Exploiting the Physical World as User Interface in Augmented Reality Applications

István Barakonyi, Dieter Schmalstieg
Graz University of Technology, Austria
{ bara | schmalstieg } @ icg.tugraz.ac.at

Abstract

This paper proposes an advanced exploitation of reality in Augmented Reality (AR) applications by turning physical objects into intelligent, responsive entities, which are equal, active partners of virtual objects. We assert that autonomous agents can be used as a user interface for AR systems; formulate an agent behavior scheme and discuss some visualization techniques used in a pilot application implementing a machine maintenance scenario.

1. Introduction

"Reality is merely an illusion, albeit a very persistent one", according to Albert Einstein. In Virtual Reality (VR) environments synthetic worlds inhabited by virtual objects act as an interface between applications and users. During the application development process the most effort is usually put into creating a faithful model of physical objects so that users have the "illusion" of seeing the "real thing". This model not only includes an accurate visual representation of all relevant object states but a careful depiction of the behavior of the respective objects as well.

A significant advantage of Augmented Reality (AR) systems over their VR counterparts is the exploitation of the physical world. Developers do not need to consider and model every single detail in applications since the details are already physically present, with infinite resolution and accuracy. The sensitive tasks of modeling appearance and behavior can be reduced to superimposing only meaningful, application-specific virtual information over real world objects. This reduction enables a better focus on efficient information visualization.

2. The physical world as output modality

Despite the obvious advantages AR environments offer, only few applications take full advantage of real world features. Classic AR applications utilize real world attributes such as position, orientation, sound, light, temperature etc. as input modalities. These attributes are measured and processed by various sensors. Based on the data delivered by sensors a virtual representation is rendered, usually as a computer graphics image and synthesized sound on top of its real world counterpart. Typical examples are tangible AR applications [1], where the user manipulates virtual objects by physically manipulating real, tangible props.

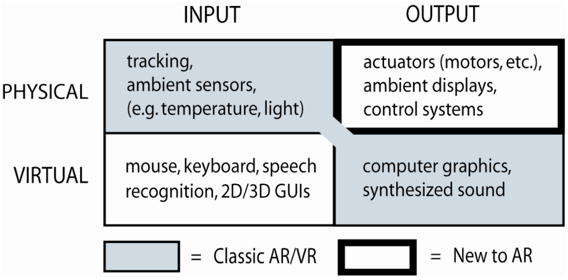

Figure 1. Input and output modalities

Actuators and control systems have been used for a long time as output communication channels in various engineering fields such as robotics. For instance the QRIO robot from Sony [2] transforms user voice commands and gestures into anthropomorphic movement by manipulating a complex network of motors. Ambient displays [3] are physical devices that transmit information on the periphery of human perception using light, sound or movement. However, neither QRIO nor ambient displays combine physical and virtual output as AR systems do.

We argue that AR applications should take advantage of the physical world by acting as an output modality, which is a hitherto unexploited concept.

3. Example scenario

Machine maintenance problems similar to the early work of Feiner et al. [4] are traditional AR scenarios.

Let us imagine that a complex machine breaks down in a factory. Well-trained technicians are not always available due to spatial, temporal or financial constraints, therefore lengthy manuals are supplied with the machine, full of illustrations concerning which button to press or which container to refill. Existing systems like the STAR project [5] provide an animated presentation by displaying a virtual manual superimposed on the real machine, however, they neither verify nor provide feedback whether the actual maintenance step has been executed correctly. A desirable feature involves continuously querying the physical system state and comparison with the demanded application state that should have been reached by the correct execution of instructions.

Another possibility is having a knowledgeable remote operator who sends instructions to the local, untrained operator from a remote location. AR provides the overlay of virtual icons, images and animated models on top of the physical machine explaining what should be done and playing synthetic sound besides the original sounds of the machine. Synchronized with the virtual information physical actions can be executed on the actual machine. With physical actions no simulation of machine behavior is necessary since the real machine does exactly what it is expected to do and gives feedback through its own engines, instruments, control displays, LEDs, etc.

A further notable issue rarely tackled in AR-based maintenance applications is industrial safety. If an untrained operator could see what *would* happen if he pulled the wrong lever before the actual execution of the action, accidents could be avoided and industrial training would be more efficient by preparing trainees for the results of possible failures.

4. Autonomous agents in AR

4.1 Agent-based UI for AR applications

To implement the previously described scenario in an AR application, we need a suitable user interface (UI). We argue that autonomous agents are a good choice for a key visual UI element as previously demonstrated in our earlier work [6]. Autonomous agents are “smart” software components which proactively make decisions based on events coming from sensors present in their environment, an internal world simulation model and some persistent knowledge. In highly dynamic environments, such as AR, this autonomy is useful because the application that the agents are embedded into does not need constant guidance from the user and thus can hide irrelevant, low-level details.

Autonomous agents in AR can be embodied as three-dimensional virtual or *physical* objects. Virtual agents in AR scenarios appear to have a solid, tangible body that can be observed from an arbitrary viewpoint, thus becoming integral parts of the physical environment. The virtual bodies represent typically animated characters but are not necessarily anthropomorphic, sometimes a simple animated arrow serves application purposes better. Animated characters receive and deliver information through multimodal channels such as animations, gestures, speech synthesis and recognition, and various sounds.

4.2 Physical objects as first-class entities

A novel and exciting aspect of AR agents is that physical objects like printers, digital pianos or interactive robots can be turned into intelligent, responsive entities that collaborate with virtual characters and users as equal, active partners. If we track and monitor relevant physical attributes and process this data, attribute changes can generate events that can be interpreted by other agents and application logic. Using network packets, infrared messages, MIDI code sequences or other means of low-level communication, physical objects can not only be queried for status information but can also be controlled by external commands that trigger actuators. Therefore physical objects act as *input and output devices* in AR spaces.

4.3 Overcoming human sensory limitations

The use of physical objects as first-class entities introduces certain constraints imposed by the limitations of our human sensors. We are unable to look through opaque objects or examine their internal structure, see in dark or foggy places, or hear distant sound sources. We also have difficulties in selecting relevant information in too much or too noisy data.

By augmenting our physical objects with virtual information we overcome these limitations. Visualization techniques described in our pilot application in Section 5 are examples how the capabilities of our senses can be enhanced.

Figure 2 illustrates the behavior scheme of autonomous agents in an AR environment. To add semantics to the visual and aural means of augmentation, our system needs a model of the physical world such as the mechanical and spatial structure of a car engine, the meaning and source of error messages generated by a printer or the safe range of data coming from sensors surveilling a production line. The system needs to maintain its knowledge of the real world, therefore the

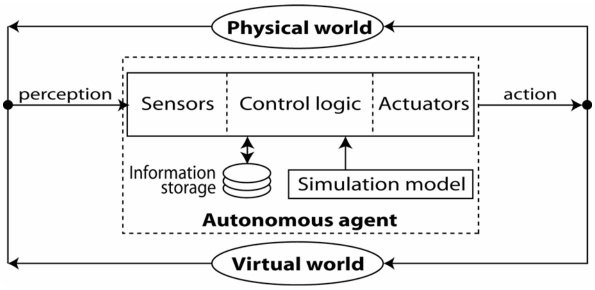

Figure 2. Behavior scheme for autonomous agents in AR applications

data received by the agent's sensors needs to be processed by some control logic or reasoning module based on the agent's internal world model. The agent stores and retrieves information to implement memory and persistency. Having interpreted sensor data, the agent can react accordingly by generating actions in the physical and virtual world.

4.4 Collaboration with AR agents

In a collaborative AR setting the attributes of visible virtual information depend on user location and orientation (e.g. only annotating machine parts currently visible to the user), and profile (e.g. novice users receive more basic information than experienced ones). Therefore, virtual objects are rendered in a user-specific way. The attributes of the information must be synchronized and shared among the users. The collaboration with physical objects is obvious, no technology is required.

5. Pilot application: AR Lego

Our pilot application called AR Lego, demonstrates how to fully utilize real world features with the help of autonomous agents. The application is fully described in previous work by the authors [6], therefore we only highlight aspects relevant to this paper's topic. AR Lego implements a machine assembly and maintenance scenario, in which two agents are employed to educate an untrained user to assemble, test and maintain machines composed of active (engines and sensors) and passive (cogwheels, gears, frames) parts. The two agents are a real LEGO Mindstorms® robot and a virtual animated repairman. The PC-based application communicates with the robot via an infrared channel, through which it sends commands to control the attributes of the active parts (e.g. engine voltage and direction, type of the sensors) and queries the current robot state (e.g. sensor values, communication channel failures, or battery level).

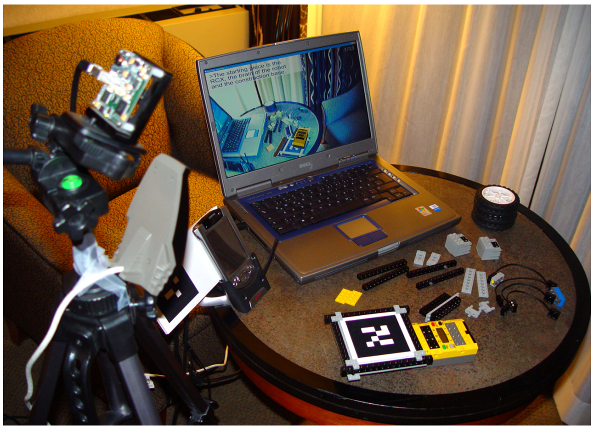

Figure 3. Work environment in AR Lego

Upon starting the application, the user has a set of LEGO® building blocks (see Figure 3). The system provides step-by-step assembly instructions as to which block to mount next and how to verify whether the user is at the correct stage in the construction. The verification of passive parts (i.e. inactive bricks) is only possible by visually comparing the appearance of the physical model to the virtual. We have built an accurate virtual 3D model of the full robot using a CAD editor and superimpose it over the real robot. It is possible to switch between a full model representation allowing a careful comparison with the physical model or just showing the next brick to be mounted to prevent obstructing the view of the real parts during construction.

Although more complex, testing whether the active parts (engines and sensors) have been mounted correctly is a more straightforward task. After mounting an engine the application instructs the robot by an infrared command to turn the engine on. If mounted in the right position and correct direction, the engine and all moving parts connected to it should behave as demonstrated by overlaid animated virtual models. Similarly, if we mounted the sensors properly, the right type and range of data should arrive from the robot. The system checks and visually reports inconsistency so that the user can go back one or more steps to double-check the construction.

6. Visual augmentation

AR Lego uses virtual objects superimposed on top of the physical robot for visual augmentation. The following techniques are used in our demo as demonstrated in the screenshots in Figure 4:

- **Labels and icons:** Labels and icons placed next to robot parts explain functionality or display information about the current robot state, e.g. explanation of the light sensor or a display of the battery level.

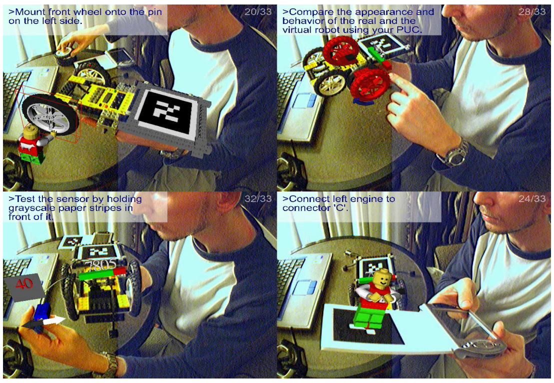

Figure 4. Screenshots from the AR Lego demo

- **Wireframe overlay:** Instead of showing a fully detailed virtual model, a wireframe model often yields better emphasis of certain features, such as indicating where and how to mount the next building block onto the robot.
- **Zooming:** Zooming onto some fine details draws the user's attention to information otherwise easily overlooked, such as the icons and numbers on the LCD display of the robot control module.
- **X-ray vision:** This often quoted "superman"-like feature of AR allows observing the internal structure of the robot or showing parts otherwise occluded by adjusting the transparency of the virtual objects rendered on top or in place of physical parts.
- **3D fisheye:** Although not implemented in our current demo, a 3D version of the well-known 2D fisheye technique would prove useful when emphasizing a certain part in a highly complex engine. The 3D fisheye tool temporarily reorganizes the virtual model by enlarging the parts around the center of focus while suppressing others further from the center.
- **Anticipation with animation:** When approaching a button, it would be useful to play animations to demonstrate what would happen if the button was really pressed, e.g. animating a network of engines revealing design or construction errors before damaging the actual device or injuring the user.
- **Gestures:** Virtual animated agents are able to perform human-like gestures over the correct physical location to visualize how to mount a certain block or connect a sensor. These gestures are the animated 3D versions of the static explanation images often seen in assembly and maintenance manuals like that of IKEA.

Although most of the visualization techniques used in our demo have been previously presented individually by others [7][8][9][10][11], their use and combination with the augmentation of physical objects and autonomous agents is novel.

7. Conclusion and future work

In this paper we have proposed a way to better exploit the real world in AR applications. This can be achieved by turning physical objects into intelligent, responsive agents and using them as input and output devices in AR environments. Various visualization techniques were used with virtual information superimposed over real world objects to enhance the limitations of our human senses. Important future work includes evaluation of the usefulness and effectiveness of these techniques in various scenarios and making experiments with other physical objects such as a MIDI synthesizer or a laser printer.

8. References

[1] Poupyrev, I., Tan, D.S., Bilinghurst, M., Kato, H., Regenbrecht, H., Tetsutani, N., „Developing a Generic Augmented-Reality Interface", *IEEE Computer* 35 (3), 2002, pp. 44-50.

[2] http://www.sony.net/SonyInfo/QRIO/

[3] Wisneski, C., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B. and Yarin, P., "Ambient Displays: Turning Architectual Space into an Interface between People and Digital Information", *Proc. of International Workshop on Cooperative Buildings (CoBuild'98)*, Darmstadt, Germany, 1998, Springer Press, pp. 22-32.

[4] Feiner, S., Maclntyre, B., Seligmann, D., "Knowledge-based Augmented Reality,", *Communcations of the ACM* 36 (7), 1993, pp. 52-62

[5] Vacchetti, L., Lepetit, V., Papagiannakis, G., Ponder, M., Fua, P., "Stable real-time interaction between virtual humans and real scenes", *Proc. of 3DIM 2003*, Banff, AL, Canada, 2003

[6] Barakonyi, I., Psik, T., Schmalstieg, D., "Agents That Talk And Hit Back: Animated Agents in Augmented Reality", *Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality 2004 (ISMAR'04)*, 2004, Arlington, VA, USA, pp. 141-150.

[7] Azuma, R., Furmanski, C., "Evaluating Label Placement for Augmented Reality View Management", *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality (ISMAR 2003)*, Tokyo, 2003, pp. 66-75.

[8] Drascic, D., Grodski, J.J., Milgram, P., Ruffo, K., Wong, P., Zhai, S., "ARGOS: A Display System for Augmenting Reality", Extended Abstract and Video, *Proc. of INTERCHI '93: Human Factors in Computing Systems*, Amsterdam, Netherlands, 1993, p. 521

[9] Bane, R., Höllerer, T., "Interactive Tools for Virtual X-Ray Vision in Mobile Augmented Reality", *Proc. IEEE and ACM International Symposium on Mixed and Augmented Reality 2004 (ISMAR'04)*, 2004, Arlington, VA, USA, pp. 231-239.

[10] Looser, J., Billinghurst, M., Cockburn, A., "Through the Looking Glass: The Use of Lenses as an Interface Tool for Augmented Reality Interfaces", *Proc. of the 2nd Int'l Conf. on Comp. Graphics and Interactive Techniques in Australasia and South-East Asia (Graphite 2004)*, Singapore, 2004, ACM Press, pp. 204-211.

[11] Carpendale, M.S.T., Cowperthwaite, D.J., Fracchia, F.D., "Distortion Viewing Techniques for 3D Data", *Proc. of the IEEE Conf. on Information Visualization (INFO-VIS'96)*, San Francisco, USA, 1996, IEEE Comp. Society Press, pp. 46-53.