Proceedings of IJCAI-97 Workshop on Animated Interface Agents
Nagoya, Japan, August 25, 1997, pp. 45–51

A Virtual Human Presenter

Tsukasa Noma
Department of Artificial Intelligence
Kyushu Institute of Technology
Kawazu, Iizuka, Fukuoka 820
JAPAN

Norman I. Badler
Center for Human Modeling and Simulation
University of Pennsylvania
Philadelphia, PA 19104-6389
U. S. A.

Abstract

A virtual human presenter is created based on extensions to the *Jack*™ animated agent system. Inputs to the presenter system are in the form of speech texts with embedded commands, most of which relate to the virtual presenter's body language. The system then makes him act as a presenter with presentation skills in real-time 3D animation synchronized with speech outputs.

1 Introduction

In our society, people make presentations to inform, teach, motivate, and persuade others. Likewise, if a virtual human agent has adequate presentation skills, he can inform us effectively and thus can be an interface agent mediating communication between user and computer.

Thalmann and Kalra made some sequences of a virtual actor acting as a television presenter[Thalmann and Kalra, 1995]. Their production, however, seems to have been performed mostly manually since they did not refer to presentation techniques.

To make a virtual human presenter usable, it should satisfy the following requirements:

(1) *Natural motion with presentation skills.*
To build credibility with users, the virtual presenter's motion should be as natural as possible. In addition, presentation skills, particularly non-verbal skills, should be modeled and embedded in the presenter system so that presentation/interface designers can make effective presentations easily.

(2) *Real-time motion generation synchronized with speech.*
If the virtual presenter acts as a personalized weatherman[Negroponte, 1995], he should report a weather forecast to users as soon as possible. If he acts as an agent in an interactive user interface, he should react immediately depending on users' input. The virtual presenter's motion generation should thus be in real-time. In addition, presentation typically consists of body motion and speech. The body motion should be synchronized with the speech.

(3) *Proper inputs for representing presentation scenarios.*
The form of inputs to the virtual presenter should enable designers to represent presentation scenarios without its detailed description.

This paper presents a virtual human presenter being developed on the *Jack*™ animated agent system[Badler *et al.*, 1993]. It is designed to meet the above requirements. Inputs to the presenter system are in the form of speech texts with embedded commands, most of which relate to the virtual presenter's body language. The system then makes him act as a presenter in real-time 3D animation synchronized with speech outputs.

In the current implementation, a single human agent acts as a presenter with a visual aid which looks like a blackboard or flip-chart. We call it a *virtual board*. Since we can map arbitrary textures on the board, it can show any texts, charts, maps, and images. The size of the board is arbitrary, but the typical size we use is 1.5 to 2 meters square. This typical size is taken from that of visual aids in daily meetings and weather maps in TV weather reports.

2 Inputs

Suppose that a virtual human agent gives a presentation with a speech in Figure 1. To enrich the presentation in our virtual presenter, commands are embedded in the speech text as an input. A sample input to the virtual presenter is shown in Figure 2. A command is a backslash followed by a word. Depending on its type, it is additionally followed by arguments enclosed in braces. For example, `\board{jackpanel}` and `\point_down{jackpanel.board.item1}` are commands in Figure 2.

In our current implementation, there are three types of commands: board commands, point commands, and gesture commands. The board command `\board{}` takes an argument specifying a virtual board as a *figure* in *Jack*. It means that the current board is to be changed to the specified one. `\point_down{}`, `\point_idxf{}`,

```
Welcome to Jack presenter.  This presenter
produces animated presentation from a speech
text in real time.  And you can insert gesture
commands in the speech text.  Then I will make
specified gestures synchronized with speech.

In the current system, we support simple
gestures, for example, giving and taking,
rejecting, and warning.  In addition to these
simple gestures, various pointing gestures are
prepared.  If a pointed site is unreachable, I
will walk to and point at it automatically.

The application area of the presenter is
potentially so vast.  For example, I can be a
weather reporter.  Hurricane Bertha is now to
the east of Florida peninsula.  It is now
going north.  New York and Philadelphia may be
hit directly.  Take care.

In the near future, a weather forecast may be
produced with your own weather reporter like
me.
```

Figure 1: A sample speech text.

`\point_back{}`, and `\point_move{}` are point commands. They are used to make the agent point at the positions specified as arguments, and can take several arguments, to which the presenter points in sequence. The gesture commands, `\gest_givetake`, `\gest_reject`, and `\gest_warning` specify simple arm-hand gestures. They do not take any arguments. The point and gesture commands are described in more detail in Section 4.

The synchronization of body motion and speech is represented by the positions of commands in the input texts. Basically, the motion specified by a command is to coincide with the utterance of a word following the command in the inputs.

To examine the adequacy of the above form of inputs to the presenter, we need to discuss two points: The first point is whether the "annotated" speech texts are appropriate for representing presentation scenarios. Presentation proceeds in parallel with spoken words in speech texts, and the major message is delivered via verbal channel. This is obvious from the terms like "visual *aids*" and "*non*-verbal communication." Speech texts can thus be a temporal axis of presentation scenarios. In addition, presenters in the real world are recommended to insert easily read indicators to coordinate the manuscript in slides, events, or times[Leech, 1993]. From the above facts, our inputs are in an appropriate form for the virtual presenter.

The second and more critical point is whether the speech texts are in an appropriate level of abstraction for an "intelligent" presenter. Some readers may consider that more abstract data, for example, weather/temperature tables for weather reports, would

```
\board{jackpanel} Welcome to Jack presenter.
This presenter produces
\point_down{jackpanel.board.item1} animated
presentation from a speech text
\point_down{jackpanel.board.item2} in real
time.  And you can insert
\point_down{jackpanel.board.item3} gesture
commands in the speech text.  Then I will make
specified gestures synchronized with speech.

\board{gesturepanel} In the current system, we
support simple gestures, for example,
\gest_givetake giving and taking, \gest_reject
rejecting, and \gest_warning warning.  In
addition to
\point_idxf{gesturepanel.board.givetake
gesturepanel.board.reject
gesturepanel.board.warning} these simple
gestures,
\point_idxf{gesturepanel.board.point} various
pointing gestures are prepared.  If a pointed
site is \gest_givetake unreachable, I will
walk to and
\point_idxf{gesturepanel.board.far} point at
it automatically.

The application area of the presenter is
potentially so vast.  For example, I can be a
weather reporter.  \board{berthapanel}
\point_idxf{berthapanel.board.bertha}
Hurricane Bertha is now to the east of
\point_back{berthapanel.board.florida} Florida
peninsula.  It is now going
\point_move{berthapanel.board.bertha
berthapanel.board.north} north.
\point_idxf{berthapanel.board.ny
berthapanel.board.phil} New York and
Philadelphia may be hit directly.
\gest_warning Take care.

In the near future, a weather forecast may be
produced with your own weather reporter
\gest_givetake like me.
```

Figure 2: A sample input to the virtual presenter.

be more desirable for inputs to the presenter. But appropriate styles of speech texts vary depending on applications, and so do the requirements for text generators. For example, speech texts for academic paper presentation are completely different from those for weather reports. Compared with speech texts, however, non-verbal presentation skills are much more independent of applications. We thus designed our input form so that our virtual human presenter can be a common tool in various areas, and application-dependent speech text generation should be put on preprocessing. This design decision in our presenter is also appropriate for a case where a presentation designer would like to edit a presentation scenario after its speech text is generated automatically.

3 Control

3.1 Control via PaT-Nets

Our virtual human presenter is controlled by PaT-Nets (Parallel Transition Networks)[Cassell *et al.*, 1994]. They are simultaneously executing finite state automata. Every clock tick, they call for action and conditionally make state transitions.

Since the existing implementation of PaT-Nets on Lisp interpreter was inappropriate for real-time animation, we re-implemented PaT-Nets in C++. Each class of PaT-Nets is defined as a derived class of class `LWNet`, which stands for Light Weight PaT-Net. Its nodes are defined with their associated actions and transition rules in its constructor. Instances of PaT-Nets are stored on a list in *Jack*, and they are scanned every tick.

Other features of the C++-PaT-Nets include:

(1) The definition of PaT-Nets is extended so that it can have multiple states at the same time. It enables us to represent simple parallel execution of actions in a single PaT-Net.

(2) A PaT-Net can send messages to other PaT-Nets (message passing), and can also wait for their reply.

In the current implementation of our virtual presenter, nine PaT-Nets are running in parallel. The structure of the nets is shown in Figure 3. Each arrow represents message passing between the nets. How they work is discussed in the following subsections.

3.2 PaT-Nets as Body Parts

In our virtual presenter, groups of body parts are assigned to individual PaT-Nets: `WalkNet`, `ArmNet`, `HandNet`, and `SeeNet`. For example, an `ArmNet` manages clavicle, shoulder, elbow, and wrist joints of a single arm. To move the arm, we do not need to directly assign their joint angles. All we have to do is to send a message to the `ArmNet`. The `ArmNet` move the joints depending on messages such as "pointing a particular position on a virtual board (via inverse kinematics)" or "taking a particular arm posture."

These PaT-Nets as body parts have an advantage over conventional animation systems where motion generators directly assign the joint angles of the whole body.

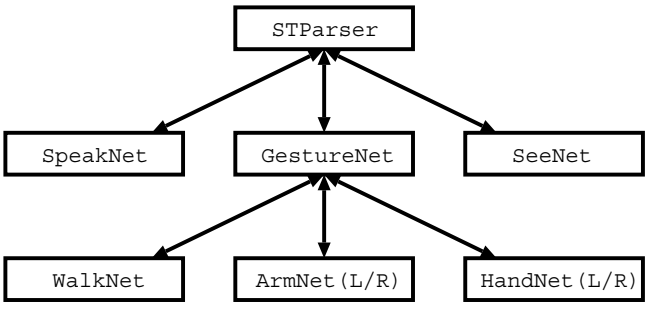

Figure 3: The PaT-Nets structure in the virtual presenter.

In the conventional systems, the motion generators need to manage the transition from the current motion to the next motion. In many cases, the human body needs to return to a "neutral" posture before the next motion, or the transitions between motions need to be defined for every pair of motions in advance. On the other hand, in our virtual presenter, if the higher-level motion generator, e.g. `GestureNet`, issues a "freeing" message to the `ArmNet`, the `ArmNet` locally moves the arm to its neutral position or another particular posture depending on the following messages. Thus the high-level motion generators do not need to worry about transitions between motions.

3.3 Parsing on a PaT-Net

In our virtual presenter, the input form is within regular language supposing that commands, (lists of) words, and braces are treated as tokens. On the other hand, PaT-Nets are finite automata. We thus parse the inputs by the highest-level PaT-Net called `STParser` in Figure 3, and make it control the whole PaT-Nets structure.

The `STParser` has 51 nodes and make transitions depending on the input tokens. It can parse the inputs sequentially in real-time, and the output animation throughput is independent of the input length.

3.4 Interfacing via Sockets

In addition to file inputs, our virtual presenter can be controlled by inputs via interprocess communication of TCP/IP socket[Stevens, 1991].

Its usage is simple. Except for opening and closing connections, only two functions are necessary for controlling the presenter: `jvp_sendstrwait()` sends a string of command-embedded speech texts to the presenter and waits for it to be fully performed by the virtual human agent. Similarly, `jvp_sendfilewait()` sends a file of command-embedded speech texts and waits for its termination. Even for inputs via socket, the agent's performance is generated in real-time.

This means that other processes/programs can use our virtual presenter as a subsystem. They can be freed from manipulating the virtual human body itself, and thus can be devoted to scenario control with speech text

generation and gesture specification. It enables us to create an interactive system with an animated virtual human agent in a simplified fashion.

4 Presentation Skills

In presentation, messages are delivered over both audio and video channels. Moreover, the effect of non-verbal communication is much greater than that of verbal messages[Bergin, 1995; Brody and Kent, 1993; Leech, 1993]. To be a good presenter, our virtual human agent should have presentation skills, particularly via non-verbal channels.

This section discusses presentation skills modeled in our virtual presenter. They are based on presentation know-hows taken from books on presentation and public speaking.

Posture. Posture is a highly visual element of presentation. Presenters should stand up straight with both feet slightly apart and firmly planted on the floor[Becker and Becker, 1994; Bergin, 1995; Brody and Kent, 1993; Kupsh and Graves, 1993; Kushner, 1996; Snyder, 1990]. This posture is said to convey confidence[Bergin, 1995; Kupsh and Graves, 1993]. Even for human presenters, arm and hand positions are serious problems when they are not in use. Brody and O'Connor advise us to let our arms hanging down naturally at our sides[Brody and Kent, 1993; O'Connor, 1996]. Our virtual presenter follows the above rules by default.

Presenters' shoulders should be oriented to the audience[Brody and Kent, 1993; Mandel, 1993], which is interpreted as the viewpoint for the virtual presenter. The shoulder orientation, however, become critical when using visual aids, like our presenters, mainly for pointing them. According to [Mandel, 1993], even if presenters need to angle away from the audience, it should not be more than 45 degrees. Thus in our virtual presenter, the angle between viewpoint and facing direction is 45 degrees by default.

Gesture. Gestures can emphasize the important points and then back up verbal messages in presentation. Rozakis[Rozakis, 1995] mentioned the 6 traditional speech gestures: giving and taking, fist, pointing, rejecting, dividing, and warning. But according to Snyder[Snyder, 1990], the fist and the *karate*-chop, which appears in "dividing," should be avoided. We thus implemented:

(1) *Giving and taking.* (Command: `\gest_givetake`)
This is placing the hand out with the palm turned upward to propose a new idea or ask for something.

(2) *Pointing.*
This is basically pointing the index finger to indicate position. It is mentioned later with its varieties.

(3) *Rejecting.* (Command: `\gest_reject`)
This is sweeping the hand, palm downward.

(4) *Warning.* (Command: `\gest_warning`)
This is placing the hand straight out like a stop sign — palm out, heel of the hand down.

With these commands, their corresponding gestures are performed at their inserted positions in the input texts.

Except for these "meaningful" gestures, our virtual presenter will move as little as possible. This is because there is a golden rule in presentation that "Don't do anything that draws attention to itself."[Becker and Becker, 1994]

Pointing. Pointing gestures are used to make quick visual references on visual aids. From pointing examples in real-world presentations, we modeled and implemented four types of pointing gestures:

(1) *Pointing the index finger.* (Command: `\point_idxf{}`)
This is the basic pointing gesture which indicates a point or small portion on the visual aid.

(2) *Pointing with the palm facing backward.* (Command: `\point_back{}`)
This gesture indicates a larger area on the visual aid than (1).

(3) *Pointing with the palm facing downward.* (Command: `\point_down{}`)
This is often used to emphasize a phrase in a text or an item on a list.

(4) *Moving the hand in the palm facing direction.* (Command: `\point_move{}`)
This is used to show a flow on a map or chart.

Pointed positions are specified as *sites* in *Jack*, and enclosed in braces. If multiple sites are specified, they are pointed one by one. Currently, these sites must be manually inserted in the figure definitions in advance.

To avoid crossing the arm over the body and then to keep the body posture open, our virtual presenter is designed to use the nearer hand to the pointed position[Brody and Kent, 1993; Mandel, 1993; O'Connor, 1996; Snyder, 1990].

As the size of visual aids gets larger, presenters cannot point on the visual aids from a fixed body position. The presenter thus need to move before pointing if he cannot point the next location from the current body position. In our presenter, (a list of) the next pointed site(s) is read in advance from the input stream, and if he needs to move, he does so before his speech reaches the point when he points. Such an anticipation gives more realism in virtual presentation, and locomotion itself makes an active stage picture and then makes him more interesting for the audience to watch[Kushner, 1996]. Locomotion for the virtual presenter is discussed in the next section.

If the presenter needs to move, choosing the left or right hand for the next pointing gesture is also an important issue. The hand choice determines the next body position and the extent of the visual aid (virtual board) blocked to the audience's view[Brody and Kent, 1993; Leech, 1993]. The hand selected for pointing is thus determined by a heuristic which minimizes both visual aid occlusion and the distance from the current body position to the next one.

Eye Contact. Many authors emphasize the importance of eye contact with the audience[Becker and

Becker, 1994; Bergin, 1995; Brody and Kent, 1993; Kupsh and Graves, 1993; Kushner, 1996; Leech, 1993; Mandel, 1993; O'Connor, 1996; Rozakis, 1995; Snyder, 1990]. In case of public presentation by real humans, presenters should vary the person they look at(e.g. [Brody and Kent, 1993]). But in case of our virtual presenter, he is designed to talk to the fixed viewpoint like the TV camera. This means that he talks to every person in the audience directly, eye to eye[Snyder, 1990].

Although the eye contact is important, the presenter must not look at the audience every time. When he points some position on visual aids, he should glance at it to direct the audience's attention to the visual aid and its pointed position[Hendricks *et al.*, 1996]. After pointing, he needs to look back immediately at the audience, and then keep the eye contact. Our presenter acts as above.

5 Locomotion

A number of approaches to human locomotion animation have been studied, and most of them generate forward locomotion along straight or curved paths (e.g. [Boulic *et al.*, 1990; Bruderlin *et al.*, 1994]). But presenters often need to step laterally or backward, and thus more broadly capable locomotion engine is desirable for our virtual presenter.

Ko and Cremer[Ko and Cremer, 1996] proposed VRLOCO locomotion engine, which has five locomotion modes — walking, running, lateral stepping, turning around, and backward stepping — and can make smooth transitions between these locomotion modes.

Applying VRLOCO to our virtual presenter, however, has two problems: The first problem is that inputs to VRLOCO are streams of body center position P_i and facing direction $\overline{F}_i$ with time t_i. On the other hand, in our virtual presenter, only the final body center position and facing direction can easily be evaluated. Thus obtaining proper streams of $(t_i, P_i, \overline{F}_i)$ from the final $(P_{final}, \overline{F}_{final})$ is itself a problem.

The second problem is that each locomotion by the virtual presenter is normally within a few steps, and within these steps, he moves his body position and changes his facing direction simultaneously. It is thus too short for applying specific locomotion modes like VRLOCO.

From the above discussions, we developed yet another locomotion engine, whose distinctive feature is that it covers forward/lateral/backward stepping and turning around in a unified fashion. The major idea of our locomotion engine is that the next step is chosen among the possible steps so that the virtual presenter's body approaches the goal $(P_{final}, \overline{F}_{final})$ as much as possible. In this locomotion engine, a step is represented by a triplet (θ_L, θ_R, d) (Figure 4).

To be concrete, the next step is determined by the following rule:

(1) If the next swing foot can be placed on the final position and in the final direction, then do so.

(2) Otherwise, if the next swing foot can be placed on the final position, then do so.

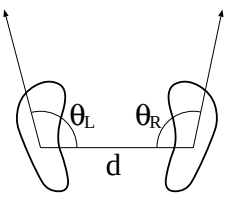

Figure 4: Representation of a step.

(3) Otherwise, if the next swing foot is not in the direction of the final body position, then change the direction into it as much as possible, and if still permitted, move the body center to the final body position as much as possible.

(4) Otherwise, move the body center to the final body position as much as possible.

We thus modeled the above rule in a PaT-Net called `WalkNet`. It has eight nodes, each of which corresponds to a single step. To calculate joint angles for each frame, we prepared the joint angles of key frames on some fixed points in $\theta_L\theta_R d$-space in advance. We then obtain the joint angles of key frames for a given (θ_L, θ_R, d) by interpolation, and finally generate the joint angles via keyframe animation techniques.

6 Results

We implemented the above system on a SGI's Onyx/RealityEngine. Inputting the speech texts with commands (e.g. Figure 2), presentation animations are generated in real-time (30 frames/sec), except for some time-consuming operations, e.g. changing to another virtual board. Images in Figure 5 are taken from the sample presentation.

For voice output, we used a Entropic Research Laboratory's TrueTalk^{TM} TTS (Text-To-Speech) system[Entropic Res. Lab., 1995] running on a SGI's Indigo2. For its control, a PaT-Net called `SpeakNet` drives the TrueTalk via TCP/IP socket. The `SpeakNet` also mimics the mouth motion by moving the jaw joint randomly during his speech. The synchronization between animation output and voice output was sufficiently satisfactory.

We also implemented an interactive weather reporter as a client of our presenter via TCP/IP socket. The client was able to control the virtual presenter in an interactive fashion.

7 Conclusions

We discussed a presentation system where a virtual human agent acts as a presenter from the inputs of speech texts with gesture-related commands. With the PaT-Net-based design, presentation animations are generated in real-time, synchronized with voice output. The presentation skills were modeled from guidelines that appear in books on public speaking. A new locomotion

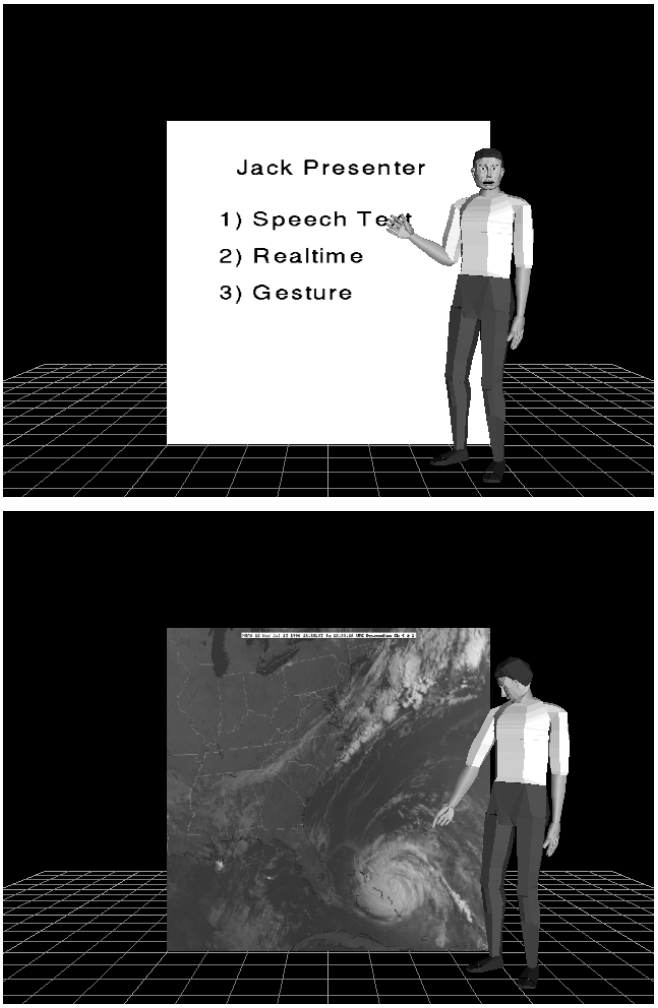

Figure 5: Images from sample presentation.

engine was developed to integrate a variety of locomotion modes. Thus, particularly with the socket interface, controlling an animated virtual human interface agent is reduced into generating speech texts with gesture specification.

An alternative design of our presenter would be using sequences of 2D video images like "video widgets and video actors"[Gibbs *et al.*, 1993]. In our view, however, 3D animation offers superior flexibility, generalized control, and future compatibility with VRML browsers. The advantages increase as applications and computer platforms improve.

In our current implementation, there is still much room for improvement: e.g. (1) supplying facial expressions, (2) enriching gesture commands, and (3) modeling further rules for presentation. And to use our presenter more effectively, site specifications in figures, speech text generation, and annotation (command embedding) should be further investigated.

Our virtual human presenter has many potential applications. For example, the agent can make sales presentations depending on consumers' preferences. Such applications will have a great impact on merchandise marketing. To make our presenter of extensive use in such areas, protocols for driving presenters over the network should be established in the future.

Acknowledgements

The first author's visit to the University of Pennsylvania is financially supported by the Japanese Ministry of Education, Science, Sports and Culture as overseas research fellow program. The satellite image of Hurricane Bertha is from NOAA / National Climatic Data Center. This research is partially supported by DARPA DAMD17-94-J-4486; U.S. Air Force through BBN F33615-91-D-0009/0008; ONR through Univ. of Houston K-5-55043/3916-1552793; DARPA SB-MDA-97-2951001 through the Franklin Institute; Army AASERT DAAH04-94-G-0220; DARPA AASERT DAAH04-94-G-0362; NSF IRI95-04372; and JustSystem Japan.

References

[Badler *et al.*, 1993] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, 1993.

[Becker and Becker, 1994] Dennis Becker and Paula Borkum Becker. *Powerful Presentation Skills*. Irwin, 1994.

[Bergin, 1995] Francis Bergin. *Successful Presentations*. Director Books, 1995.

[Boulic *et al.*, 1990] Ronan Boulic, Nadia Magnenat Thalmann, and Daniel Thalmann. A global human walking model with real-time kinematic personification. *The Visual Computer*, 6(6):344–358, 1990.

[Brody and Kent, 1993] Marjorie Brody and Shawn Kent. *Power Presentations*. Wiley, 1993.

[Bruderlin *et al.*, 1994] Armin Bruderlin, Chor Guan Teo, and Tom Calvert. Procedural movement for articulated figure animation. *Computers and Graphics*, 18(4):453–461, 1994.

[Cassell *et al.*, 1994] Justine Cassell, et al. Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *Proc. SIGGRAPH 94*, pages 413–420, July 1994.

[Entropic Res. Lab., 1995] Entropic Research Laboratory. *TrueTalk Programmer's Manual*, 1995.

[Gibbs *et al.*, 1993] Simon Gibbs, Chiristian Breiteneder, Vicki de Mey, and Michael Papathomas. Video widgets and video actors. In *Proc. UIST '93*, pages 179–185, Nov. 1993.

[Hendricks *et al.*, 1996] William Hendricks, Micki Holliday, Recie Mobley, and Kristy Steinbrecher. *Secrets of Power Presentations*. Career Press, 1996.

[Ko and Cremer, 1996] Hyeongseok Ko and James Cremer. VRLOCO: real-time human locomotion from positional input streams. *Presence*, 5(4):367–380, 1996.

[Kupsh and Graves, 1993] Joyce Kupsh and Pat R. Graves. *How to Create High-Impact Business Presentations*. NTC Business Books, 1993.

[Kushner, 1996] Malcolm Kushner. *Successful Presentations for Dummies*. IDG Books Worldwide, 1996.

[Leech, 1993] Thomas Leech. *How to Prepare, Stage, and Deliver Winning Presentations*. 2nd ed., AMACOM, 1993.

[Mandel, 1993] Steve Mandel. *Effective Presentation Skills*. Revised ed., Crisp Publications, 1993.

[Negroponte, 1995] Nicholas Negroponte. *Being Digital*. Random House, 1995.

[O'Connor, 1996] J. Regis O'Connor. *High-Impact Public Speaking for Business and the Professionals*. NTC Publishing Group, 1996.

[Rozakis, 1995] Laurie E. Rozakis. *The Complete Idiot's Guide to Speaking in Public with Confidence*. Alpha Books, 1995.

[Snyder, 1990] Elayne Snyder. *Persuasive Business Speaking*. AMACOM, 1990.

[Stevens, 1991] W. Richard Stevens. *UNIX Network Programming*. Prentice-Hall, 1991.

[Thalmann and Kalra, 1995] Nadia Magnenat Thalmann and Prem Kalra. The simulation of a virtual TV presenter. In *Computer Graphics and Applications (Proc. Pacific Graphics '95)*, pages 9–21, World Scientific, August 1995.