Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Using Augmented Reality to Interact with an Autonomous Mobile Platform

Björn Giesler Tobias Salb Peter Steinhaus Rüdiger Dillmann
University of Karlsruhe (TH)
Chair for Industrial Applications of Informatics and Microsystems (IAIM)
Karlsruhe, D 76128, Germany

Abstract— **To allow users without special knowledge to interact with robots, it is desirable to make interaction methods as intuitive as possible. This goal is in many cases difficult to achieve since data flow from the robot to the human is limited, especially if free locomotion of both the human and the robot are required. Therefore, new communication channels need to be created. In this paper, we propose the use of an Augmented Reality display together with a wearable, wirelessly networked computer to achieve this goal. This system makes it possible to overlay planning, world model and sensory data provided by the robot over the wearer's field of view. We discuss the system architecture, interaction methods and experimental results. We demonstrate an example application for rapid prototyping of a warehouse transport system using the Augmented Reality system and a mobile platform. The user can create a topological map in an unknown environment on-the fly by setting and manipulating map nodes. This is done by pointing at the floor with a special interaction device, and issuing voice commands. The map is shown to the user as an augmentation of the real world view. Additionally, the robot's path planning data is visualized.**

I. Introduction

Interaction between two humans is characterized by rich communication channels: Language, facial expression, hand and whole body gestures, etc. Especially when instructing someone to do a physical task, much of this richness is needed to convey correct understanding of the instructions, ability (or inability) to follow them, requirements that need to be fulfilled in advance, and other information important to the instructor.

In service robotics, few of these channels exist today in a way expected by humans, since they require the robot to have a face, articulate arms and body, and speech. Additionally, much of the information (such as paths, world model, etc.) are difficult to convey using traditional methods at all. The usual form of interaction is a central control computer where sensory, world model and planning data is visualized. Since this is in most cases a stationary computer system, using it is unsatisfactory in situations where the user must move around.

There have been approaches to construct humanoid features such as faces for service robots, the most prominent of which is probably MIT's Kismet [1]. Such approaches are highly interesting for human-like interaction, and can be used to visualize correct understanding of instructions and (in-)ability to perform a task. However, they do not solve the problem of visualizing sensor, planning and world model data, which is very helpful especially in interaction with mobile robots.

To overcome this problem, an Augmented Reality system has been developed at the IAIM that can be used for human-robot interaction. Augmented Reality (AR) is the concept of overlaying computer-generated image elements (often: elements of 3D computer simulation) over the user's field of view, thus "augmenting" his/her view of reality by illusionary objects. In the presented system, this technique is used to visualize the topological and geometrical maps of the autonomous mobile platform ODETE, as well as laser scanner readings and planning data. In the future, we will use the system to instruct our humanoid robot ALBERT, and incorporate world model visualization and manipulation.

II. State of the Art

There has been much research in the field of AR in the past years (an excellent overview and update is given in [2], [3]). The applicability of AR is examined in domains as diverse as intraoperative visualization of surgical data [4], urban planning [5] and interactive user manuals [6]. As diverse as the field of applications is the number of techniques involved; the two perhaps most important distinctions from a technical point of view lie in the choice of the visualization and tracking systems employed. Visualization systems currently in use can be can distinguished into one of three categories:

- **Projector-based.** [7] An LCD projector is used to project computer-generated images onto a surface. Because of the bulk and weight of current LCD projectors, these systems are mostly used in static showcases.
- **Video See-through Goggles (VST).** [8] This term is often just an euphemism for Virtual-Reality goggles that have been retrofitted with cameras that are used to merge a view of the outside world with the computer-generated images. These systems often offer excellent contrast and clarity of the imagery and can take calibration from camera calibration, but suffer from the disadvantage of using the optical system of the cameras involved, which is different from that of the user's eye.[1] The result is the possibility of headaches and motion sickness. In addition, the hand-eye coordination capabilities of the user suffer, making VST systems inadequate for fine manipulation

[1]However, a study conducted in [9] has shown only seven in 40 test persons to feel any ill effects from long-term use of the VST system; thus, while the system clearly is not ready for daily use, it is viable as an experimentation device.

0-7803-8232-3/04/$17.00 ©2004 IEEE

tasks (such as intraoperative display) or long-time use. However, because of their ready availability they make up excellent experimentation systems.

- **Optical See-through Goggles (OST).** [10], [11] These systems use goggles similar to those used in VST applications, with the difference that the user receives a view of the environment through a semi-transparent mirror upon which the computer-generated imagery is projected. OST systems offer a view in the user's own optical system, but suffer from the calibration steps necessary every time the system is worn (such as described in [12]), as well as a tradeoff between contrast of simulated images and brightness of world view. An additional disadvantage is that OST goggles have all but disappeared from the market. Because of the optical system, however, OST is the only pursuable way where fine manipulation is required.

From this list, it is easy to see that there is a tradeoff involved in current solutions between the working area covered by the display system (projectors vs. goggles), as well as a tradeoff between quality of the display and accuracy of the depiction of reality (VST vs. OST). With these tradeoffs in place, choosing the right display device is very much dependent on the problem domain.

One of the most prominent problems in AR is *tracking*, or the determination of the user's *pose* (position and orientation of gaze). This is eminently important as its inversion is required to produce the illusion of objects that remain constant in space. Some commercial tracking systems are available on the market, solutions based on magnetic fields such as the Polhemus [13] or Ascension [14] systems, or camera-based approaches such as the POLARIS [15] tracker. In VST systems, tracking is often based on uniquely identifiable, artificial landmarks or fiducials, in systems such as ARToolkit [8], VIS-Tracker [16], SCR [17] and others (an excellent comparative study can be found in [18]). Markerless tracking, as described in [19], [20], is still too slow to be used in real-time applications, but that will change as processor speeds and technologies evolve.

Important aspects of tracking systems are the *working area* they cover as well as the *accuracy* they provide. In general, it can be said that these two criteria are inversely proportional, leading to yet another tradeoff: Applications that require a high degree of precision will have a small working area; applications that require large working areas will have low accuracy. This tradeoff can be met by combining several tracking methods and using multi-sensor fusion methods. Sec. III-B describes our approach to this problem.

III. General Approach

At the IAIM, an AR system has been developed in a joint effort by the medicine and robotics groups, to be used in intraoperative visualization [11] and human-robot interaction. Because of the diverse nature of the target disciplines, the system has been designed as modular as possible, to make it easily adaptable to different visualization and tracking systems. The system design is described in III-A. The system has already proven its viability in medical applications.

To demonstrate the applicability of AR to the field of human-robot interaction, a real-world application has been chosen; that of introducing a mobile robot into a warehouse for transport tasks. Such an endeavor usually requires mapping of the target area, possibly introducing artificial landmarks, careful mapping of pickup and delivery locations, and many other tedious tasks. Especially for commercial applications (such as those developed by one of our partners in the MORPHA project, propack data [21]), it is often desirable to rapidly prototype and demonstrate such an application to examine its usability.

The presented application enables the user to interactively construct a topological map of paths and nodes in the warehouse by walking around and pointing at the floor. The map can be manipulated and a robot can be instructed to move along the map. The planned path is visualized, and the user can inspect possible intersecting points with workers' paths, possibly closing doors, etc., thus quickly obtaining a rough estimate for the efficiency of the solution.

A. Software Design

As can be seen from section II, there are some issues in AR that cannot right now be solved in the general case because of technical and fundamental problems. Therefore, a core system has been developed that is modular enough to be easily reconfigured for different applications.

To remain as modular and portable as possible, the system uses a plugin approach, in which system components are loaded at runtime depending on what kind of setup is required by the application. Components belong to the following four core classes:

- **Tracker.** Either an interface to a hardware tracking system, such as the NDI Polaris [15], or a position-determining component implemented in the module itself, such as a camera-based tracker. The Tracker component used determines in large part the accuracy and the operation area of the system.
- **Renderer.** A software interface to the OpenGL/OpenInventor graphics system, containing view calibration and deciding between OST or VST (using a camera component).
- **Camera.** An optional interface to whatever video interface the operating system provides. Cameras serve images that can be used as source data by Tracker and/or Renderer components.
- **World.** A World is the actual application that the system runs. Worlds hold information about the world model and use tracking information to update the OpenInventor visualization in each cycle of the run loop.

This organization has the additional advantage of keeping the main system very small and portable, since the operating-system dependent parts are encapsulated in separate plugins.

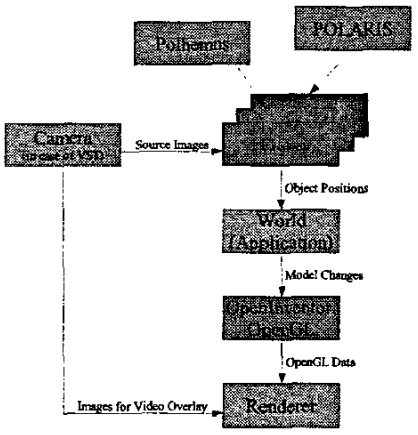

Fig. 1. The software components and their interaction. Blue: exchangeable plugin conponents; red: external components

B. Tracking Device Abstraction

It is often desirable to use several tracking systems at once, for example to achieve high accuracy in localized areas (such as a workbench) while retaining a large overall working area. To make this possible, the system distinguishes between *tracked features* and *tracked objects*. Tracked features are core units of specific tracking systems, such as ARToolkit fiducials, POLARIS retro-target clusters, POLHEMUS/Ascension magnetic sensors, etc. Tracked objects are real-world objects such as a magic wand or the user's head. Tracked features are affixed to tracked objects, and their geometric relationship to the coordinate origin of the respective tracked object is expressed as homogeneous transforms. This approach makes it possible to track an object by several tracking devices simultaenously or alternatively, and use methods of sensor fusion to refine the tracked object's location in space. Figure 2 shows an example of a tracked object with several tracked features.

C. Hardware Requirements

The software runs on a standard Intel PIII/800 PC running Linux; OpenGL visualization is done with an ATI Radeon 9600 card, and as framegrabbers two standard "TV" cards are used. Although the hardware setup is relatively stationary at the moment (the user has an operational radius of about three meters around the computer, due to the cabling involved), future work will be directed to developing or selecting a wearable computer powerful enough to run the software.

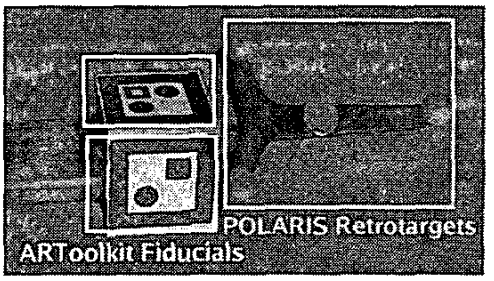

Fig. 2. A Tracked Object with ARToolkit and POLARIS fiducials.

D. Application Setup

For the presented application, rapid prototyping and manipulation of topological maps and control of a mobile platform navigating in these maps, the following requirements have been identified:

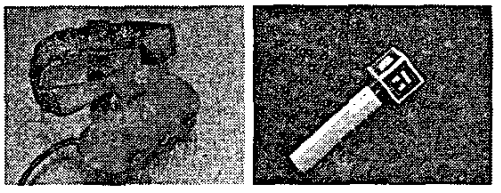

Fig. 3. Our VST goggles and the "Magic Wand" used for interaction

1) Tracking: Wide-area tracking is necessary; optimally, the user should be able to walk around a large warehouse and have full tracking coverage. On the other hand, high accuracy is secondary since the problem domain does not call for it and the robot localization is relatively inaccurate in itself (it has an accuracy of about 5cm). We have chosen the ARToolkit [8] as our main tracking system.

The ARToolkit uses camera images to identify square fiducials (as seen in fig. 2) with unique patterns in the center. It is able to determine the fiducials' poses in relation to the camera; given a known location of the fiducials in space, the inversion of this pose can be used to determine the camera's pose in space. To use this method for large-area tracking, many fiducials must be distributed in the working area and their position in space must be determined, which is lengthy and error-prone to do by hand.

To reduce this effort, a method has been developed to calibrate all fiducials in relation to a single origin fiducial, which marks the origin of the tracking coordinate system. The method iteratively refines the fiducial positions through the following algorithm:

1) Every fiducial f_i is assigned a weight w_i, with $w_0 = \infty$ for the origin fiducial f_0 and $w_i = 0 \forall i \neq 0$. If a fiducial's weight becomes larger as a certain w_{lim}, it is considered calibrated; i.e. its pose in the world coordinate system is known.

2) A new camera image is taken in every step; all visible fiducials are identified.
3) The user's pose P is calculated from all *calibrated* fiducials in the image using a least-squares approach. The weight of the camera w_c is calculated as the sum of the weights of all *calibrated* fiducials in the image. The interim poses F_i' of all *visible* and identified fiducials in the image are calculated from P.
4) If $w_c < w_{lim}$, the camera's pose is considered to be too uncertain to use in calibration. Otherwise,
5) Each of the identified fiducials is assigned an interim weight $w_i' = \sum$ pixels in fiducial$/\sum$ pxiels in image.
6) Each fiducial pose F_i is updated to $\frac{w_i F_i + w_i' F_i'}{w_i + w_i'}$. Each fiducial's weigt w_i is updated to $w_i + w_i'$.

The user's experience in using this method is that of doing a slow camera sweep over all fiducials starting at the origin. The fiducials are color-marked in the user's view with colors between red and green representing their weight. The process takes less than one minute for markers distributed over a $25m^2$ area, depending on the frame rate, and is very intuitive to use. Fig. 4 shows an example run.

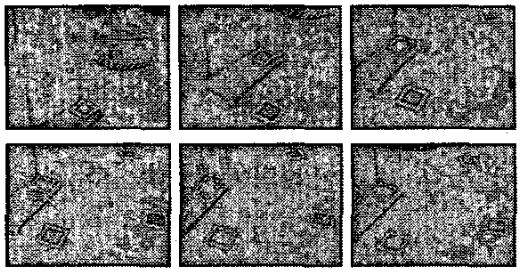

Fig. 4. Calibrating fiducials beginning at a designated origin fiducial. Green fiducials are calibrated, red ones are not. The shades in between designate the system's certainty about their position.

2) Visualization: Lighting conditions and user experience make a VST system desirable, especially since fine hand-eye coordination is secondary in the problem domain.

Because of availability issues and ease of combination with the ARToolkit tracking system, for the robot interaction application a VST head-mounted display is used that can be seen on the left in figure 3 (this is a prototype display manufactured by German company Trivisio [22]).

3) Interaction: Since the primary interaction method is pointing to certain locations on the floor and performing actions at those locations (setting/manipulating map nodes, telling the robot to move to the indicated location), a pointing device together with a command-passing metaphor is required. We have chosen a simple "magic wand" (visible on the right in fig. 3) that is also tracked using ARToolkit markers. The "magic wand" has three buttons which can be queried via a Bluetooth interface; a facility not used by the presented application.

To determine the action that should be taken when the user points to a location using the "wand", the speech recognition system ViaVoice [23] by IBM is used. Speech recognition is done in a simple phrase-recognition mode, where phrases such as *"Set a map node here"* are stored and recognized as a whole. This is inflexible but sufficient for the present application, and works very well.

IV. Using the System to Control a Mobile Platform

The goal of the presented application is to create a topological map in an unknown environment on-the-fly by setting and manipulating map nodes. To make pointing and manipulation easier, a cursor is projected on the floor (visible as the yellow cube in fig. 5) when the user points down with the magic wand. The cursor marks the point of intersection between an imaginary ray emanating from the wand (in z coordinate direction) with the floor plane.[2] The resulting sensation is that of very intuitive feedback between wand action and cursor movement, just like moving a mouse on a desktop computer.

A. Manipulating the Topological Map

In the current system, a map node can be created at the cursor position by speaking the word *"Set"*. Map nodes that have been set can be connected to other map nodes by pointing at the origin node and saying *"Connect this node..."*. After that, the selected node will change color to purple and the cursor will be followed by a purple map edge. Moving the cursor over a target node and saying *"...with this node"* will form a connection between the two nodes, representing a map edge that the robot can move upon. Fig. 5 shows the map-generation process from the user's point of view.

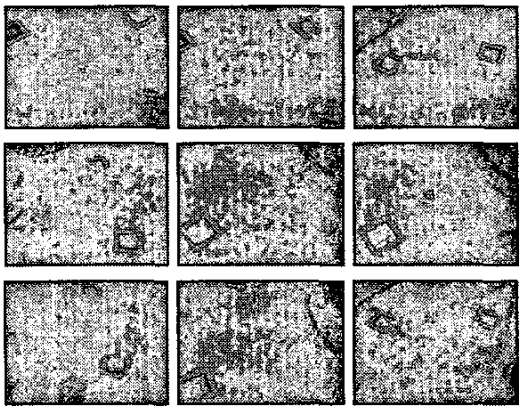

Fig. 5. Interactive generation of a topological map. Light blue nodes are currently selected by the cursor, violet color marks nodes currently being connected.

Map nodes that have been set can be moved to new locations (*"Move this node..." / "...to here"*) and deleted, as shown in fig. 6; deleting a node also deletes all edges leading up to it. Deleting edges separately is currently not implemented.

[2]If the user is not pointing at the floor (meaning, the intersection point would be farther than 10m away from the user's feet, or in negative z from the magic wand), the cursor is hidden from view, and voice commands have no effect.

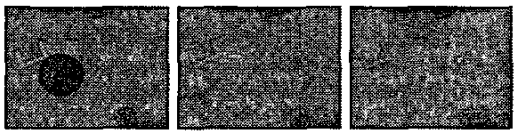

Fig. 6. Deleting a map node set by accident.

B. Visualization of Planned Paths

After a topological map has been created, the robot can be instructed to move along it to certain map nodes. This can be done simply by pointing at the map node and telling the robot to *"go to this node"*. This command triggers the robot's planning algorithm, finding a path in the map. If the robot is not currently on a map node, it drives (with collision avoidance) to the nearest node and begins planning from there. When the planning algorithm has found a path, it is demonstrated to the user by marking the nodes and edges in the path in red color (fig. 7), and the robot begins driving along the path. When it reaches the goal node, the nodes and edges are re-set to blue color, demonstrating to the user that the task has been successfully performed.

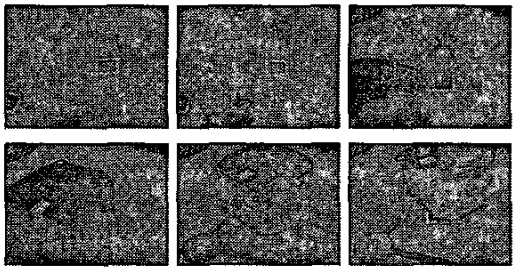

Fig. 7. Instructing the robot to visit a map node. The planned path that the robot will take is marked in red.

While the robot is following a path, it may find certain nodes unreachable or edges unpassable because they are blocked by obstacles. In the current implementation, the robot will try to plan a new path around the obstacle rather than trying to drive around it freely. If a path is blocked, the robot drives back to the last-visited node and begins constructing a new plan from there. As can be seen from fig. 8, the user is kept up-to-date about the current plan at all times.

C. Free-Drive Mode

Of course, it is also possible to instruct the robot to visit indicated positions without constructing a map first, by pointing at the position and instructing the robot to *"go there"*. A free-drive mode has also been investigated in which the robot tries to move to the cursor position immediately, but the non-omnidirectional nature (requiring a turn at every direction change) and the slow speed of the current platform made this impractical. Free-drive mode will be examined again soon with a new omnidirectional platform currently under construction at the IAIM.

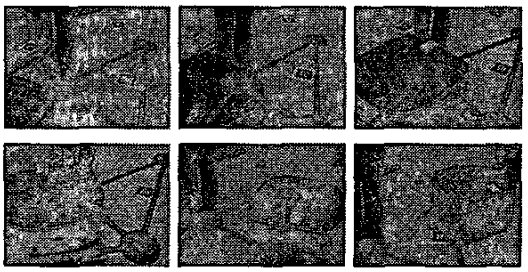

Fig. 8. The robot finds a map edge blocked by an obstacle, retraces its steps to the last-visited node, creates a new plan to the goal and follows it. The user is kept informed of the robot's path at all times.

V. Future Work

The current application of our AR system in robotics, as presented in this paper, is still a very simple one. However, it serves well to demonstrate both the potential of AR in robot instruction in general and the usability of our specific AR system in this domain. In the following months, we will investigate the benefits of using AR in the field of *Programming by Demonstration* (PbD), using our humanoid robots ALBERT and ARMAR [24]. Paths that will be examined are the following:

Visualization of robot intention. AR is excellently suited to visualize geometrical properties, such as planned trajectories for a robot arm or grip points for a dextrous hand. Therefore, the system will be used to refine existing interaction methods by providing the user with richer data about the robot's intention.

Integration with a rich environmental model. We will examine the benefits of selecting objects for maniuplation simply by pointing at them, together with immediate feedback to the user about whether the correct object has been selected, etc.

Manipulating the environmental model. Since robots are excellent at refining existing geometrical object models from sensor data but very bad at creating new object models from unclassified data themselves, we will try to *integrate* the human user into the robot's sensory loop. The user will receive a view of the robot's sensor data and will be able to use interactive devices to perform cutting and selection actions on the sensor data to help the robot to create new object models from the data itself.

There is also some work to be done on the system itself, most prominently the ARToolkit tracking module, which works well most of the time but sometimes fails for one or more frames because of image noise or changes in lighting, resulting in jumps and sudden tilts in the generated data. This will be solved by integrating some simulated inertia into the tracked features. A full-featured voice recognition system is also desirable, as is a method for creating an on-the-fly 3D environment map to determine occlusions between the environment and simulation elements (so that the map would be occluded by the robot driving over it, or humans walking across it; the visual effect without occlusions, as seen in the

pictures in this paper, can be jarring).

VI. Summary and Conclusion

In this paper, an Augmented Reality system has been presented that is independent of application domain and tracking, visualization and interaction devices. We have demonstrated an application for this system that allows interaction with and instruction of an autonomous mobile robot. We believe that using AR for human-robot interaction holds much promise and will continue research in this field.

VII. Acknowledgements

This research was performed at the Institute for Computer Design and Fault Tolerance (IRF), chair for Industrial Applications of Informatics and Microsystems (IAIM), Prof. Dr.-Ing. R. Dillmann, University of Karlsruhe, Germany. The work is being funded in part by the German Federal Ministry of Education and Research, in the context of the project MORPHA [25].

References

[1] C. Breazeal and B. Scassellati, "How to build robots that make friends and influence people," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, 1999.

[2] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments 6*, vol. 4, pp. 355–385, August 1997.

[3] R. T. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47, November/December 2001.

[4] T. Salb, O. Burgert, T. Gockel, J. Brief, S. Hassfeld, J. Mühling, and R. Dillmann, "Risk reduction in craniofacial surgery using computer-based modeling and intraoperative immersion," in *Proceedings of Medicine Meets Virtual Reality 10*, January 2002.

[5] H. Ishii, J. Underkoffler, D. Chak, B. Piper, E. Ben-Joseph, L. Yeung, and Z. Kanji, "Augmented urban planning workbench: Overlaying drawings, physical models and digital simulation," in *Proc. IEEE Int'l Symposium on Mixed and Augmented Reality*, October 2002, pp. 203–211.

[6] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. Navab, "Markerless tracking for AR: A learning-based approach," in *Proc. IEEE Int'l Symposium on Mixed and Augmented Reality*, October 2002, pp. 295–304.

[7] H. Hoppe, S. Däuber, J. Raczkowsky, H. Wörn, and J. Moctezuma, "Intraoperative visualization of surgical planning data using video projectors," in *Medicine Meets Virtual Reality (MMVR)*, J. W. et. al., Ed. Newport Beach, CA: IOS Press and Ohmsha, 2001, pp. 206–208.

[8] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. IEEE International Workshop on Augmented Reality*, 1999, pp. 125–133.

[9] C. S. Özbek, "Spielerische Evaluierung eines Augmented Reality Systems," Master's thesis, University of Karlsruhe (TH), July 2003.

[10] Y. Genc, M. Tuceryan, and N. Navab, "Practical solutions for calibration of optical see-through devices," in *Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, September 2002, pp. 169–175.

[11] T. Salb, J. Brief, T. Welzel, B. Giesler, S. Hassfeld, J. Mühling, and R. Dillmann, "INPRES (intraoperative presentation of surgical planning and simulation results) — augmented reality for craniofacial surgery," in *SPIE Electronic Imaging. International Conference on Stereoscopic Displays and Virtual Reality Systems*, J. M. et. al., Ed. San Jose, CA: spie, Januar 2003.

[12] M. Tuceryan and N. Navab, "Single point active alignment method (SPAAM) for optical see-through hmd calibration for AR," *Presence: Teleoperators and Virtual Environments*, vol. 11, no. 3, pp. 259–276, June 2002.

[13] "Polhemus Corporation Homepage," http://www.polhemus.com.

[14] "Ascension Corporation Homepage," http://www.ascension-tech.com.

[15] "Northern Digital, Inc.: The Polaris Tracking System," http://www.ndigital.com/polaris.html.

[16] L. Naimark and E. Foxlin, "Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker," in *Proc. IEEE Int'l Symposium on Mixed and Augmented Reality*, October 2002, pp. 27–36.

[17] X. Zhang, Y. Genc, and N. Navab, "Mobile computing and industrial augmented reality for real-time data access," in *Proc. 7th IEEE Int'l Conference on Emerging Technologies and Factory Automation*, 2001.

[18] X. Zhang, S. Fronz, and N. Navab, "Visual marker detection and decoding in AR systems: A comparative study," in *Proc. IEEE Int'l Symposium on Mixed and Augmented Reality*, October 2002, pp. 97–106.

[19] G. Simon and M.-O. Berger, "Reconstructing while registering: A novel approach for markerless augmented reality," in *Proc. IEEE Int'l Symposium on Mixed and Augmented Reality*, October 2002, pp. 285–294.

[20] S. J. P. Kar Wee Chia, Adrian David Cheok, "Online 6dof augmented reality registration from natural features," in *Proc. IEEE Int'l Symposium on Mixed and Augmented Reality*, October 2002, pp. 305–313.

[21] "propack data GmbH Company Homepage," http://www.propack-data.de.

[22] ""trivisio corporate homepage"," http://www.trivisio.de.

[23] "IBM ViaVoice Homepage," http://www.ibm.com/viavoice.

[24] R. Dillmann, M. Ehrenmann, P. Steinhaus, O. Rogalla, and R. Zöllner, "Human friendly programming of humanoid robots - the german collaborative research center," in *Proc. of the International Advanced Robotics Programme*, December 2002.

[25] German Ministry of Education and Research, "Morpha project home page," http://www.morpha.de.