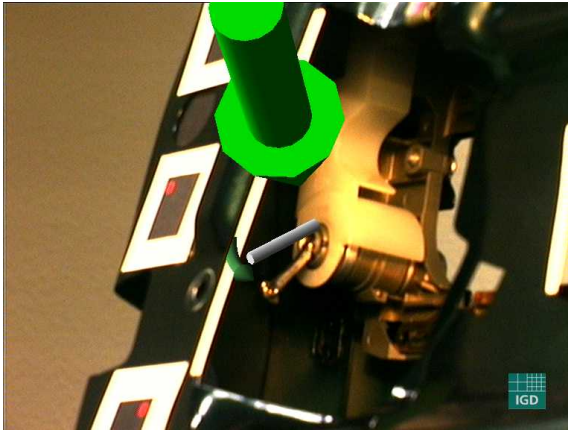


Figure 10: Lever inside the door to be pushed to right position

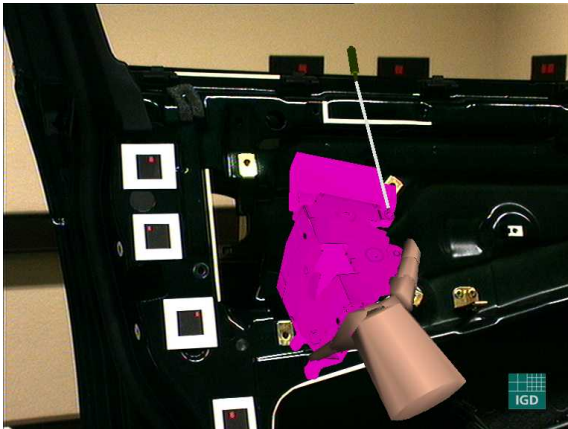


Figure 11: Gripping instruction



Figure 12: Moving instruction

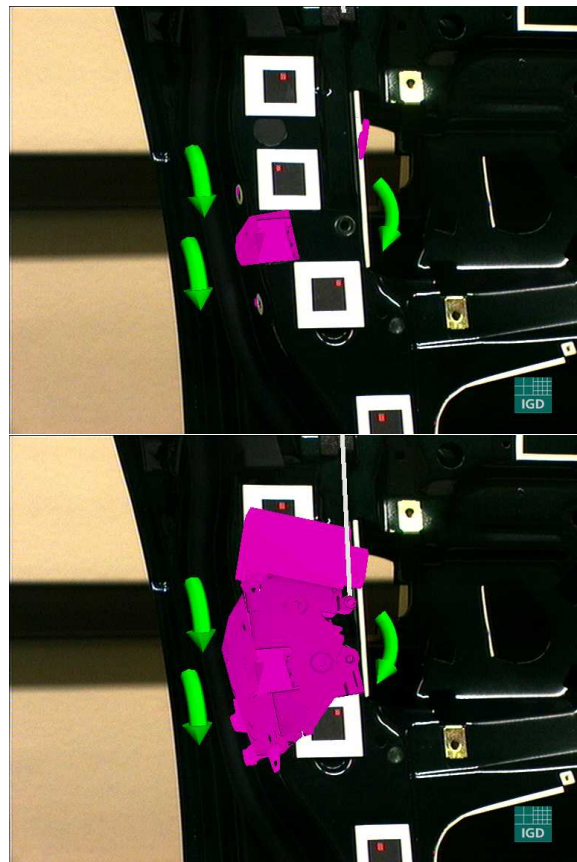


Figure 13: Screw-fixing instruction with and without occluding door