The Visual Computer

VHD: a system for directing real-time virtual actors

Gaël Sannier[1], Selim Balcisoy[2], Nadia Magnenat-Thalmann[1], Daniel Thalmann[2]

[1] MIRALab, University of Geneva, 24 rue du Général Dufour, CH 1211 Geneva, Switzerland
[2] Computer Graphics Laboratory, Swiss Federal Institute of Technology, EPFL, LIG, CH 1015 Lausanne, Switzerland

In this paper, we present a new system dedicated to the real-time production of multimedia products and TV shows involving multiple virtual actors and presenters. This system, named VHD (Virtual Human Director), may also be used for character animation on high-level systems and interaction with Web-based applications. The user can create virtual stories by directing all actions in real time. It is similar to a movie producer directing actors during filming but with more control given to the director in the sense that everything is decided and controlled by the same person. To make this complicated task possible, in a useable way, we based the interaction on the high-level control of the virtual actor.

Key words: Authoring tool – Virtual humans – Real-time facial animation – Body animation – Motion capture – Virtual speech – Virtual presenter

1 Introduction

Though virtual human models have been in existence for many years (Badler et al. 1993; Magnenat-Thalmann and Thalmann 1991), they have been mainly used for research purposes to enable the simulation of human movements and behaviors. Only recently has there been any encouraging interest from outside the academic world, especially for multimedia products and TV applications. Traditional character animation techniques are time-consuming tasks without any real-time possibilities. Design and animation of such characters is a time-demanding operation as it is done using a general-purpose animation system like Softimage, Alias-Wavefront, Maya, or 3D StudioMax. Furthermore, character animation systems developed for high quality graphical output cannot interact with any other media spaces such as web-based applications, television, or telecommunication.

Traditional multimedia systems are systems that handle different forms of data, such as text, audio, or video. Until recently, the Web has represented the typical traditional multimedia system with HTML text, gif images, and movies. Then came VRML the Virtual Reality Modeling Language: a way of creating 3D scenes allowing the Web users to walk through 3D spaces. VRML supports the integration of virtual reality with the World Wide Web with the goal of broadening access to VR environments via the WWW infrastructure. Generally, virtual environments define a new interface for networked multimedia applications. Users will be able to move in virtual spaces where they can find many virtual objects, including virtual humans. These virtual objects should be multimedia objects. For example, a virtual human should not only be a 3D graphical object, but also an object able to speak or emit sound. A virtual dog should not only be able to move, but also to bark. Virtual humans have potential applications in entertainment and business products such as film, computer games, and as TV talk-show hosts. New applications are involved with new ways of interacting between real and virtual entities.

VHD is a system which aims to provide interactive control of real-time virtual humans. In this paper, after giving an overview of the system, we present the body and facial animation modules. Then we describe the user-friendly interface, and finally we present some results, where we tested VHD with broadcasting partners of the European Project VISTA.

The Visual Computer (1999) 15:320–329

2 System overview

Controlling virtual humans is still a tedious task. A lot of work has been done in the area of providing easy control of their behavior in the virtual environment. Nevertheless, if typical systems are mainly focussed on the immersive aspects, just a few are interested in controlling the virtual actors for the purpose of "virtual story" elaboration. However, with The JACK system, Badler et al. (1993) focussed their work on providing realistic behavioral control of human models. Based on a similar framework, Motivate (http://www.motion-factory.com/home.html) was developed for the creation of 3D games. In Improv (Perlin and Goldberg 1996) provide tools to define the behavior of their animated actors. Their goal is more to "educate" virtual actors, being able to react by themselves to real humans or virtual events, than to give to the user the possibility of having specific control of each virtual human. The goal of our system is to provide tools that will allow a user to create any scenario involving realistic virtual humans. One other key aspect is to keep the ability to react in real-time to some external events (for example, a virtual presenter interviewing a real guest) while showing a high-quality rendered virtual human. This system is very similar to what is used by a real puppeteer, except that more control is given to the user due to the complexity of animation that can be triggered.

In this paper, we do not discuss animation techniques as in (Kalra et al. 1998), but we present a new kind of system dedicated to the real-time production of multimedia products and TV shows. This system, named VHD may also be used for character animation on high-end systems and interaction with Web-based applications.

VHD focuses on two important issues:

- *Fully integrated virtual humans with facial and body animation, and speech.*
- *A straightforward user interface for designers and directors.*

VHD system provides a range of virtual human animation and interaction capabilities integrated into one single environment. Our software architecture allows the addition of different interfaces from distinct environments. VHD actors can be controlled via a standard TCP/IP connection over any network by using our virtual human message protocol as shown in Fig. 1. Using this protocol, possible high-level AI software can also be coded to control multiple actors.

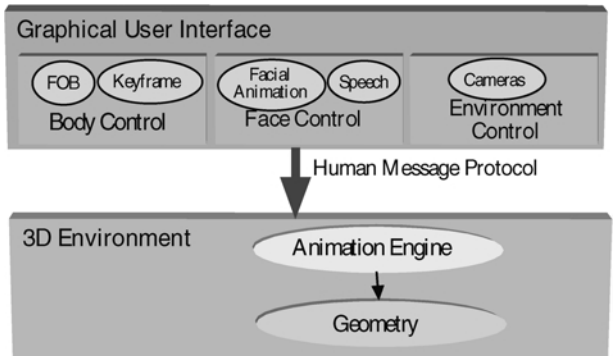


Fig. 1. System overview

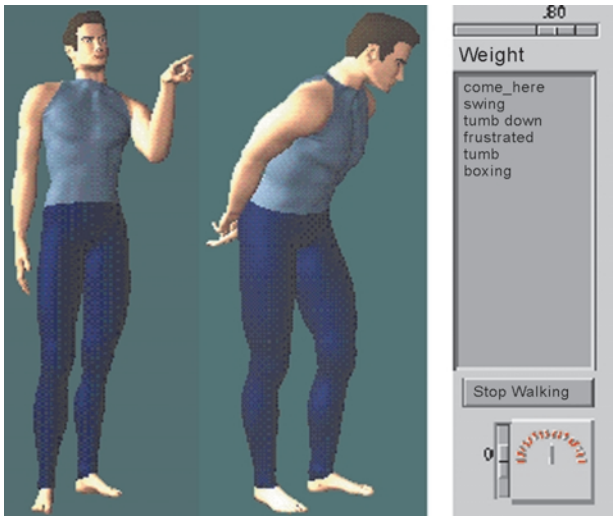
3 Real-time animation modules for body

In VHD, we have two types of body motion: predefined gestures and task-oriented motion. Predefined gestures are prepared using keyframe animation and motion capture. For task oriented motions like walking we use motion motors.

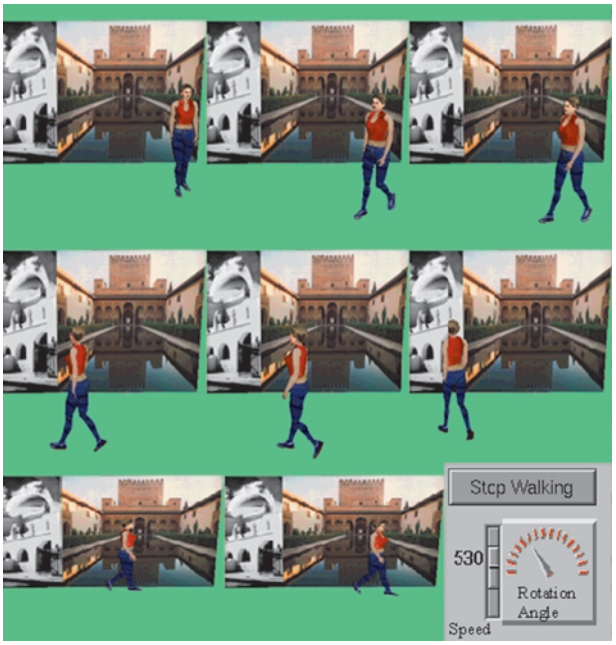
3.1 Motion capturing and predefined postures

A traditional way of animating virtual humans is playing keyframe sequences. We can record specific human body postures or gestures with a magnetic motion capturing system and an anatomical converter (Molet et al. 1996), or we can design human postures or gestures using the TRACK system (Boulic et al. 1994).

Motion capturing can be best achieved by using a large number of sensors to register every degree of freedom of the real body. Molet et al. (1996) consider that a minimum of 14 sensors are required to manage a biomechanically correct posture. The raw data coming from the trackers has to be filtered and processed to obtain a usable structure. Our software permits the conversion of raw tracker data into joint angle data for all 75 joints in the skeleton.



2



3

Fig. 2. Keyframe examples with user interface
Fig. 3. Walking example with walking interface

TRACK is an interactive tool for the visualization, editing and manipulation of multiple track sequences. To record an animation sequence we create key positions from the scene, then store the 3D parameters as 2D tracks of the skeleton joints. The stored keyframes, from the TRACK system or magnetic tracker, can be used to animate the virtual human in real time. We use predefined postures and gestures to perform realistic hand and upper body gestures for interpersonal communications. Figure 2 presents some predefined postures and a body animation part of the user interface.

Users can trigger gestures and postures from a list on the main interface. Moreover, users can automate the activation of gestures and postures from the interface. A good example is automating "neutral" postures: usually when nothing is happening, e.g., when an actor is not performing any animation, the actor looks "frozen". It is not easy for a user to activate random postures to move the actor slightly every few seconds. To overcome this problem we developed a programmable posture trigger, where a user can decide the interval and randomness of a posture. The right-most image in Fig. 1 shows details from the user interface. The upper slider "Weight" defines the weight of current active keyframe in case of a motion merge with other body animation. Under the slider there is a selectable list of available keyframes.

3.2 Motion motors

We used one motion motor (Boulic et al. 1990) for the body locomotion. Current walking motors enable virtual humans to travel in the environment using instantaneous velocity of motion. One can compute the walking cycle-length and time from which the necessary skeleton joint angles can be calculated for animation. This instantaneous speed-oriented approach has influenced VHD user interface design, where user is directly changing the speed. On the other hand VHD also supports another module for controlling walking, where users can control walking with simple commands like "WALK_FASTER" or "TURN_LEFT". This simple interface allows specific user interfaces to talk with VHD more easily.

A possible user interface is a touch phone to control walking. In this case setting speed and direction is impossible with our first method. The latter one solves this problem in a natural way where we can map our commands to each key on the number pad. Figure 3 includes snapshots from a walking session in one picture.

In many cases actors are expected to perform multiple body motions like waving while walking. Our agent-based software (Boulic et al. 1992) coordinates multiple AGENTS with multiple ACTIONS. AGENTLib also manages natural motion blending.

VHD is built on the AGENTLib library for body animation, motion planning and motion blending.

4 Real-time animation and speech for face

For virtual actors and presenters, facial animation is essential. In this section we will describe our method for animating a 3D-face model. The animation is driven by high-level actions such as "smile", "surprise", which also control the head deformations. For the speech part, different approaches have been proposed. Pearce et al. (1986) use a string of phonemes to generate the corresponding animation of the 3D face, while Cohen and Massaro (1993) start with an English text as input to generate the speech. In our system we combine both elements to generate speech. We extended our facial animation model to speech capabilities by combining these facial animations with the output of a text-to-audio system at the phoneme level.

4.1 Real-time facial animation system

Globally, the process of face animation is decomposed into several layers of information as shown in Fig. 4.

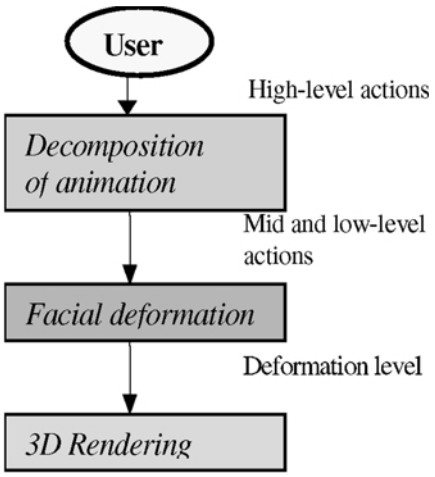


Fig. 4. Layer of information to perform facial animation

- The high-level actions concern the emotions, the sentences and the head movements of the virtual actor. The animator completely directs the virtual clone with actions from this level.
  Emotions are interpreted as an evolution of face over time. It is defined as starting from a neutral state, passing through a sequence of visible changes, and returning to a neutral state. A library of standard emotions can be defined, including smile, anger, surprise, fear, etc., but specific emotions like "virtual twitch" can also be created to personalize the virtual human.
  For speech animation, each sentence is decomposed into phonemes. To each phoneme corresponds a viseme, which is the phoneme counterpart in terms of facial animation. A total of 44 visemes are used. Examples of visemes are given in Fig. 5. From the animation point of view, a sentence is a sequence of temporized visemes. Nevertheless, during a conversation, a human mouth is not able to articulate precisely each phoneme: visemes are blended with one another through time. In order to smooth the animation, we mix the beginning of a viseme with the end of the previous one.
  Finally, the head movements can be controlled by setting the intensity of motion over time.
- The mid-level actions can be defined as expressions of the face. They are considered as a facial snapshot modulated in time and intensity to make the high-level actions. A facial snapshot is composed by a set of low-level action units.
- The low-level actions are defined as 63 face regions. Each region corresponds to a facial muscle. An intensity value is associated to each region describing its deformation. These intensities are called 'Minimum Perceptible Actions' (MPA). For each frame of the facial animation, an array of 63 MPAs is provided, defining the state of the face at this frame.
- The deformation of the face is performed using Rational Free Form Deformation (RFFD) applied on regions of the mesh corresponding to the facial muscles (Kalra et al. 1992). The role of our facial animation library is to compute, at every activation of high-level action, a list of MPA frames. To get the face deformation at a given time, this library composes, one by one, every MPA of the time-right frame of each activated action. It allows the mixing of high-level actions in real-time (for example smiling while speaking). The result-

Fig. 5. A virtual head model with its corresponding visemes for the indicated words

ing array of MPA is transmitted to the face animation module that computes the deformation of the mesh. This regions-based method is totally mesh/head independent, as long as the 63 regions are well defined for each object.

4.2 Speech

The input text is converted into temporized phonemes using a text-to-speech synthesis system. In this case we are using the Festival Speech Synthesis System that is being developed at the University of Edinburgh (Black and Taylor 1997). It also produces the audio stream that will be subsequently played back in synchronization with the facial animation system. Using the temporized phonemes, we are able to create the facial animation by concatenating the corresponding visemes through time. Most of time, we use the set of phonemes present in the Oxford English Dictionary. Nevertheless, an easy extension to any language could be done by designing the corresponding visemes. As a case study, we also extend our model to the German language.

The synchronization of face movement with sound is initially done by starting the two processes at the same time. In order to keep synchronization during all the speech, we use the sound playback as a time reference. By knowing beforehand the total length of the sound, the number of frames of the animation (given by the temporized visemes), and the current time, the synchronization can be done easily by skipping frames in case of delay.

5 Software issues on integration

In order to be able to optimize quality, a computer can be completely dedicated to the rendering using IRIS Performer (Rohlf and Helman 1994) libraries, while another one is used only for the interface. The constraint, for the interface, is to remotely control the animation in real time.

For that purpose, a very simplified client-server protocol was established as shown in Fig. 6. The interface communicates directly with the 3D application using TCP/IP sockets over a network. It allows the computing of a real-time full-screen rendered animation while also having a whole screen dedicated to the interface. In order to make the connection possible between the interface and the 3D application, a communication protocol has been established. As a consequence, any interface using the same protocol for communication is able to control the 3D environment.

A new concept can be developed based on the idea of a networked PC-based interface as shown in Fig. 7. Each client of this network will have control of one actor. A simplified lower-quality rendering could be displayed on each PC. A specific client would control the cameras. The server of this network, obtaining all the data from all its clients, could use the communication protocol for directing the 3D application in order to render a high quality output of the scene in real time. Such a PC connection is planned for the near future using a JAVA server with VRML worlds, where home users connect to a multi-user VRML media space. VHD can

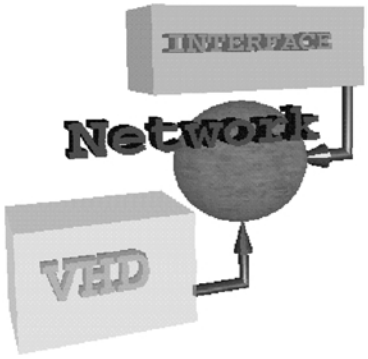
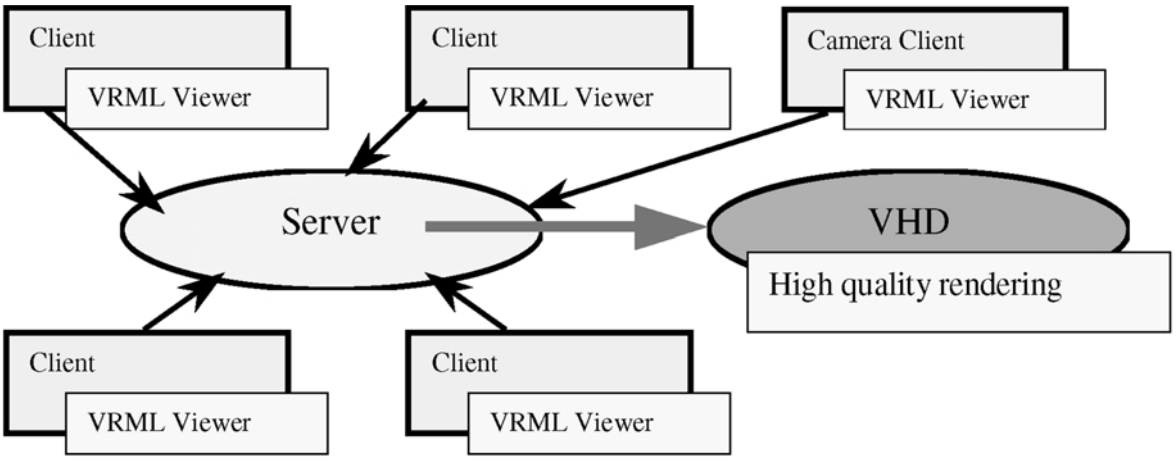


Fig. 6. Simplified client-server system between the interface and the 3D application

Fig. 7. A client-server network linked with VHD output

6



7

be used as a flying camera over a multi-user world. The Director can shoot interesting interactions with a virtual camera.

6 Interface design issues

In the previous sections, we described a system for animating virtual humans with regard to quality constraints. We will now present aspects of interactivity developed in this system.

The issue of easily controlling a virtual face was raised in (Magnenat-Thalmann et al. 1998). The goal of the VHD interface is an extension of this concept. VHD provides an easy way to control multiple actors and cameras in real time. Thus, the user can create virtual stories by directing all the actions in real time. It is similar to a producer directing actors during shooting but with more control given to the producer in the sense that everything is decided and controlled by the same person. To make this complicated task possible and useable, we provide high-level control of the virtual actors. Tools were developed to be able to predefine actions in advance so that during real-time playing, the producer can concentrate on the main guidelines of his scenario (sentences for example).

Only high-level actions can be used with the interface. It allows control of the speech by typing/selecting simple text-based sentences. The facial animation is controlled by pre-recorded sequences. These facial animations can be mixed in real time and also mixed with speech. Control of the body is done through keyframing and motion motors for walking.

As soon as we have many virtual actors to control, the number of available actions will make the task

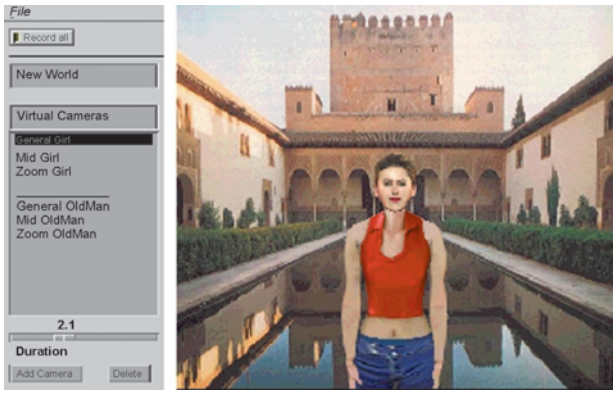


Fig. 8. Interface setting the camera position to "General View"

of the director more and more complex. In order to ease this complicated task for real-time interaction, we provide several tools for pre-programming actions in advance. The user can give a time after which the action will be played. Nevertheless, the idea is more useful for repeating actions. For example, we can program the eye blinking of an actor every five seconds. However, not all the actions can be programmed this way. As the goal is to be able to play the scenario in real time, we want to let the control of the main actions to be in the hands of the director. Nevertheless, a lot of actions result from a few main events. Let's consider the sentence "I am very pleased to be here with you today". The user may want to have the virtual actor smiling after something like one second (while saying: "pleased") and move the right hand after 1.5 seconds (saying: "here"). So the idea is to pre-program actions to be played after the beginning of a main event which is the sentence. Then, just by selecting the main actions, complex behavior of the virtual actors will be completely determined. However, the user will still be able to mix other actions to the pre-programmed ones.

Basic virtual camera tools are also given to the user. New camera positions can be fixed interactively from the interface. Cameras can be attached to virtual humans, so that we can have a total shot and a close-up of a virtual actor whatever his/her position is on the virtual scene. During real time, the list of camera positions is available on the interface, and the user can switch from one camera to the other just by clicking on it. An interpolation time can be easily set to provide zooming and traveling options between two camera positions. Figure 8 shows a snapshot of camera control interface. The cameras are also considered as an extension of the actions. They can be programmed in advance so when an actor says a sentence, the camera can be programmed to go directly to a given position.

In order to improve the building of virtual stories, a scripting tool was developed for recording all the actions being directed to an actor. Each action is recorded into a file with its activation time. Then, the user can adjust the timing of the sequence and play it again in real time. During playback, new actions can be triggered. As the saving is done independently for each actor and for the cameras, one can program each actor separately. At the end, all the scripts can be played at once. The other important aspect is to be able to program background actors. Then, only important actors will be directed "live".

7 Results

We tested our software extensively within the European project VISTA, with broadcasting partners, to produce TV programs. After several tests with designers and directors, our concept of one single integrated platform for virtual humans obtained good feedback. Our interface has a straightforward concept and the high level of control allows designers to control virtual humans in a natural way. We included several traditional camera operations into our interface to allow directors to use VHD without a steep learning curve.

This case study presents how VHD can be used to make interactive events. The first stage is compilation of the scenario. Designers then create virtual humans. According to the scenario, the production team records necessary predefined body gestures, facial animations and sentences. After this, the first trials may start with VHD. The director generates specific camera shots and checks all parts of the scenario. The director and the production team record and edit the complete scenario with the scripting capabilities of VHD. Some complex parts of a real-time show can be recorded as a script to have a perfectly synchronized result. Today, the main interest is in the mixing of real and virtual elements in the same environment. Production houses use several virtual set applications to attract view-

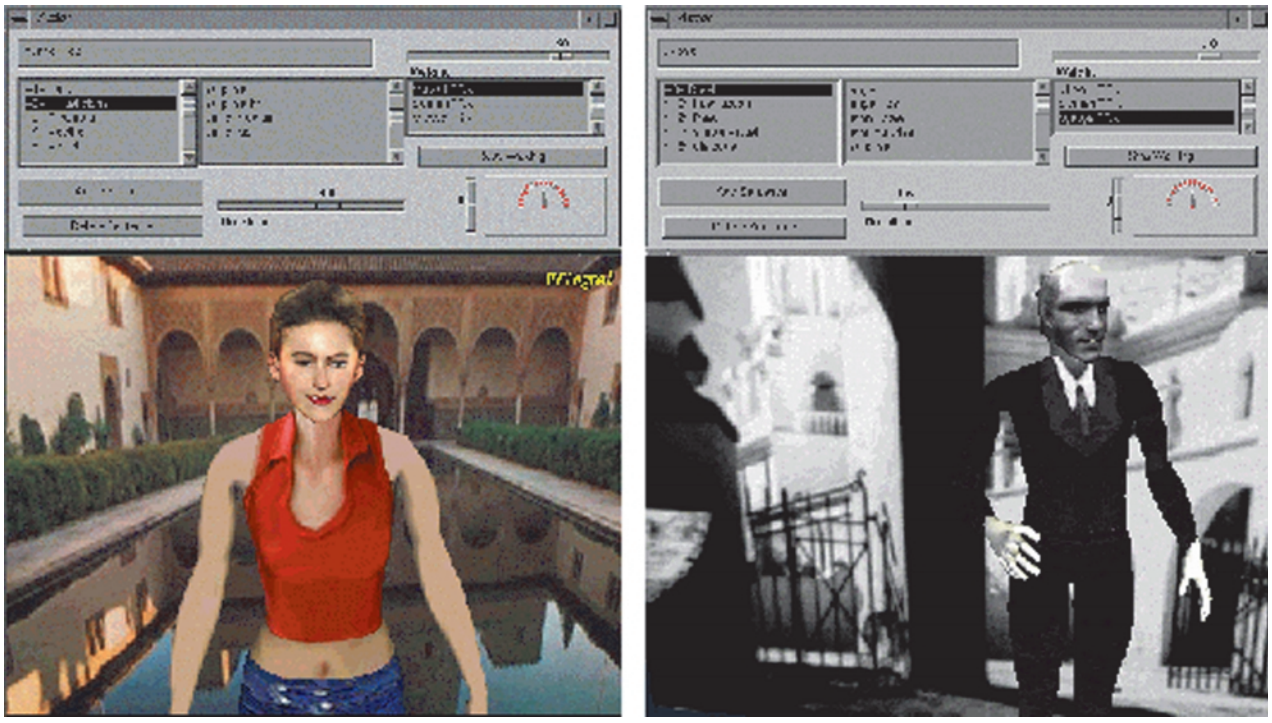


Fig. 9. Examples of the interface controlling virtual actors

ers by using mixed environments. Virtual sets include multiple components like a rendering system, an animation management system and a camera calibration system. In our work with broadcasting companies we had to embed VHD into such a large system. Preset positions for static camera shots and other environmental parameters are set by the interface. A major problem was synchronization of real and virtual cameras (when the real camera switches from position A to B). To have correct synchronization we developed a serial port interface for VHD. Through this interface a real camera switch sends the number of the active real camera to VHD. As such a connection can be used for dynamic cameras, we are planning to develop a dynamic camera tracking protocol in the near future.

Figure 9 displays snapshots from a VHD session where designers are testing a conversation between two virtual actors.

In all the design steps of VHD we tried to keep real time as our main target. We used multiple processes for communication, sound rendering (speech) and animation. We used a 4-processor SGI Onyx2 with a single Infinite Reality Pipe for timings in the first diagram of Table 1. VHD renders and animates up to 4 actors at acceptable rates. The frame size is fixed to PAL size.

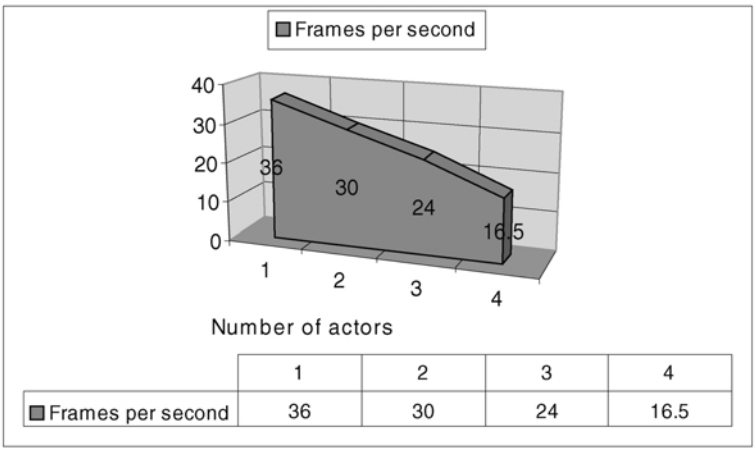
We also tested VHD with a close-up virtual head made of 2313 triangles and fully textured. The second diagram in Table 1 presents the number of frames per second we got on our trials on the following computers using full screen (1280×1024) and a 400×500 pixel window:

1- O2 R5000 at 200 MHz
2- Octane R10000 at 195 MHz
3- Onyx2 2x R10000 at 195 MHz

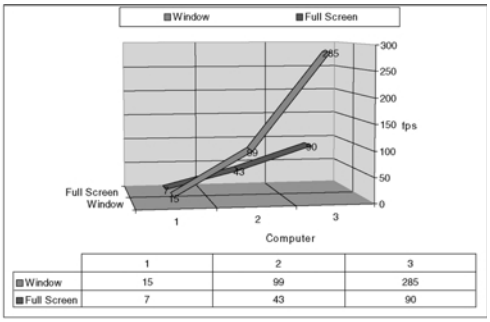
8 Conclusion and future work

In this paper we have presented a novel system for integrated virtual human animation. Our software, VHD, has been developed to enable a non-computer scientist to interact with virtual humans in several

Table 1. Real-time performance diagrams



	1	2	3	4
Frames per second	36	30	24	16.5



	1	2	3
Window	15	99	285
Full Screen	7	43	90

ways. A designer can create a full story by directing, and animating several virtual actors. A home user can get connected via their home PC to a studio and control one virtual actor (this option requires special PC software). A director and a team of artists can create a complex event, where virtual actors interact with real actors.

For many AI programs VHD can be used to create a visual environment. VHD also provides an open platform to extend the capabilities of virtual humans without any major difficulties.

In the near future we are planning to add new capabilities to VHD. Several high-level motion/task controls and editing facilities will be integrated to create mini scripts. Automatic grasping of virtual objects will be included. We have already started to integrate an AI system to trigger reactions to facial emotions, sentences, and pre-recorded keyframes for the body. We will also look for integration with computer vision-based tracker software to have a fully integrated augmented reality system. In order to improve the audio control of our virtual actors, we are working for direct-speech animation control with the use of a microphone.

Acknowledgements. The authors would like to thank MIRALab and LIG teams for their support, and Frank Alsema from VPro for his intuitive ideas and constructive criticism. We would also like to thank Chris Joslin for proof-reading this document. The European Project VISTA and the Swiss National Foundation for Scientific Research supported this research.

References

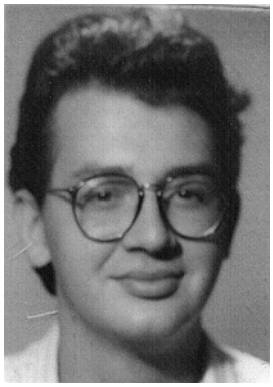
1. Badler NI, Philips BC, Webber LB (1993) Simulating humans. Computer Graphics Animation and Control, Oxford University Press
2. http://www.motion-factory.com/home.html
3. Perlin K, Goldberg A (1996) Improv: A system for scripting interactive actors in virtual worlds. Comput Graph (SIGGRAPH '96 Proceedings) pp 205–216
4. Magnenat-Thalmann N, Thalmann D (1991) Complex models for animating synthetic actors. IEEE Comput Graph Appl 11(5):32–44
5. Kalra P, Mangenat-Thalmann N, Moccozet L, Sannier G, Aubel Á, Thalmann D (1998) Real-time animation of realistic virtual humans. IEEE Comput Graph Appl 18(5):42–56
6. Molet T, Boulic R, Thalmann D (1996) A real-time anatomical converter for human motion capture. In: Boulic R, Herdon G (eds) Eurographics Workshop on Computer Animation and Simulation, Springer, Wien, pp 79–94, ISBN 3-211-828-850
7. Boulic R, Huang Z, Mangenat-Thalmann N, Thalmann D (1994) Goal oriented design and correction of articulated figure motion with the TRACK system. Comput Graph 18(4):443–452
8. Boulic R, Thalmann D, Magnenat-Thalmann N (1990) A global human walking model with real-time kinematic personification. Visual Comput 6(6):344–358
9. Boulic R, Becheizaz P, Emering L, Thalmann D (1992) Integration of motion control techniques for virtual human and avatar real-time animation. Proc. VRST '97, pp 111–118, ISBN 0-89791-953-*x*
10. Pearce A, Hill DR, Wyvill B (1986) Speech and expression: A computer solution for face animation. Proc. Graphics Interface '86, Vision Interface '86, pp 136–140

11. Cohen MM, Massaro DW (1993) Modeling coarticulation in synthetic visual speech, Magnenat-Thalmann N, Thalmann D (eds) Models and techniques in computer animation Springer, Tokyo, pp 139–156
12. Kalra P, Mangili A, Mangnenat-Thalmann N, Thalmann D (1992) Simulation of facial muscle actions based on rational free form deformation. Proc Eurographics '92, pp 59–69
13. Black A, Taylor P (1997) Festival speech synthesis system: System documentation (1.1.1). Technical Report HCRC/TR-83, Human Communication Research Center, University of Edinburgh
14. Rohlf J, Helman J (1994) IRIS performer: A high performance multiprocessing toolkit for real-time 3D graphics. Proc. SIGGRAPH '94, ACM Press
15. Magnenat-Thalmann N, Kalra P, Escher M (1998) Face to virtual face. Proc Comput Animation '98

DANIEL THALMANN is a pioneer in research on virtual humans. His current research interests include real-time virtual humans in virtual reality, networked virtual environments, artificial life, and multimedia. He is coeditor-in-chief of the Journal of Visualization and Computer Animation, member of the editorial board of the Visual Computer, the CADDM Journal (China Engineering Society) and Computer Graphics (Russia). He is cochair of the EUROGRAPHICS Working Group on Computer Simulation and Animation and member of the Executive Board of the Computer Graphics Society. Daniel Thalmann was member of numerous Program Committees, Program Chair of several conferences and chair of the Computer Graphics International '93, Pacific Graphics '95, and ACM VRST '97 conferences. He has also organized 4 courses at SIGGRAPH on human animation. He has published more than 200 papers in Graphics, Animation, and Virtual Reality. He is coeditor of 25 books, and coauthor of several books including: Computer Animation: Theory and Practice and Image Synthesis: Theory and Practice. He is also codirector of several computer-generated films with synthetic actors including a synthetic Marilyn Monroe shown on numerous TV channels all over the world.

NADIA MAGNENAT–THALMANN has pioneered research into virtual humans over the last 20 years. She studied psychology, biology and chemistry at the University of Geneva and obtained her PhD in computer science (cum laude) in 1977. In 1989 she founded MIRAlab, an interdisciplinary creative research laboratory at the University of Geneva. Some recent awards for her work include the 1992 Moebius Prize for the best multimedia system awarded by the European Community, "Best Paper" at the British Computer Graphics Society congress in 1993, election to the Academy of Image by the Belgium Television in Brussels in l993, and election as a Member at the Swiss Academy of Technical Sciences, in l997. She is president of the Computer Graphics Society and chair of the IFIP Working Group 5.10 in computer graphics and virtual worlds.

GAEL SANNIER is a computer scientist who has studied in Lyon (master degree in Computer Graphics in 1996). He is now working at the University of Geneva as a research assistant in MIRALab participating in research being done there. His main tasks are user-friendly texture-fitting methods and virtual presenters for TV applications.



SELIM BALCISOY is currently research assistant in the Computer Graphics Laboratory at Swiss Federal Institute of Technology (EPFL). He graduated from ETHZ in Electronics in 1996. His research interests include mixing real and synthetic worlds, augmented reality and vision, and human action recognition.