

The Mobile Animator: Interactive Character Animation in Collaborative Virtual Environments

Mario Gutiérrez, Frédéric Vexo, Daniel Thalmann

Virtual Reality Lab (VRlab)
Swiss Federal Institute of Technology Lausanne (EPFL)
Lausanne Switzerland CH-1015
{Mario.Gutierrez, Frederic.Vexo, Daniel.Thalmann}@epfl.ch

Abstract

We have designed a mobile -PDA-based- interface for real-time control of virtual characters in multiuser semi-immersive Virtual Environments -using a large rear-projection screen. The proof-of-concept implementation we present shows the potential of handheld devices as powerful interfaces to Virtual Reality applications. This technique eliminates the display of floating menus and other widgets over the simulation screen. A brief discussion on the advantages and disadvantages of using a handheld for 3D interaction is presented as well.

1. Introduction: Interactive control of objects and characters in VE applications

It is a well-known fact that the main objective of a VE application is to immerse the user into the simulation, there are several definitions for this concept, a simple and popular one is: "creating the illusion of being there". Immersion requires different technologies to create a multisensorial representation of the VE and at the same time provide a way for the user to communicate -interact- within the virtual world. Depending on the technology applied, we can be fully or partially immersed into the VE.

Full immersion is typically achieved by means of Head Mounted Displays (HMD). The semi-immersive approach consists in positioning one or more users in front of a large rear-projection screen displaying the virtual world. Stereo glasses and 3D surround sound enhance the experience.

The problem with full-immersion is that the HMD isolates the user from the real world and makes it difficult to share the experience and communicate with other people. The limited field of view leads to spacial orientation prob-

lems and motion sickness : the cyber-sickness effect [21], [26].

Research has been aimed at tackling the spacial orientation and other related problems [5]. However, after several years of experimentation, the semi-immersive approach has gained popularity due to the possibilities it provides for direct interaction and communication between users. It has been observed that a semi-immersive VE improves the overall user experience [23]. Moreover, this kind of systems can be used as collaborative tools by the design and development teams, making the implementation of a VE application more interactive and effective. At this point is important to clarify that we consider the term immersion in a broader sense than the "classical" VR approach. In this work we talk about "immersion in the task". We aim at creating an environment that eases the control and setup of virtual scenes letting several users work simultaneously.

At VRlab we have tested the semi-immersive approach with different interfaces. The interaction mechanisms inside a VE are also a key component to create the "illusion of being there", the sense of presence [28]. Interaction includes activities like changing the point of view (camera management), navigating through the 3D world, and performing specific actions involving objects and/or virtual characters -either autonomous or user-controlled avatars.

In the IST-JUST European project [25] -a VE for health emergency decision training- head orientation tracking with a magnetic sensor for camera management and navigation was used in combination with voice commands to interact with semi-autonomous characters. The sensor on the head revealed to be uncomfortable, concerning the voice commands, the intervention of a human operator was required to ensure the correct performance of the system.

In a later work, we implemented a multimodal interface called "The Magic Wand" [9], we replaced the head tracker by a less cumbersome device, similar to a 3D mouse. We continued to use voice commands, since they are a very in-

tuitive way for selecting among preset actions. The problem of speech recognition was partially solved by limiting the number of recognizable phrases to a small single-word vocabulary.

However, this interface does not perform so well in more open scenarios, due to the inherent difficulties of natural language processing. In effect, some situations may require full control of the virtual character, e.g. there are games where the characters must adopt a very specific pose in order to reach the goal.

For instance, in "The Enigma of the Sphinx" [2] -a VR game developed to demonstrate the use of the "magic wand" interface- the user must mimic the postures displayed by one virtual character and also direct another one to adopt a particular pose [1]. The problem was solved with the combination of the 3D pointing device and a set of voice commands (the names of the body parts to be moved). While this can be entertaining, is not always the best way to specify a detailed character posture or execute more complex tasks.



Figure 1. User interacting on a semi-immersive VE using "The Magic Wand".

Interactive simulations require fine-tuning of the character's posture, in particular at design time. As we have mentioned, a semi-immersive VE can be used since the early stages of development for experimenting and observing the effects of the VE on the test users -the designers themselves and the rest of the development team.

Our objective was to provide a low-level control -joint-wise- of the characters in a multiuser Virtual Environment. For instance, one user could modify the posture of the upper body while another one adjusts the position and orientation of the same character, the changes are immediately reflected on the projection screen.

Each user would need access to an independent view and a way to interact with. The problem of interaction with objects in VEs has been approached by means of 3D/2D widgets and other interaction techniques: [10], [29], [4]. With this work we start exploring a new alternative based on handheld devices or PDAs (Personal Digital Assistant). One of main advantages of this approach is the absence of floating menus and other user interface (UI) components over the simulation screen. Several interaction techniques have been proposed to avoid superposing UI components: [24], [4]; but they have been implemented in the context of full-immersion applications and are not suitable for our semi-immersive system.

In the following section we justify our choice of handheld devices as the hardware platform for our animation interface. The remaining sections will describe the system in detail -data representation, interface, architecture. The paper ends with a brief discussion of our results and an overview of the further improvements of this work.

2. Handheld devices as interfaces to virtual environments

Handheld devices have attracted the attention of researchers in the field of Virtual Reality even before the required hardware were available.

One of the first studies aimed at using a mobile device as an interface to a VE is the work of Fitzmaurice et. al. [14]. Actually, the authors implemented their own palmtop device¹. A 4-inch color monitor with an attached 6D input device was used to navigate and interact through the virtual world. No additional display was used, the small monitor was the only "window" into the VE. The experiment showed the feasibility of using a handheld display in a VE and its advantages: user mobility, intuitive way to interact with the computer. However by that time (1993) the available offer of true handhelds was very limited and there was no hardware capable to communicate with the main system and display an independent interface -their system was physically connected to a workstation. Besides, this experiment was more oriented to replace normal size monitors by small screens for navigation and interaction. It was different from our concept of an interaction tool for collaboration in an immersive VE.

The work of Watsen, Darken and Capps [27] was one of the first implementations of a true handheld-based interface to a Virtual Environment. The authors implemented a 2D interface on a 3Com PalmPilot© to be used while standing inside a CAVE system. Basic scene and object manipulation were done through slider controls on the handheld.

¹ The term palmtop is considered here as equivalent to "handheld": a device that can be held in the palm of the hand

While this prototype confirmed the benefits of a mobile interface to VE, the scenario where they applied it -CAVE-opposed some problems to the use of a handheld: being a full-immersion system, the CAVE made it difficult for the user to switch between the surrounding 3D world and the handheld.

More recent research has continuously focused on the implementation of 2D GUIs for basic interaction: camera control and scene management (object picking, navigation through a simplified representation of the 3D environment) [18], [12]. Emphasis has been made on the portability and mobility of the GUI itself. Some other problems starting to be targeted are peer-to-peer communications between handhelds, together with client-server (handheld-main graphic workstation) interactions to control the VE and interact with other users at the same time [13].

Other authors still consider the PDA as a virtual viewer. In the Touch-Space system [8] the PDA is optically tracked and used to display virtual objects superposed on the real world.

The PDA may serve as a tool to reinforce human-human contact in a virtual environment. Indeed, this is a very important point to consider, apart from immersing the users into the simulation, we want them to keep in contact with each other to share their impressions and work together.

The missing component was an efficient interface to give full control of the characters and other objects in the scene. The research work we are aware of has always focused on basic manipulation of 3D objects, excluding articulated characters: translation, rotation and camera management by means of 2D controls.

We decided to go one step forward and created a 3D graphical representation of the characters and objects in the Virtual Environment right into the handheld. We believe the best way to interact with a virtual object is through a 3D representation controllable by virtual manipulators. Similar to the animation tools included in commercial packages like Maya [22] or 3DS MAX [1]. This is why we called our interface "the mobile animator", we tried to mimic the well tested interfaces used by professional animators and adapted them to a mobile device.

The following section describes our handheld interface in detail: the object representation, the virtual manipulators, the data coding/transmission and the system architecture.

3. Implementing the mobile animator

3.1. Object representation

The fundamental assumption we have made when designing the mobile animator concerns the interface. We decided to use a 3D representation of the objects in the handheld display instead of 2D controls -sliders, buttons, menus-

because this seems to be the most intuitive way to have the low level control we are looking for.

This choice is justified by the confirmed efficacy of the commercial animation tools. Professional animators manipulate characters and objects through virtual 3D manipulators, see figure 2. This interface can be more intuitive than a 2D GUI, in particular when we have direct feedback on a high resolution 3D projection. In fact, for this first prototype we targeted the manipulation of virtual characters only.

The next question was about the level of detail of such 3D representation. The immediate answer could be to reproduce with the same resolution and detail the characters as they are rendered in the VE. This option has been discarded for several reasons.

First, the computing power available on the currently most advanced PDA -Pocket PC using the Intel XScale©microprocessor- is not enough to render the high resolution geometries we project on the large screen.

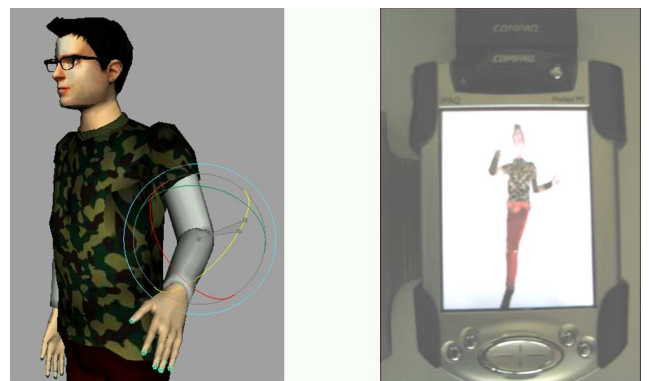


Figure 2. Virtual manipulator for joint orientation and high resolution character on IPAQ©.

We have implemented a testbed application to visualize the same virtual characters used in our VE applications - articulated characters with an average size of 15k polygons- in an IPAQ PocketPC© [16], see figure 2. We didn't obtain an interactive frame rate, our best results were about 5 frames per second, not enough for real-time manipulation.

Moreover, even if the hardware performance is expected to increase in the next months/years, the screen of the PDA is too small to let the user appreciate realistic 3D objects. Augmenting the resolution of such small displays won't help too much, and increasing the size wouldn't make sense because those devices would be transformed into Tablet or Notebook PC's, impossible to hold them on one hand.

Here it is pertinent to explain the way we model the virtual characters we work with (virtual humans, most of them). Indeed we use the standard representation defined by

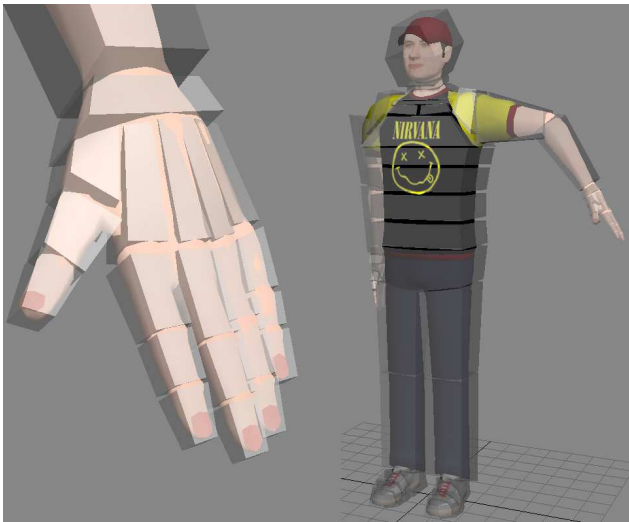


Figure 3. Bounding-box representation on Bones Pro©.

the MPEG-4 FBA specification [6], [19], [20]. The foundation of this is the hierarchical structure or skeleton used to animate the virtual humans. The skeleton has been standardized by the H-Anim group [17] and adopted by MPEG. A full H-Anim skeleton contains close to 80 joints with their corresponding segments (associated geometry). The most complex parts due to the number of joints involved are the hands and the trunk.

The choice was to provide the mobile users with a simplified representation and let the realistic rendering for the large projection screen.

How simple should the representation be? Once more we took the professional tools as a reference. Bones Pro© [3] is a tool intensively used by our design team for character posing and animation. The bounding-box representation it uses to manipulate the 3D objects is rather common and intuitive, see figure 3.

There is another frequently used representation: a skeleton built by "balls and sticks". We discarded it because it doesn't give enough information about the actual dimensions of the character limbs.

The bounding boxes help avoiding collisions and impossible postures while manipulating the skeleton -even if no collision detection is performed, having the notion of the actual dimensions is welcomed. However, the reduced dimensions of the handheld's display still prevents us to make an exact copy of such representation for virtual characters.

To keep the representation simple we introduced different levels of detail. In the default level, the shoulders, trunk and hands are simplified. If the user needs fine control, the

simplified sections can be zoomed to access the full set of joints. The figure 4 shows the simple model compared to the detailed model of the upper trunk.

3.2. Interactive control of the characters

The graphical interface on the PDA allows the user to modify the posture of the character (joint orientations), the orientation of the character in the handheld display, and its position/orientation in the Virtual Environment, see figure 5. If the VE contains more than one virtual character, the interface presents a list of names to pickup the character to modify.

Each joint is fully modifiable through the stylus by pointing on the segment (bounding box) and dragging it to the desired position. Each joint has 3 degrees of freedom -Euler angles-, they are controlled by mapping the 2D coordinates of the stylus

In fact, only two rotation axes are modifiable at a time, the third one is accessible by dragging the stylus while pressing on a button configured by the user. The PDA has several built in buttons whose function is customizable.

In a similar way, the character can be viewed from different points of view, by pressing the button preset for camera management and dragging the stylus until the model is in the desired orientation. This makes more comfortable the manipulation of certain joints, difficult to access from the default orientation.

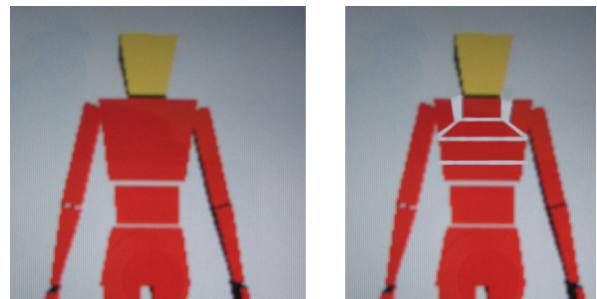


Figure 4. Simple model and a detailed view of the upper trunk.

While the user can modify at will the orientation of the character on the PDA screen, this parameter remains unchanged in the Virtual Environment. Only the changes on the joints are reflected. To modify the position/orientation in the virtual world a virtual manipulator is used in combination with the directional pad on the device.

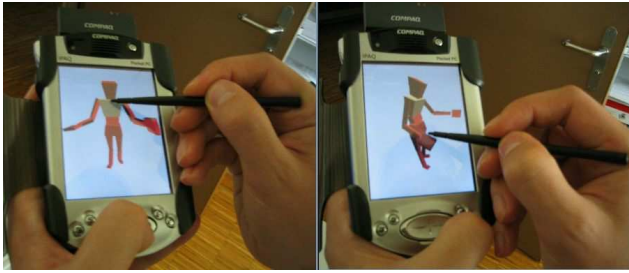


Figure 5. Interactive character control on the PDA: joint orientations and viewpoint.

3.3. System architecture

The Mobile animator is the front-end of a 4-layered system. As shown in the figure, the system architecture has 4 main components:

- **VE system:** software running the interactive Virtual Environment application to project on the large screen.
- **Network layer:** software component in charge of managing the communications between the Handheld device(s) and the Rendering System. This component can reside in the rendering workstation or on a different computer.
- **Character Data:** automatic digital model adaptation.
- **Handheld interface:** The user interface to the VE.

The VE system is an interactive 3D graphics application that receives information from the network layer to update the character's posture/position/orientation. It sends information concerning the characters in the scene for the network layer to adapt it and deliver it to the PDA.

The network layer contains a multi-threaded TCP sockets application able to manage concurrent connections between one or more PDAs and the rendering system.

The Character Data component is responsible of retrieving the character's definition from the VE application and delivering it to the PDA when requested. It queries the rendering system to get the data defining the virtual characters present in the VE: skeleton and associated geometry, position and orientation in the space. With this information it prepares the simplified bounding-box representation of each of them: it replaces the polygonal mesh of each character's segment (body part) by its corresponding bounding-box. The segments corresponding to the hands and trunk are simplified as described before.

In order to manage concurrent communications, the network layer blocks the access to a character on a per-parameter basis: One or more users can interact with the

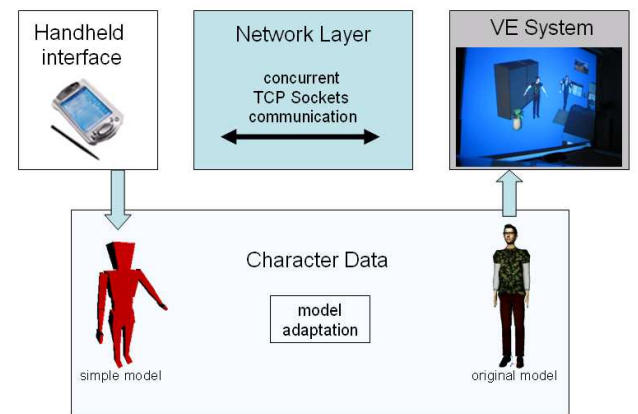


Figure 6. System Architecture.

same virtual character, but they are not allowed to modify the same joint or the position/orientation on the VE at the same time.

The Handheld interface communicates with the network layer to retrieve the list of available characters on the scene and sends the updated information generated by the user.

3.4. Message coding and transmission

Since we work with MPEG-4 characters, the natural way to codify the animation and send it to the VE would be by means of the MPEG-4 BAP (Body Animation Parameters) files.

A BAP file contains the euler angles to apply to each joint -up to 3 rotation angles- per animation frame. The BAP unit is defined as:

$$BAP = angleInRadians * 100000/\pi$$

It has the following structure:

BAP mask: 0 or 1 for each of the possible 296 BAPs indicating whether the BAP is used (masked or unmasked). The BAP mask precedes each BAP data line. BAP data description: each frame is described as one line of integer BAP values starting with the frame number. A 0 value for any BAP indicates the default position.

The following is an example of a bap file called new01.bap with a frame rate of 25 fps and 1 frame of animation for one joint:

```
3.2 new01.bap
25 1 1 0 0 0 ...
1542
```

MPEG-4 defines a compression algorithm [7], but it would require more computer power than the PDA can provide, considering that it is at the same time processing the 3D rendering, user interaction and network communications.

We could use BAP frames without compression but this would require sending more than 296 characters per frame, per joint, and this is inefficient. Since we modify only one joint at a time, we can simplify this by sending only the character ID, the joint ID and the euler angles for each frame, this reduces the required information to transmit.

Below is an example of an animation frame.

```
01 55 1500 1600 0
```

We are animating the joint 55 of the character with ID 01 and the euler angles to apply are the last 3 numbers.

In our first test we noticed a discontinuous animation on the PDA, the reason was the network communication. The handheld wasn't able to keep sending data and rendering the 3D model at an interactive frame rate. We decided to privilege the user interaction on the PDA by reducing the data transmission rate -sending 1 frame of animation for every 3 rendered frames. The problem is that the handheld does not send enough data and hence the animation on the VE application isn't as fluid as on the PDA.

To solve this problem we have two alternatives: interpolate between two animation frames, inducing a 3 frame latency; or extrapolating the data based on the already received information.

To ensure the animation accuracy we decided to interpolate. This is the best solution since the latency isn't noticed by the user while he's manipulating the character on the PDA, and for the rest of the collaborators, it isn't a problem.

The next section describes the implementation of our system.

4. System implementation and results

We have set up a multiuser Virtual Environment based on a large rear-projection screen. However, the system can be used with a conventional monitor as well. The VE application is based on a MPEG-4 animation engine programmed in Java and using OpenGL hardware acceleration [15]. The application allows navigating through a 3D scene inhabited by virtual humans. The system runs as a java applet with TCP sockets for communication with the network layer on a PC workstation with dual Intel® Xeon CPU at 1.7Ghz, an nVidia Quadro®4 graphics card and 1GB RAM.

The network communications management and simplification of the virtual characters is implemented as a java application running on the same PC.

The handheld interface is programmed in C++, the 3D graphics rendering is done by the DieselEngine® library [11]. The choice of DieselEngine was based on the flexibility of its API, which is similar to DirectX, and its overall performance. There is another popular graphics library for

Pocket PC devices: pocketGL, even if it is supposed to emulate basic OpenGL functionality, porting OpenGL code is not so transparent. This is not the case for DieselEngine, in fact this library is multi-platform and is better optimized for the two main CPU models found in the Pocket PC market: StrongARM® and XScale®. At the time we implemented this prototype pocketGL was optimized only for the [older] StrongARM® processor.

We have tested the application on two different models of the IPAQ Pocket PC®: 3850 (StrongARM® processor at 206Mhz) and 3970 (XScale® at 400Mhz), obtaining a frame rate of 18 and 27 frames per second, respectively. This improvement on the frame rate on the fastest CPU confirms that DieselEngine is able to take advantage of the new XScale architecture. According to our own observations -trying demos on different CPUs- and comments on discussion forums, pocketGL does not perform better on XScale CPUs -in the best case frame rate remains unchanged. The network communication was done with a 802.11b wireless compact flash card, a PCMCIA card was tested as well, no significant changes in the performance was noticed.

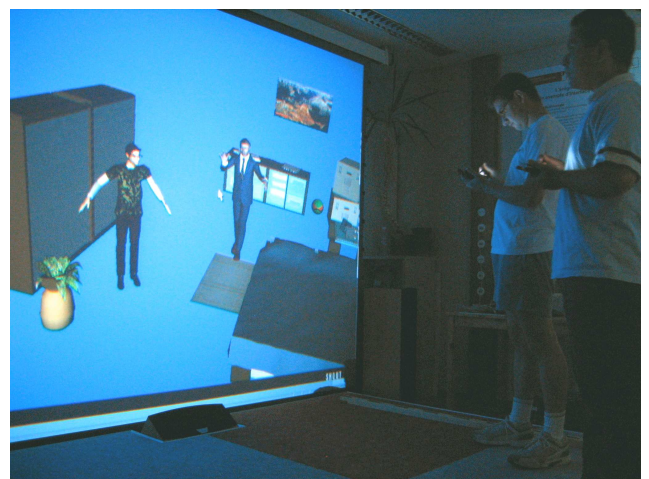


Figure 7. Two users interacting in the collaborative Virtual Environment.

Tests with two users in front of the projection screen reported good results in terms of user-friendliness and effectiveness of the interface, see figure 7.

The system truly enables more than one person to interact in a shared Virtual Environment. The main issue he have observed during a first informal evaluation was the division of the user's attention between the handheld and the simulation screen. Some users looked at the large screen only after they had finished posing their character on the PDA, to check the final result. Other participants selected one joint

on the handheld and turned their attention to the main screen while moving the stylus. Direct contact between participants and absence of UI components on the main simulation screen were appreciated by the test users. A more detailed study on user acceptance and efficiency will be necessary to validate the results.

5. Conclusions and further work

The system we have presented allows for multiuser interaction in a semi-immersive Virtual Environment. The handheld interface showed to be an efficient way to modify the posture and location of virtual characters in a Virtual Environment, without requiring additional applications/interfaces to be open, the main simulation screen remains undisturbed.

The interface could be improved by using inverse kinematics controls -as complement to the forward kinematics already implemented-, collision detection between character segments, and validation of joint limits.

Additionally, the mobile interface can be used not only for posing characters, but for interacting with every object in the Virtual Environment. We are currently working on a representation of virtual objects that will enable controlling the object's properties -position, orientation, visual attributes- and functionalities -animation, behavior. The system under development will present on the handheld a simplified representation of the virtual world for scene navigation and object interaction.

6. Acknowledgments

The authors want to thank Pascal Porte and Alexis Touon for their significant contribution on the development of the presented application. This work has been supported by the Swiss Federal Office for Education and Science in the framework of the European project IST-WorldStudio.

References

- [1] 3ds max. 3ds max, animation modeling and rendering software. <http://www.discreet.com/products/3dsmax/>.
- [2] T. Abaci, R. de Bondeli, J. Ciger, M. Clavien, F. Erol, M. Gutierrez, S. Noverraz, F. Vexo, and D. Thalmann. The enigma of the sphinx. In *International Conference on CYBERWORLDS, to appear*, 2003.
- [3] BonesPro. Skeletal deformation system for 3ds max <http://www.di-o-matic.com/products/max/bonespro/>.
- [4] D. Bowman. *Interaction Techniques for Common Tasks in Immersive Virtual Environments*. PhD thesis, Georgia Institute of Technology, June 1999.
- [5] D. Bowman, E. Davis, A. Badre, and L. Hodges. Maintaining spatial orientation during travel in an immersive virtual environment. In *Presence: Teleoperators and Virtual Environments*, volume 8, pages 618–631, 1999.
- [6] T. Capin, E. Petajan, and J. Ostermann. Efficient modeling of virtual humans in mpeg-4. *IEEE International Conference on Multimedia and Expo (ICME)*, 2, 2000.
- [7] T. Capin, E. Petajan, and J. Ostermann. Very low bitrate coding of virtual human animation in mpeg-4. *IEEE International Conference on Multimedia and Expo (ICME)*, 2, 2000.
- [8] A. Cheok, X. Yang, Z. Ying, M. Billinghurst, and H. Kato. Touch-space: Mixed reality game space based on ubiquitous, tangible, and social computing. In *Personal and Ubiquitous Computing*, volume 6, january 2002.
- [9] J. Ciger, M. Gutierrez, F. Vexo, and D. Thalmann. The magic wand. In *Proceedings of Spring Conference on Computer Graphics 2003*, pages 132–138.
- [10] D. Conner, S. Snibbe, K. Herndon, D. Robbins, R. Zeleznik, and A. van Dam. Three dimensional widgets. In *Symposium on Interactive 3D Graphics*, March 1992.
- [11] DieselEngine. Graphics library <http://www.inmarsoftware.com/>.
- [12] E. Farella, D. Brunelli, M. Bonfigli, L. Benini, and B. Ricco. Using palmtop computers and immersive virtual reality for cooperative archaeological analysis: the appian way case study. In *International Conference on Virtual Systems and Multimedia (VSMM)*, Gyeongju, Korea, 2002.
- [13] E. Farella, D. Brunelli, M. Bonfigli, L. Benini, and B. Ricco. Multi-client cooperation and wireless pda interaction in immersive virtual environment. In *Euromedia 2003 Conference*, Plymouth, United Kingdom, 2003.
- [14] G. W. Fitzmaurice, S. Zhai, and M. H. Chignell. Virtual reality for palmtop computers. *ACM Transactions on Information Systems (TOIS)*, 11(3):197–218, 1993.
- [15] M. Gutierrez, F. Vexo, and D. Thalmann. A mpeg-4 virtual human animation engine for interactive web based applications. In *Proceedings of the IEEE International Workshop on Robot and Human and Interactive Communication*, pages 554–559. Berlin, Germany, September 2002.
- [16] M. Gutierrez, F. Vexo, and D. Thalmann. Controlling virtual humans using pdas. In *The 9th International Conference on Multi-Media Modeling (MMM'03)*. Taipei, Taiwan, January 2003.
- [17] H-anim. The humanoid animation working group. <http://www.h-anim.org>.
- [18] L. Hill and C. Cruz-Neira. Palmtop interaction methods for immersive projection technology systems.
- [19] ISO/IEC 14496-1:1999. Coding of audio-visual objects: Systems, amendment 1, december 1999.
- [20] ISO/IEC 14496-2:1999. Coding of audio-visual objects: Systems, amendment 1, december 1999.
- [21] F. Mantovani. *Towards Cyber-Psychology: Mind, Cognitions and Society in the Internet Age*, chapter VR learning: Potential and Challenges for the Use of 3D Environments in Education and Training. IOS Press, 2003.
- [22] Maya. 3d animation and effects software. <http://www.alias.com/eng/products-services/maya/index.shtml>.

- [23] M. Mine. Towards virtual reality for the masses: 10 years of research at disney's vr studio. In *Proceedings of the workshop on Virtual environments 2003*, pages 11–17. ACM Press, 2003.
- [24] J. Pierce, A. Forsberg, M. Conway, S. Hong, and R. Zeleznik. Image plane interaction techniques in 3d immersive environments. In *Symposium on Interactive 3D Graphics*. Providence, Rhode Island, April 1997.
- [25] M. Ponder, B. Herbelin, T. Molet, S. Schertenleib, B. Ulicny, G. Papagiannakis, N. Magnenat-Thalmann, and D. Thalmann. Immersive vr decision training: Telling interactive stories featuring advanced virtual human simulation technologies. In *9th Eurographics Workshop on Virtual Environments (EGVE) 2003, to appear*, 2003.
- [26] A. van Dam, D. H. Laidlaw, and R. M. Simpson. Experiments in immersive virtual reality for scientific visualization. volume 26, pages 535–555. Elsevier Science Ltd., August 2002.
- [27] K. Watsen, R. Darken, and W. Capps. A handheld computer as an interaction device to a virtual environment. In *Proceedings of Third International Immersive Projection Technology Workshop (IPT 1999)*, Stuttgart, Germany, 1999.
- [28] B. Witmer and M. Singer. Measuring presence in virtual environments: A presence questionnaire. In *Presence: Teleoperators and Virtual Environments*, volume 7, pages 225–240, 1998.
- [29] R. Zeleznik, K. Hemdon, D. Robbins, N. Huang, T. Meyer, N. Parker, and J. Hughes. An interactive 3d toolkit for constructing 3d widgets. In *SIGGRAPH*, August 1993.