Digital Actors for Interactive Television

Nadia Magnenat Thalmann
MIRALab, University of Geneva

Daniel Thalmann
Computer Graphics Lab, EPFL

Abstract

In this paper, we define the concept of digital actors and show their importance in the future interactive digital television and multimedia. We discuss the applications of these digital actors in the new multimedia services especially for training and education. We summarize the main techniques to create and control these digital actors and emphasize the importance to develop new tools for the control and the autonomy of these actors. An integrated and functional system for the animation of digital actors is presented. In particular, we describe the integration of motion control techniques, autonomy based on synthetic sensors and facial communication.

Keywords: digital actors, autonomy, motion control, facial communication and animation

1. Why digital actors?

Traditional television means that the viewer may only decide which program he/she wants to watch. With the new developments of digital and interactive television and multimedia products, the viewer will be more and more able to interact with programs and this will lead to individual programs for each viewer.

Autonomous digital actors are important in the multimedia industry where an interactively use of the functionality means an immediate asset. Each Film- and TV producer will be interested to develop new features and programmes where the public will be involved interactively. The authors, editors and publishers of interactive TV programs, CD-I's and CD-ROM's exploiting increasingly interactivity need such a system.

There is a need for systems that provide designers with the capability for embedding simulated humans in games, multimedia titles and film animations. It will make the technology developed for producing computer-generated films available to the entertainment industry.
In the games market, constant innovation is required in order to prevent sales of games from falling off. Convincing simulated humans in games have been identified by the industry as a way of giving a fresh appearance to existing games and enabling new kinds of game to be produced.

There is increasing use of animation in film production including production of video assets for multi-media titles. Providing a capability for simulating people will extend the range of uses of 3-D graphics animation.

The ability to embed the viewer in a dramatic situation created by the behaviour of other, simulated, digital actors will add a new dimension to existing simulation-based products for the education and entertainment on interactive TV. These authorship tools will also provide means for the designer to construct scenarios based on these digital actors.

2. State-of-the-Art in Digital Actors

2.1. Motion Control of Digital Actors

Motion control is the heart of computer animation. In the case of a digital actor, it essentially consists in describing the evolution over time of the joint angles of a hierarchical structure called skeleton.

A general motion control system should be a combination of various techniques: motion capture \([2]\), keyframe \([3]^{4,5,6}\), inverse kinematics \([7]^8\), physically based methods like direct and inverse dynamics \([9]^{10}\), spacetime constraints \([11]^{12,13}\), functional models of walking \([14]^{15,16}\) (Fig.1) and intelligent grasping \([17]^{18}\) (Fig.2).

Integration of different motion generators is vital for the design of complex motion where the characterization of movement can quickly change in terms of functionality, goals and expressivity. This induces a drastic change in the motion control algorithm at multiple levels: behavioral decision making, global criteria optimization and actuation of joint level controllers. By now, there is no global approach which can reconfigure itself with such flexibility.

2.2. Smart autonomous digital actors

For applications like complex games or interactive drama, there is not only a need for motion control but also a way of providing autonomy or artificial smartness to these digital actors. By smartness we mean that the actor does not require the continual intervention of a viewer. Smart actors should react to their environment and take decisions based on perception systems, memory and reasoning. With such an approach, we should be able to create simulations of situations such as digital actors moving in a complex environment they may know and recognize, or playing ball games based on their visual and touching perception.

The need for the digital actors to have smart behaviour arises from two considerations:
in film animations, the more autonomous behaviour that is built into the digital actors, the less extra work there is to be done by the designer to create complete scenarios in interactive games, autonomous human-like behaviour is necessary in order to maintain the illusion in the viewer that digital actors are real ones.

This kind of approach is sometimes called "behavioral animation". For example, Reynolds [19] studied in detail the problem of group trajectories: bird flocks, herds of land animals and fish schools. In the Reynolds approach, each bird of the flock decides its own trajectory without animator intervention. The animator provides data about the leader trajectory and the behavior of other birds relatively to the leader. Haumann and Parent [20] describe behavioral simulation as a means to obtain global motion by simulating simple rules of behavior between locally related actors. Lethebridge and Ware [21] propose a simple heuristically-based method for expressive stimulus-response animation. Wilhelms [22] proposes a system based on a network of sensors and effectors. Ridsdale [23] proposes a method that guides lower-level motor skills from a connectionist model of skill memory, implemented as collections of trained neural networks.

Digital actors should be equipped with visual, tactile and auditory sensors. These sensors are used as a basis for implementing everyday human behaviour such as visually directed locomotion, handling objects, and responding to sounds and utterances. We first introduced the concept of synthetic vision [24] as a main information channel between the environment and the digital actor. Reynolds [25] more recently described an evolved, vision-based behavioral model of coordinated group motion. Xu and Terzopoulos [26] Also Badler et al. [28] reported research on Terrain Reasoning for Human Locomotion.

Digital actors should also be equipped with the ability to navigate past obstacles towards designated destinations. Fig. 3 shows an example of vision-based obstacle avoidance. This is necessary for simulating digital actors walking purposefully in a realistic environment containing objects and other digital actors. Different techniques are required depending on whether the objective is in view or not. In the latter case, some form of internal map of the digital actor's environment must be consulted in order to determine which way to go.

3. Human communication in the Virtual World

3.1. Facial communication between digital actors and viewers

This section discusses real time interaction using visual input from a human face. It describes the underlying approach for recognizing and analyzing the facial movements of a real performance. The output in the form of parameters describing the facial expressions can then be used to drive one or more applications running on the same or on a remote computer. This enables the user to control the graphics system by means of facial expressions. This is being used primarily as a part of a real-time facial animation
system, where the synthetic actor reproduces the animator's expression. This offers interesting possibilities for teleconferencing as the requirements on the network bandwidth are low (about 7 Kbit/s).

In performance driven facial animation, the method enables recognition of facial expressions of a real person which are appropriately mapped as controlling parameters to simulate facial expressions of a synthetic actor in real time. In other applications, the extracted parameters can provide real time estimates of positions and orientations in a virtual scene. The system requires a video camera (CCD) input and extracts motion parameters through a small set of visually tracked feature points.

3.2. Recognition of facial expressions

Recognition of facial expressions is a very complex and interesting subject. However, there have been numerous research efforts in this area. Mase and Pentland [29] apply optical flow and principal direction analysis for lip reading. Terzopoulos and Waters [30] reported on techniques using deformable curves ("snakes") for estimating face muscle contraction parameters from video sequences. Waters and Terzopoulos [31] modeled and animated faces using scanned data obtained from a radial laser scanner and used muscle contraction parameters estimated from video sequences. Saji et al. [32] introduced a new method called "Lighting Switch Photometry" to extract 3D shapes from the moving face. Kato et al. [33] use isodensity maps for the description and the synthesis of facial expressions. Most of these techniques do not extract information in real time. There are some implementations of recognition of facial expressions which use colored markers painted on the face and/or lipstick [34] [35] [36]. However, use of markers is not always practical and methods are needed to enable recognition without them. In another approach Azarbayejani et al. [37] use extended Kalman filter formulation to recover motion parameters of an object. However, the motion parameters include only head position and orientation. Li et al. [38] use the Candid model for 3D motion estimation for model based image coding. The size of the geometric model is limited to only 100 triangles which is rather low for characterizing the shape of a particular model. Magnenat-Thalmann et al. [39] propose a real time recognition method based on "snakes" as introduced by Terzopoulos and Waters [28]. The main drawback of this approach, is that the method relies on the information from the previous frame in order to extract information for the next one. This can lead to accumulation of error and the "snake" may completely lose the contour it is supposed to follow. To improve the robustness we adopt a different approach, where each frame can be processed independently from the previous one.

Accurate recognition and analysis of facial expressions from video sequence requires detailed measurements of facial features. Currently, it is computationally expensive to perform these measurements precisely. As our primary concern has been to extract the features in real time, we have focused our attention on recognition and analysis of only a few facial features.

3.3. Facial animation
There has been extensive research done on basic facial animation and several models have been proposed. In the early models proposed by Parke [40, 41], he has used a combination of digitized expressions and linear interpolation of features such as eyelids and eyebrows and rotations for jaw. Platt and Badler [42] have proposed a model that simulates points on the skin, muscle, and bone by a set of interconnected 3D network of points using arcs between selected points to signify relations. Waters [43] proposes a muscle model which is not specific to facial topology and is more general for modifying the primary facial expression. In this model, muscles are geometric deformation operators which the user places on the face in order to simulate the contraction of the real muscles. Magnenat Thalmann et al. [44] introduced a way of controlling the human face based on the concept of abstract muscle action (AMA) procedures. An AMA procedure is a specialized procedure which simulates the specific action of a face muscle. Terzopoulos and Waters [45] have extended the Waters model, using three layered deformable lattice structures for facial tissues. The three layers correspond to the skin, the subcutaneous fatty tissue, and the muscles. Kalra et al. [46] propose the simulation of muscle actions based on Rational Free Form Deformations (RFFDs). Recently several authors have provided new facial animation techniques based on the information derived from human performances [47, 48, 29]. Finally, Cassell et al. [49] have described a system which automatically generates and animates conversations between multiple human-like agents with appropriate and synchronized speech, intonation, facial expressions, and hand gestures.

4. Our digital actor system

4.1. Architecture of the system

The heart of the system described in this paper is part of the European project HUMANOID, a Parallel Realtime System for Virtual Humans (Fig.4 and Fig.5). A prototype of the system has been implemented including all functions described in this paper. Integration of the various components have been ensured by a general common architecture described in the appendix.

In order to solve the problem of real-time motion adaptation, we have designed a generalization of the hierarchical structure of the environment. On this base, a new human model has been developed [50] and the associated motion control modules are now available (keyframe, inverse kinematics, dynamics [51]). The general architecture and functionality of the TRACK system have been described in a recent article [52]. Our walking model has also been ported on the new environment in both centralized and distributed architectures. A prototype of grasping process [17] has been also developed and integrated into TRACK. The system takes into account the blending of various motion generators and the time modulation of predefined motions in order to fit the requirements of real-time interaction.

4.2. Synthetic sensors
Our digital actors are equipped with visual, tactile and auditory sensors. The simulation of the touching system consists in detecting contacts between the digital actor and the environment. For auditive aspects, we developed a framework for modeling a 3D acoustic environment with sound sources and microphones. Now, our virtual actors are also able to hear.

The most important perceptual subsystem is the vision system [23]. In our approach, synthetic vision provides the actor with a realistic information flow from the environment. To simulate human behavior, i.e. the way a human reacts to his/her environment, we should simulate the way the actor perceives the environment.

Artificial vision is an important research topic in robotics, artificial intelligence and artificial life. But the problems of 3D recognition and interpretation are not yet generally solved. With synthetic vision, we do not need to address these problems of recognition and interpretation. The same reasoning may be applied to the simulation of the auditory system.

For synthetic vision, each pixel of the vision input has the semantic information giving the object projected on this pixel, and numerical information giving the distance to this object. So, it is easy to know, for example, that there is a table just in front at 3 meters. With this information, we can directly deal with the problematic question: "what do I do with such information in a simulation system?"

A vision based approach for digital actors is a very important perceptual subsystem and is for example essential for navigation in virtual worlds. It is an ideal approach for modeling a behavioral animation and offers a universal approach to pass the necessary information from the environment to the digital actor in the problems of path searching, obstacle avoidance, and internal knowledge representation with learning and forgetting characteristics.

We model the digital actor brain with a visual memory and a limited reasoning system allowing the digital actor to decide his motion based on information. Our approach is based on Displacement Local Automata (DLA), similar to scripts [53] for natural language processing. A DLA is a black box which has the knowledge allowing the digital actor to move in a specific part of his environment. The controller is the thinking part of our system; it makes decisions and performs the high-level actions. In an unknown environment, it analyzes this environment and activates the right DLA. In the simple case of a known environment, the controller directly activates the DLA associated with the current location during the learning phase. From information provided by the controller, a navigator builds step by step a logical map of the environment.

More complex problems come when the digital actor is supposed to know the environment, which means the introduction of a digital actor memory. Using his vision, the digital actor sees objects and memorize them, based on an octree representation. Then, he may use this memory for a reasoning process. For example, a recursive algorithm allows a path to be found from the digital actor to any position avoiding the
obstacles based on his memory. The digital actor should also be able to remember if there is no path at all or if there are loops as in a maze. Once a digital actor has found a good path, he may use his memory/reasoning to take the same path. However, as new obstacles could have been added on the way, the digital actor will use his synthetic vision to decide the path, reacting to the new obstacles.

To illustrate the capabilities of the synthetic vision system, we have developed several examples. First, a digital actor is placed inside a maze with an impasse, a circuit and obstacles. The digital actor's first goal is a point outside the maze. After some time, based on 2D heuristic, the digital actor succeeds in finding his goal. When he has completely memorized the impasse and the circuit, he avoided them. After reaching his first goal, he had nearly complete visual octree representation of his environment and he could find again his way without any problem by a simple reasoning process. A more complex example is concerned with the simulation of vision-based tennis playing. Tennis playing (Fig.6) is a human activity which is mainly based on the vision of the players. In our model, we use the vision system to recognize the flying ball, to estimate its trajectory and to localize the partner for game strategy planning.

4.3. Facial animation recognition / synthesis

In this section, we describe a prototype system for communication between a digital actor and a real person. As shown in Fig. 7, the system is mainly a dialog coordinator program with facial data as input channel and facial animation sequences as output channel. The dialog coordinator program can just reproduce the facial expressions of the real person on the virtual actor (Fig. 8) or it can make the virtual actor responding to some emotions by a specific behavior: e.g. to be angry when the user smiles.

Our recognition method relies on the "soft mask," which is a set of points adjusted interactively by the user on the image of the face. Using the mask, various characteristic measures of the face are calculated at the time of initialization. Color samples of the skin, background, hair etc., are also registered. Recognition of the facial features is primarily based on color sample identification and edge detection. Based on the characteristics of a human face, variations of these methods are used in order to find the optimal adaptation for the particular case of each facial feature. Special care is taken to make the recognition of one frame independent from the recognition of the previous one to avoid accumulation of error. The data extracted from the previous frame is used only for the features that are relatively easy to track (e.g. the neck edges), making the risk of error accumulation low. A reliability test is performed and the data is reinitialized if necessary. This makes the recognition very robust. The method enables extraction of the following facial features.

* vertical head rotation (nod)

* horizontal head rotation (turn)
* head inclination (roll)

* eyes aperture

* horizontal position of the iris

* eyebrow elevation

* horizontal distance between the eyebrows (eyebrow squeezing)

* jaw rotation

* mouth aperture

* mouth stretch/squeeze

More details on the recognition method for each facial feature may be found in Pandzic et al. [54].

Facial animation, as any other animation, typically involves execution of a sequence of a set of basic facial actions. Each basic facial motion parameter, called a Minimum Perceptible Action (MPA) [55] has a corresponding set of visible movements of different parts of the face resulting from muscle contraction. Muscular activity is simulated using rational free form deformations [56]. We can aggregate a set of MPAs and define expressions and phonemes. Further these can be used for defining emotion and sentences for speech. Animation at the lowest level, however, is specified as a sequence of MPAs with their respective intensities and time of occurrence. The discrete action units defined in terms of MPAs can be used as fundamental building blocks or reference units for the development of a parametric facial process. Development of the basic motion actions is nonspecific to a facial topology and provides a general approach for the modeling and animation of the primary facial expressions. In our facial model the skin surface of the face is considered as a polygonal mesh. It contains 2500-3000 polygons to represent the shape of the model. Hence, the model considered has sufficient complexity and irregularity to represent a virtual face, and is not merely represented as a crude mask as considered in many other systems.

For the real time performance driven facial animation the input parameters to the facial animation are the MPAs. These MPAs have normalized intensities between 0 and 1 or -1
The analysis of the recognition module is mapped appropriately to these MPAs. In most cases the mapping is straightforward. Due to the speed constraint we have concentrated on only few parameters for the motion. This reduces the degrees of freedom for the animation. However, we believe that complete range of facial motion is practically not present in any particular sequence of animation. To mimic the motion of a real performance only a set of parameters is used.

With the input from real performance we are able to reproduce individual particular feature on the synthetic actor's face (e.g. raising the eyebrows, opening the mouth etc.) in real time. However, reproducing these features together may not faithfully reproduce the overall facial emotion (e.g. smile, surprise etc.). In order to achieve this a better interpreting/analyzing layer between recognition and simulation may be included. Use of a real performance to animate a synthetic face is one kind of input accessory used for our multimodal animation system. The system can basically capture the initial template of animation from real performance with accurate temporal characteristics. This motion template then can be modified, enhanced and complemented as per the need by other accessories for the production of final animation.

5. Performance and real-time

Our purpose is to provide a real-time animation system dedicated to applications centered on human modeling and control. The Hardware configuration include a SGI INDIGO II extreme workstation connected to a Transputer network for the computation of well identified tasks with high computational cost. The distribution of Computational load has been decided for the dynamics simulation, the deformation process and the collision detection. It has already been clearly identified that the human model should be distributed on several transputers to carry these tasks. A clear requirement is that this partition of the human model in the transputer local memory is the same for the different tasks. In a distributed implementation, the motion blending is managed on the local hierarchy maintained by the transputer. The motion of the root of this hierarchy depends on the motion of other body parts located on other transputers (e.g. the motion of the arm root depends on the motion of the torso). So, for some motion generators (inverse kinematics, dynamics) and for collision detection it is necessary to import the root motion to ensure consistency of motion state and collision. Figure 9 shows the corresponding data flow.

For the facial animation recognition, we use a professional CCD camera. We have undertaken extensive tests of the recognition system with various persons using it for real time facial animation, walkthrough and object manipulation. Our real time facial expression recognition system is capable of extracting adequate quantity of relevant information at a very satisfying speed (10 frames/s). The system also checks the conditions. If the user positions himself in such a way that the recognition cannot proceed (e.g. if he leaves the camera field of view or turns away from the camera) the system issues a warning sign, and an appropriate signal to the application(s). At this time, the major drawback is that the recognition doesn't work equally well for all the people. In particular, the bald people cannot use our system. The users with pale blond hair and
eyebrows may have problems. The people with irregular haircuts and hair falling on the forehead have to use some hairpins to straighten the hair. We are exploring better techniques to make the program more general.

For real time performance driven facial animation the number of parameters are limited by the features extracted from the recognition module. We intend to use some heuristics and rules to derive more information from the extracted features to improve the qualitative performance of the animation of the virtual actor. To add realism for the rendering we also intend to add texture information which will be captured from the real performance. As the recognition method for facial expressions in our system does not use any special markers or make-up, it may easily be used in a multimedia environment and interactive TV with camera input facilities. No "training" period is necessary for the system. The system is adequately fast, reasonably robust and adaptable for a new user with quick initialization. We believe that facial interaction will have it's place among the interaction techniques in the near future. At this time no visual speech acquisition has been planned.

6. Future work

In the near future, we intend to create interactive and immersive real-time simulations of our smart virtual actors. These actors will be able:

to move from one place to another by walking, bypassing, jumping or climbing obstacles.

to move objects in the Virtual Space

Locomotion, processing of the obstacles and grasping will be based on the three synthetic sensors: vision, audition and touch.

The simulation will be performed in Virtual Environments allowing the participant (real human) to move the obstacles and obstacles in the Virtual Space using a VR-device (DataGlove).

Acknowledgments

The authors would like to thank Ronan Boulic, Zhyong Huang, Prem Kalra, Hansrudi Noser and Igor Pandzic for their participation to the projects. Part of the research described in this paper is supported by "Le Fonds National Suisse pour la Recherche Scientifique", and the ESPRIT R & D project HUMANOID (OFES).

References

Appendix: Architecture of the DIGITAL ACTORS system
Fig.A1 shows the general organization of our system with the associated libraries SCENELIB, BODYLIB, ANIMALIB and AGENTLIB. This appendix describes these libraries and their relationships.

The library SCENElib is dedicated to the design of general purpose 3D hierarchy. The basic building block is called the node_3D. By default it just maintains information of positioning a frame in 3D space. Obviously, SCENElib is dedicated to design and handle a flexible representation of motion propagation.

The library BODYlib is dedicated to the design of a specialized 3D hierarchy with a fixed skeleton-like topology. It maintains the low level information of a UNIT which is part of a general purpose SCENE. Figure A2 presents the relationship between SCENELIB and BODYLIB. The purpose of the BODY data structure is to maintain a topological tree structure for a vertebrate body with predefined mobility, a corresponding volume discretisation with mass distribution and a corresponding envelope. A general mechanism allows to customize the skeleton structure at two levels, either at a high level with a small set of scaling parameters or at the low level of the position and orientation of the various articulations defined in the SKELETON data structure. In both cases the modifications are propagated to the lower level structure of the volume and envelope.

A deformation function can be triggered to compute skin surface according to the body posture. A BODY is the only entity which can compute the deformations of its external surface. This is the most computationally expensive process of an animation with BODY entities. The body deformation is based on current position and joint angles of the skeleton. We use a layered model based on 3 interrelated levels:

* the underlying articulated skeleton hierarchy composed of only articulated line segments whose movements are controlled with the JOINT data structure. It may be animated using motion generators.

* a layer is composed of metaball primitives attached to the JOINT of the skeleton. By transforming and deforming the metaballs, we can simulate the gross behavior of bones and muscles.

* the skin surface of the body automatically derived from the position and shape of the first and second layer. Internally, we define every part of the body as a set of B-spline patches, then tessellate the B-spline surfaces into a polygonal mesh for smooth joining different skin pieces together and final rendering.

The purpose of ANIMALIB is to manage the integration of various sources of motion for a BODY data structure in particular or more generally for 3D hierarchical entities. An ANIMA data structure (Fig. A3) is designed to carry on that function. An application dedicated to the animation of a complex environment with multiple human figures will
manage as many ANIMA data structures as animated human figures and animated sub-
hierarchies. An ANIMA maintains and coordinates various entities. The most important
one is the GENERATOR. This generic entity is designed to facilitate the plug in of
various motion control modules into a common framework for motion integration. The
following control modules have been implemented:

**KFR:** keyframe module, used on one hand as a GENERATOR but also as a means to
specify input, to record output and to blend sampled motions

**INVK:** inverse Kinematics, for an open chain, drives the joint values from a
specification of the position variation of an end effector.

**DYN:** Direct Dynamics, simulates the motion of an articulated structure holding solid
node_3d, from the specification of forces and torques over time.

**WALK:** accept a high level input expressed either in terms of a normalized velocity or
as a position to reach in the plane.

**GRASP:** grasping with one or two hands with motion by inverse kinematics and
keyframing (perception and decision are not considered at that level)

ANIMALIB is an integration scheme of various motion GENERATORs, it does not
handle directly the function of producing various kind of motion; it just mixes them and
is responsible of the final update at the scene level. It eventually corrects the resulting
motion with the Coach-Trainee method.

The goal of the TRACK application is to manage the motion control and combination of
one to many human models including the following GENERATORs: keyframing,
dynamics, inverse kinematics, walking and grasping.

The INTEGRATOR function is used to mix various motion sources. Each of them is
associated with a weight which can vary over time with a KFR sequence. Then, the sum
of all the weighted motion is evaluated, recorded in a KFR sequence and applied to the
general hierarchical 3D structure. Finally the COLLISION-DETECTOR provides the
information of self-collisions and collision with the environment.

The distributed scheme of animation is necessary in the perspective of behavioral
animation where each agent acts autonomously. In such a way, the ANIMA data structure
handles all the necessary information for the processing of the motion of the agent. As a
simple way to capture the purpose of the ANIMA with respect to the AGENT, we can
say that:

an ANIMA is responsible of the "reflex control".

an AGENT is responsible of the "reflexive control"
However, the ANIMA data structure and integration scheme remains at the low level of control (Kinematics, Dynamics, Inverse Kinematics and Dynamics, output of functional models). Higher level control paradigm should be explicitly managed in a higher level entity which we will therefore refer to as an AGENT. Basically, an AGENT is responsible of organizing and handling the PERCEPTION information, deriving the DECISION making process and managing the COORDINATION of motion within an ANIMA entity.

AGENTLIB provides high-level functions for the behavior and autonomy of digital actors. Future developments are planned in this library. Today, it already provides the following features: vision, auditory and tactile sensors, navigation and a few task-level functions supported at a lower level by ANIMALIB, especially for walking and grasping.

Figure captions

Fig.1 Walking

Fig.2. Intelligent real-time object grasping

Fig.3. Vision-based obstacle avoidance

Fig.4. Two digital actors

Fig.5. A digital actress

Fig.6. Tennis playing

Fig.7. A facial communication system

Fig.8. Real-time performance animation based on video

Fig.9. Data flow of surface information with (a) and without (b) real-time requirement

Fig.A1 : different levels of information management from SCENE to AGENT

Fig.A2 : BODYLIB is build only over SCENELIB

Fig.A3 : The ANIMA data structure showing the internal components
Nadia Magnenat Thalmann is full Professor of Computer Science at the University of Geneva, Switzerland and Adjunct Professor at HEC Montreal, Canada. She has served on a variety of government advisory boards and program committees in Canada. In 1987, she was nominated woman of the year by the Montreal community in Quebec. She has received several awards, including the 1985 Communications Award from the Government of Quebec, the Moebius Award from the European Community in 1992, and the British Computer Society Award in 1993. Dr. Magnenat Thalmann received a BS in
psychology, an MS in biochemistry, and a Ph.D in quantum chemistry and computer graphics from the University of Geneva. She has written and edited several books and research papers in image synthesis and computer animation and was codirector of the computer-generated films Dream Flight, Englantine, Rendez-vous à Montreal, Galaxy Sweetheart, IAD, Flashback, Still Walking and Fashion Show. She has served as a chairperson of Graphics Interface '85, Computer Graphics International (CGI'88), and the annual workshop and film festival in Computer Animation held in Geneva. She is co-editor-in-chief of the Visualization and Computer Animation Journal, associate editor-in-chief of the Visual Computer, editor of the Computational Geometry Journal and the CADDM journal. She is the President of the Computer Graphics Society (CGS).

Daniel Thalmann is currently full Professor and Director of the Computer Graphics Laboratory at the Swiss Federal Institute of Technology in Lausanne, Switzerland. He is also adjunct Professor at the University of Montreal, Canada. He received his diploma in nuclear physics and Ph.D in Computer Science from the University of Geneva. He is coeditor-in-chief of the Journal of Visualization and Computer Animation, member of the editorial board of the Visual Computer, the CADDM Journal (China Engineering Society) and Computer Graphics (Russia). He is cochair of the EUROGRAPHICS Working Group on Computer Simulation and Animation and member of the Executive Board of the Computer Graphics Society. Daniel Thalmann was member of numerous Program Committees, Program Chair of several conferences and chair of the Computer Graphics International '93 Conference. Daniel Thalmann's research interests include 3D computer animation, image synthesis, virtual reality and digital media. He has published more than 150 papers in these areas and is coauthor of several books including: Computer Animation: Theory and Practice and Image Synthesis: Theory and Practice. He is also codirector of several computer-generated films with synthetic actors.


[52] R. Boulic., Z. Huang, D. Thalmann (1994) "Goal Oriented Design and Correction of Articulated Figure Motion with the TRACK System," Computer and Graphics, 1994


