Virtual Humans for Representing Users in Virtual Environments

Tolga K. Capin[1], Igor Sunday Pandzic[2],
Nadia Magnenat Thalmann[2], Daniel Thalmann[1]

[1]Computer Graphics Laboratory
Swiss Federal Institute of Technology
CH1015 Lausanne, Switzerland
{capin,thalmann}@lig.di.epfl.ch

[2]MIRALAB
Centre Universitaire d'Informatique
University of Geneva
24 rue de Général-Dufour
CH1211 Geneva 4, Switzerland
{ipandzic,thalmann}@cui.unige.ch

Abstract

Virtual environments define a new interface for networked multimedia applications. The sense of "presence" in the virtual environment is an important requirement for collaborative activities involving multiple remote users working with social interactions. Using virtual actors within the shared environment is a supporting tool for presence. In this paper, we present a shared virtual life network with virtual humans that provides a natural interface for collaborative working.

Keywords: Multimedia, Virtual Life, Computer Animation, Networked Environments, Virtual Actors.

1. Introduction

Increasing hardware and network performance together with the software technology make it practical to use environments which support interactive human collaboration, with more realistic and immersive interfaces. A variety of applications can be developed which consist of networked virtual environments that include different media in a single stream, in real time. For this purpose, there is a need to provide mechanisms to give the users the impression of presence in this virtual world. This can be achieved by interactive modelling and animation of human-like actors to represent their participants.

The virtual environment should allow effective collaboration among users at different sites, for example distant users can have virtual workspaces to collaboratively design a product. This necessitates incorporating different media in the complex environment, such as image, real-time movies, 3D objects. Instead of having different applications or windows for each medium, the 3D virtual environment should be able to provide easy mechanisms to integrate these applications.

It is desirable to have a realistic representation of the users in the 3D world. This can be achieved with two aspects: the visual appearance of the actors similar to their users in the virtual environment, and realistic mechanisms for interaction of these actors with the environment.

The visual appearance of the virtual actors is an important requirement, as it helps the users to show their intensions as well as it provides an appropriate mechanism for providing realistic interaction with the environment.

In addition to visual realism, behavioral realism of virtual humans is also a significant requirement especially for systems that are based on human collaboration, such as Computer Supported Cooperative Work (CSCW) systems. Networked CSCW systems require that the shared environment should: provide a comfortable interface for gestural communication, support awareness of other users in the environment, provide mechanisms for different modes of interaction (synchronous vs. asynchronous, allowing to work at different times in the same environment), supply mechanisms for customized tools for data visualization, protection and sharing.

There has been a considerable amount of research for modelling and animation of 3D virtual humans [BCHK95] We have developed an articulated model that realistically simulates the degrees of freedom and the joint limits, and tools for animating the figures in real-time. The model also contains realistic body surfaces attached as envelops to the 3D skeleton; and real-time automatic motion generators for actions such as walking and grasping.

There has been increasing interest in the area of networked virtual environments recently [ZJ94][MZPBZ94][S94][GS94]. These systems are centralized on the task level, and few work has been done on supporting the presence of the users through interactive representation and manipulation of virtual actors, to give the user the sense of presence with their whole body. The aim of the VLNET system is to provide mechanisms for supporting virtual actors for interactive user participation in a networked 3D virtual multimedia environment by:

- representing the participants by the realistic-looking virtual human actors, similar to their users,
- mechanisms for the users to interact with each other in the natural interface via facial interaction and body gestures of their virtual actors,
- mechanisms for the users to interact with the rest of the virtual environment through complex and realistic behaviors such as walking for navigation, grasping, etc.
- user-customized tools for editing the picked objects, depending on the object type (e.g. images, free-form surfaces)
- autonomous actors with special tasks to help and interact with the participants.

The paper starts with the properties of the system: the environment, methods for modelling and animation of virtual actors in this environment, facial interaction support, communication model. Then we discuss the implementation aspects of the system and list current applications in use. Finally, we present our concluding remarks and expectations for future improvements.

2. The Shared Environment

The VLNET system supports a networked shared virtual environment that allows multiple users to interact with each other and their surrounding in real time. The users are represented by 3D virtual human actors, with realistic appearances and articulations. The actors have similar appearance and behaviors with the real humans, to support the sense of presence of the users in the environment.

The environment incorporates different media; namely sound, 3D models, facial interaction among the users, images represented by textures mapped on 3D objects, and real-time movies. The environment works as a general stream to include these different media (Figure 1).

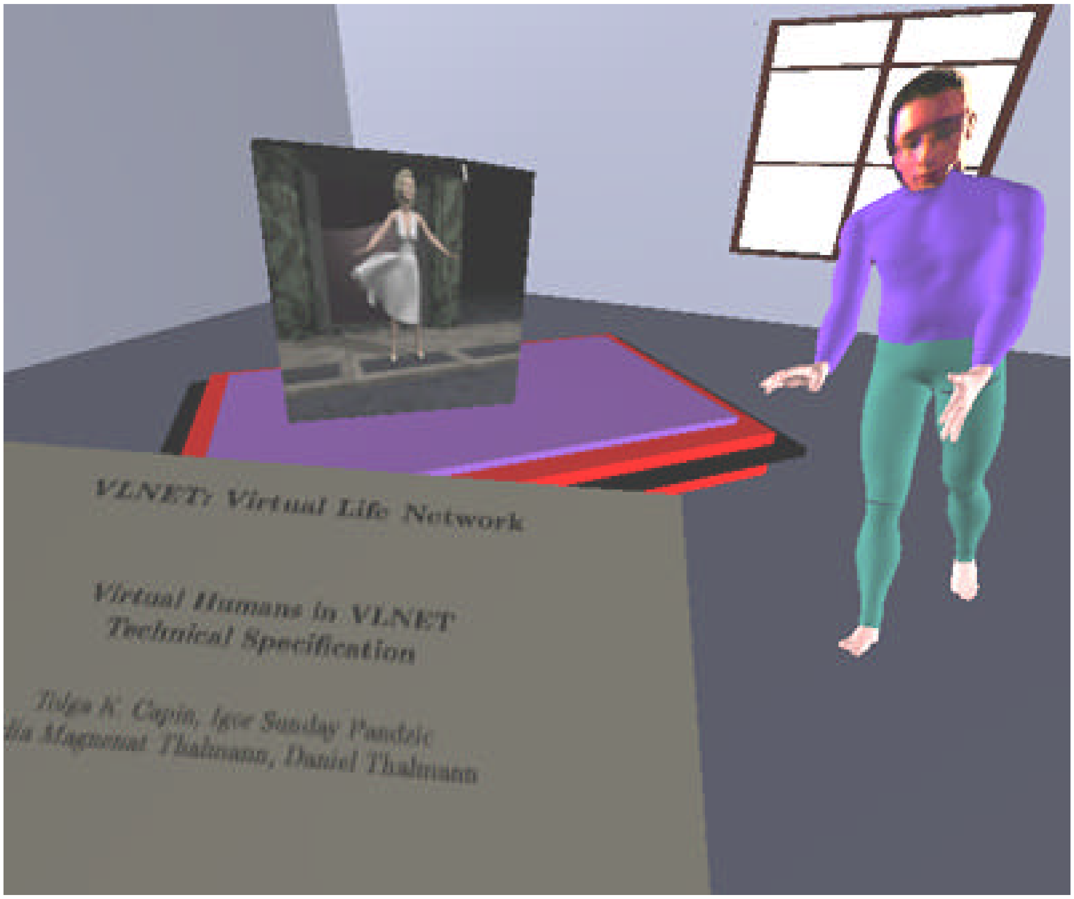

Figure 1: The environment integrates different media in a single stream

The objects in the environment are classified in two groups: fixed (e.g. walls) or free (e.g. a chair). Only the free objects can be picked, moved and edited. This allows faster computations in database traversal for picking. In addition to the virtual actors representing users, various types of objects can be included: simple polygonal objects, image texture-mapped polygons (e.g. to represent three-dimensional documents, or images in the environment), etc. In order to reposition or edit an object, the user has to pick the object, using her virtual actor. Each type of object has a user-customized program corresponding to the type of object, and this program is spawned if the user picks and requests to edit the object. Figure 1 shows a general view from an example environment.

In addition to user-guided actors, the environment is also extended to include fully autonomous human actors which can be used as a friendly user interface to different services such as navigation. Autonomous actors can also be used in order to represent the currently unavailable partners, allowing asynchronous cooperation between distant partners.

3. Virtual Actors

The participants are represented by 3D virtual human bodies in the environment. In addition, there exist virtual actors with self-determined actions. We make difference of these types by calling the former as “guided actors” and the latter as “autonomous actors”. Although they have different motion control methods, they should have basically similar representation in the 3D virtual world. In this section, we discuss the representation of the virtual body, and the methods provided to control the guided actors, and self-control of autonomuous actors.

3.1. Representation of Virtual Actors

In order to provide maximum immersion for using virtual actors, it is important to use the most realistic human representation possible with real time considerations. Therefore, it is prefered to use the deformed body surfaces as envelops attached to the human skeleton, rather than simple representations such as basic geometric primitives. It is also necessary to simulate the articulations of the body, i.e. the joints to connect the body limbs with realistic constraints.

Having these requirements in mind, we have exploited the anthropometrically based human models. There models were developed for the ESPRIT Humanoid I project, in order to parametrically produce different human bodies. The specifications of the human models are:

- 17 limbs for representing the whole human body (head, pelvis, thorax, abdomen, left and right upper leg, lower leg, foot, upper arm, lower arm, hand)
- 74 degrees of freedom
- different levels of detail for representing the body surface
    - high resolution (40000 triangles)
    - low resolution (10000 triangles)
    - rough resolution (2500 triangles)

3.2. Control of Guided Virtual Actors

In the environment, each participant is represented by a 3D virtual body that resembles the user. Each user sees the virtual environment through the eyes of her body. The user controls the movement of her virtual actor by various input devices, such as mouse, spaceball. Stereo display devices such as shutter glasses and head-mounted display can also be used for more realistic feeling in the environment.

It is not desirable to see solid-looking floating virtual actors in the environment, rather it is required to make use of articulations in the body as similar as possible to the real motions. For this purpose, motion control methods should be used for guiding virtual actors. A motion control method for virtual humans can be assumed as a function that accepts user input and behavioral specifications; and outputs the angle values of the skeleton joints.

There are numerous methods for controlling motion of synthetic actors. The motion control method can be classified according to the type of information it privileged in animating the synthetic actor. The nature of the privileged information for the motion control of actors falls into three categories:

> - The first approach corresponds to methods heavily relied upon by the animator: rotoscopy, shape transformation, keyframe animation. Synthetic actors are *locally controlled* by the input of geometrical data for the motion.
>
> - The second way is based on the methods of kinematics and dynamics. The input is the data corresponding to the complete definition of motion, in terms of forces, torques, constraints. The task of the animation system is to obtain the trajectories and velocities by solving equations of motions. Therefore, it can be said that the actor motions are *globally controlled*.
>
> - The third type of animation is called behavioral animation and takes into account the relationship between each object and the other objects. The control of animation can also be performed at task-level, but one may also consider the actor as an *autonomous creature*. The behavioral motion control of the actor is provided by providing high-level directives indicating a specific behavior without any other stimulus.

Each category can be used for guiding virtual actors in the virtual environment, however it is prefered to provide the most appropriate interface for controlling different motions. No method alone is convenient to provide a comfortable interface to accomplish all the motions, therefore it is necessary to combine various techniques for different tasks.

For control of guided virtual actors, we make use of the combination of these methods. The methods rely on the participant to use the input devices to guide the actor, by specifying the next key posture for the body. Using this next posture, the behavioral mechanisms are used for generating the realistic motions.

For the current implementation, we use the behaviors of walking for navigation and grasping for picking. We also include a set of gestures for representing different intentions. This set of behaviors can easily be extended, however these behaviors are sufficient to perform everyday activities, providing minimum set of behaviors to attend virtual meetings.

For navigation, the guidance is achieved as follows. The participant uses the input devices to update the transformation of the eye position of the virtual actor. This local control is used by computing the incremental change in the eye position, and using the walking motor to update the joint angles. The walking motor is based on the Humanoid walking model, guided by the user interactively or automatically generated by a trajectory. This model includes kinematical personification depending on the inidividuality of the user [BMTT90]. The joint values are set for walking, based on the mathematical parameterization coming from biomechanical experimental data. See [BPCH95] for detailed description of the walking motor.

For grasping behavior, we make use of the inverse kinematics within the right arm of the virtual actor. The participant uses the input devices to update the position and orientation of the virtual actor's hand. Based on this input, the realistic posture of the right arm is computed within the joint limit constraints. We use simple chain of the right arm for inverse kinematics for real-time motion of the virtual actor. Although we could apply a physically correct method, our concern is more on the visual appearance of the grasping motion.

3.3. Autonomous Virtual Actors

We also include additional virtual autonomous actors in the environment. The autonomous actors are connected to the system in the same way as human participants, and enhance the usability of the environment by providing services such as replacing missing partners, providing services such as helping in navigation. As these virtual actors are not guided by the users, they should have sufficient behaviors to act autonomously to accomplish their tasks. This requires building behaviors for motion, as well as appropriate mechanisms for interaction.

Animation of autonomous actors is an active area of research [T94]. A typical behavioral animation system consists of three key components:

- the perceptual system, that senses the environment, and provides the perception information about the objects that the actor can interact with,
- the organism system that is concerned with the rules, skills, motives, drives and memory,
- the locomotor system that is responsible for using motors to generate motion.

These components are not independent of each other, rather they are connected by links. It is an active research area on how to model the connectivity between these components.

The perceptual system should be realistic by providing the limitations of sensing the surrounding, and should be improved through the synthetic vision [RMTT90] as what the actor would see in the real world.The organism system should consider the different goals with the inference mechanism in order to achieve motions.

In the current implementation, the autonomous actors are used for helping in navigation, for finding the position of different objects in the environment. The participant has to approach the autonomous actor to communicate with it. When the autonomous actor detects the approaching user using the perception mechanism, it begins to "listen". The user can then send the name of the object to the actor. The autonomous actor follows the path to that object using its internal database, requiring the participant to follow himself. Therefore the perception system is included to detect the participant, the locomotor system for the walking behavior in the environment, and the organism system consists of the local database of the virtual actor for the objects' positions. Note that we make use of the same walking motor to compute the joint values for autonomous actors as guided actors.

3.4. Facial Gestures

Face is one of the main streams of interaction among humans for representing intentions, thoughts and feelings; hence including facial expressions in the shared virtual environment is almost a requirement for efficient collaboration. Although it is also possible to utilize a videoconferencing tool among the users in a separate window, it is more appropriate to display the facial gestures of the users in the face of their actors in 3D virtual world in order to provide a better interaction among users.

We include the facial interaction by texture mapping the image containing the user's face on the virtual actor's head. To obtain this, the subset of the image that contains the user's face is selected from the captured image and is sent to other users. To capture this subset of image, we apply the following method: initially the background image is stored without the user. Then, during the session, video stream images are analyzed, and the difference between the background image and the current image is used to determine the bounding box of the face in the image. This part of the image is compressed using the SGI Compression Library MVC1 compression algorithm. Finally, the image is sent to the other users after compression. There is a possibility to send uncompressed grey-scale images instead of using compression, which is useful if the participants' machines are not powerful enough to perform compression and decompression without a significant overhead. However with all the machines we used this was not necessary. If this option is used, the compression can be turned on/off on the sending side, and the receiving side recognises automatically the type of incoming images.

At the receiving side, an additional service program is run continuously in addition to the VLNET program: it continuously accepts the next images for the users and stores in the shared memory. The VLNET program obtains the images from this shared memory for texture mapping. In this way, communication and simulation tasks are decoupled, decreasing the overhead by waiting for facial communication.

Currently, we are using the simplified objects for representing the head of users' virtual actors (Figure 2). This allows better three-dimensional appearence than simple primitives such as ellipsoid, sphere. It would also be possible to use more complex heads for the faces, however this requires additional task of topologically adjusting the texture image to the face of the virtual actor, to match the correesponding parts of the 3D face and image of the user.

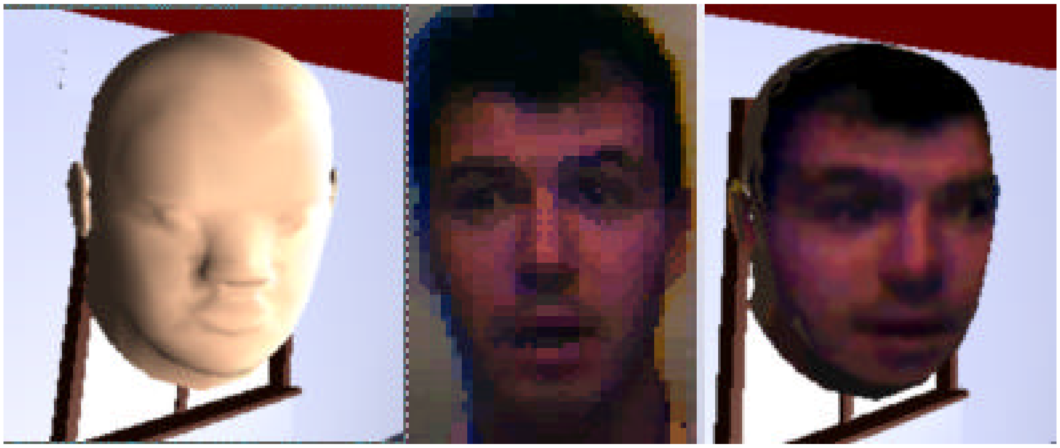

Figure 2: Mapping of the face to the 3D virtual actor

4. Communication Architecture

We exploit a distributed model of communication, therefore each user is responsible for updating its local set of data for the rendering and animation of the objects.

There is always one user that determines the environment. The other users are "invited" and do not need to specify any parameters, all the environment data is initially loaded over the network to the local machine when the user is connected to the shared environment. There exists one server responsible of transmitting the actions to the participants.

The communication is asynchronous. The information about the users' actions are transmitted to the server as the actions occur. The actions can be changing the position or orientation of actors, as well as grasping or releasing an object. The actions are sent to the other users by the server in terms of new orientations of the updated objects in space, as well as other possible changes such as modification to the objects.

Note that the architecture requires the broadcasting of the data from the server to all the users in the system. This can create a bottleneck if there are a lot of users in the environment. To overcome this problem, we plan to exploit a communication mechanism that makes use of the geometric coherence of interactions among the virtual actors in the three-dimensional environment. This solution is based on the aura and nimbus concepts, proposed by Fahlen and Stahl [FS94] in order to emphasize the awareness among the entities in the virtual environment. *Aura* refers to the subspace where an object has potential to interact with others. In order for two objects to interact, their auras should intersect. Furthermore, if the auras intersect, then a test whether the *focus* of the first object intersects with the *nimbus* of the second object. Focus represents the subspace where the object draws its attention. Nimbus refers to the space where the object makes an aspect of itself available to other users. If the focus of the first user intersects with the nimbus of the second object, then it is assumed that the user is attracted to the object.

We make use of the aura and nimbus concepts as follows: When the data is to be sent from the server to a participant, the sending program tests if the nimbus of the sending user intersects with the focus of the other user's virtual actor. The intersection means that the actors are near to each other, therefore the server sends the change to the other user. If there is no intersection with one other actor's focus, it can be assumed that the actor is too far and does not need the extensive knowledge of the source user, therefore the change is not sent every time step. However, for consistency, it is necessary to send the local position data every k frames. The k value could be computed using the distance between the two actors, however we assume a constant k for the initial implementation.

5. Implementation Issues

For the virtual environment to be realistic, our system should be fast enough in providing feedback; otherwise it is not comfortable to use. Therefore, we currently make use of the state-of-the-art technology discussed below to achieve utmost performance available, but it is widely accepted that these platforms will be used by a large community in a few years' period. There are three ways for speeding-up the system performance: display, communication, simulation.

For fast display, we make use of the IRIS Performer environment [RH94]. Performer provides an easy-to-use environment to develop real-time graphics applications. It can extract the maximum performance of the graphics subsystem, and can utilize parallel processors in the workstation for graphics operations. Therefore, it is an appropriate platform to increase the performance of display part of the system. In addition, it provides efficient mechanisms for collision detection, and supports a variety of popular file types.

The network overhead can also have a significant effect, especially with increasing number of users. Therefore, it is important to provide low-latency high-throughput connections. The ATM network is one of the most promising solutions to the network problem, therefore we are experimenting our system over the ATM pilot network, provided to the Swiss Federal Institute of Technology and University of Geneva, by Swiss Telecom (Figure 3). The ATM technology, based on packet switching using fixed-length 53-byte cells, allows to utilize traffic rates for videoconferencing, video-on-demand, broadcast video. The quality of service is achieved on demand, and guarantees a constant performance. The network has full logical connectivity at the virtual path level and initially supports PDH 34 Mbit/s and SDH 155 Mbit/s links. The pilot provides point to point links in the first phase. Multipoint links are supposed to be added in the future, allowing more efficient multiple-user virtual environments.

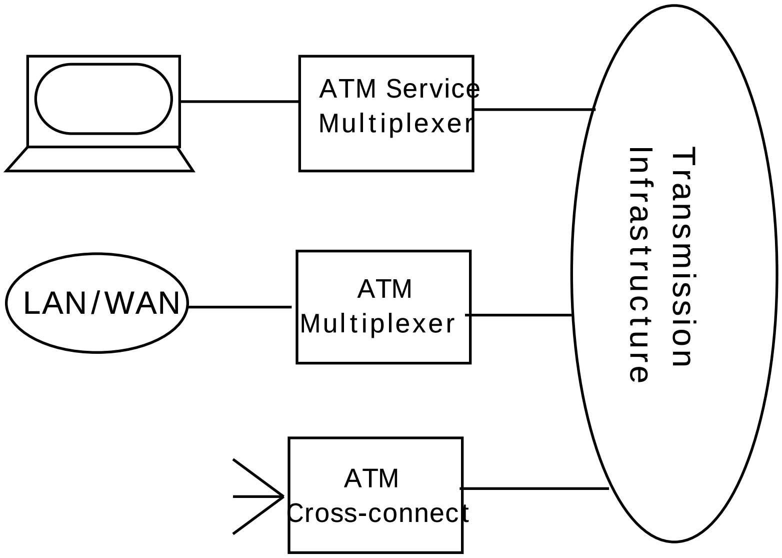

Figure 3: Network Topology, ATM Overlay Network

In addition to the display and network parts, the simulation part should also be performed efficiently using appropriate mechanism. We make use of the HUMANOID system for modelling and real-time animation of virtual actors [BCHK95]. The HUMANOID environment supports the following facilities:

- real-time manipulation of virtual actors on a standard graphics workstation,
- a way of accelerating the calculations using a dedicated parallel machine or by distributing the calculation load on several workstations
- a flexible design and management of multiple humanoid entities
- skin deformation of a human body, including the hands and the face
- a multi-layer facial animation module
- collision detection and correction between multiple humanoid entities
- several motion generators and their blending: keyframing, inverse kinematics, dynamics, walking and grasping

6. Applications

As already discussed, VLNET is a general-purpose system. As various widely-used file formats are supported, it is easy to create a shared environment consisting of already developed models with other computer modelling programs, such as AutoCad, Inventor, etc.

In this section, we present some experimental applications currently available with our system:

Teleshopping: The VLNET system is currently used by *Chopard Watches, Inc.* , Geneva; to collaboratively view and present the computer-generated models of the recently-designed watches with the remote customers and colleagues in Singapore, and Geneva. The models had already been developed using AutoDesk program, and these were easily included in the virtual environment, with the help of 3D Studio reader for Performer.

Business: Experiments are going on for building a virtual room involving distant users to be able to have a meeting, with the aid of images and movies to be able to discuss and analyze the results.

Figure 4: Applications of the VLNET Environment

Entertainment: The VLNET environment is also used for playing chess between different remote partners; and puzzle solving by two users. These models were created using the IRIS Inventor and WaveFront packages.

Interior design: Currently experiments are continuing on furniture design by the customer and the sales representative to build a virtual house. The model was created using the WaveFront package.

7. Conclusions and Future Work

In this paper, we have presented a system which provides an efficient and visually effective way for human collaboration. We believe that it will give a satisfactory interface for virtual environments allowing interactive cooperation.

However, further improvements remain to include physical models for interaction, more visual realism by incorporating already-developed body deformations module, natural language interface with the virtual actors, sound rendering. These new properties are not difficult to integrate, however require more powerful processors than what is available today.

Also, it is an open research area to add more behaviors to the virtual actors, and provide mechanisms to generate gestures of virtual bodies automatically depending on the user's intentions.

We also plan to integrate our system with the framework of dVS and DIVE to be able to support more platforms for the environment.

Acknowledgements

The research was partly supported by ESPRIT project HUMANOID (P 6079), Swiss National Foundation for Scientific Research, Silicon Graphics, l'Office Fédéral de l'Education et de la Science and Department of Economy of the City of Geneva. We would like to thank Riccardo Camiciottoli for his 3D Studio driver for Performer.

References

[BCHK95] Boulic R., Capin T., Huang Z., Kalra P., Lintermann B., Magnenat-Thalmann N., Moccozet L., Molet T., Pandzic I., Saar K., Schmitt A., Shen J., Thalmann D., "The Humanoid Environment for Interactive Animation of Multiple Deformable Human Characters", *Proc. Eurographics '95*, 1995.

[BMTT90] Boulic R., Magnenat-Thalmann N. M.,Thalmann D. "A Global Human Walking Model with Real Time Kinematic Personification", *The Visual Computer,* Vol.6(6),1990.

[BPCH95] Boulic R., Becheriaz P. Capin T., Huang Z., Molet T., Shen J., Kalra P., Moccozet L., Pandzic I., "Intelligent Motion Control: Walking and Grasping. Walklib: Technical Manual", ESPRIT 6709 HUMANOID Project Report, 1995.

[CPBS94] Cassell J., Pelachaud C., Badler N., Steedman M., Achorn B., Becket T., Douville B., Prevost S., Stone M., "Animated Conversation: Rule-Based Generation of Facial Expression Gesture and Spoken Interaction for Multiple Conversational Agents", *Proc. SIGGRAPH'94*, 1994.

[FS94] Fahlen L.E., Stahl O., "Distributed Virtual Realities as Vehicles for Collaboration", *Proc. Imagina '94*, 1994.

[GS94] Gisi M. A., Sacchi C., "Co-CAD: A Collaborative Mechanical CAD System", *Presence: Teleoperators and Virtual Environments,* Vol. 3, No. 4, 1994.

[MT94] Magnenat-Thalmann N., "Tailoring Clothes for Virtual Actors", *Interacting with Virtual Environments,* MacDonald L., Vince J. (Ed), 1994.

[MZPBZ94] Macedonia M.R., Zyda M.J., Pratt D.R., Barham P.T., Zestwitz, "NPSNET: A Network Software Architecture for Large-Scale Virtual Environments", *Presence: Teleoperators and Virtual Environments*, Vol. 3, No. 4, 1994.

[PKMTT94] Pandzic I.S., Kalra P., Magnenat-Thalmann N., Thalmann D., "Real-Time Facial Interaction", *Displays*, Vol 15, No 3, 1994.

[RH94] Rohlf J., Helman J., "IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics", *Proc. SIGGRAPH'94*, 1994.

[RMTT90] Renault O., Magnenat-Thalmann N., Thalmann D., "A Vision-based Approach to Behavioral Animation", *The Journal of Visualization and Computer Animation*, Vol.1, No.1, 1990.

[RWSS94] Rich C., Waters R.C., Strohecker C., Schabes Y., Freeman W.T., Torrance M. C., Golding A.R., Roth M., "Demonstration of an Interactive Multimedia Environment", *IEEE Computer,* Vol. 27, No. 12, 1994.

[S93] Swiss Telecom, "ATM Pilot Services and User Interfaces", *Swiss Telecom,* 1993.

[S94] Stansfield S., "A Distributed Virtual Reality Simulation System for Simulational Training", *Presence: Teleoperators and Virtual Environments,* Vol. 3, No. 4, 1994.

[ST95] Shen J., Thalmann D, "Interactive Shape Design Using Metaballs and Splines", Implicit Surface95, Eurographics Workshop on Implicit Surfaces, Grenoble, France, 1995

[T94] Thalmann D., "Automatic Control and Behavior of Virtual Actors", *Interacting with Virtual Environments*, MacDonald L., Vince J. (Ed), 1994.

[TWA94] Travis D., Watson T., Atyeo M., "Human Psychology in Virtual Environments", *Interacting with Virtual Environments*, MacDonald L., Vince J. (Ed), 1994.

[ZJ94] Zeltzer D., Johnson M., "Virtual Actors and Virtual Environments", *Interacting with Virtual Environments*, MacDonald L., Vince J. (Ed), 1994.