7. International Immersive Projection Technologies Workshop
9. Eurographics Workshop on Virtual Environments (2003)
J. Deisinger, A. Kunz (Editors)

The Invisible Person - Advanced Interaction Using an Embedded Interface

Thomas Psik,[1†] Krešimir Matković[2‡], Reinhard Sainitzer[3§], Paolo Petta[4¶] and Zsolt Szalavari[3‖]

[1] Institute for Design and Assesement of Technology, Vienna University of Technology, Austria,
http://www.media.tuwien.ac.at
[2] VRVis Research Center in Vienna, Austria, http://www.vrvis.at
http://www.media.tuwien.ac.at
[3] Imagination Computer Services GesmbH, Vienna, Austria,
http://www.imagination.at
[4] Austrian Research Institute for Artificial Intelligence, Vienna, Austria,
http://www.ai.univie.ac.at/oefai

Abstract

In this paper we describe an advanced user interface enabling even playing games in an immersive virtual environment. There are no common input devices, users presence in the environment, movements, and body postures are the available tools for interaction. Furthermore, a publicly accessible installation in the Vienna Museum of Technology implementing such an advanced environment is described. In this installation computers are completely hidden, and it is one of the most popular exhibits in the museum, which has been accessed by more than 200,000 visitors since September 1999.

1. Introduction

In this paper we describe the new "Invisible Person" interactive installation which is on exhibit in the "Technisches Museum Wien" (TMW , Vienna Museum of Technology). It is an interactive, immersive virtual environment installation with completely hidden computers. The system is placed in a publicly accessible place, where a lot of users access the system. This results in much higher demands on stability and robustness than those required for a lab prototype. The installation consists of a stage, with a sizeable (2.5 x 3.8 m) display on one side. Figure 1 shows the actual installation in the TMW. A camera is placed above the display, and the video of the stage is shown in real time on the screen, creating some sort of mirror effect. The mirror image does not only show the mirrored stage, but includes "The Invisible Person" living in the mirror (it is "invisible" only in the real world). "The Invisible Person" (IP) tries to interact with the users in real time. The only interaction tools available for both are their movements and full body postures. No facial gestures, hand gestures or use of fingers are traced. This makes it simpler for the user and also lessens the computational demand on the system.

Actually, this setup has been in the museum since September 1999, when the original IP installation was set up. Petta et al. [11] described the original installation. It was up and running from September 1999 to June 2001. The original installation was based on the "ALIVE" system[4] from the MIT Media Labs. The ALIVE uses a virtual mirror metaphor and an additional virtual character, who looks like a dog and interacts with them as well. The users can play with that virtual dog by throwing virtual sticks, who it brings back, and additional "doggy" behavior was implemented as well.

A similar installation also based on the virtual mirror metaphor is "The Augmented Man" introduced by Stricker et al.[12]. This installation has a more artistic approach. There was a quite sizable time lag (approx. 4 seconds) between the movements of humans and "The Augmented Man". Direct

† e-mail:tpsik@pop.tuwien.ac.at
‡ e-mail:Matkovic@VRVis.at
§ e-mail:sainitzer@imagination.at
¶ e-mail:paolo@ai.univie.ac.at
‖ e-mail:szalavari@imagination.at

and immediate interaction was not possible, but reflections about time and space came to the minds of the visitors. The main difference between "The Augmented Man" and "The Invisible Person" is that "The Augmented Man" lives in the near past while "The Invisible Person" populates the Present.

For the original installation in the TMW a child-like virtual character was designed. An artificial intelligence (AI) module controlled the behavior of the virtual character forming IP. The IP installation allows real-time interaction. Vistitors movements are displayed immediately and interaction with the IP is direct and instant. This real-time requirement puts hard constraints on the realization of the system and even more on the design of the AI that has to react within just a moment. The reaction time of the IP is about one second, which is not disturbing because of the childish look of the character. This look also helps to reduce the expectations about the complexity of the interaction.

An important feature of all characters living in virtual environments is their "intelligence". There is a lot of research done on modeling emotional and inteligent virtual characters populating virtual environments. Petta et al.[11], ALIVE system[4], and Tu and Terzopoulos[13], describe the emotional model of virtual characters. The AI-module was improved for the new installation. The description of the new model is beyond the scope of this paper. We concentrate on the interaction between user and IP. The new emotional model of the IP will be described in a forthcoming paper.

The installation was a great success and has been a big attraction for children and grown ups. More than 200 000 visitors have interacted with the IP since the original installation was setup in September 1999. In this paper we describe a new user-IP interaction scheme, which makes it possible for users to step beyond simple interaction, and even play games with the IP. None of the common input devices like keyboards, mouse or trackballs are used. The only input device is the users body and their presence in the environment. Additionally, the actual application will be described, in order to give the reader a brief insight into the complexity of such an interactive real-time immersive virtual environment installation.

2. Main Idea

With the experience gained from the first installation we wanted to enhance the interface of the previous system in the TMW. Adding some new forms of advanced interaction was our main goal. The system in the TMW was realized to allow direct communication between an artificial intelligent (AI) and humans without the need of custom input devices. The enhancement of the interaction should use only the available interface. To find out what an "advanced interaction" really means we classified the interaction schemes,that are provided by available immersive virtual environments (IVE).

2.1. Classification of Interactions

To illustrate the difference to other systems we will introduce a classification for interaction schemes that are provided in installations like the one we describe. To provide a deeper understanding of our proposed classification we will give some examples.

The simplest way of interaction is situation-based reaction. As the AI perceives a specific situation every user action is replied to with a meaningful reaction. Under certain circumstances this leads to a new interaction scheme. We will describe an example of a simple interaction the greeting situation, that was used in the first system. When a new user enters the stage the IP would display a greeting gesture. This is a direct reaction to the user entering the stage. After that a new interaction situation will evolve.

A more sophisticated interaction scheme is needed for meaningful communication. For that purpose more specific situations can be introduced. Certain user actions are considered and special interaction situations are added to the repertoire of the AI. By adding more situation steps, and therefore generating longer interaction patterns, the interaction scheme becomes more complex. Thus, by connecting several simple interaction schemes, a higher level of interaction can be realized. The greeting situation can be used to give a good example for such a medium interaction scheme: as the user enters the stage the IP greets him, but then the interaction is prolonged. The IP waits for the user to also greet him back, to detect a response on his reaction (new situation). If this is the case before the timeout is reached, the user is rewarded with some happy gestures from the IP (another situation). There are two drawbacks in this approach. Generalized interaction situations are more reusable than specific situations, which only make sense under certain conditions. And therefore much of the design work has to be invested into interaction schemes, which will not be used much of the time. Additionally, if the users do not understand the specific interaction schemes implemented, a communication cannot be established.

When trying to solve even more complex interaction patterns this approach will not work. Games serve as a good example for this advanced interaction scheme. E.g. when playing "TicTacToe", both players are allowed to make their marks on a game board. But some additional rules apply while the game continues. Each player has to wait until the other has made his decision and no one is allowed to mark a marked field. Additional rules are added as "meta" rules are considered such as timeouts, game won, no place left etc. Such an interaction scheme generates too many situation-reaction combinations to be solved using a combination of simple interaction schemes.

2.1.1. Interaction by Playing Games.

As the original system proved to be a well-working environment supplying human-computer interface with hidden

Figure 1: *Pictures of the installation displaying the working scheme of the virtual mirror. Displaying the IP and the users*

computers, enhancement of the system by adding more advanced interaction was desired.

From the little fieldwork we had done, we learned that in general the cognition of the interaction situation and the internal state of the IP is not an easy task for the users. Because the reaction of the IP is based on the situation, the internal and emotional state of the IP and on the action of the user itself, the reaction of the IP was often not understood by the users. When adding even more complexity to the interaction the confusion of the users was expected to grow. Therefore we decided to use the well-known interaction situation of game playing, so that the situation and the meaning of the users' actions are easily understandable. The only unknown issue left is the internal state of the IP. We tried to make it easier to understand for the users by enhancing the animation repertoire and developing a better emotional texture.

We found that the minimum interaction tools players must have for common games are selection and mark. Thus all players must be able to select a game field. After that selection is done, a "marking" action referred to by us as "click" must be possible. Both interaction tools must be very reliable to encourage the users to play and not frustrate themselfs with a non-working interface. We had to introduce a new graphical element to the system serving as an interface for the game-relevant communication between the IP and the users.

2.1.2. Game-pads.

Bowman and Hodges[2] proposed that the guidelines of Norman[3] should be applied to interaction objects in virtual environments (VE). The game-pads we have used in the installation satisfy all four criteria which are: affordance, feedback, constraints and good mappings.

To satisfy the affordance criteria, the object must be able to inform the user of the way it can be used. Usage of the game-pads is demonstrated by the IP when it makes the first move. The IP uses the same posture that is registered by the system for a user "click". Repeating the gesture, if the user is not marking a field within a certain time and giving additional hints for usage. Feedback is realized by highlighting the game-pads when players moves over or stand on them. They change both color and shape as they are highlighted or marked, displaying a clear difference in their state. Constraints refer to the limitations on the use of the objects that help the user to use them in a proper way. As the game-pads do not give any feedback to players who are not allowed to make a selection, this supports the users in understanding that they have to wait until the IP has made his mark. Good mapping requires that the conceptual model, on which an object is based, is easily understood in the specific usage model. The game-pads are based on an easily understandable metaphor of some floor element that is sensitive to people standing on or moving over it, as they exist e.g. in underground stations by the escalators.

For the enhanced system in the TMW we came up with two different types of games. The first type deals with games like "TicTacToe" or "Senso" that allow only one player to be active at a time. The second type is more complex, by allowing all participating parties to select pads at the same time. We found a generic representation of the information that is needed for the AI to play with the users.

3. Application

The original system described by Petta et al.[11] was hosted by the TMW from September 1999 to June 2001. During that time the IP installation had great success and was visited by approx. 200.000 users. As a new part of the museum was opened for the public, a redesign of the installation was planned. The board of the museum wanted the installation to be developed further in the direction stated by the first prototype, enhancing the interface and gaining a more "playful" experience.

To enhance the system some major changes were implemented. First of all, two instead of one camera were used to detect the movement of the users. The visual appearance of the virtual character was enhanced by introducing single mesh animations and the emotion texture was redesigned for easier perception of the emotional state of the IP. Based on the better quality of the vision system which means better information about the people on stage, a working 3D occlusion could now be implemented in the system.

A PC was added to the system to host a web-server. This web-server generates a live video-stream (rtsp://ip.tmw.at/encoder/live.rm), so that internet visitors are able to see what is happening on the stage. In addition the web-server holds an archive of the pictures that were taken by the IP. The pictures are now available in a internet archive to prolong the interaction experience to more than a few minutes. Since October 2001 the installation of the new "Invisible Person" has been up and running.

3.1. General Description

The system is divided into four modules. The vision-module analyzes two live camera video streams and generates information blocks which are used by the AI to make decisions about new interactions. The AI-module decides what animations the virtual character should display. The game-module holds all information and algorithms about the games, detaching the game-knowledge from the emotional and interaction algorithms of the AI. The render-module displays the animations and the additional graphics needed for the games, additionally the rendering of the occlusion is performed by this module.

These modules communicate over a 100 MBit network connection. The TCP/IP protocol was chosen to establish reliable communication. Different internal protocols where designed to allow compact and lean information-flow.

The vision-module delivers information about the user's position and current posture to the AI and the game module. The render-module is supplied with occlusion-masks and depth information.

The AI-module sends control information to the game-module and advises the render-module to display the appropriate animations and emotion-textures. Together with the virtual character from the render-module the AI forms a symbiosis of body and mind: "The Invisible Person".

The game-module provides information about the game status for the AI. The graphic elements of the games are controlled by communicating with the render-module. As the games and their specific rules must be separated from the implementation of an artificial intelligence, the game algorithms were implemented in a separate module. To keep communication straight and simple, a generic representation for the game information communication was developed.

The render-module sends position and posture of the virtual character to both the AI-module and the game-module. The game-module and the AI-module need this information to merge the user input - from the vision-module - and the character input - from the render-module - to get an overall view of the interaction.

We will now describe each module in detail to give a deeper insight into the interfaces and scope of each module.

3.2. Game-module

Because the game-module was realized as an independent module, careful consideration of the interfaces was required. The inputs for the game-module are position and posture data of the users from the vision system, the position and action of the virtual character from the render-module and control messages from the AI. As the information had to be distributed via a network protocol, the interfaces had to be general and compact. The AI has control over the game-module via the control-interface. This interface allows the AI to start and stop a game. Additionally, several parameters of the game can be set, such as the difficulty level and which player has the first move.

The outputs from the game-modules are game-status information including overall game score, a game history and end causes. This information is processed by the AI to update the emotional model. Possible moves for both the users and the IP were distributed by specifying regions (2D boxes) and a rating for these regions. A timeout was added to enable time-critical games (such as volleyball). If a player has to wait, no moves are available. As both user and IP moves can be set at the same time, real-time games that allow both parties to play can be realized.

Avoid regions for the IP are supplied to give the AI a hint about the game-board, and when to avoid it. This is necessary to let the user have a look at the game-board and choose his game-pad without beeing disturbed by the IP. The game module does not interfere with the AI system itself, that is, the game-module does not make any decisions. It simply supplies the AI with information so that the AI can decide what to do.

For the render-module a different interface was needed to ensure the correct visualization of the game-pads and other graphical gimmicks.

3.2.1. Adding Games.

We wanted the systems to be independent, to be able to add more games later, without needing to change the AI system, so the interface between AI, game-module and render-module had to be general and versatile. Although the AI decides when and what type of game to start, selection between the different games available is done by the game-module.

Thus more games can be added without changing the AI

system. Our abstract interface allows the creation of very different games. As the information contains only regions, ratings and timeouts, games without game-pads are also possible. As a matter of fact photo-shooting - the IP can make pictures of the users, if it likes - was also implemented as game.

3.3. Artificial Intelligence

The description of the whole AI-module is beyond the scope of this paper. The module is described by Novak [9] as part of her master-thesis. We will only go into detail about the game-part within the AI-module.

Although the control of the game itself is done by the game-module the "meta" control of the game-module is performed by the AI. That means that the AI decides when and what type of game it wants to play. It may also decide to close a game that is underway. This can have several reasons, like the other player leaves, or the AI has a "very bad day" and just loses interest in the game. What is important is that the AI has full control and the game-module just delivers information, but does not make any decisions.

The AI can decide if it wants to win or lose the game. When a game is active the AI has several possibilities of playing the game, depending on the information from the game-module. The IP can walk into a game-region and display a specified action, if there are no regions available, the "avoid region" tells the AI which region on the game-board should be avoided. So the AI can decide to leave and wait outside the game-board. Because the user can choose to end a game just by walking out of the stage, IP can also end the game whenever it wants.

The region and rating-interface permits many different games to be implemented. The IP just walks into a certain region, that it freely chooses from the regions defined by the game-module. This can be even be a region that has a very bad rating for the IP. Then it displays a specified action defined by the game-module. This action has a specific meaning in the game context, for most games this is the "click" posture to mark a game-pad, but this interface has also been used to display the gesture of pressing the button on the camera to make a picture. Using such a general representation of the game moves allows us to add more games without changing the algorithms in the AI-module.

The rating information for the user region is supplied to the IP to enable it to give a feedback while it waits for the user to move. Depending on its internal state it can encourage the users with appropriate gestures to make good or bad moves.

3.4. Vision

The main task of the implemented vision system is to find the silhouettes of users seen from the front, to estimate the users' position and posture on the stage. The silhouettes from the front camera are needed to implement efficient occlusion in rendering. The position on the stage is used to add depth information (needed for occlusion) to the occlusion masks and to supply the AI with the users' position. This information is also used by the AI to avoid collisions with the users. The posture recognition is necessary for user interaction with the system[5]. As stated above, selection and click are necessary to implement user interaction within our games. Our idea is to use an embedded interface where user position will be used for selection, and a predefined posture that is used for the click. The posture chosen is "two arms spread apart", since this posture showed to be the most reliable one.

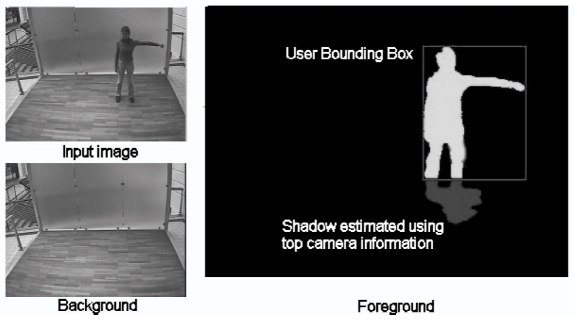

Figure 2: *Vision module extracts the foreground pixels from the input image, and estimates the shadow using the top camera information*

The vision system runs on a dual Pentium PC running the Linux operating system with two video grabber cards and two cameras. The first camera is the same one used for the output video, and it is placed in front of the stage, just above the display screen. The second camera is positioned above the stage, pointing downwards, and this camera is used by the vision system only. As stated above, the whole system is intended to run in a publicly accessible museum hall, which has a frosted glass roof. Due to the frosted glass roof the lighting conditions change practically all the time. These changing lighting conditions, the demand to design a stable and robust system for operation in a publicly accessible place and the demand to design a real-time system were the three most challenging requests to the vision module.

3.4.1. Basic Vision.

The basic vision is implemented using the OpenCV library from Intel[10]. The basic task is to determine which pixels belong to the foreground (user) and which to the background (stage). The basic principle used is background extraction, which is a common technique in people-tracking systems. The idea is to build a background image using a running average of the input images. The current image is then compared to the estimated background image. The pixels where the difference is greater than a certain threshold are consid-

ered to belong to the foreground. The threshold is estimated per pixel using the standard deviation of its brightness distribution over the last 200 frames.

Once the pixels are classified, the background image is updated using different update rates for background (faster update) and foreground (slower update) pixels. The foreground updating is necessary due to constantly changing lighting conditions. The foreground pixels contribute to the background as well, in order to make the system more robust in case of error. If a sudden change of lighting is interpreted as a foreground object it will become part of the background, and will not remain foreground "forever". On the other hand, if a user stands still for a long period of time, he or she will become part of the background, and once he/she moves there will be two foreground objects.

The basic vision described above is a standard technique [7]. We added some new features to the vision system in order to fulfill our requirements.

3.4.2. Advanced Vision.

Due to constant lighting changes and large range of possible lighting levels, the camera iris system must adjust itself automatically. If a sudden lighting change occurs, for example a cloud is passing by and covers the sun, the pixel intensity in the background pixels can change so much that system assumes the whole background to be foreground. In order to overcome this problem, a kind of software iris balance is implemented. The light is measured on five spots, which are unlikely to be covered by the users, and the current image is made darker or brighter depending on the readings in these measuring areas.

The vision system runs parallel for each camera, using the algorithm described above. When the mask images from the front camera and from the top camera are extracted, both mask images are used to compute user masks as seen from front camera and user positions on the stage. The original front camera mask very often contains the shadow. The shadow can be cut out using the information from the top camera. The user position is known, and the floor level in the front mask can be computed using the user position information. Figure 2 illustrates the result of estimating the background, the foreground and shadow in the foreground image. Furthermore, the top camera information is used to divide the front blob in cases where two users are standing one beside the other from the front camera's point of view, but they are actually standing apart (one behind and beside another). Once the mask images are merged, a new final mask image is produced containing masks with depth information. These masks are used for occlusion and posture recognition.

3.4.3. Posture Recognition.

Besides the depth information for the masks, a posture is assigned to each mask as well. There is a predefined set of postures from which one (or "none" if the posture is not recognized) is chosen and added to the mask. The bounding rectangle of the mask is divided into 25 sub-rectangles, and a fill ratio is computed for each sub-rectangle. The fill ratio is the ratio of foreground pixels and total pixel count in a sub-rectangle. These fill ratios are compared with the fill ratios of predefined posture masks, and the posture with the smallest mean square error is considered to be the right one. If the mean square error of the closest match exceeds the maximum allowed threshold, the posture is considered to be unrecognized. The system recognizes postures such as left arm stretched out, right arm stretched out, both arms stretched out, left leg stretched out, etc.

3.5. Rendering

The hardware used for the rendering system is a SGI Octane with video option and some additional graphic processors for texturing and advanced rendering schemes. This graphic stream is the only output for the users of the system. It consists of a video background, using the video stream from the front camera, the IP and the occlusion masks. The main part of the rendering bandwidth was used for the realtime video stream of the background to display the "virtual mirror" environment. Then the virtual character is rendered using a single mesh and a hardware-generated texture. At last the occlusion is added to the scene. The system was tuned to produce a realtime (approximately 15 frames per second) graphic. The rendering module was implemented using the "SGI Open Inventor" library shipped with the SGI Octane System. This allows the use of a scene graph and triggers an OpenGL rendering.

3.5.1. Occlusion.

A camera located above the screen records the users on stage. The captured pictures are superimposed by the computer-generated-virtual character and displayed on the screen. Simply adding the virtual character is not enough. The virtual character can be (at least partly) occluded by persons on stage. Ignoring the occlusion would lead to a visually inconsistent scene, which is very confusing for users.

We have chosen a simple but fast occlusion-culling algorithm, the accuracy of which is sufficient for our needs. The algorithm assumes that the users' hands or legs are nearly in line with the body. A flat silhouette approximates the representation of each recognized user. The silhouette is located at the measured position of the user and is oriented parallel to the screen. All the space behind the silhouette is marked as occupied by the user. After that, the virtual character and additional geometry are displayed. Because all the space behind the recognized users is marked as occupied, parts of the computer-generated geometry lying behind a user will not be drawn. Although this is a simple approach, it significantly enhances the visual quality of the picture (see Figure 6).

The vision module is responsible for computing the positions and silhouettes of the persons on stage. This information is already needed for the artificial intelligence module and for posture recognition. So there is no additional load for the vision module. The silhouettes as well as the positions are sent to the rendering module via a network protocol.

3.5.2. Single Mesh.

The original system used rigid body animation for the virtual character. The visual appearance was enhanced by introducing a system of bones and skin. The bones are linked together with joints and form the "skeleton" of the virtual character, which is basically a hierarchy of rotation and translation operations. A single mesh of polygons and vertices defines the skin of the character. This mesh is defined using neutral angles of the joints.

The position of each vertex of the mesh is influenced by one or two bones. Most vertices are influenced only by one specific bone. For example a vertex, which is located at the middle of the forearm of the character keeps its relative position to the bone. A vertex located at the elbow is influenced by the bone of the upper arm and the forearm. Weights specify the degree of influence of the two bones. The resulting position of the vertex is a weighted average of the influences.

In this way the mesh is deformed according to the current joint angles of the defined skeleton. By using proper weights, the skin in the range of joints is deformed smoothly and realistically. Compared to the former rigid body animation, the animation of the virtual character using bones and skin significantly enhances the appearance and the naturalness of movements.

3.5.3. Game-board and Game-pads.

The Game-pads were designed to have enough information to function independently of the rest of the system. No supervision is necessary, the game pads will highlight or even change their state if any player interacts with them. The game-board has knowledge about the connection between the pads and it also distributes information from the game to the pads (active player, initialize). Most of the game-boards realized used the Game-pads.

4. Performance

As described above we have implemented a real-time system. To enable real-time interaction a system had to be developed that recognizes user actions and replies to these actions in a short period of time. As most of the described user actions can only be detected at the end of the interaction (like "double click" can only be detected at the end of the second click) some delay in the response is system immanent. The vision-module updates the information about the people on the stage with a mean rate of 10 information blocks per second. The render-module is able to produce about 15-18 fps (depending on the complexity of the occlusion masks). The emotion texture is refreshed every frame, this is possible by using palette cycling to generate the motion and color change effects. The time slots for the AI-module depend on the length of the chosen animations. As we want to display natural movements the AI-module is not allowed to interrupt an ongoing animation. For that reason only very short animations (from 0,4 seconds to 1,2 seconds) are used most of the time.

This means that the IP is able to react to a user movement or action within 0,4 to 1,4 seconds. A collision between a user and the IP (meaning that they occupy the same "virtual" place on stage) can not be avoided at all times, but as the character was modeled with a childish look the reaction time is understandable for the users.

The visual feedback of the game-pads is shorter though. As soon as a person (including the IP) moves over a game-pad it changes its color and shape (which means, depending on the frame rate of the render-module and the detection rate of the vision-module, about 0,1 seconds). The reaction to a click action is determined by the detection of the user action (normally about 0,6 seconds after rising both arms while standing on a game-pad).

To enhance the performance of the system and reduce the time lag between detection of a user movement and reaction some improvements could be thought of. The frame rate of the render-module should be raised to 25-30 fps simply by using newer hardware. Some method of interrupting ongoing animations have to be found, or existing animations could still be reduced or divided into more (and therefore shorter) parts. And finally the update rate of the vision-module could be enhanced by using newer hardware and increasing the CPU power.

5. Conclusion

We have implemented and installed an advanced user interface in an immersive virtual environment accessible for a large public. Some extra work had to be done to develop the lab prototype to be functional in a public environment. By utilizing well-known metaphors and analogies the user interface was understood by almost all users. We have shown that it is possible to implement an advanced interaction scheme in an immersive virtual environment, which is well perceived by everyday users, satisfying all four interface design requirements described by Bowman and Hodges[2] and realizing a stable and reliable setup to simulate a "mouse input" without requiring the user to handle any input device. The idea of using this advanced interface as an input for games originated from the environment in which the system was installed. The "playful" interaction between technology and humans is a widely used way to transport knowledge in a museum. For modern museums the "interaction" between

exhibited objects and visitors is a way of getting people to be really involved with the themes. Especially school classes populated the system with great exitement during their visit to the museum.

The public accepted the installation and tried to communicate with the IP with anticipation. Most of the users' cognitive power is absorbed by the IA and its effort to communicate with the users. The actual interface to the game is not the main intellectual task for the users. Most of them try to understand what the IA has to tell them. By giving the users an understandable metaphor of "someone" living in the mirror, they do not concentrate on learning the usage of the interface. So the simplest learning scheme "imitation" is performed by nearly all visitors. The hardest task for the users is to recognize that they are playing a game. Once users made that cognitive step, the interfaces were well understood.

The most difficult part was developing a vision system which would function in a non-controllable light environment.

6. Future Work

Now that the system has been proved to be a good approach to the difficult task of user interfaces in an immersive virtual environment, more work has to be done on ethnological studies. More insight of how different users perceive the interface and how understandable the interaction scheme is for them could be gained from investigating and recording. This should give some input on how to improve the system.

It is planned to add more games to enlarge the variety of the system. Games like "memory", "football" or some sort of "volleyball" could easily be implemented. Even more advanced games like "chess" are possible.

6.1. Generic Gameboard.

Using this installation without the IP would allow its use for interaction between human players. New forms of gameboards could be developed or already available ones could be used. Any place that is equipped with the hardware would become a generic game-board. Although this idea seems to be challenging, arguments that justify such an enormous technical effort for "just" playing normal board-games like "chess" have yet to be found.

If you make a trip to Vienna, do not forget to visit the "Technisches Museum Wien" - Vienna Museum of Technology and the "The Invisible Person" installation. Just say hello and play a game with him and maybe, if you let him win, you will get a picture of yourself standing on the stage communicating with an artificial intelligence.

Acknowledgements

The authors thank Vienna Museum of Technology for making it possible to realize this project and for their help in developing the concept of the installation. Furthermore, we thank Gudrun Novak for her help in implementing the software. This work was partly sponsored by the project Atelier - IST-2001-33064, which is part of the Future and Emerging Technologies (FET) activity of the Information Society Technologies (IST) research program funded by the EU. The Atelier project is also part of the Disappearing Computer initiative.Parts of this work were carried out in the scope of applied research at the VRVis Research Center in Vienna (`http://www.VRVis.at/`), Austria, which is funded by an Austrian governmental research program called K plus. The Austrian Research Institute for Artificial Intelligence acknowledges basic financial support by the Austrian Federal Ministry for Education, Science, and Culture.

References

1. K.Ch. Posch and D.W. Fellner. The Circle-Brush Algorithm. *Transactions on Graphics*, **18**(1):1–24, 1989.
2. Bowman, D. A., and Hodges, L. F. User interface constraints for immersive virtual environment applications. *Tech. Rep. 95-26*, 1995
3. D., Norman *The Design of Everyday Things*. Doubleday, Ney York ,1990.
4. P. Maes, B. Blumberg, T. Darrel, and A. Pentland. The alive system: Full-body interaction with animated autonomous agents. *ACM Multimedia Systems*, **5**:105-112, 1997.
5. W. Freeman, D. B. Anderson, P. A. Beardsley, C. N. Dodges, M. Roth, C. D. Weissman, W. S. Yerazunis, H. Kage, K. Kyuma, Y. Miyake, and K. Tanaka. Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications*, **18**(3):42-53, 1998.
6. L. Emering, R. Boulic, S. Balcisoy, and D. Thalmann. Real-time interactions with virtual agents driven by human action identification. *First ACM Conf. on Autonomous Agents'97*, 1997.
7. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence* **19**(7):780-785, 1997.
8. , C. Wren, et. al., Perceptive Spaces for Performance and Entertainment: Untethered Interaction using Computer Vision and Audition, *Applied Artificial Intelligence*, **11**(4):267–284, June 1997.
9. Novak, G. *Inside Invisible Person 2, Robustes und kompetentes Interaktionsverhalten eines synthetischen Agenten in einem öffentlichen Exponat*. Master's thesis, University of Vienna, 2002.
10. Opencv–Intel Open Source Compter Vision Library, http://www.intel.com/research/mrl/research/opencv/.
11. Petta, P., Staller, A., Trappl, R., Mantler, S., Szalavari,

Z., Psik, T., and Gervautz, M. Towards Engaging Full-body Interaction. *Proceedings of the 8th International Conference on Human-Computer Interaction (HCI International '99)*, 1999.

12. Stricker, D., Fröhlich, T., and Söller-Eckert, C. The Augmented Man. *Proceedings of IEEE and ACM International Symposium on Augmented Reality 2000*, 30–36, 2000.

13. Tu, X., and Terzopoulos, D. Artificial fishes: Physics, locomotion, perception, behavior. *Computer Graphics, Annual Conference Series*, **28**:43–50, 1994

Figure 3: *Pictures of the installation displaying the working scheme of the virtual mirror. Displaying the IP and the users*

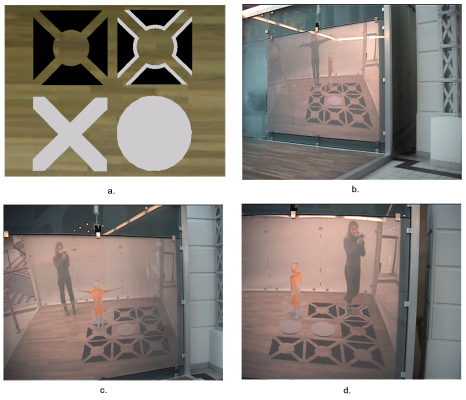

Figure 4: *a. Game pads shoving default pad, selected pad, clicked pads; b. User has selected a pad and just makes a click; c. IP makes a click; d. The user selects next pad*

Figure 5: *IP holding the camera and taking a photo of visitors*

Figure 6: *Examples from the installation, showing the occlusion effect*