Participant, User-Guided and Autonomous Actors in the Virtual Life Network VLNET

Daniel Thalmann Tolga K. Capin
Computer Graphics Laboratory
Swiss Federal Institute of Technology
CH 1015 Lausanne, Switzerland
{thalmann, capin}@lig.di.epfl.ch

Nadia Magnenat Thalmann Igor Sunday Pandzic
MIRALAB, CUI
University of Geneva
24 rue de Général-Dufour
CH 1211 Geneva 4, Switzerland
{thalmann, ipandzic, }@cui.unige.ch

Abstract

VLNET is a shared virtual life network with virtual humans that provides a natural interface for collaborative working. Virtual actors play a key role in VLNET. Three kinds of virtual actors may coexist in a VLNET scene: participant, guided and autonomous actors. A participant actor is a virtual copy of the real user or participant; his movement is exactly the same as the real user. A guided actor is an actor completely controlled in real-time by the user. An autonomous actor is an actor who may act without intervention of the user.

Keywords: Virtual Life, Virtual Reality, Telecooperative Work, Computer Animation, Networked Multimedia, Virtual Actors.

1. Introduction

There have been an increasing interest in the area of networked virtual environments recently [1] [2] [3] [4]. These systems are centralized on the task level, and less work has been done on supporting the psychological aspects [5] [6], including the sense of *presence* in the shared environment through gestural communication with other users or virtual agents and the representation of the whole body.

Providing a behavioral realism is a significant requirement for systems that are based on human collaboration, such as Computer Supported Cooperative Work (CSCW) systems. Networked CSCW systems also 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 in different times in the same environment), supply mechanisms for customized tools for data visualization, protection and sharing.

Virtual Reality can provide a powerful mechanism for networked CSCW systems, by its nature of emphasizing the *presence* of the users in the virtual environment. This can be accomplished through the support of:

- representing the users and special-purpose service programs by 3D virtual actors in the virtual environment,
- 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)

With Virtual Reality, new interfaces and 3D devices allow us a complete immersion into virtual worlds or at least a direct and real-time communication with them. But at the same time, researchers have been able to create plants, trees, and animals and research in human animation has led to the creation of synthetic actors with complex motion. Moreover, a new field based on biology has tried to recreate the life with a bottom approach, the so-called *Artificial Life* approach [7]. Based on these considerations, we have developed the VLNET system [8 9]. In VLNET, we try to integrate these techniques in order to create truly Virtual Worlds with autonomous living actors with their own behavior and real people should be able to enter into these worlds and meet their inhabitants. The ultimate objective is to build intelligent autonomous virtual humans able to take a decision based on their perception of the environment. Fig. 1 shows the foundations of VLNET.

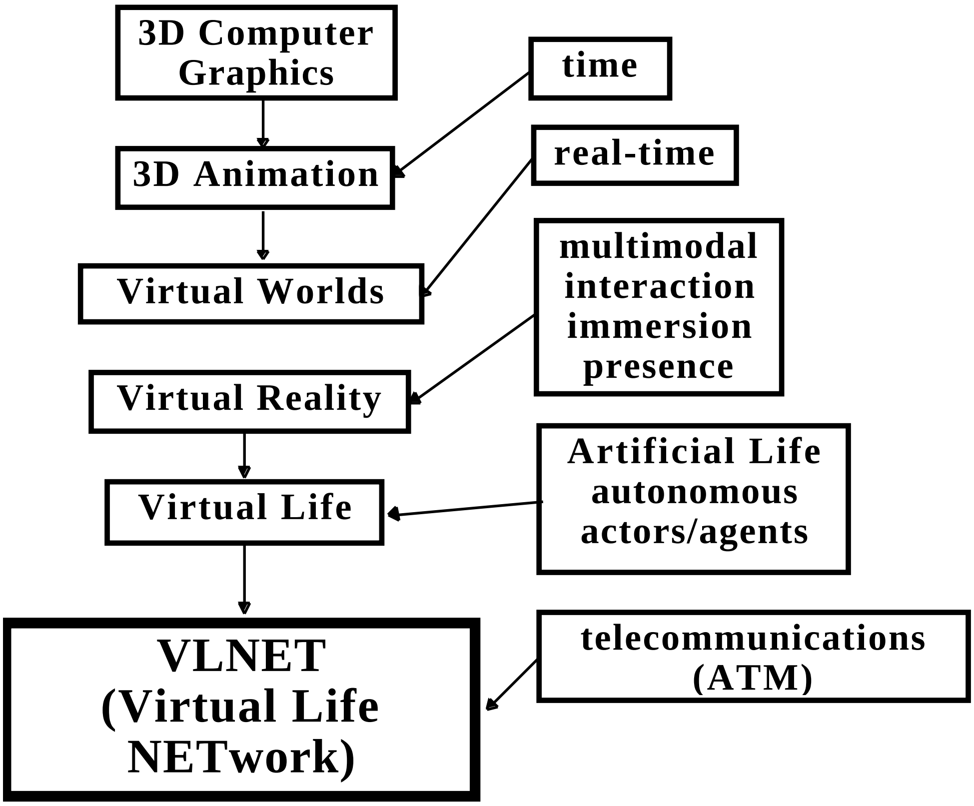

Fig. 1 Evolution towards VLNET

2. The VLNET system

The VLNET (Virtual Life NETwork) 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, which serve as agents to interact with the environment and other agents. The agents have similar appearance and behaviors with the real humans, to support the sense of presence of the users in the environment. In addition to user-guided agents, the environment can also be extended to include fully autonomous human agents which can be used as a friendly user interface to different services such as navigation. Virtual humans can also be used in order to represent the currently unavailable partners, allowing asynchronous cooperation between distant partners.

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. Instead of having different windows or applications for each medium, the environment integrates all tasks in a single 3D surrounding, therefore it provides a natural interface similar to the actual world. The environment works as a general-purpose stream, allowing the usage of various models for different applications.

The objects in the environment are classified into 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, the types of objects can be: simple polygonal objects, image texture-mapped polygons (e.g. to include three-dimensional documents, or images in the environment), etc. Once a user picks an object, he or she can edit the object. 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. Fig. 2 shows a general view of an example environment.

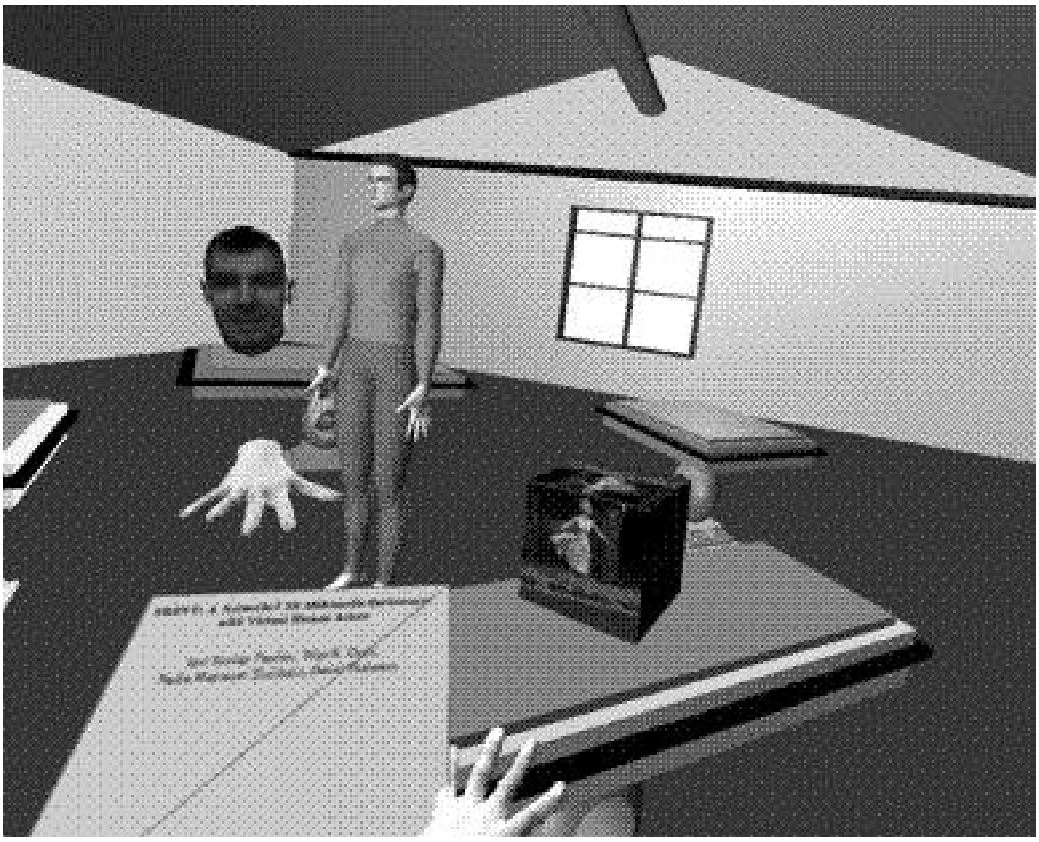

Fig. 2 The Environment integrates different media in a single stream

3. A classification of real-time virtual creatures

As virtual actors play a key role in VLNET, we will first try to clarify the concept of virtual actor. We define a virtual (or synthetic) actor as a human-like entity with an abstract representation in the computer. A real-time virtual actor is a virtual actor able to act at the same speed than a real person. Virtual Reality, Interactive Television, and Games require real-time virtual actors. In VLNET, three types of real-time virtual actors may coexist in the same shared environment.

- **Participant actors**
  A participant actor is a virtual copy of the real user or participant. His movement is exactly the same as the real user.

- **Guided actors**
  A guided actor is an actor completely controlled in real-time by the user.

- **Autonomous actors**
  An autonomous actor is an actor who may act without intervention of the user.

In the next sections, the role of each type of real-time virtual actor is explained in more details.

4. Participant actors

Representing participants by a virtual actor in the virtual environment is an important factor for presence. This becomes even more important in multi-user environments, where effective interaction among participants is a contributing factor to presence. This is the Self-representation in the Virtual World. Even with limited sensor information, a virtual human frame can be constructed in the Virtual World, that reflects the activities of the real body. Slater and Usoh [10] indicates that such a body, even if crude, heightens the sense of presence. The virtual actor is required to have a natural-looking body and be animated correlated to the actual body. It corresponds to a real-time form of traditional rotoscopy. Traditional rotoscopy in animation consists of recording the motion by a specific device for each frame and using this information to generate the image by computer. Using the terminology we introduced previously [11], we call *real-time rotoscopy method* a method consisting of recording input data from a VR device in real-time allowing to apply at the same time the same data to a graphics object on the screen. For example, when the animator opens the fingers 3 centimeters, the hand in virtual scene does exactly the same.

For a complete representation of the participant actor’s body, it should have the same movements as the real participant body for more immersive interaction. This can be best achieved by using a large number of sensors to track every degree of freedom in the real body, however this is generally not possible due to limitations in number and technology of the sensing devices. Therefore, the tracked information should be connected with behavioral human animation knowledge and different motion generators in order to "interpolate" the joints of the body which are not tracked.

5. Guided actors

Guided actors are actors which are driven by the user but which do not correspond directly to the user motion. They are based on the concept of *real-time direct metaphor* [9] a method consisting of recording input data from a VR device in real-time allowing to produce effects of different nature but corresponding to the input data. There is no analysis of the meaning of the input data. To understand the concept, we may take an example of traditional metaphor: *the puppet control.* A puppet may be defined as a doll with jointed limbs moved by wires or strings. Similarly glove-puppets are dolls of which the body can be put on the hand like a glove, the arms and head being moved by the fingers of the operator. In both cases, human fingers are used to drive the motion of the puppet.

In VLNET, the best example of actor guidance is guided navigation. 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 estimating the rotation and velocity of the center of body. The walking motor [12] uses the instantaneous velocity of motion, to compute the walking cycle length and time, by which it computes the necessary joint angles. The motor is based on the walking model, guided by the user interactively or automatically generated by a trajectory. This model can also include kinematics personification depending on the individuality of the user, and it is based on the mathematical parameterization coming from biomechanical experimental data. The sensor information for walking can be obtained from various types of input devices like DataGlove, or SpaceBall. Another good example for local control for walking is the virtual treadmill as proposed by Slater et al. [13].

The arm motion for picking an object is a similar problem to walking: given 6 degrees of freedom (position and orientation) of the sensed right hand with respect to body coordinate system, the arm motor should compute the joint angles within the right arm. This is more difficult than simple inverse kinematics problem, because there are generally multiple solutions for the joint angles to reach to the same position, and the most realistic posture should be selected. In addition, the joint constraints should be considered for setting the joint angles.

For real-time purposes, we exploit a three-link chain with 7 degrees of freedom within the right arm, with the right shoulder as the root. The arm motor first uses heuristics to obtain approximate arm joint angles, given the position of the right hand. This is done with the help of a precomputed table that divides the normalized volume around the body to discrete number of subvolumes (e.g. 4x4x4), and stores a mapping from subvolumes to joint angles in the right arm. This mapping was calculated by experimental observation of a real subject, using a DataGlove and sensors attached to the right arm. In the first step, the current position of the right hand is used to find the subvolume it resides with respect to the body, and the joint values are set using this mapping. In the second step, starting from the approximated posture, the arm motor applies normal inverse kinematics within the arm, considering the joint constraints. As this step starts from a more correct posture, it is expected to converge faster, and generate more realistic postures. The disadvantages of the current method are the extra storage of subvolume data structure, which becomes more difficult to store and experimentally obtain with finer discretization of the volume. Currently, we are working on obtaining a more accurate mathematical parametrization of the arm movement, without using the discretization of the space into big subvolumes.

6. Autonomous actors

An autonomous system is a system that is able to give to itself its proper laws, its conduct, opposite to an heteronomous systems which are driven by the outside. Guided actors as introduced in Section 5, are typically driven by the outside. Including autonomous actors that interact with participants increases the real-time interaction with the environment, therefore we believe that it contributes to the sense of presence in the environment. The autonomous actors are connected to the VLNET system in the same way as human participants, and also 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.

In order to introduce autonomous actors in VLNET, we have taken advantages of our active research in this area for multimedia[14] and interactive television applications [15]. Our autonomous virtual actors are able to have a behavior, which means they must have a manner of conducting themselves. Behavior is not only reacting to the environment but should also include the flow of information by which the environment acts on the living creature as well as the way the creature codes and uses this information. Behavior of autonomous actors is based on their perception of the environment. Perception is defined as the awareness of the elements of environment through physical sensation. In order to implement perception, virtual actors should be equipped with visual, tactile and auditory sensors. These sensors should be used as a basis for implementing everyday human behavior such as visually directed locomotion, handling objects, and responding to sounds and utterances. For synthetic audition, in a first step, we model a sound environment where the synthetic actor can directly access to positional and semantic sound source information of a audible sound event. A simulation of the haptic system consists in detecting contacts between the virtual actor and the environment. But, the most important perceptual subsystem is the vision system. A synthetic vision [16] based approach for virtual humans is a very important perceptual subsystem and is for example essential for navigation in virtual worlds. It is an ideal approach for modeling a behavioral animation and offers a universal approach to pass the necessary information from the environment to the virtual human in the problems of path searching, obstacle avoidance, and internal knowledge representation with learning and forgetting. Using hardware developments like the Reality Engine, it is possible to give a geometric description of 3D objects together with the viewpoint and the interest point of a synthetic actor in order to get the vision on the screen. This vision may then be interpreted like the virtual actor vision. This is our approach as described in several papers.

7. Visual appearance

7.1 Body representation

For realistic modeling of human shapes, we make use of deformed body surfaces attached to the human skeleton, rather than simple geometric primitives representing body links with simple skeleton. The original model is based on metaballs and splines [17] and allows parametric representation of different human bodies. The human skeleton that we use is based on anatomical structure of real skeleton without compromising real-time control; and consists of 74 degrees of freedom without the hands, with additional 30 degrees of freedom for each hand. The skeleton is represented by a 3D articulated hierarchy of joints, each with realistic maximum and minimum limits. We approximate the body links by 16 deformed surfaces (head, pelvis, thorax, abdomen, left and right upper leg, lower leg, foot, upper arm, lower arm, hand). We use three levels of detail for the body surfaces: with a total of 40000 triangular facets with high resolution, 10000 facets for low resolution, and 2500 facets for rough resolution.

7.2 Face representation

Face is one of the main streams of interaction among humans for representing intentions, thoughts and feelings [18]; hence including facial expressions increases interaction especially in multi-user environments. 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 on 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 sent to the other users after optional compression and the receiving side recognizes 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 (Fig. 3). This allows better three-dimensional appearance than simple primitives such as ellipsoid, sphere. The main drawback of the facial module is that it is not possible to incorporate it with conventional stereo devices such as HMD, as the face cannot be captured. However, it provides a good medium of interaction in the lack of these devices.

Fig. 3 The mapping of the face to the virtual actor

8. Applications

As already discussed in Section 1, VLNET is a general-purpose system. We are experimenting the system for various types of applications where interaction between multiple participants is an important factor. In this section, we give examples of the applications of VLNET system.

Teleshopping: The VLNET system has been used for an experience with Chopard Watches Inc., Geneva; to collaboratively view and present the computer-generated models of the recently-designed watches (Fig. 4) with the remote customers and colleagues in Singapore, and Geneva. The experience was shown between Geneva and Singapore during Telecom ‘95. The models have been developed using AutoDesk program, and these were easily included in the virtual environment, with the help of 3D Studio reader for Performer.

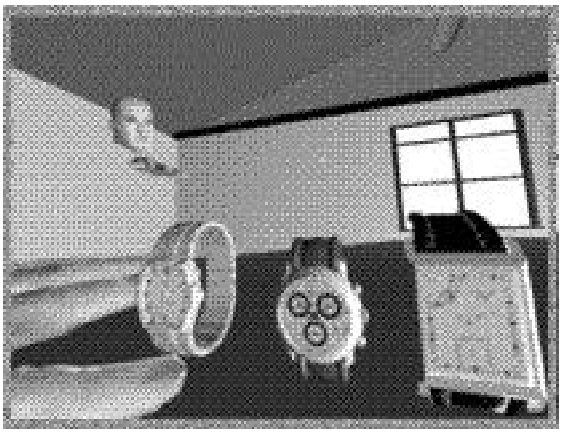

Fig. 4 Application: manipulation of watches

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.

Entertainment: The VLNET environment is also used for playing chess (Fig. 5) between different remote partners; and 3D puzzle (Fig.6) 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.

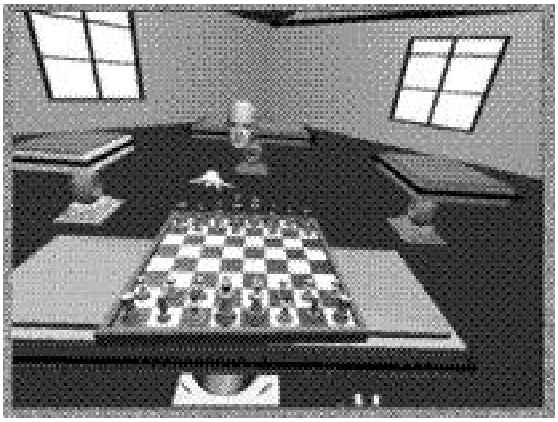

Fig. 5 Chess

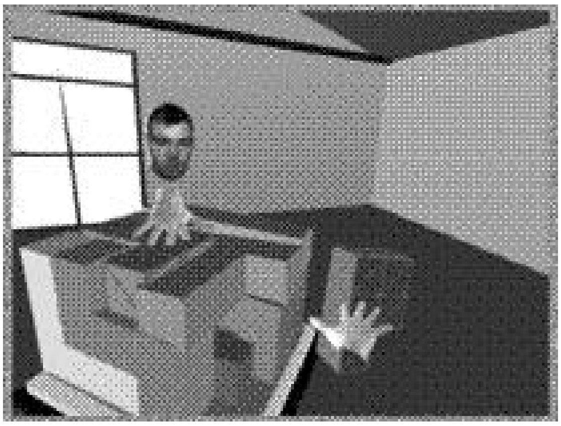

Fig. 6 3D puzzle

9. Implementation

For the virtual environment to be realistic, our system should be fast enough in providing feedback. Therefore, in the VLNET system, we currently make use of the state-of-the-art technology to achieve utmost performance available, but it is widely accepted that these platforms will be used by a large community in a few years' time. There are three ways for speeding-up the system performance: display, communication, simulation.

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. The architecture requires the broadcasting of the data from the server to all the users in the system. To overcome the problem of bottleneck when there are a lot of users, we take advantage 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 [19] in order to emphasize the awareness among the entities in the virtual environment.

The network overhead can 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 [20]. 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 expected to be added in the future, allowing more efficient multiple-user virtual environments.

The simulation part should also be performed efficiently using appropriate mechanism. We make use of the HUMANOID system for modeling and real-time animation of virtual actors [21]. 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

For fast display, we use the IRIS Performer environment [22]. 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 use parallel processors in the workstation for graphics operations. Therefore, it is an appropriate platform to increase the performance of display part of the system.

10. Future Work

Further improvements are to include physical models for interaction, more visual realism by incorporating already-developed body deformations modules, natural language interface with the virtual actors, and sound rendering. We also intend to incorporate more complex behaviors and communication between the three different types of virtual actors.

Acknowledgment

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 the Department of Economy of the State of Geneva. We would like to thank Riccardo Camiciottoli for his 3D Studio driver for Performer.

References

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

2 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.

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

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

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

6 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.

7 Magnenat Thalmann N., Thalmann D., "Artificial life and Virtual Reality", John Wiley and Sons, 1994.

8 Pandzic I., Çapin T., Magnenat Thalmann N., Thalmann D., "VLNET: A Networked Multimedia 3D Environment with Virtual Humans", Proc. Multi-Media Modeling MMM `95, Singapore (to appear)

9 Capin T., Pandzic I.S., Magnenat-Thalmann N., Thalmann D. "Virtual Humans for Representing Participants in Immersive Virtual Environments", Proc. FIVE Conference, London (to appear)

10 Slater and Usoh (1994) "Body centred Interaction in Immersive Virtual Environments", in: (Magnenat Thalmann and Thalmann, eds) Artificial Life and Virtual Reality, John Wiley, Chichester, 1994, pp.1-10

11 Thalmann D., "Using Virtual Reality Techniques in the Animation Process" in: Virtual Reality Systems (Earnshaw R, Gigante M, Jones H eds), Academic Press, pp.143-159.

12 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.

13 Slater M., Steed A., Usoh M., "The Virtual Treadmill: A Naturalistic Method for Navigation in Immersive Virtual Environments", Proceedings of the First Eurographics Workshop on Virtual Environments, 1993.

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

15 Magnenat Thalmann N., Thalmann D., "Digital Actors for Interactive Television", Proc. IEEE, August 1995.

16 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.

17 Shen J., Thalmann D, "Interactive Shape Design Using Metaballs and Splines", Implicit Surface ‘95, Eurographics Workshop on Implicit Surfaces, Grenoble, France, 1995

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

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

20 Swiss Telecom, "ATM Pilot Services and User Interfaces", Swiss Telecom, 1993.

21 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.

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