The Virtual Human as a Multimodal Interface

Daniel Thalmann
Computer Graphics Lab, EPFL
Lausanne, Switzerland
thalmann@lig.di.epfl.ch
http://ligwww.epfl.ch

ABSTRACT

This paper discusses the main issues for creating Interactive Virtual Environments with Virtual Humans emphasizing the following aspects: creation of Virtual Humans, gestures, interaction with objects, multimodal communication.

Keywords

Virtual Humans, Gestures, Action Recognition, Multimodal Communication

1. INTRODUCTION

As our world is three-dimensional, 3D Virtual Environments are now essential in multimedia systems. Virtual Reality techniques have introduced a wide range of new methods and metaphors of interaction with these Virtual Environments. Virtual Environments are generally composed of static and dynamic Virtual Entities and may include 3D sound. Inside these Virtual Environments, Virtual Humans are a key technology that can provide Virtual presenters, Virtual Guides, Virtual actors, and be used to show how humans should act in various situations (see Figure 1).

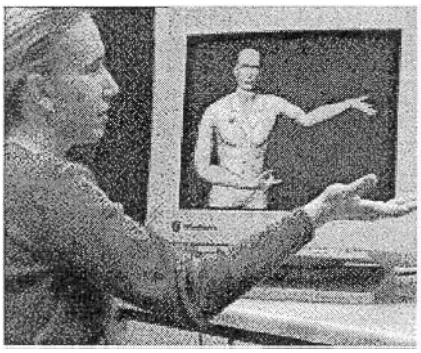

Figure 1. A Virtual Human Interface

They may even become Virtual substitutes in the near future. A virtual substitute is an intelligent computer-generated agent able to act instead of the real person and on behalf of this person on the network. The virtual substitute has the voice of the real person and his or her appearance. He/she will appear on the screen of the workstation/TV, communicate with people, and have predefined behaviours planned by the owner to answer to the requests of the people.

The merging of recent developments in Virtual Reality, Human Animation and Video Analysis techniques has led to the integration of Virtual Humans in Virtual Reality, our interaction with these virtual humans, and our self representation as a clone or avatar or participant in the Virtual World. Interaction with Virtual Environments may be at various level of user configuration. A high end configuration could involve an immersive environment where users would interact by voice, gesture and physiological signals with virtual humans that would help them explore their digital data environment, both locally and over the Web. For this, Virtual Humans should be able to recognize gestures, speech and expressions of the user and answer by speech and animation. The ultimate objective in creating realistic and believable virtual actors is to build intelligent autonomous virtual humans with adaptation, perception and memory. These actors should be able to act freely and emotionally. Ideally, they should be conscious and unpredictable. This paper discusses the main issues for creating Interactive Virtual Environments with Virtual Humans emphasizing the following aspects:

- Creation of Virtual Humans
- Facial and body gestures
- Interaction with objects
- Multimodal communication

2. CREATION OF VIRTUAL HUMANS

Real-time representation and animation of virtual human figures has been a challenging and active area in Computer Graphics since early eighties. Typically, an articulated structure corresponding to the human skeleton is needed for the control of the body posture. Structures representing the body shape have to be attached to the skeleton, and clothes may be wrapped around the body shape. Typically, the skeleton is represented by a 3D articulated hierarchy of 50 or 70 joints, each with realistic maximum and minimum limits. The skeleton is encapsulated with geometrical, topological, and inertial characteristics of different body limbs. The body structure has a fixed topology template of joints, and different body instances may be created by specifying scaling parameters.

In order to represent the body shape, it is necessary to attach to the skeleton a structure which should be deformed during the motion. A solution is to represent only the skin as a surface; the problem is how to generate realistic shapes for any configuration of the skeleton. Another solution is to simulate the inner layers: muscles,

AVI 2000, Palermo, Italy.

bones, soft tissues. This approach is too expensive for real-time actors. A compromise solution is to approximate the muscles and the bones using primitives like ellipsoids or more generally kinds of implicit surfaces, called in the jargon of computer graphics, metaballs, blobs or soft objects. The skin or body shape is generated from the inner structure. One way of doing this is to cut the metaballs into cross sections and generate the skin from these sections. Figure 2 shows the principle.

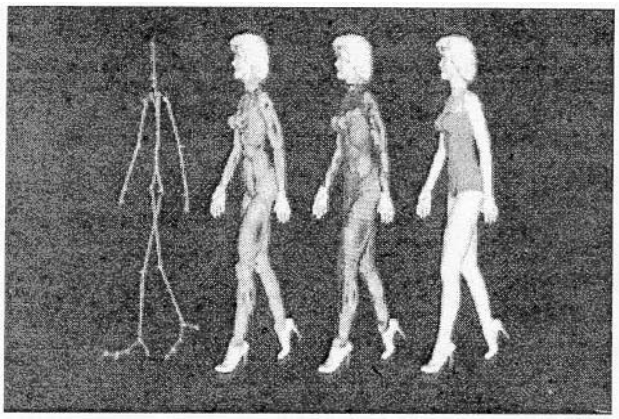

Figure 2. Modeling human body: skeleton, metaballs, metaballs with wireframe skin, rendered skin and clothes

For creating face shapes, tedious methods based on digitizing techniques have been first replaced by methods based on local deformations. For example, a realistic human character may be produced with a method similar to the modelling in clay, work which essentially consists of adding or eliminating bits of material, and turning the object around when the shape has been set up. An elegant solution is the use of a sculpting software based on an interactive 3D input device like the Spaceball.

More recently, new methods coming from Computer Vision have appeared. The goal of these methods is to automatically fit a complex facial animation model to image data obtained using orthogonal photographs [1] or regular video cameras [2] as opposed to sophisticated sensors such as laser range finders. For example, Fua et al. [3] starts with a set of stereo image pairs or a video sequence, and first extract stereo data and silhouette data. The system then fits to this image data the facial animation model described in Section 3. This model should allow us to produce sophisticated and realistic animations of a specific person in an automated fashion and using only cheap sensors.

In the example in Figure 3, the images have been registered using standard interactive photogrammetric techniques. We first compute disparity maps for each consecutive pair of images in the video sequence, derive clouds of 3D points and fit local surface patches to these raw 3D points. Finally, we use a least-square adjustment technique to fit the animation mask to the 3--D data.

3. FACIAL AND BODY GESTURES

For multimodal system, body animation play important role. Body motion control is normally performed by animating a skeleton. Using geometric techniques, the skeleton is locally controlled and defined in terms of coordinates, angles, velocities, or accelerations. We define two types of body motions: predefined gestures and task-oriented motion.

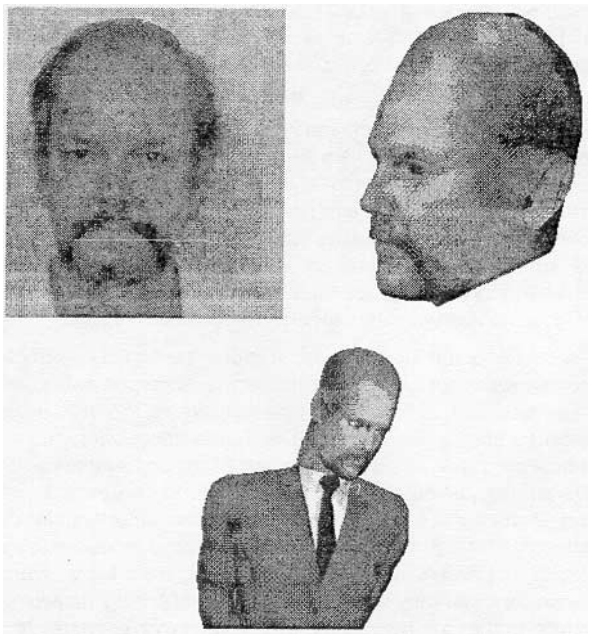

Figure 3. Facial Reconstruction from video sequences

For predefined gestures, we can record specific human body postures or gestures with a magnetic motion capturing system and an anatomical converter, or we can design human postures or gestures using a keyframe system [4]. Motion capturing can be best achieved by using a large number of sensors to register every degree of freedom of the real body. For example, Molet et. al. [5] discuss that a minimum of 14 sensors are required to manage a bio-mechanically correct posture. The raw data coming from the trackers has to be filtered and processed to obtain suitable animation parameters like the MPEG-4 compatible Body Animation Parameters (BAP). Motion capture is the simplest approach, but key frame animation is still another popular technique in which the animator explicitly specifies the kinematics by supplying keyframes values (see Figure 4) whose "in-between" frames are interpolated by the computer.

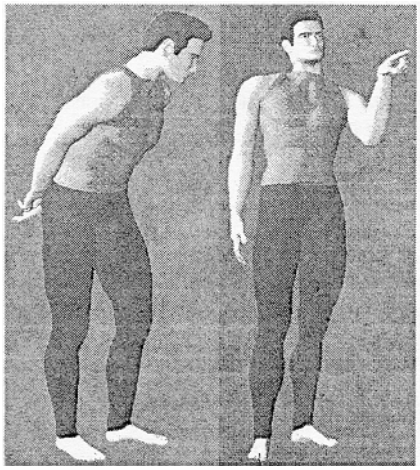

Figure 4. Keyframes

If keyframe animation is in real-time, the design of the keys should be prepared in advance, which may be a severe limitation for interactive and VR applications. However, with motion capture, this is the most common approach used in video-games.

For task oriented motions, specific motion motors should be used. Inverse kinematics is a technique coming from robotics, where the motion of links of a chain is computed from the end link trajectory. Geometric methods are efficient and may easily be performed in real-time. There is a need for methods of automatic motion like physics-based methods. Physical data are provided and the motion is obtained by solving the dynamic equations. However, physical data are often hard to find and some physics-based methods may be very time-consuming and unstable.

In fact, for distributed VR applications as well as interactive entertainment applications like interactive drama or games, what is now essential is the use of motor functions for basic actions typically like walking, grasping, and, depending on the application, facial animation. The use of motor functions will be different for guided and autonomous actors. However, in both cases, the motor functions are essential. These motor functions are more powerful than playing previously-recorded motions: they are generally based on approximations coming from biomechanical experiments, and they attempt to consider different parameters of the motion they are responsible for, in order to give parametrized motion (for example step length in walking as a function of velocity). Globally, the process of face animation is decomposed into several layers of information as shown in Figure 5.

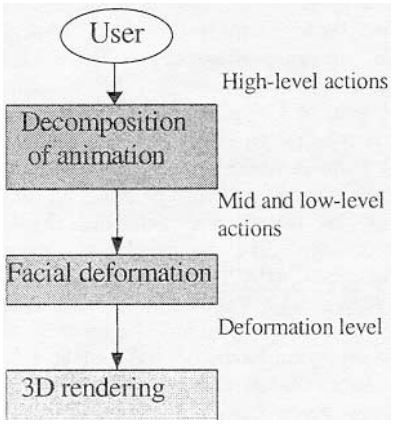

Figure 5. Decomposition into Multilayers

The high-level actions concern the emotions, the sentences and the head movements of the virtual actor. 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 [6].

4. INTERACTION WITH OBJECTS

The necessity to model interactions between an object and a virtual human appears in most applications of computer animation and simulation. Such applications encompass several domains, as for example: virtual autonomous agents living and working in virtual environments, human factors analysis, training, education, virtual prototyping, and simulation-based design. A good overview of such areas is presented by Badler [7]. An example of an application using agent-object interactions is presented by Johnson et al [8], whose purpose is to train equipment usage in a populated virtual environment.

Commonly, simulation systems perform agent-object interactions for specific tasks. Such approach is simple and direct, but most of the time, the core of the system needs to be updated whenever one needs to consider another class of objects.

To overcome such difficulties, a natural way is to include within the object description, more useful information than only intrinsic object properties. Some proposed systems already use this kind of approach. In particular, the object specific reasoning [9] creates a relational table to inform object purpose and, for each object graspable site, the appropriate hand shape and grasp approach direction. This set of information may be sufficient to perform a grasping task, but more information is needed to perform different types of interactions.

Another interesting way is to model general agent-object interactions based on objects containing interaction information of various kinds: intrinsic object properties, information on how-to-interact with it, object behaviors, and also expected agent behaviors. The smart object approach, introduced by Kallmann and Thalmann [10 11] extends the idea of having a database of interaction information. For each object modeled, we include the functionality of its moving parts and detailed commands describing each desired interaction, by means of a dedicated script language. A feature modeling approach [12] is used to include all desired information in objects. A graphical interface program permits the user to interactively specify different features in the object, and save them as a script file. Figure 6 shows an example.

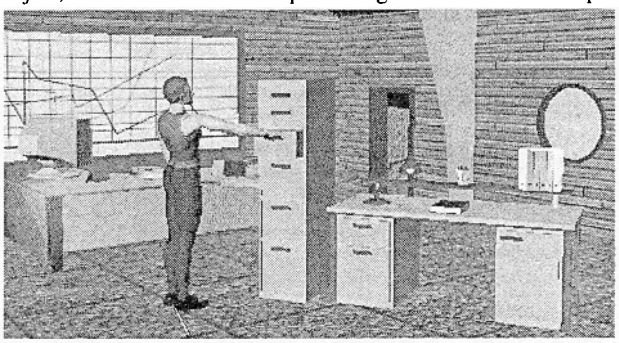

Figure 6. Interacting with objects

5. MULTIMODAL COMMUNICATION

5.1 Action Recognition through magnetic sensors

Most of today's virtual environments are populated with some kind of autonomous life-like agents. Such agents follow a pre-programmed sequence of behaviours that exclude the user as a participating entity in the virtual society. In order to make inhabited virtual reality an attractive place for information exchange and social interaction, we need to equip the Virtual

Humans with some perception and interpretation skills. An important skill is human action recognition. By opposition to human-computer interfaces (HCI) that focus on speech or hand gestures, we have proposed a full-body integration of the user..

One goal in virtual reality (VR) is to simulate real world situations. As a consequence, the interface has to facilitate the real world interaction paradigms. However, reproducing every single detail of an interaction is not necessarily a desirable goal. Grasping an object is such an example. It is relatively easy to copy a performer's hand motions onto virtual hands. Even collision detection between objects and the virtual hands can be mastered [13] [14]. Nevertheless, experience shows that grasping a virtual object is still more difficult than in real life. The problem is the poor quality of the feedback loop. Neither can we manage realistic 3D-vision sensation, nor do we have realistic force feedback devices. We think that grasping an object with detailed finger control is not necessarily desirable. Instead of asking the user to perform a precision grasp, we ask the user to perform a grasp gesture. This action is recognized by the system and the object snaps to his/her hands. Of course, in real life, objects generally do not behave like magnets, but this can be an acceptable compromise between the realism of a task performance and the convenience of the interface. Other interaction tasks that can be advantageously simplified through action recognition events are walking, sitting or simply looking around by turning the head. In most VR systems, such actions are triggered by some hand posture recognition events. We claim that such actions can be more naturally triggered through a full-body action recognition system.

As technology improves, Virtual Reality interfaces based on body actions and expressions will become more and more important. In the domain of games, an evident application is the control of the hero by body actions. The robotics domain is another area where such an interface presents an attractive issue, especially because telepresence is a hot topic nowadays. In this paper, we discuss an action model along with a real-time recognition system and show how it may be used in Virtual Environments. The real people are of course easily aware of the actions of the Virtual Humans through VR tools like Head-mounted displays, but one major problem to solve is to make the virtual actors conscious of the behaviour of the real people. Virtual actors should sense the participants through their virtual sensors. Such a perceptive actor would be independent of each VR representation and he could in the same manner communicate with participants and other perceptive actors. The real time constraints in VR demand fast reaction to sound signals and fast recognition of the semantic it carries. For the interaction between virtual humans and real ones, gesture recognition is a key issue. As an example, Emering et al. [15] produced a fighting between a real person and an autonomous actor. The motion of the real person is captured using a Flock of Birds. The gestures are recognised by the system and the information is transmitted to the virtual actor who is able to react to the gestures and decide which attitude to do.

The Action Recognition Algorithm recognition process exploits a multi-level action model in order to perform in real-time (at least 10 frames/sec). First, the Candidate Action Set (in short CAS) is initialized with the whole action database. Then, the motion capture system provides the user's posture directly in terms of joint angles [MBT96]. This information drives the action selection process at the five levels of the action model. The ARA retains only those actions which match the current action's characteristic or which do not define any action primitive at that level. The interest is to have low dimension matching at the higher level, with a large action database, while matching between high dimension data is made only on a small number of candidates (Fig. 3). Futhermore, the hierarchical approach overcomes the limitation resulting from the extreme simplicity of the matching algorithm while allowing real-time performance. For each new body posture sample the body movement is analyzed to derive the average velocity of the Center of Mass (CoM) and End Effectors (Ees) over a short period of time. The recognition algorithm proceeds in a uniform fashion for all the levels of the action model. For each level, first the actions that do not define any action primitive for that level just bypass it. Second, it evaluates the action primitives triggered by the current motion. Then the remaining candidate actions exhibiting these action primitives in their definition (CoM level) or fulfilling their boolean expression (EEs level) are selected. If no action remains from the candidate set, the algorithm stops and reports an "unknown action" as output. Otherwise, the resulting CAS becomes the input of the next lower level. When no motion is detected, the gesture level reports it. Figure 7 shows an example.

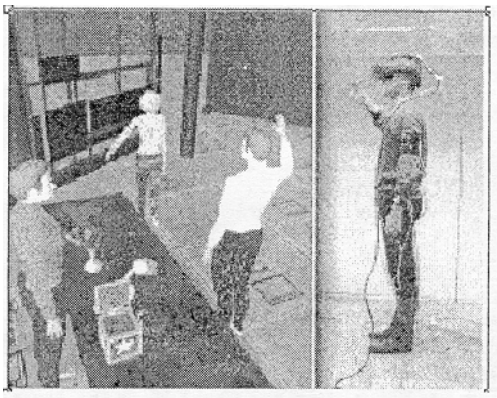

Figure 7. Action recognition

5.2 Realtime Video-Based Tracking

Problems with magnetic sensors is the interface with cumbersome wires or body attached sensors for VR multi-user shared environments. A way of avoiding this situation is to used standard video camera to track the motion. Normally, camera-based methods -in this context - use complex algorithms for image processing and are computing intensive but this is not the case with the proposed approach [16] which allows real-time tracking on a standard SG machine. The *classical* image processing approach would be to search the trackers looking for their colour and/or texture in the entire image over all pixels -with a time-complexity dependent on the frame size - and only afterwards to apply some model-based constraints to determine the motion. The model-based constraints are applied directly to the user's model - these constraints are used to generate several possible positions - and afterwards the best among these models is considered to be the next user's position. In this way the required computations are reduced dramatically while only few hundreds of pixels are processed in the image *regardless* of the frame size. With the proposed approach, several new models (or estimations of the next state) are generated based on model-based constraints, then a block called BEST MATCH selects the highest resemblance result that will represent the next state in which the

rendered model is going to be. The only assumptions we are making are: the camera model has to be known - in order to render the user's model from our camera's point of view - and the position of the source light has also to be known. A perfect modelling of the environment is not very useful. The model we used for the camera is a pinhole [17] and it matches all usual cameras. This model may be characterised by a projection matrix computed for a certain 3D reference system. The used reference system is a calibration grid and a program was written to automatically compute the projection matrix relatively to the white reference system in the image. Subsequently any 3D point referenced to this system can be rendered by the computer and retrieved in the 2D image stream coming from the camer. The user's model is specified by his/her sizes(e.g. length of arms, distance between shoulders) and the model for the source light is specified by giving its relative position to the user and characteristics. In order to track the user, he/she will wear 4 trackers attached to the joints (2 for the wrists and 2 for the elbows) and his position is assumed to be fixed in front of the camera. The model we used has two fixed points - represented with crosses and corresponding to the shoulders - and the four trackers. Figure 8 shows examples.

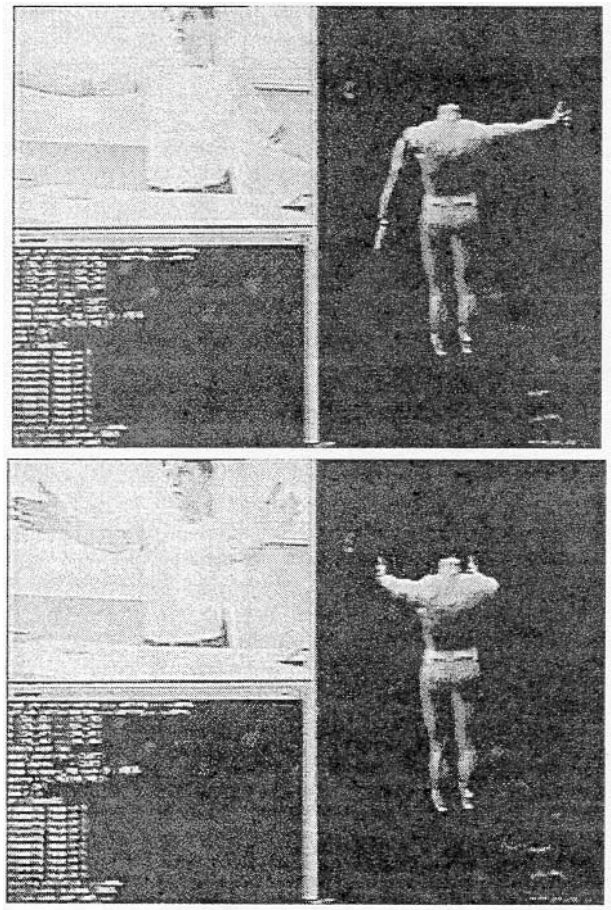

Figure 8. Video-based action recognition

5.3 Facial communication

What gives its real substance to face-to-face interaction in real life, beyond the speech, is the bodily activity of the interlocutors, the way they express their feelings or thoughts through the use of their body, facial expressions, tone of voice, etc. Some psychological researches have concluded that more than 65 percent of the information exchanged during a face-to-face interaction is expressed through nonverbal means [18]. A VR system that has the ambition to approach the fullness of real-world social interactions and to give to its participants the possibility to achieve a quality and realistic interpersonal communication has to address this point; and only realistic embodiment makes nonverbal communication possible.

5.4 Speech Synthesis and Recognition

A considerable part of human communication is based on speech. Therefore, a believable virtual humanoid environment with user interaction should include speech synthesis and speech recognition. For speech synthesis, in the VHD system (see Section 6), the input text is being converted into temporized phonemes using a text-to-speech synthesis system. In this case, the Festival Speech Synthesis System from the University of Edinburg [19] is used. It produces also the audio stream that will be subsequently playback in synchronization with the facial animation system. Using the temporized phonemes, it is possible to create the facial animation by concatenating the corresponding visemes through time. The facial animation system is based on the system described in Section 3. The set of phonemes is limited to the ones used in the Oxford English Dictionary, but an easy extension to any language could be done by designing the corresponding visemes.

In order to improve real time user interaction with autonomous actors we extended the L-system interpreter [20] with a speech recognition feature that transmits spoken words, captured by a microphone, to the virtual acoustic environment by creating corresponding sound events perceptible by autonomous actors. This concept enables us to model behaviors of actors reacting directly to user-spoken commands. For speech recognition, we use POST, the **P**arallel **O**bject **o**riented **S**peech Toolkit [21], developed for designing automatic speech recognition. It can perform simple feature extraction, training and testing of word and sub-word Hidden Markov Models with discrete and multi Gaussian statistical modeling. We use a POST application for isolated word recognition. The system can be trained by several users and its performance depends on the number of repetitions and the quality of word capture. This speech recognizing feature was recently added to the system and we don't have much experience with its performance. First tests, however, with a single user training, resulted in a satisfactory recognition rate for a vocabulary of about 50 isolated words.

6. THE VIRTUAL HUMAN DIRECTOR SYSTEM

The Virtual Human Director (VHD) system [22] is a multimodal system jointly developed by EPFL (Computer Graphics Lab) and University of Geneva (MIRALab). It 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 provides a range of virtual human animation and interaction capabilities integrated into one single environment. The software architecture (see Figure 9) 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.

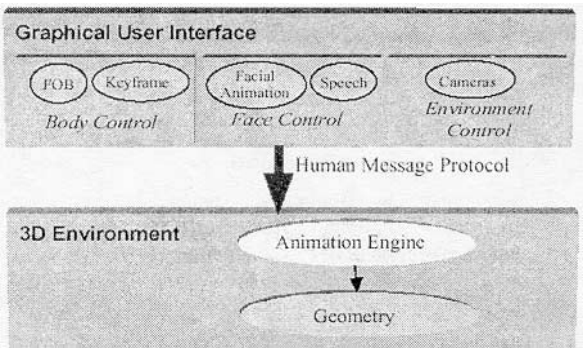

Figure 9. The VHD architecture

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.

We used one motion motor for the body locomotion. This walking motor was developed by Boulic [23]. 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 10 includes snapshots from a walking session.

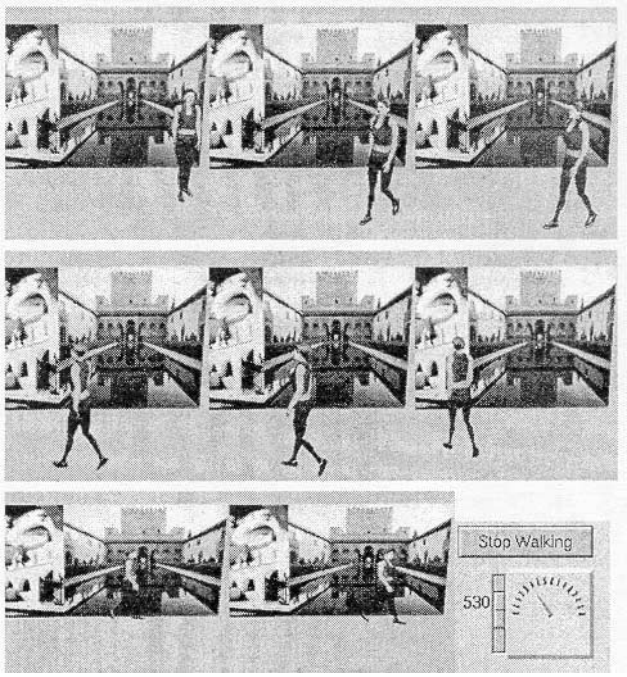

Figure 10. A walking example

7. ACKNOWLEDGMENTS

The authors would like to thank people who contributed to the various projects, especially Dr. Pascal Fua, Selim Balcisoy, Luc Emering, Michal Ponder, and Marcelo Kallmann. The research was partially sponsored by the Swiss National Research Foundation.

8. REFERENCES

1 W.S. Lee, N. Magnenat Thalmann, Head Modeling from Pictures and Morphing in 3D with Image Metamorphosis based on triangulation, Proc. Captech98 (Modelling and Motion Capture Techniques for Virtual Environments), (Springer LNAI LNCS Press), Geneva, 1998, pp.254-267.

2 P. Fua, R. Plaenkers, and D. Thalmann, From Synthesis to Analysis: Fitting Human Animation Models to Image Data, Proc. Computer Graphics International, Canmore, Alberta, Canada, June 1999.

3 P.Fua, Face Models from Uncalibrated Video Sequences, in: N.Magnenat-Thalmann, D.Thalmann (eds), "Modeling and Motion Capture Techniques for Virtual Environments" , Lecture Notes in Artificial Intelligence, No1537, Springer, 1998, pp.214-228.

4 R. Boulic et. al. Goal Oriented Design and Correction of Articulated Figure Motionwith the TRACK system, Computer and Graphics, Pergamon Press, Vol. 18, No. 4., pp. 443-452, 1994.

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

6 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.

7. N. Badler, "Virtual Humans for Animation, Ergonomics, and Simulation", IEEE Workshop on Non-Rigid and Articulated Motion, Puerto Rico, June 97.

8. W. L. Johnson, and J. Rickel, "Steve: An Animated Pedagogical Agent for Procedural Training in Virtual Environments", Sigart Bulletin, ACM Press, vol. 8, number 1-4, 16-21, 1997.

9. L. Levison, Connecting Planning and Acting via Object-Specific reasoning, PhD thesis, Dept. of Computer & Information Science, University of Pennsylvania, 1996.

10. M. Kallmann, D. Thalmann, Modeling Objects for Interaction Tasks, Proc. Eurographics Workshop on Animation and Simulation, Springer, 1998

11. M. Kallmann, D.Thalmann, A Behavioral Interface to Simulate Agent-Object Interactions in Real-Time, Proc. Computer Animation 99, IEEE Computer Society Press.

12. J.J. Shah, and M. Mäntylä, "Parametric and Feature-Based CAD/CAM", John Wiley & Sons, inc. 1995, ISBN 0-471-00214-3.

13 R. Boulic, S. Rezzonico, D. Thalmann, 1996. Multi-Finger Manipulation of Virtual Objects, Proc. ACM Symposium on Virtual Reality Software and Technology VRST'96, Hong-Kong, 1996, pp 67-74.

14 S. Rezzonico, R. Boulic, Z. Huang, N. Magnenat-Thalmann, D. Thalmann, 1995. Consistent Grasping in Virtual Environment based on the Interactive Grasping Automata, in Virtual Environment. Martin Goebel Editor, pp 107-118, Springer-Verlag Wien, October 1995.

15. L. Emering, R. Boulic, D. Thalmann, Interacting with Virtual Humans through Body Actions, IEEE Computer Graphics and Applications, 1998 , Vol.18, No1, pp.8-11.

16 C. Ciressan, D.Thalmann, 3D Model-Based Upper-Body Tracking Using Monocular Camera, internal report, Computer Graphics Lab, EPFL.

17 E.D. Dickmann and V. Graefe, "Applications of dynamic monocular machine vision", Machine vision and Applications ,1:241-261,1988.

18. M. Argyle, Bodily Communication, New York: Methuen & Co., 1988.

19 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.

20 H. Noser, D.Thalmann, A Rule-based Interactive Behavioral Animation System for Humanoids, IEEE Transactions on Visualization and Computer Graphics, Vol.5, No4, pp.281-307.

21 Hennebert J. and Delacrétaz D.P., (1996) POST: Parallel Object-Oriented Speech Toolkit, Proc. ICSLP 96, Philadelphia

22 G. Sannier, S. Balcisoy, N. Magnenat-Thalmann, D. Thalmann, VHD: A System for Directing Real-Time Virtual Actors, The Visual Computer, Springer, Vol.15, No 7/8, 1999, pp.320-329.

23 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.