The Role of Virtual Humans in Virtual Environment Technology and Interfaces

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

Abstract

The purpose of this paper is to show the importance of Virtual Humans in Virtual Reality and to identify the main problems to solve to create believable Virtual Humans.

Introduction

We may identify several areas [1] where autonomous virtual humans are essential:

Virtual people for Inhabited Virtual Environments. Their role is very important in virtual environments with many people, like virtual airports or even virtual cities. In the next few years, we will see a lot of Humanoids or Virtual Humans in many applications. These virtual humans will be more and more autonomous. They will also tend to become intelligent.

Virtual substitutes. 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.

Virtual medical assistance. Nowadays, it seems very difficult to imagine an effective solution for chronic care without including the remote care of patients at home by a kind of Virtual Medical Doctor. The modelling of virtual patient with correspondence to medical images is also a key issue and a basis for telesurgery.

The ultimate reason for developing realistic-looking Virtual Humans is to be able to use them in virtually any scene that re-creates the real world. However, a virtual scene -- beautiful though it may be -- is not complete without people.... Virtual people, that is. Scenes involving Virtual Humans imply many complex problems we have been solving for several years [2]. With the new developments of digital and interactive television [3] and multimedia products, there is also a need for systems that provide designers with the capability for embedding real-time simulated humans in games, multimedia titles and film animations. In fact, there are many current and potential applications of human activities that may be part of a VR system involving virtual humans:

- simulation based learning and training (transportation, civil engineering etc.)
- simulation of ergonomic work environments
- virtual patient for surgery, plastic surgery
- orthopedy and prostheses and rehabilitation

- plastic surgery
- virtual psychotherapies
- architectural simulation with people, buildings, landscapes and lights etc.
- computer games involving people and "Virtual Worlds" for Lunaparks/casinos
- game and sport simulation
- interactive drama titles in which the user can interact with simulated characters and hence be involved in a scenario rather than simply watching it.

But mainly, telepresence is the future of multimedia systems and will allow participants to share professional and private experiences, meetings, games, and parties. Virtual Humans have a key role to play in these shared Virtual Environments and true interaction with them is a great challenge. Although a lot of research has been going on in the field of Networked Virtual Environments, most of the existing systems still use simple embodiments for the representation of participants in the environments. More complex virtual human embodiment increases the natural interaction within the environment. The users' more natural perception of each other (and of autonomous actors) increases their sense of being together, and thus the overall sense of shared presence in the environment.

Fig.1 Virtual Humans

But, the modelling of Virtual Humans is an immense challenge as it requires to solve many problems in various areas. Table 1 shows the various aspects of research in Virtual Human Technology. Each aspects will be detailed and the problems to solve will be identified.

Face and body representation
Avatar functions
Motion control
High-level behavior
Interaction with objects
Intercommunication
Interaction with user
Collaborative Virtual Environments

Crowds
Rendering
Standards
Applications

Table 1. Aspects of research in Virtual Humans

Face and body representation

Human modelling is the first step in creating Virtual Humans. For head, although t is possible to create them using an interactive sculpting tool, the best way is to reconstruct them from reality. Three methods have been used for this:

1) Reconstruction from 2D photos [4]
2) Reconstruction from a video sequence [5]
3) Construction based on the laser technology

The methods could be used for body modelling, but the main problem is still with the body deformations which has been addressed by many researchers, but is still not 100% solved.

Concerning facial expressions in Networked VEs, four methods are possible: video-texturing of the face, model-based coding of facial expressions, lip movement synthesis from speech and predefined expressions or animations. Believable facial emotions are still very hard to obtain.

Main problem to solve: realistic body and face construction and deformations

Avatar functions

The avatar representation fulfils several important functions:

1) the visual embodiment of the user
2) means of interaction with the world
3) means of sensing various attributes of the world

It becomes even more important in multi-user Networked Virtual Environments [6], as participants' representation is used for communication. This avatar representation in NVEs has crucial functions in addition to those of single-user virtual environments [7 8]:

1) perception (to see if anyone is around)
2) localisation (to see where the other person is)
3) identification (to recognise the person)
4) visualisation of others' interest focus (to see where the person's attention is directed)
5) visualisation of other's actions (to see what the other person is doing and what is meant through gestures)
6) social representation of self through decoration of the avatar (to know what the other participants' task or status is)

Using articulated models for avatar representation fulfils these functionalities with realism, as it provides the direct relationship between how we control our avatar in the virtual world and how our avatar moves related to this control, allowing the user to use his/her real world experience. We chose to use complex virtual human models aiming for a high level of realism, but articulated "cartoon-like" characters could also be well suited to express ideas and feelings through the nonverbal channel in a more symbolic or metaphoric way.

Main problem to solve: easy way of directing an avatar

Motion control

The main goal of computer animation is to synthesize the desired motion effect which is a mixing of natural phenomena, perception and imagination. The animator designs the object's dynamic behavior with his mental representation of causality. He/she imagines how it moves, gets out of shape or reacts when it is pushed, pressed, pulled, or twisted. So, the animation system has to provide the user with motion control tools able to translate his/her wishes from his/her own language.

In the context of Virtual Humans, a Motion Control Method (MCM) specifies how the Virtual Human is animated and may be characterized according to the type of information it privileged in animating this Virtual Human. For example, in a keyframe system for an articulated body, the privileged information to be manipulated is the angle. In a forward dynamics-based system, the privileged information is a set of forces and torques; of course, in solving the dynamic equations, joint angles are also obtained in this system, but we consider these as derived information. In fact, any MCM will eventually have to deal with geometric information (typically joint angles), but only geometric MCMs explicitly privilege this information at the level of animation control.

Many MCMs have been proposed: motion capture, keyframe, inverse kinematics, dynamics, walking models, grasping models, etc.. But, no method is perfect and only combination of blending of methods can provide good and flexible results.

Main problem to solve: flexible reuse, combination, and parameterisation of existing movements

High-level behavior

Autonomous Virtual Humans should be able to have a behaviour, which means they must have a manner of conducting themselves. Typically, the Virtual Human should perceive the objects and the other Virtual Humans in the environment through virtual sensors [9]: visual, tactile and auditory sensors. Based on the perceived information, the actor's behavioural mechanism will determine the actions he will perform. An actor may simply evolve in his environment or he may interact with this environment or even communicate with other actors. In this latter case, we will consider the actor as a interactive perceptive actor.

Perception through Virtual Sensors

The actor-environment interface, or the synthetic sensors, constitute an important part of a behavioral animation system. As sensorial information drastically influences behavior, the synthetic sensors should simulate the functionality of their organic counterparts. Due to real-time constraints, we did not make any attempt to model biological models of sensors. Therefore, synthetic vision only makes efficient visibility tests using SGI's graphics rendering hardware that produces a Z-buffered color image representing an agent's vision. A tactile point-like sensor will be represented by a simple function evaluating the global force field at its position. The synthetic "ear" of an agent will be represented by a function returning the on-going sound events. What is important for an actor's behavior is the functionality of a sensor and how it filters the information flow from the environment, and not the specific model of the sensor.

Another aspect of synthetic sensor design is its universality. The sensors should be as independent as possible from specific environment representations and they should be easily adjustable for interactive users. For example, the same Z-buffer based renderer displays the virtual world for an autonomous actor and an interactive user. The user and the autonomous actors perceive the virtual environment through rendered images, without knowing anything about the internal 3D environment representation or the rendering mechanism.

The sense of touch plays also an important role for humans. In order to model this sense, we use a physically based force field model. This model is close to reality, as the real sense of touch also perceives collision forces. By adding a physically-based animation of objects, we can extend the force field model to a physically based animation system where touch sensors correspond to special functions evaluating the global force field at their current position. This approach also solves the response problem of collisions, as they are handled automatically by the physics-based evolution system if both colliding objects - sensor and touched object - exert for example short range repulsion forces. We opted for a force field-based model to represent the sense of touch, as it integrates itself naturally into the physically-based particle system we already use for physical and behavioral animation. Of course, for real-time applications, the number of sensors and particles should be small, as the evolution of the particle system is computationally expensive. Another disadvantage of this approach is that the touch sensors only "sense" geometrical shapes that are explicitly bounded by appropriate force fields. Another difficulty arising due to the force field model is the fact that the parameterization of the force fields, the numerical integration of the system of differential equations, the time step and the speed of moving objects depend on each other, and that the adaptation of all parameters for a stable animation is not always trivial. All these parameters need to be manually tuned in the L-system definition.

As pointed out above, we use a sound event framework for controlling the acoustic model of the animation system. The sound event handler maintains a table of the on-going sound events. Consequently, one can immediately model synthetic hearing by simple querying of this table of on-going sound events. An interactive user can also produce sound events in the virtual environment via a speech recognition module. Through a sound event, an autonomous actor can directly capture its semantic, position and emitting source.

The Vision System

In our implementation of the vision-based approach to behavioral animation, the synthetic actor perceives its environment through a small window in which the environment is rendered by the computer from the actor's point of view. Rendering is based on Z-buffer techniques. The Z-buffer consists of an array containing the depth values of the pixels of the image. The algorithm uses these Z-buffer values for efficient rendering of 3D scenes. Renault et al. [1990] used the Z-buffering hardware graphics of workstations for efficiently rendering a bitmap projection of the actor's point of view. The color of an object is unique and serves the purpose of identifying the semantics of an object in the image. This synthetic vision was used to create an animation

involving synthetic actors moving autonomously in a corridor, and avoiding objects as well as other synthetic actors.

As an actor can access Z-buffer values of the pixels - corresponding to the distances of the objects' pixels to the observer -, their color, and its own position, it can therefore locate visible objects in the 3D environment. This local information is sufficient for some local navigation. For global navigation, however, a visual memory is useful in order to recognize dead-ends problems, such as searching for the exit to a maze. We modeled visual memory by a 3D occupancy octree grid, similar to a technique described in [Roth-Tabak and Jain 1989]. In this space grid, each pixel of an object, transformed back to 3D world coordinates, occupies a voxel. By comparing, in each frame, the rendered voxels in the visual field with the corresponding pixel of the vision window, we can update the visual memory by eliminating voxels having disappeared in the 3D world. Consequently, the visual memory reflects the state of the 3D dynamic world as perceived by the synthetic actor.

The concept of synthetic vision with a voxelized visual memory is independent of 3D world modeling. Even fractal objects and procedurally-defined and rendered worlds without 3D object database can be perceived as long as they can be rendered in a Z-buffer-based vision window. We use synthetic vision in conjunction with a visual memory, for environment recovery, for global navigation, for local navigation optimization and for object recognition through color coding in several behaviors. The reconstruction of the perceived environment by the "visual memory" of an actor, and its use in global navigation is published in [Noser et al. 1993; Noser et al. 1995].

The Hearing Sensors

The hearing sensor of an actor corresponds to the table of the currently active sounds provided by the sound event handler representing the propagation medium. From this table the actor retrieves the complete information regarding each event consisting of the sound identifier, source and position. The same principle as for the other synthetic sensors also applies to the hearing sensor. We need to define special functions usable in the conditions of production rules, and returning useful information. We implemented functions that return on-going identifiers of sound events and sound sources.

The Tactile Sensors

Ideally, geometrical collision detection between surfaces should be used for the modeling of tactile sensors. However, as a typical L-system environment is composed of a large number of objects, and as there is no geometrical database of the 3D objects, traditional collision detection is not the best solution for a tactile sensor model. As we already have a force field environment integrated in the L-system, we use a force field approach to model tactile sensor points. All we need to do is define a function that can evaluate the amount of the global force field at a given position. This amount can be compared with a threshold value that represents, for instance, a collision. With this function, even wind force fields can be sensed. Traditional collision detection between surfaces can cause a large number of collisions, and it will not always be easy to model the behavioral response. With the definition of only one or few sensor points attached to an actor, this behavioral response is easier to control, and calculation time is reduced, which is important for real-time applications. We can also associate a particle having an appropriate force field with a sensor point that will act automatically on other particles. Thus, an actor can "sense" and manipulate other particles.

In order to use tactile information for behavior modeling with production rules, the force field sensing function must be usable under the conditions of the production rules during the derivation phase of the symbolic object. During interpretation of the symbolic using a query symbol, the turtle position can be copied into the parameter space of the symbol. Consequently, the turtle position, given by the x, y, and z coordinates, is available in the parameters x, y, and z of the query symbol for the force field function. This force field function returns the amount of force felt at the position of the turtle. Therefore, the force can be used in conditions that trigger certain behaviors represented by production rules.

When the turtle position is available in the parameter space of a symbol, it can of course also be used for geometrical collision detection, coded within the condition expressions of production rules. If the parameter y corresponds to the y coordinate of the turtle, a condition, such as y<0, for example, detects a collision of the turtle when the ground is situated at y = 0, and gravity is acting in the y down direction.

Speech Recognition
A considerable part of human communication is based on speech. Therefore, a believable virtual humanoid environment with user interaction should include speech recognition. In order to improve real time user interaction with autonomous actors we extended the L-system interpreter 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 **T**oolkit [Hennebert and Delacrétaz 1996], developed for designing automatic speech recognition. POST is freely distributed to academic institutions. 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.
A high level behavior uses in general sensorial input and special knowledge. A way of modeling behaviors is the use of 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. Abstraction mechanisms to simulate intelligent behaviours have been discussed in the AI (Artificial Intelligence) and AA (Autonomous Agents') literature. Several methods have been introduced to model learning processes, perceptions, actions, behaviours, etc, in order to build more intelligent and autonomous virtual agents.

Interaction with objects

The necessity to model interactions between an object and a virtual human agent (here after just referred to as an agent), 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 [10]. An example of an application using agent-object interactions is presented by Johnson et al [11], 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 [12] 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 [13 14] 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 [15] 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.

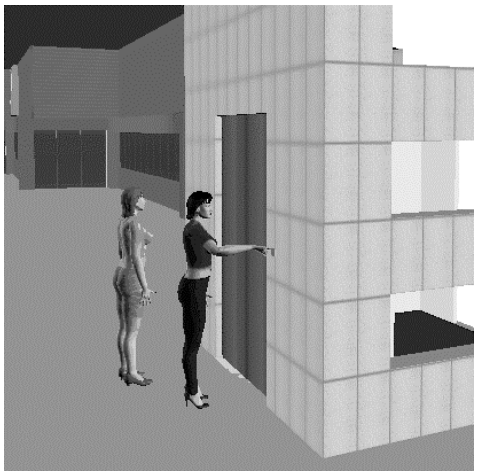

Fig.2. Interaction with objects

Intercommunication

Behaviours may be also dependent on the emotional state of the actor. A non-verbal communication is concerned with postures and their indications on what people are feeling. Postures are the means to communicate and are defined by a specific position of the arms and legs and angles of the body. This non-verbal communication is essential to drive the interaction between people without contact or with contact.

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 [16]. 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.

Fig.3. Intercommunication

Interaction with user

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. Perceptive actors and participants may easily be. 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. For the interaction between virtual humans and real ones, gesture recognition is a key issue.

To date, basically two techniques exist to capture the human body posture in real-time. One uses video cameras which deliver either conventional or infrared pictures. This technique has been successfully used in the ALIVE system (cf. [1]) to capture the user's image. The image is used for both the projection of the participant into the synthetic environment and the extraction of cartesian information of various body parts. If this system benefits from being wireless, it suffers from visibility constraints relative to the camera and a strong performance dependence on the vision module for information extraction.

The second technique is based on sensors which are attached to the user. Most common are sensors measuring the intensity of a magnetic field generated at a reference point. The measurements are transformed into position and orientation coordinates and sent to the computer. This raw data is matched to the rotation joints of a virtual skeleton by the means of an anatomical converter (cf. [2]). This is the approach we use currently for our interactive VR testbeds.

Figure 1 shows a snapshot of a life participant with ten sensors used to reconstruct the avatar in the virtual scene. The participant performs fight gestures which are recognized by the virtual opponent [3]. The latter responds by playing back a pre-recorded keyframe sequence. The most disturbing factors of this system are the setup time to fix all the sensors and the wires hanging around during the animation. However wireless systems are already available which solve these problems. For the interactive part of the VR testbed we developed a model of body actions as base of the recognition system.

By analyzing human actions we have detected three important characteristics which inform us about the specification granularity needed for the action model. First, an action does not necessarily involve the whole body but may be performed with a set of body parts only. Second, multiple actions can be performed in parallel if they use non-intersecting sets of body parts. Finally a human action can already be identified by observing strategic body locations rather than skeleton joint movements. Based on these observations, a top-down refinement paradigm appears to be appropriate for the action model. The specification grain varies from coarse at the top level to very specialized at the lowest level. The number of levels in the hierarchy is related to the feature information used. At the lowest level, we use the skeleton degrees of freedom (DOF) which are the most precise feature information available (30-100 for a typical human model). At higher levels, we take advantage of strategic body locations like the center of mass and end effectors, i.e. hands, feet, the head and the spine root.

Human activity is composed of a continuous flow of actions. This continuity makes it difficult to define precise initial and in-between postures for an action. As a consequence we consider only actions defined by a gesture, a posture or a gesture followed by a posture.

The action model of relies on cartesian data and joint angle values. For a given action this data is not unique but depends on the anatomical differences of the live performers. A solution is, to normalize all cartesian action data by the body height of the performer. This is reasonable as statistical studies have measured a significant correlation between the body height and the major body segment lengths. Also it's important to choose adequate reference coordinate frames. We use three coordinate systems (cf. Figure 2): the Global, the Body and

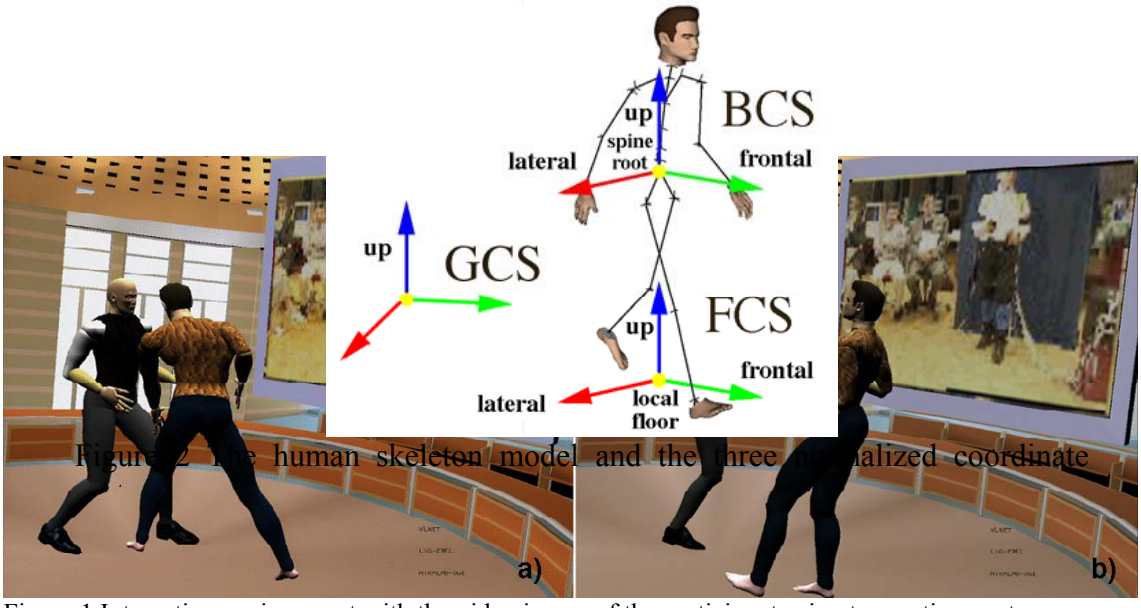

Figure 2 The human skeleton model and the three normalized coordinate

Figure 1 Interactive environment with the video image of the participant using ten motion capture sensors (right of a & b), his avatar (middle of a & b) and the virtual opponent (left of a & b)

the Floor Coordinate System (in short *GCS*, BCS, *FCS*). The BCS is attached to the spine

base of the body and its up axis is aligned with the main spine direction. The FCS is located at the vertical projection of the spine root onto the floor level and reuses the GCS Up axis.
Each level of the action model defines action primitives. At the gesture level (Table 1, levels 1 & 2) an action primitive is the detection of the motion of the center of mass (CoM) or an End Effector (EE) along a specific direction. They are of the form (CoM, velocity direction) or (EEi, velocity direction), where i denotes one of the end effectors. If the average motion is above a normalized threshold, the velocity direction is assigned to one of the following values: upward, downward, forward, backward, leftward, rightward. For the head end effector it's preferable to use rotation directions of a 'look-at' vector rather than velocity directions, in order to specify messages like 'yes' or 'no'. Additionally a *not_moving* primitive detects a still CoM or EE. Thus the gesture of an action is described by an explicit boolean expression of gesture primitives, e.g.:

'Body downward motion' = (CoM, downward)
'Walking motion' = ((spine_root, forward) AND (left foot, forward))
OR ((spine_root, forward) AND (right foot, forward))

At the posture level (Table 1, levels 3, 4 & 5) an action primitive is the cartesian position of the CoM or the EE's or the joint values of the body posture. As it is not convenient to specify position or joint information explicitly, we use a 'specify-by-example' paradigm: we build a database of posture prototypes and extract the posture primitives automatically.

During the recognition phase we keep a trace of the actions which are potential candidates for the recognition result. Initially this Candidate Action Set (*CAS*) is a copy of the complete action database. Then the candidate selection is performed sequentially on the five levels of the action model, starting with level 1 (Figure 3). Here we take advantage of the hierarchical nature of the action model: at the higher levels the CAS is large, but the data to be analyzed is small and therefore the matching costs are low. At the lower levels a higher matching cost is acceptable because the CAS has considerably shrunken.

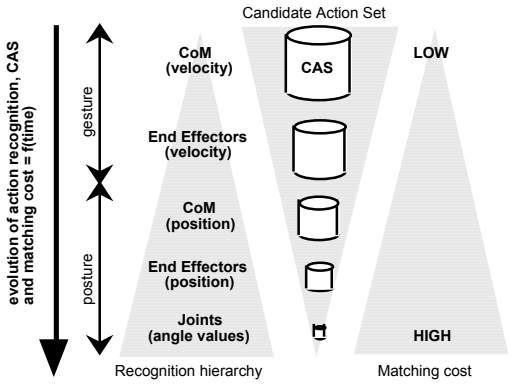

Figure 3 Action Recognition structure

- Gesture Levels Matching. Here we compute the current gesture primitives of the avatar and evaluate the actions' gesture definition. All action candidates whose boolean expression of gesture primitives results in a *False* value are removed from the CAS. Action candidates without a gesture definition remain in the CAS.

- Posture Levels Matching: For all the actions in the CAS, the algorithm computes the squared distance between their stored final CoM position and the current CoM position. The selection retains all the candidate actions for which the distance is smaller than a selectivity radius *R* given by:

$$R = Min + (1-S)*(Max-Min)$$

Min and *Max* are the smallest and largest squared distances. *S* is a normalized selectivity parameter within [0,1]. The same algorithm is applied to the levels four and five with possibly a different selectivity factor. The only difference resides in the dimension of the cartesian vector: 3D for the CoM, 18D for the EEs (concatenation of 6 end effector 3D positions) and 74D for the joints (74 degrees of freedom of the body model). Note that this algorithm always selects at least one posture among the posture candidates. In practice it means that the posture database should always contain some elementary postures: if in a VR session the participant is standing still most of the time then the database should contain a 'stand still' posture even if this posture is not used as recognition result in the application. Always selecting a winner posture is a desirable property in interactive environments because participants are more tolerant to an action mis-interpretation than ignorance. Nevertheless it is possible to add a test which discards winner postures if the deviation between the current and the database posture is beyond some threshold.

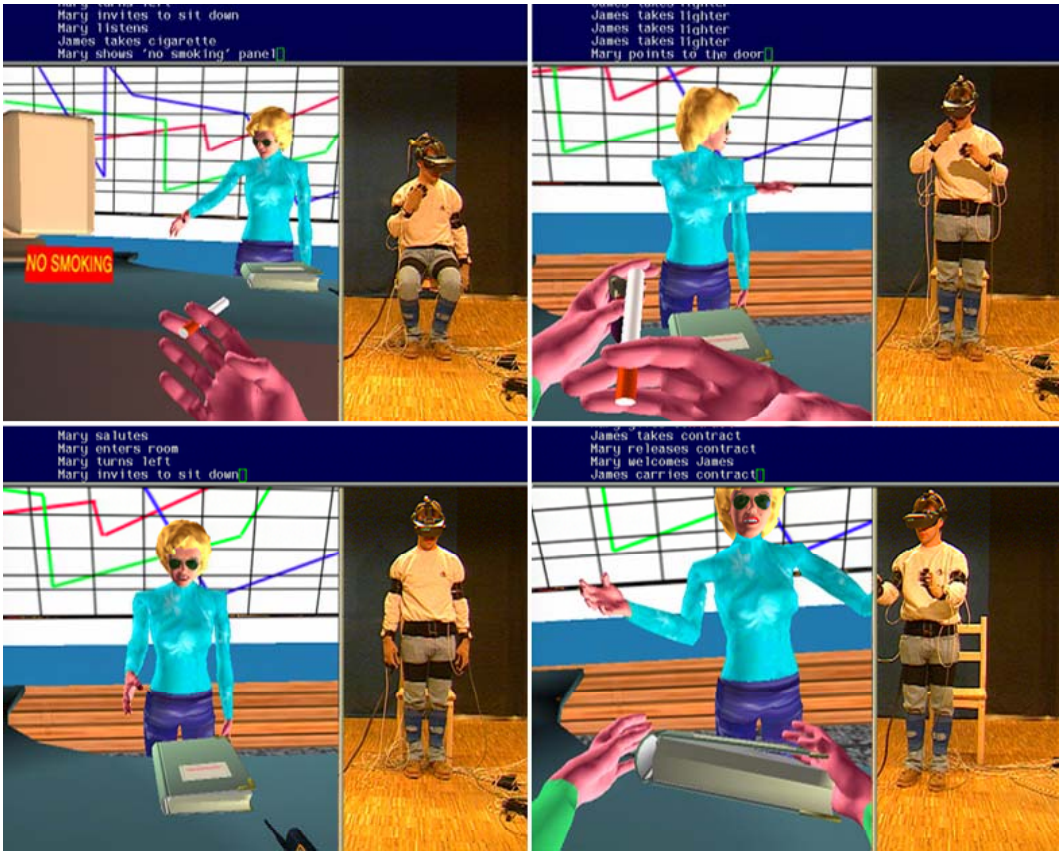

Figure 4. A virtual office environment with the participant (right), his avatar (virtual camera view) and a digital character (cyan). The animation automata is driven by the recognition events of the performers actions.

If at any level the CAS happens to get empty, the algorithm reports ‘unknown action’ as output. Simultaneous actions can be detected as long as they act on complementary body parts. For example a 'right hand phoning' action can be defined by a posture involving the right arm and the neck (levels four and five). Another action is 'walking' defined by a (CoM, forward) primitive and the 'walking motion' expression. So, whenever the performer walks while phoning, both actions are recognized due to non-intersecting sets of body parts used for both actions.

Figure 4 shows an interactive office environment with an avatar and an autonomous character: the employer. The employer's decision automata is completely coordinated by the recognition feedback of the performer's actions. For example the employer insists energetically on the fact of a non-smoking area if the performer wants to lit his cigarette. If the performer throws away his cigarette the employer invites him to take the contract files. The motions of the employer character consist of pre-recorded keyframe sequences, inverse kinematics and a procedural walking and grasping motors.

As an example, Boulic et al. [17] 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.

Specific problems of Networked Virtual Environments

Inserting virtual humans in the NVE is a complex task [6]. The main issues are:

1) selecting a scalable architecture to combine these two complex systems,
2) modeling the virtual human with believable appearance for interactive manipulation,
3) animating it with minimal number of sensors to have maximal behavioral realism,
4) investigating different methods to decrease the networking requirements for exchanging complex virtual human information.

Particularly, controlling the virtual human with limited input information is one of the main problems. For example, a person using a mouse will need extra input techniques or tools to exploit the functionalities of his embodiment. In this paper, we survey these tools that help a user with desktop VR configuration, we did not consider full tracking of the body using magnetic trackers, although this approach can be combined with limited tracking of the participant's arms.

Crowds

An accepted definition of crowd is that of a large group of individuals in the same physical environment, sharing a common goal (e.g. people going to a rock show or a football match). The individuals in a crowd may act in a different way than when they are alone or in a small group [18].

Although sociologists are often interested in crowd effects arising from social conflicts or social problems [19] the normal behavior of a crowd can also be studied when no changes are expected.

There are, however, some other group effects relevant to our work which are worth mentioning. Polarization occurs within a crowd when two or more groups adopt divergent attitudes, opinions or behavior and they may argue or fight even if they do not know each other. In some situations the crowd or a group within it may seek an adversary. The sharing effect is the result of influences by the acts of others at the individual level. Adding is the name given to the same effect when applied to the group. Domination happens when one or more leaders in a crowd influence the others.

Our goal [20] is to simulate the behavior of a collection of groups of autonomous virtual humans in a crowd. Each group has its general behavior [21] specified by the user, but the individual behaviors are created by a random process through the group behavior. This means that there is a trend shared by all individuals in the same group because they have a pre specified general behavior.

Main problem to solve: define collective behaviors while keeping individualities

Fig.4. Crowds

Areas of applications

Conclusions and recommendations

Telepresence is the future of multimedia systems and will allow participants to share professional and private experiences, meetings, games, parties. The concepts of Distributed Virtual Environments are a key technology to implement this telepresence. Using humanoids within the shared environment is a essential supporting tool for presence. Real-time realistic 3D avatars will be essential in the future, but we will need interactive perceptive actors to populate the Virtual Worlds. 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. But, how far are we from such a ideal situation? Our interactive perceptive actors are able to perceive the virtual world, the people living in this world and in the real world. They may act based on their perception in an autonomous manner. Their intelligence is constrained and limited to the results obtained in the development of new methods of Artificial Intelligence. However, the representation under the form of virtual actors is a way of visually evaluating the progress. In the future, we may expect to meet intelligent actors able to learn or understand a few situations.

References

1. D.Thalmann, L.Chiariglione, F.Fluckiger, E.H. Mamdani, M.Morganti, J.Ostermann, J.Sesena, L.Stenger, A.Stienstra, Report on Panel 6: From Multimedia to Telepresence, Expert groups in Visionary Research in Advanced Communications, ACTS, European Commission, 1997.
2. N. Magnenat Thalmann, D. Thalmann, Complex Models for Animating Synthetic Actors, IEEE Computer Graphics and Applications, Vol.11, No5, 1991, pp.32-44.

3. N. Magnenat Thalmann N., Thalmann D. (1995) Digital Actors for Interactive Television, Proc. IEEE, Special Issue on Digital Television, Part 2, July 1995, pp.1022-1031.
4. W.S.Lee, N.Magnenat-Thalmann, Head Modeling from Pictures and Morphing in 3D with Image Metamorphosis Based on triangulation, in: Modelling and motion Capture Techniques for Virtual Environments, Lecture Notes in Artificial Intelligence, 1537, Springer, 1998.
5. P. Fua, R. Plankers and D. Thalmann, From Synthesis to Analysis: Fitting Human Animation Models to Image Data, Proc. CGI 99, IEEE Computer Society Press, 1999
6. T. K. Capin, I.S. Panzic, N. Magnenat-Thalmann, D. Thalmann Avatars in Networked Virtual Environments, John Wiley and Sons, 1999.
7. T. K. Capin, I.S. Panzic, N. Magnenat-Thalmann, D. Thalmann, Virtual Human Representation and Communication in VLNET Networked Virtual Environment, IEEE Computer Graphics and Applications, March 1997.
8. S. D. Benford et al., « Embodiments, Avatars, Clones and Agents for Multi-user, Multi-sensory Virtual Worlds », Multimedia Systems, Berlin, Germany: Springer-Verlag, 1997.
9. D. Thalmann, Virtual Sensors: A Key Tool for the Artificial Life of Virtual Actors, Proc. Pacific Graphics '95, Seoul, Korea, August 1995, pp.22-40.
10. N. N. Badler, "Virtual Humans for Animation, Ergonomics, and Simulation", IEEE Workshop on Non-Rigid and Articulated Motion, Puerto Rico, June 97.
11. 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.
12. L. Levison, Connecting Planning and Acting via Object-Specific reasoning, PhD thesis, Dept. of Computer & Information Science, University of Pennsylvania, 1996.
13. M. Kallmann, D. Thalmann, Modeling Objects for Interaction Tasks, Proc. Eurographics Workshop on Animation and Simulation, Springer, 1998
14. M.Kallmann, D.Thalmann, A Behavioral Interface to Simulate Agent-Object Interactions in Real-Time, Proc. Computer Animation 99, IEEE Computer Society Press (to appear)
15. J.J. Shah, and M. Mäntylä, "Parametric and Feature-Based CAD/CAM", John Wiley & Sons, inc. 1995, ISBN 0-471-00214-3.
16. M. Argyle, Bodily Communication, New York: Methuen & Co., 1988.
17. 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.
18. M. E. Roloff. "Interpersonal Communication - The Social Exchange Approach". SAGE Publications, v.6, London. 1981.
19. J. S. McClelland. The Crowd and The Mob. Book printed in Great Britain at the University Press, Cambridge. 1989.
20. S.R. Musse, C. Babski, T. Capin, D. Thalmann, Crowd Modelling in Collaborative Virtual Environments, ACM VRST /98, Taiwan
21. S.R. Musse, D. Thalmann, A Model of Human Crowd Behavior: Group Inter-Relationship and Collision Detection Analysis. Proc. Workshop of Computer Animation and Simulation of Eurographics'97, Sept, 1997. Budapest, Hungary.