An Interactive Interface for Directing Virtual Humans

Gael 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

Abstract. Research on Virtual Humans spans from body animation to speech generation. In many cases research systems are isolated software pieces and can only be used after a steep learning curve. In this paper a novel system is presented which integrates a large repertoire of Virtual Human research in one piece of user-friendly software, Virtual Human Director (VHD). This software achieves two important goals without any decrease in real-time performance or graphics quality. One goal is the integration of all the existing real-time virtual human technology from two research laboratories into open software architecture. The other one is to develop an intuitive user interface, which allows artists and directors to work with Virtual Humans.

1. Introduction

Though virtual human models have been in existence, they have been used mainly for research purposes to enable the simulation of human movements and behaviors. Only recently there has been any encouraging interest from outside the academic world. Virtual humans have potential applications in entertainment and business products such as films, computer games, and as TV talk-show host. New applications are involved with new ways of interacting between real and virtual entities.
On the other hand, research on virtual humans produces several solid software components. In this paper, a novel system, VHD, is proposed for integrating these software components. 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 into our software. VHD actors can be controlled via a standard TCP/IP connection over any network by using our virtual human message protocol. Possible high-level AI software can also be coded to control multiple actors.

We first describe briefly the real-time animation modules for body and face. Then we will describe the user interface. Finally, we present some results, where we tested VHD with broadcasting partners of the European Project VISTA.

2. 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.

2.1 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 [7], or we can design human postures or gestures using the TRACK system [8].

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. [7] discuss 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 standard HUMANOID skeleton [6].

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 used predefined postures and gestures to perform realistic hand and upper body gestures for interpersonal communications. Figure 1 presents some predefined postures.

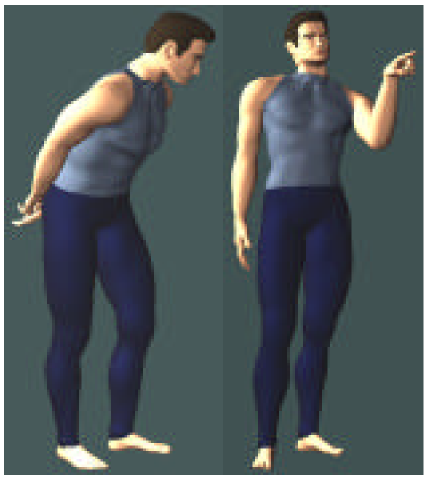

Figure 1. Keyframe examples

2.2 Motion motors

We used one motion motor for the body locomotion. This walking motor was developed by Boulic [10]. Current walking motor enables 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 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”. Figure 2 includes snapshots from a walking session in one picture.

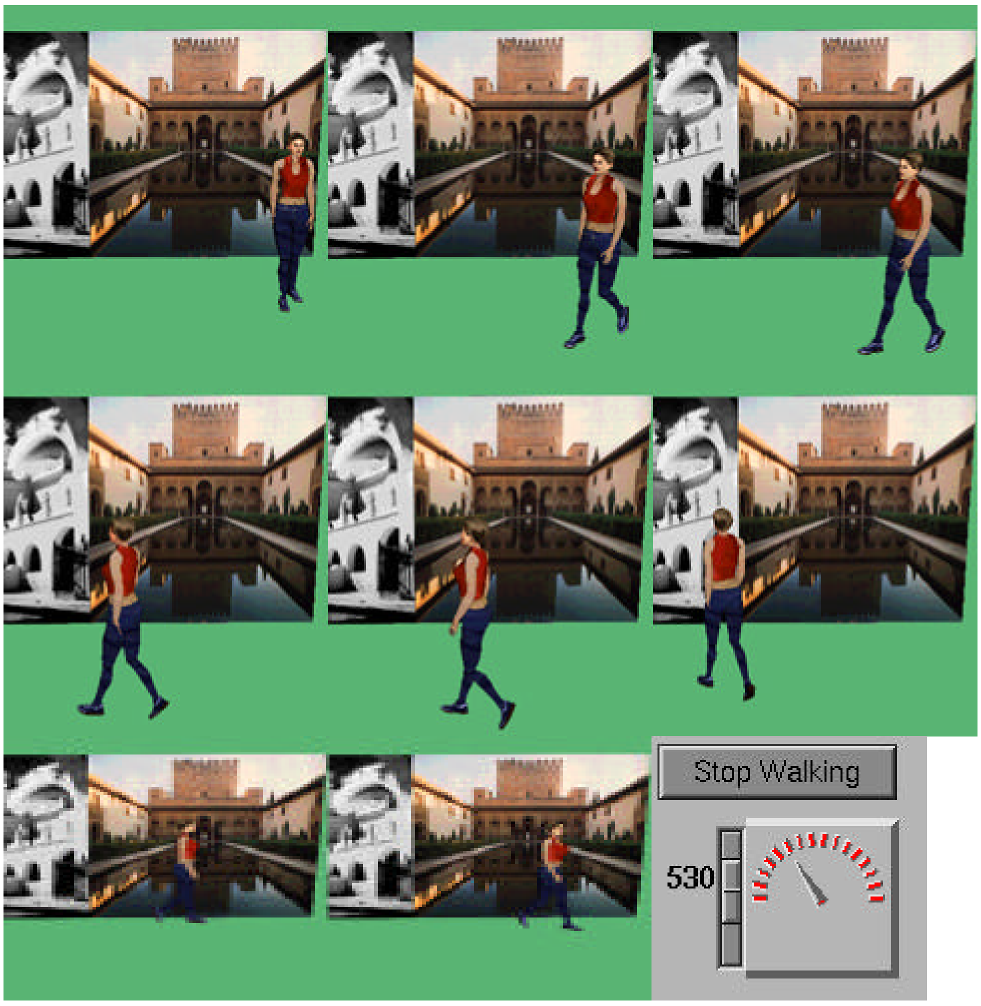

Figure 2. Walking example with walking interface

In many cases actors are expected to perform multiple body motions like waving while walking. AGENTLib [10] 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.

3. Real-time animation and speech for face

In this section we will describe our method for animating a 3D-face model. Different approaches have been proposed for the synthesis of talking heads. Pearce et al. [1] uses a string of phonemes to generate the corresponding animation of the 3D-face, while Cohen and Massaro [2] start with an English text as input to generate the speech. In our system we combine both elements to generate the speech. The animation is driven by high-level actions such as "smile", "surprise", which also control the head deformations. We extended our model to speech capabilities by combining these facial animations with the output of a text-to-audio system at the phoneme level.

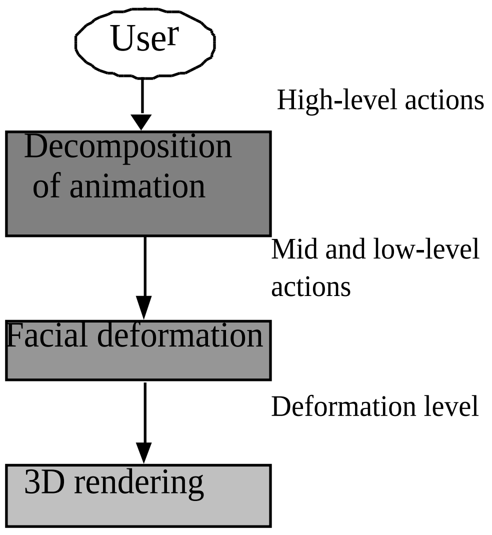

Figure 3. Layer of information to perform the facial animation.

3.1. Real-time facial animation system

Globally, the process of face animation is decomposed into several layers of information as shown in Figure 3.

The high-level actions concern the emotions, the sentences and the head movements of the virtual actor. The animator direct completely the virtual clone with actions from this level. Emotions are interpreted as an evolution of face over time. It is defined as starting from 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 the 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 Figure 4. From the animation point of view, a sentence is just a sequence of temporized visemes. Finally, the head movements can be controlled by the variation of their intensity over time.

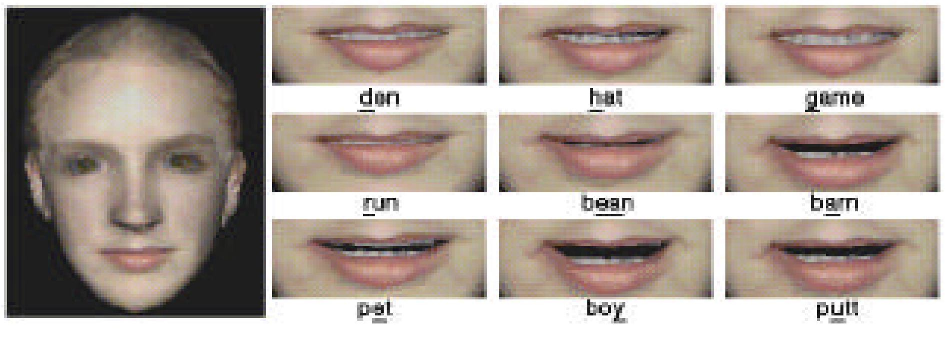

Figure 4. A virtual head model with its corresponding visemes for the indicated words.

The mid-level actions can be defined as expressions of the face. They are considered as facial snapshot modulated in time and intensity to make the high-level actions. A facial snapshot is composed by a set of low-level actions unit.

The low-level actions are defined as 63 regions of the face. 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 [3]. The role of our facial animation library is to compute, at every activation of high-level action, a list of

MAPs' frame. 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 resulting 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.

3.2. Speech

The input text is being 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 Edinburg [4]. It produces also the audio stream that will be subsequently playback 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. The facial animation system is based on the system described by Kalra [5]. We have limited the set of phonemes to the ones used in the Oxford English Dictionary, but an easy extension to any language could be done by designing the corresponding visemes.

The synchronization of face movement with sound is initially done by starting the two processes at the same time. In order to keep the 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.

4. Software issues on integration

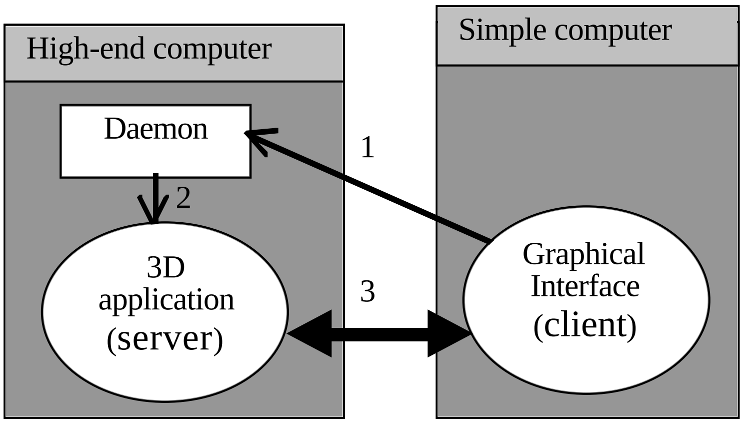

Figure 5. Simplified client-server system between the interface and the 3D application.

In order to be able to optimize the quality, a computer can be completely dedicated to the rendering aspect using IRIS Performer [11] 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 Figure 5. A daemon is run on the computer dedicated to the animation (a high-end computer). This daemon launches the 3D-application (arrow 2 of Figure 5) each time it gets a connection from a user (arrow 1 of Figure 5). The interface then communicates directly with the 3D-application using sockets (arrow 3 of Figure 5). 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. It results that 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. Each client of this network will have the control of one actor. A simplified lower-quality rendering would be displayed on each PC. A specific client would control the cameras. The server of this network, getting all the data of 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.

5. Interface design issues

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

The goal of the VHD interface was to provide an easy control of 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 a shooting but with more controls 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 the real-time playing, the director 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 animations can be mixed in real-time and also mixed with the facial animation. Control of the body is done through keyframing and motion motors for the walking.

As soon as we have many virtual actors to control, the number of available actions will make the task of the director more and more complex. In order to ease this complicated task for the 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 plus a value between zero and two seconds. However all the actions cannot be programmed this way. As the goal is to be able to play 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. 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 played on an actor. Then, the user can adjust the sequence and play it again in real-time. As the saving is done independently for each actor and for the cameras, one can program the actors one by one, and play all the script together at the end. The other idea is to program background actors in order to be able to pay more attention to the main action being played in real-time.

6. Results

We tested our software extensively within 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 got a good feedback. Our Interface has a straightforward concept and highest 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 any steep learning curve. Figure 6 display snapshots from a VHD session.

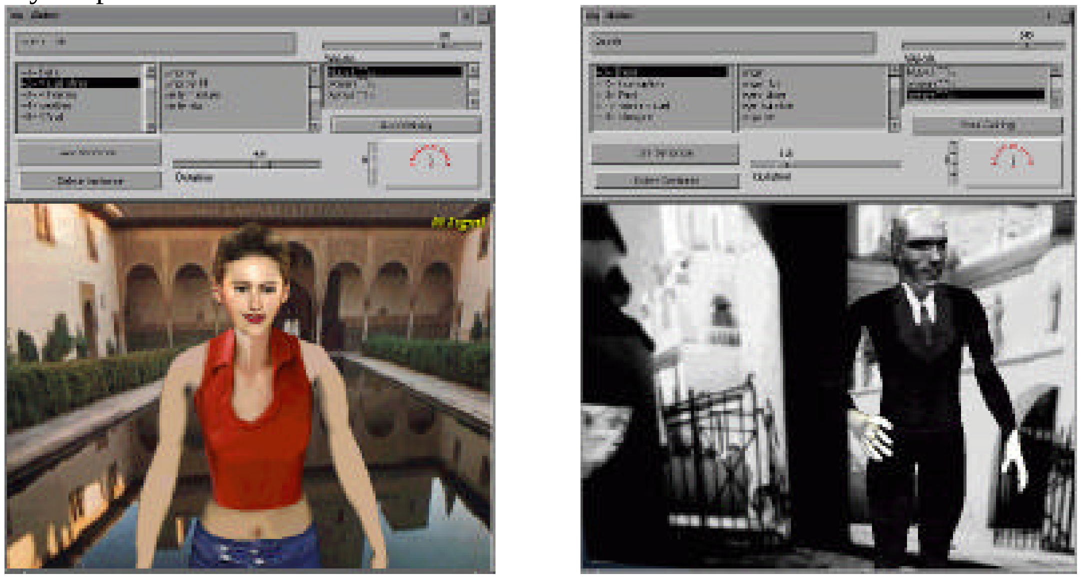

Figure 6. Examples of interface controlling virtual actors.

7. Conclusion and Future Work

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

- a designer can create a full story by directing, animating several virtual actors,
- a home user can get connected by their home PC to a studio and control one virtual actor (this option requires special PC software), or
- 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 provides also an open platform to extend capabilities of virtual humans without major difficulties.

In near future we are planning to add new capabilities to VHD:

- several high level motion/task controlling and editing facilities to create mini scripts,
- automatic grasping of virtual objects,
- integrated an AI system to trigger reactions to facial emotions, sentences, pre-recorded keyframes for body,
- integration with computer vision based tracker software to have a fully integrated augmented reality system, and
- direct-speech animation control with the use of a microphone.

8. 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 proofreading this document. This research was supported by the European Project VISTA and Swiss National Foundation for Scientific Research.

References

[1] A. Pearce, B. Wyvill, G. Wyvill, D. Hill (1986) Speech and Expression: A Computer Solution to Face Animation, Proc. Graphics Interface ’86, Vision Interface ’86, pp. 136-140.

[2] M. M. Cohen, D. W. Massaro (1993) Modeling Coarticulation in Synthetic Visual Speech, eds N. M. Thalmann, D. Thalmann, Models and Techniques in Computer Animation, Springer-Verlag, Tokyo, pp. 139-156.

[3] P. Kalra, A. Mangili, N. Magnenat Thalmann, D. Thalmann (1992) Simulation of Facial Muscle Actions Based on Rational Free Form Deformation, Proc. Eurographics ’92, pp. 59-69.

[4] A. Black, P. Taylor (1997) Festival Speech Synthesis System: System Documentation (1.1.1), Technical Report HCRC/TR-83, Human Communication Research Center, University of Edinburgh.

[5] P. Kalra (1993) An Interactive Multimodal Facial Animation System, Ph.D. Thesis, Ecole Polytechnique Federale de Lausanne.

[6] R. Boulic, Z. Huang, J. Shen, T. Molet, T. Capin, B. Lintermann, K. Saar, D. Thalmann, N. Magnenat-Thalmann, A. Schmitt, L. Moccozet, P. Kalra, I. Pandzic (1995) A System for the Parallel Integrated Motion of Multiple Deformable Human Characters with Collision Detection, Proc. EUROGRAPHICS'95 Maastricht, Computer Graphics Forum, 14(3), pp. 337-348.

[7] T. Molet, R. Boulic, D. Thalmann (1996) A Real Time Anatomical Converter for Human Motion Capture, Eurographics Workshop on Computer Animation and Simulation, R. Boulic and G. Hegron (Eds.), ISBN 3-211-828-850, Springer-Verlag Wien, pp. 79-94.

[8] R. Boulic, Z. Huang, N. Magnenat Thalmann, D.Thalmann (1994) Goal Oriented Design and Correction of Articulated Figure Motion with the TRACK system, Computers and Graphics, Pergamon Press, Vol. 18, No. 4., pp. 443-452.

[9] R. Boulic, D. Thalmann, N. Magnenat-Thalmann (1990) A Global Human Walking Model with Real-Time Kinematic Personification, The Visual Computer, Vol. 6, No. 6, pp. 344-358.

[10] R. Boulic, P. Becheiraz, L. Emering, D. Thalmann (1997) Integration of Motion Control Techniques for Virtual Human and Avatar Real-Time Animation, Proc. VRST'97, pp. 111-118, ISBN 0-89791-953-x.

[11] J. Rohlf, J. Helman (1994) IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics, Proc. SIGGRAPH’94, ACM Press.