Playing Games through the Virtual Life Network

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

[1] Computer Graphics Laboratory
Swiss Federal Institute of Technology (EPFL)
CH1015 Lausanne, Switzerland

{noser, capin,thalmann}@lig.di.epfl.ch

[2] MIRALAB - CUI
University of Geneva
24 rue du Général-Dufour
CH1211 Geneva 4, Switzerland
{ipandzic,thalmann}@cui.unige.ch

Abstract

Simulating autonomous virtual actors living in virtual worlds with human-virtual interaction and immersion is a new challenge. The sense of "presence" in the virtual environment is an important requirement for collaborative activities involving multiple remote users working with social interactions. Using autonomous virtual actors within the shared environment is a supporting tool for presence. This combination of Artificial Life with Virtual Reality cannot exist without the growing development of Computer Animation techniques and corresponds to its most advanced concepts and techniques. In this paper, we present a shared virtual life network with autonomous virtual humans that provides a natural interface for collaborative working and games. We explain the concept of virtual sensors for virtual humans and show an application in the area of tennis playing.

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

1. Introduction

One of the most challenging Virtual Reality applications are real-time simulations and interactions with autonomous actors especially in the area of games and cooperative work. In this paper, we present VLNET which is a shared virtual life network with virtual humans that provides a natural interface for collaborative working and games. 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. As a typical example, we will consider tennis playing as shown in Figure 1.

Using VLNET, we will show how it is possible to have players who may be participants, guided or autonomous. For example, a participant could play tennis or chess against another participant or an autonomous actor. A 3D puzzle may be solved by two autonomous actors or a participant with the help of a guided actor. VLNET brings together four essential technologies: networked Computer Supported Cooperative Work (CSCW), Virtual Reality, Artificial Life, and Computer Animation.

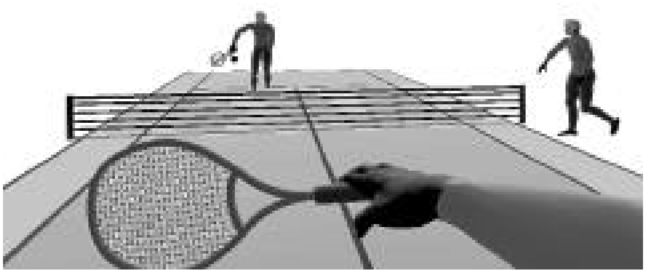

Figure 1. Tennis

In the next Section, we will present an overview of the system, then we will discuss the three types of actors, emphasizing the autonomous actors. Section 5 explains the simulation of virtual sensors for these autonomous actors. Section 6 presents tennis playing and navigation based on these sensors. Section 7 is dedicated to the problem of making autonomous virtual actors aware of participants. Finally, some aspects of implementation are described.

2. The System Architecture

A typical environment for game playing through VLNET is shown in Figure 2. For the real time tennis game simulation with synthetic actors we use a behavioral L-system interpreter. This process shares with the participant client through VLNET clients and the VLNET server the environment elements important for the simulation, as the tennis court, the tennis ball, an autonomous referee, the autonomous player and the participant. The representation of the participant in the L-system interpreter is reduced to a simple racket whose position is communicated to the other clients through the network at each frame. The racket position of the autonomous actor is communicated at each frame to the user client where it is mapped to an articulated, guided actor. This guided actor is animated through inverse kinematics according to the racket position. The referee is also represented by a guided articulated actor in the user client getting its position at each frame from the L-system animation process. The ball movement is modeled according to physical laws in the animation system and communicated to all clients through the network.

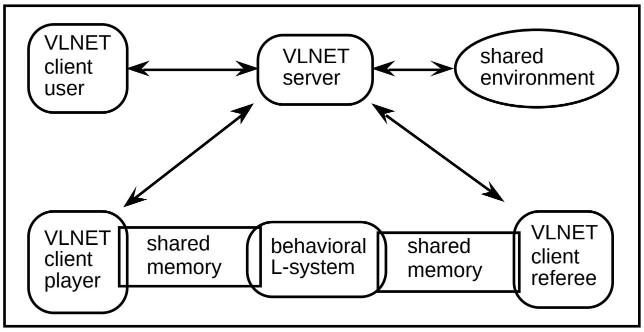

Figure 2. The system architecture

2.1 The VLNET System

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 [1, 2, 3, 4] 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.

The VLNET [5, 6] (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 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 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, 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.

2.2 The Behavioral L-system

The environment of the autonomous actors is modeled and animated with L-systems which are timed production systems designed to model the development and behavior of static objects, plant like objects and autonomous creatures. They are based on timed, parameterized, stochastic, conditional environmentally sensitive and context dependent production systems, force fields, synthetic vision and audition. More details about our L-system-based animation may be found in [7, 8, 9, 10]. Original L-systems [11] were created as a mathematical theory of plant development with a geometrical interpretation based on turtle geometry. Our behavioral L-system [12] is based on the general theory about L-Grammars described in this work.

An L-system is given by an axiom being a string of parametric and timed symbols, and production rules specifying how to replace corresponding symbols in the axiom during the time evolution. The L-system interpreter associates to its symbols basic geometric primitives, turtle control symbols or special control operations necessary for an animation. We define the non generic environment as a tennis court directly in the axiom of the production system. The generic parts as growing plants are defined by production rules having only their germ in the axiom. Actors are also represented by a special symbol.

3. Participant, user-guided and autonomous real-time 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 as 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. 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 is connected with behavioral human animation knowledge and different motion generators in order to "interpolate" the joints of the body which are not tracked.

- **Guided actors**
  A guided actor is an actor completely controlled in real-time by the user. 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 uses the instantaneous velocity of motion, to compute the walking cycle length and time, by which it computes the necessary joint angles. The arm motion for picking an object as a chess pawn (see Figure 3) 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.

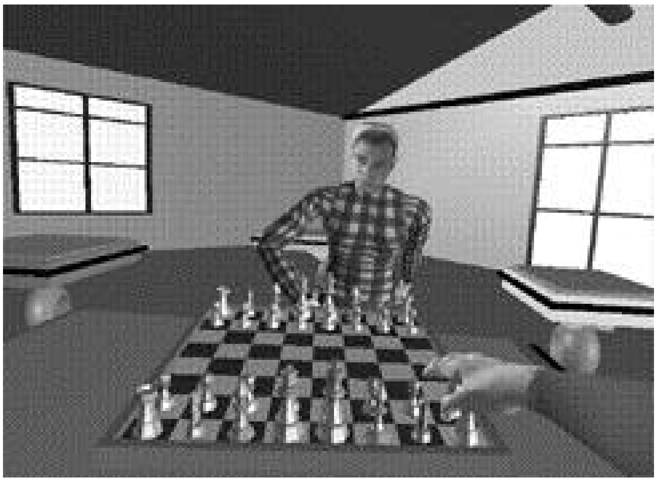

Figure 3. Chess

- **Autonomous actors**
  An autonomous actor is an actor who may act without intervention of the user. In the next sections, the role of the autonomous real-time virtual actor is explained in more details.

4. Autonomous Actors

An autonomous system is a system that is able to give to itself its proper laws, its conduct, opposite to heteronomous systems which are driven by the outside. Guided actors 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 [13] 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.

Simulating autonomous actors should be based on biology and is directly a part of Artificial Life as well as simulation of plants, trees, and animals. This kind of research is also strongly related to the research efforts in autonomous agents [14] and behavioral animation as introduced by Reynolds [15] to study the problem of group trajectories: flocks of birds, herds of land animals and fish schools. Haumann and Parent [16] describe behavioral simulation as a means to obtain global motion by simulating simple rules of behavior between locally related actors. Wilhelms [17] proposes a system based on a network of sensors and effectors. Ridsdale [18] proposes a method that guides lower-level motor skills from a connectionist model of skill memory, implemented as collections of trained neural networks. More recently, genetic algorithms were also proposed by Sims [19] to automatically generate morphologies for artificial creatures and the neural systems for controlling their muscle forces. Tu and Terzopoulos [20] described a world inhabited by artificial fishes.

Our autonomous virtual actors [21] 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 as it will be described in details in the next section.

5. Virtual Sensors

The problem of simulating the behavior of a virtual actor in an environment may be divided into two parts: 1) provide to the actor a knowledge of his environment, and 2) to make react him to this environment.

The first problem consists of creating an information flow from the environment to the actor. This synthetic environment is made of 3D geometric shapes. One solution is to give the actor access to the exact position of each object in the complete environment database corresponding to the synthetic world. This solution could work for a very " small world ", but it becomes impracticable when the number of objects increases. Moreover, this approach does not correspond to reality where people do not have knowledge about the complete environment.

Another approach has been proposed by Reynolds [13]: the synthetic actor has knowledge about the environment located in a sphere centered on him. Moreover, the accuracy of the knowledge about the objects of the environment decreases with the distance. This is of course a more realistic approach, but as mentioned by Reynolds, an animal or a human being has always around him areas where his sensitivity is more important. Consider, for example, the vision of birds (birds have been simulated by Reynolds), they have a view angle of 300º and a stereoscopic view of only 15º. The sphere model does not correspond to the sensitivity area of the vision. Reynolds goes one step further and states that if actors can see their environment, they will improve their trajectory planning.

More generally, in order to implement perception, virtual humans 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. Simulating the haptic system corresponds roughly to a collision detection process. But, the most important perceptual subsystem is the vision system. A vision 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. In the next sections, we describe our approach for the three types of virtual sensors: vision, audition and haptic.

5.1 Virtual vision

Although the use of vision to give behavior to synthetic actors seems similar to the use of vision for intelligent mobile robots [22, 23], it is quite different. This is because the vision of the synthetic actor is itself a synthetic vision. Using a synthetic vision allow us to skip all the problems of pattern recognition and distance detection, problems which still are the most difficult parts in robotics vision. However some interesting work has been done in the topic of intelligent mobile robots, especially for action-perception coordination problems. For example, Crowley [24], working with surveillance robots states that "most low level perception and navigation tasks are algorithmic in nature; at the highest levels, decisions regarding which actions to perform are based on knowledge relevant to each situation". This remark gives us the hypothesis on which our vision-based model of behavioral animation is built.

We first introduced [25] the concept of synthetic vision as a main information channel between the environment and the virtual actor. Reynolds [26, 27] more recently described an evolved, vision-based behavioral model of coordinated group motion, he also showed how obstacle avoidance behavior can emerge from evolution under selection pressure from an appropriate measure using a simple computational model of visual perception and locomotion. Tu and Terzopoulos [20] also use a kind of synthetic vision for their artificial fishes. In [29] , each pixel of the vision input has the semantic information giving the object projected on this pixel, and numerical information giving the distance to this object. So, it is easy to know, for example, that there is a table just in front at 3 meters. With this information, we can directly deal with the problematic question: "what do I do with such information in a navigation system?" The synthetic actor perceives his environment from a small window of typically 30x30 pixels in which the environment is rendered from his point of view. As he can access z buffer values of the pixels, the color of the pixels and his own position he can locate visible objects in his 3D environment. This information is sufficient for some local navigation.

We can model a certain type of virtual world representation where the actor maintains a low level fast synthetic vision system but where he can access some important information directly from the environment without having to extract it from the vision image. In vision based grasping for example, an actor can recognize in the image the object to grasp. From the environment he can get the exact position, type and size of the object which allows him to walk to the correct position where he can start the grasping procedure of the object based on geometrical data of the object representation in the world. This mix of vision based recognition and world representation access will make him fast enough to react in real time. The role of synthetic vision can even be reduced to a visibility test and the semantic information recognition in the image can be done by simple color coding and non shading rendering techniques. Thus, position and semantic information of an object can be obtained directly from the environment world after being filtered.

5.2 Virtual audition

In real life, the behavior of persons or animals is very often influenced by sounds. For this reason, we developed a framework for modeling a 3D acoustic environment with sound sources and microphones. Thus, our virtual actors are able to hear [10]. For example, tennis playing with sound effects (ball-floor and ball-racket collision) and simple verbal player - referee communication has been realized. The synthetic actor can directly access to positional and semantic sound source information of an audible sound event. This allows him to localize and recognize sound sources in a reliable way and to react immediately.

5.3 Virtual tactile

At basic level, human should sense physical objects if any part of the body touches them and gather sensory information. This sensory information is made use of in such tasks as reaching out for an object, navigation etc. For example if a human is standing, the feet are in constant contact with the supporting floor. But during walking motion each foot alternately experiences the loss of this contact.

Traditionally these motions are simulated using dynamic and kinematic constraints on human joints. But there are cases where information from external environment is needed. For example when a human descends a stair case, the motion should change from walk to descent based on achieving contact with the steps of the stairway. Thus the environment imposes constraints on the human locomotion. We propose to encapsulate these constraints using tactile sensors [28] to guide the human figure in various complex situations. In our work, the sphere multi-sensors have both touch and length sensor properties, and have been found very efficient for synthetic actor grasping problem (see Figure 4).

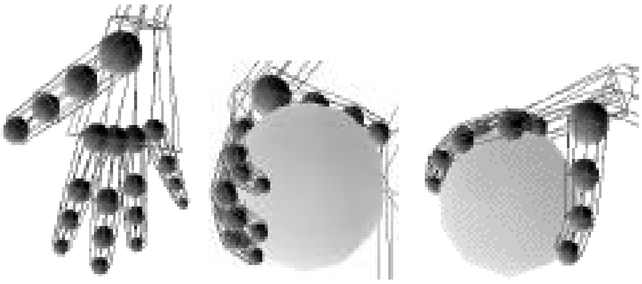

Figure 4. Tactile sensors

Another type of tactile sensor has been implemented in the L-system animation system where parts of the environment are modeled by a force field based particle system which, for example, animates the tennis ball dynamics of the tennis game. To simulate the tactile sensor we defined a function used also in conditions of production rules evaluating the global force field at the position given by the parameters x, y, z of the query symbol **?** [29]. The query symbol **?** makes part of environmentally sensitive L-systems and sets its parameters x, y and z during the interpretation step of the formal symbol string to the actual position of the turtle. Thus, at the following time step, this position data can be used through the parameters in the conditions of the production rules. The force field function returns the amount of the global force field at the actual turtle position and by comparing this value with a threshold value, a kind of collision detection with force field modeled environments can be simulated.

6. Perception-based behaviors

Synthetic vision, audition and tactile allow the actor to perceive the environment. Based on this information, his behavioral mechanism will determine the actions he will perform. Actions may be at several degrees of complexity. An actor may simply evolve in his environment or he may interact with this environment or even communicate with other actors. As we emphasize the aspect of game, sensor-based tennis playing, game judging and navigation are the most important behaviors. A behavior can be composed of other behaviors and basic actions. A basic action is in general a motor procedure allowing an actor to move. A high level behavior uses often sensorial input, special knowledge and basic actions or other high level behaviors. To model behaviors we use an automata approach. Each actor has an internal state which can change each time step according to the currently active automata and its sensorial input. To control the high level behavior of an actor we use a stack of automata for each actor. At the beginning of the animation the user furnishes a sequence of behaviors (a script) and pushes them on the actor's stack. When the current behavior ends the animation system pops the next behavior from the stack and executes it. This process is repeated until the actor's behavior stack is empty. Some of the behaviors use this stack too, in order to reach subgoals by pushing itself with the current state on the stack and switching to the new behavior allowing them to reach the subgoal. When this new behavior has finished the automata pops the old interrupted behavior and continues. This stack based behavior control helps an actor to get more autonomous and to create his own subgoals while executing the original script.

6.1 Tennis playing based on vision, audition and tactile sensors

We modeled a synthetic sensor based tennis match simulation for autonomous players and an autonomous referee, implemented in the L-system based animation system. The different behaviors of the actors are modeled by automata controlled by an universal stack based control system. As the behaviors are severely based on synthetic sensors being the main channels of information capture from the virtual environment we obtain a natural behavior which is mostly independent of the internal environment representation. By using this sensor based concept the distinction between a digital actor and an interactive user merged into the virtual world becomes small and they can easily be exchanged as demonstrated with the interactive game facility.

The autonomous referee judges the game by following the ball with his vision system. He updates the state of the match when he "hears" a ball collision event (ball - ground, ball - net, ball - racket) according to what he sees and his knowledge of a tennis match, and he communicates his decisions and the state of the game by "spoken words" (sound events).

A synthetic player can also hear sound events and he obeys the decisions of the referee. The player's game automata uses synthetic vision to localize the ball's and his opponent's position and he adaptively estimates the future ball -racket impact point and position. He uses his partner's position to fix his game strategy and to plan his stroke and his path to the future impact point.

6.2 Vision-based navigation

In cooperative work as a 3D puzzle (Figure 5), actors have to walk through the virtual space and it requires a navigation system. More generally, the task of a navigation system is to plan a path to a specified goal and to execute this plan, modifying it as necessary to avoid unexpected obstacles [23].

We may distinguish two types of navigation methods: The **local navigation** algorithm uses the direct input information from the environment to reach goals and sub-goals given by the **global navigation** and to avoid unexpected obstacles. The local navigation algorithm has no model of the environment, and doesn't know the position of the actor in the world. In our approach, local navigation is based on the concept of Displacement Local Automata (DLA). These DLAs work as a black box which has the knowledge to create goals and sub-goals in a specific local environment. They can be thought of as low-level navigation reflexes which use vision, reflexes which are automatically performed by the adults.

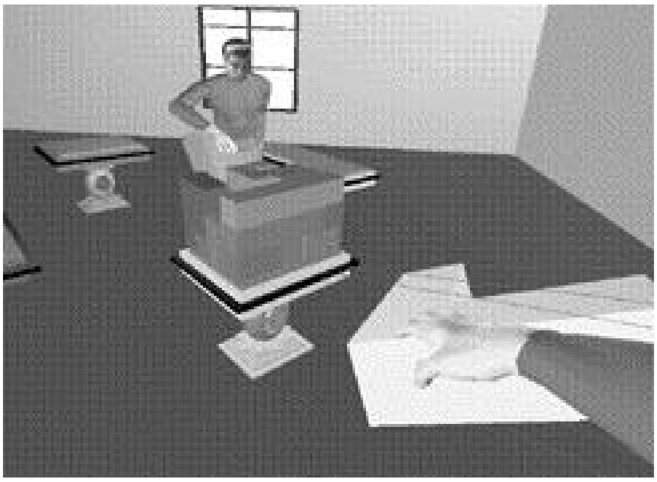

Figure 5. 3D puzzle

The global navigation needs a prelearned model of the environment to perform path-planning. This model is constructed with the information coming from the sensory system. Most navigation systems developed in robotics for intelligent mobile robots are based on the accumulation of accurate geometrical descriptions of the environment. Kuipers et al. [30] give a nearly exhaustive list of such methods using quantitative world modeling. In robotics, due to low mechanical accuracy and sensory errors, these methods have failed in large scale area. We don't have this problem in Computer Graphics because we have access to the world coordinates of the actor, and because the synthetic vision or other simulations of perception systems are more accurate. Elfes [31] proposed a 2D geometric model based on grid but using a Bayesian probabilistic approach to filter non accurate information coming from various sensor positions. Roth-Tabak [32] proposed a 3D geometric model based on a grid but for a static world. In our approach [33], we use an octree as the internal representation of the environment seen by an actor because it offers several interesting features. With an octree we can easily construct voxelized objects by choosing the maximum depth level of the subdivision of space. Detailed objects like flowers and trees do not need to be represented in complete detail in the problem of path searching. It is sufficient to represent them by some enclosing cubes corresponding to the occupied voxels of the octree. The octree adapts itself to the complexity of the 3D environment, as it is a dynamic data structure making a recursive subdivision of space. The octree has to represent the visual memory of an actor in a 3D environment with static and dynamic objects. Objects in this environment can grow, shrink, move or disappear.

7. Making virtual actors aware of real ones

The participants are of course easily aware of the actions of the virtual humans through VR tools like Head-mounted display, but one major problem to solve is to make the virtual actors conscious of the behavior of the participants. Virtual actors should sense the participants through their virtual sensors (Figure 6).

We have seen that virtual vision is a powerful tool in modeling virtual autonomous actors. Such actors can have different degrees of autonomy and different sensing channels to the environment

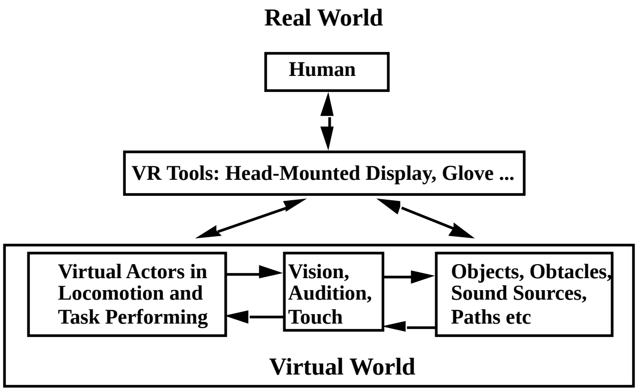

Figure 6. Real and virtual sensors

Let us now consider the case of a participant playing tennis with an autonomous virtual human. The participant can participate in VR by the head-mounted display and the earphones. He cannot get any internal VR information. His only source of knowledge from the VR is communicated by the vision, the sound, and some tactile sensory information. His behavior is strongly influenced by this sensory input and his proper intelligence. In order to process the vision of the autonomous virtual actor in a similar way than the vision of the participant, we need to have a different model. In this case, the only information obtained by the autonomous virtual actor will be through the virtual vision looking at the participant actor. Such an autonomous virtual actor would be independent of each VR representation (as a human too) and he could in the same manner communicate with human participants and other autonomous virtual actors.

For virtual audition, we encounter the same problem as in virtual vision. The real time constraints in VR demand fast reaction to sound signals and fast recognition of the semantic it carries.

Concerning the tactile sensor, we may consider the following example: the participant places an object into the Virtual Space using a CyberGlove and the autonomous virtual actor will try to grasp it and put it on a virtual table for example. The actor interacts with the environment by grasping the object and moving it. At the beginning of interactive grasping, only the hand center sensor is active. The six palm values from the CyberGlove are used to move it towards the object. Inverse kinematics update the arm postures from hand center movement. After the sensor is activated, the hand is close enough to the object final frame. The hand center sensor is deactivated and multi-sensors on hand are now used, to detect sensor object collision.

8. Implementation

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 actors in the 3D 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. 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. 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. An experience was held during Telecom ‘95 between Singapore and Geneva using ATM.

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 [34]. The HUMANOID environment supports all the facilities real-time manipulation of virtual actors on a standard graphics workstation.

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 [35] 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.

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.

L-systems are given by ASCII text files which can be created and edited by any text editor. Our L-system animation system is capable to read several L-system text files and to interpret them in parallel during an animation. One L-system text file typically models for example the growth and topology of a plant or a tree or contains the rules of some actor behavior.

For fast display, we use the IRIS Performer environment [36]. 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.

9. Future Work

Further improvements are to include physical models for interaction, 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.

Acknowledgments

The research was partly supported by ESPRIT project HUMANOID-2, Swiss National Foundation for Scientific Research, l'Office Fédéral de l'Education et de la Science, and the Department of Economy of the State of Geneva.

References

1. 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.

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

3. Stansfield S., "A Distributed Virtual Reality Simulation System for Simulational Training", Presence: Teleoperators

and Virtual Environments, Vol. 3, No. 4, 1994.

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

5. Pandzic I., Çapin T., Magnenat Thalmann N., Thalmann D., "VLNET: A Networked Multimedia 3D Environment with Virtual Humans", Proc. Multi-Media Modeling MMM `95, World Scientific, Singapore

6. Thalmann D., Capin T., Magnenat-Thalmann N., Pandzic I.S., " Participant, User-guided, and Autonomous Actors in the Virtual Life Network VLNET ", Proc. ICAT/VRST '95, Chiba, Japan, November 1995, pp.3-11.

7. Noser H, Thalmann D, Turner R, *Animation based on the Interaction of L-systems with Vector Force Fields*, Proc. Computer Graphics International '92, pp. 747-761

8. Noser H. Thalmann D., *L-System-Based Behavioral Animation*, Proc. Pacific Graphics 93, Aug. 1993, World Scientific Publishing Co Pte Ltd, pp. 133-146

9. H. Noser, D. Thalmann, *Simulating Life of Virtual Plants, Fishes and Butterflies*, in: Magnenat Thalmann N. and D. Thalmann, Artificial Life and Virtual Reality, John Wiley, 1994.

10. Noser H., Thalmann D., S*ynthetic Vision and Audition for Digital Actors*, Computer Graphics Forum, Vol. 14. Number 3, Proc. Eurographics '95, pp. 325 -336, 1995

11. P. Prusinkiewicz, A. Lindenmaer, The Algorithmic Beauty of Plants (1990), Springer Verlag

12. Noser H., Thalmann D., The Animation of Autonomous Actors Based on Production Rules, Proc. Computer Animation '96, IEEE Computer Society Press, 1996

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

14. Maes P. (ed.) "Designing Autonomous Agents", Bradford MIT Press, 1991.

15. Reynolds C. "Flocks, Herds, and Schools: A Distributed Behavioral Model", Proc.SIGGRAPH '87, Computer Graphics, Vol.21, No4, 1987, pp.25-34

16. Haumann D.R., Parent R.E. "The Behavioral Test-bed: Obtaining Complex Behavior from Simple Rules", The Visual Computer, Vol.4, No 6, 1988, pp.332-347.

17. Wilhelms J. "A "Notion" for Interactive Behavioral Animation Control", IEEE Computer Graphics and Applications , Vol. 10, No 3 , 1990, pp.14-22

18. Ridsdale G. "Connectionist Modelling of Skill Dynamics", Journal of Visualization and Computer Animation, Vol.1, No2, 1990, pp.66-72.

19. Sims K. "Evolving Virtual Creatures", Proc. SIGGRAPH '94, 1994, pp. 15-22.

20. Tu X., Terzopoulos D. "Artificial Fishes: Physics, Locomotion, Perception, Behavior", Proc. SIGGRAPH '94, Computer Graphics, 1994, pp.42-48.

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

22. Horswill I. "A Simple, Cheap, and Robust Visual Navigation System", in: From Animals to Animats 2, Proc. 2nd Intern. Conf. on Simulation of Adaptive Behavior, MIT Press, 1993, pp.129-136.

23. Tsuji S, Li S. "Memorizing and Representing Route Scenes", in: Meyer J.A. et al. (eds) From Animals to Animats, Proc. 2nd International Conf. on Simulation of Adaptive Behavior, MIT Press, 1993, pp.225-232.

24. Crowley J.L. "Navigation for an Intelligent Mobile Robot", IEEE journal of Robotics and Automation, Vol. RA-1, No. 1, 1987, pp 31-41.

25. Renault O., Magnenat Thalmann N., Thalmann D. (1990) "A Vision-based Approach to Behavioural Animation", The Journal of Visualization and Computer Animation, Vol 1, No 1, pp 18-21.

26. Reynolds C.W. "An Evolved, Vision-Based Behavioral Model of Coordinated Group Motion", in: Meyer J.A. et al. (eds) From Animals to Animats, Proc. 2nd International Conf. on Simulation of Adaptive Behavior, MIT Press, 1993, pp.384-392.

27. Reynolds C.W. "An Evolved, Vision-Based Model of Obstacle Avoidance Behavior", in: C.G. Langton (ed.), Artificial Life III, SFI Studies in the Sciences of Complexity, Proc. Vol. XVII, Addison-Wesley, 1994.

28. Huang Z., Boulic R., Magnenat Thalmann N., Thalmann D. (1995) "A Multi-sensor Approach for Grasping and 3D Interaction", Proc. CGI '95

29. P. Prusinkiewicz, M. James, R. Mech, *Synthetic Topiary* , SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series, 1994, pp. 351-358.

30. Kuipers B., Byun Y.T. "A Robust Qualitative Approach to a Spatial Learning Mobile Robot", SPIE Sensor Fusion: Spatial Reaoning and Scene Interpretation, Vol. 1003, 1988.

31. Elfes A. "Occupancy Grid: A Stochastic Spatial Representation for Active Robot Perception", Proc. Sixth Conference on Uncertainty in AI, 1990.

32. Roth-Tabak Y. "Building an Environment Model Using Depth Information", Computer, 1990, pp 85-90.

33. Noser H., Renault O., Thalmann D., Magnenat Thalmann N. (1995) "Navigation for Digital Actors based on Synthetic Vision, Memory and Learning", Computers and Graphics, Pergamon Press, Vol.19, No1, pp.7-19.

34. 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", Computer Graphics Forum, Vol.14. No 3, Proc.Eurographics '95, 1995.

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

36. Rohlf J., Helman J., "IRIS Performer: A High

Performance Multiprocessing Toolkit for Real-Time 3D Graphics", Proc. SIGGRAPH'94, 1994.