Design of a virtual human presenter

Design of a Virtual Human Presenter

Tsukasa Noma
Kyushu Institute of Technology

Liwei Zhao and Norman I. Badler
University of Pennsylvania

In our society, people make presentations to inform, teach, motivate, and persuade others. Appropriate gestures effectively enhance the expressiveness and believability of such presentations. Likewise, a virtual human agent who adequately performs gestures during presentations can inform us effectively and serve as an interface agent mediating communication between user and computer.

Although Cassell et al.[1] animated multiple conversational agents with facial expressions, gestures, and spoken intonations in a rule-based fashion, their animation generation wasn't in real time. Moreover, the researchers didn't provide appropriate tools for using such gesticulating agents. On the other hand, several examples of presentations by lifelike agents have been reported in the literature. Thalmann and Kalra,[2] for example, produced some sequences of a virtual actor performing as a television presenter, but their work mostly appears to have been created manually. André et al.[3] developed a presentation agent system for the World Wide Web, but their work focused mainly on automated planning of presentation scripts on an outline level. Additionally, they drew the agent in a 2D cartoon style. The problems of generating gestures from presentation scenarios and of developing tools for gesticulated presentation by virtual humans thus remain relatively unexplored, particularly for 3D human body models.

In this article, we present a virtual human presenter system[4,5] that we developed on the Jack animated-agent system, which was developed at the Center for Human Modeling and Simulation at the University of Pennsylvania.[6] Jack provides a 3D graphical environment for controlling articulated figures, including detailed human models.

Requirements

We designed the presenter system to satisfy the following requirements:

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

- *Real-time motion generation synchronized with speech.* For example, if the virtual presenter acts as a personalized weatherman, he should report a weather forecast to users as soon as possible.[7] If he acts as an interface agent in an interactive system, he should react immediately depending on a user's input. The virtual presenter's motion generation should thus be in real time, preferably synchronized with speech.

- *Proper inputs for representing presentation scenarios.* The form of inputs to the virtual presenter should enable us to represent presentation scenarios without a detailed movement description. The input form should offer controllability of the presentation agent and, at the same time, take advantage of motor action, such as locomotion.

- *Widespread system applicability.* Despite rapidly growing expectations for animated lifelike interface agents like our presenter, the agents aren't yet widely used for practical or business applications. Limited use results mainly because such agents are not sufficiently supported in current animation/interface systems. For example, programmers lack proper tools for developing applications with such animated lifelike agents.

We created a virtual human presenter who accepts speech texts with embedded commands as inputs. The presenter acts in real-time 3D animation synchronized with speech.

To satisfy these requirements, we designed our presenter system as follows: The system accepts system inputs as text with embedded commands, most of which relate to the presenter's gestural motions. The system then makes the presenter speak the text in synchronization with his actions. He can perform various gestures with presentation skills based on published convention for gestures, presentations, and public speaking. In terms of applicability, our presentation system also serves as a programming toolkit for interface agents and a presentation system for the Web.

As currently implemented, the presenter system supports two presentation styles: presentation with 2D visual aids and presentation with 3D environments. With 2D visual aids, a virtual human agent acts as a presenter with a prop that looks like a blackboard or flip

1 A sample speech text.

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 spontaneously.

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.

2 A sample input to the virtual presenter.

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

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

chart. We call it a virtual board: It can display arbitrary images, texts, charts, and maps. The board size is arbitrary, but the typical size we use is 1.5 to 2 meters square. This size typifies that of real visual aids used in meetings and of weather maps in TV weather reports.

In a presentation with 3D environments, an agent can walk around and refer to objects in the room by gesturing and pointing. Typical objects might be tables, chairs, doors, and pictures on the wall.

Input texts

Suppose that a virtual human agent gives a presentation with the speech shown in Figure 1. Commands, embedded in the text, animate the presentation. Figure 2 shows a sample of command-embedded text, which would be a system input. A command is preceded by a backslash, and depending on its type, is additionally followed by arguments enclosed in braces.

Our current presenter implementation features two types of commands: board and various gesture commands. The board command `\board{}` applies to visual-aid presentations and specifies a virtual board with two arguments. It means that the system must change the board's current texture to the image file specified by the first argument. The second argument is the name of a Virtual Board Mapping (VBM) file, which maps from position names to normalized x,y-coordinates on an image for the virtual board.

The other commands in Figure 2 are gesture commands. `\point_idxf{}`, `\point_back{}`, and `\point_move{}` represent pointing gestures and make the agent point at the positions specified as arguments. (The gesture command `\point_idxf{}` tells the presenter to point his or her index finger.) If several argu-

ments are given, the presenter points to each in sequence. `\gest_givetake`, `\gest_reject`, and `\gest_warn` specify simple arm-hand gestures and thus take no arguments. We describe these gestures later.

The position of commands in the input texts represent the synchronization of body motion and speech. Basically, a command specifies a motion to coincide with the utterance of a word following the command in the inputs.

Two questions help us evaluate the adequacy of these system inputs. First, are the annotated speech texts appropriate for representing presentation scenarios? Presentation proceeds in parallel with spoken words in speech texts, and the major message is delivered verbally. This is obvious from terms like "visual aids" and "nonverbal communication." Speech texts can thus be a temporal axis for presentation scenarios. In addition, human presenters are advised to insert easily read indicators to coordinate the manuscript in slides, events, or times.[8] On the basis of these facts, our inputs are appropriate for the virtual presenter.

The second, more crucial question is: Are the speech texts at an appropriate level of abstraction for an "intelligent" presenter? Perhaps more abstract data—for example, weather/temperature tables for weather reports—would be more desirable for inputs to the presenter. But appropriate styles of speech texts vary depending on applications, as 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, nonverbal presentation skills are much more application-independent. We thus designed our input form so that our virtual human presenter can be a common tool for the application-independent nonverbal gestures, whereas application-dependent speech text generation should be done by preprocessing.

Furthermore, our input form is also appropriate for automatic gesture selection, in which the presenter automatically selects gestures and performs them with the utterance. Our solution detects the presence of the corresponding concepts in the raw (unmarked) text stream and automatically inserts gesture commands solely on the basis of words used.

Gesticulation for presentation

A virtual human presenter requires such skills as arm, hand, and head gestures. To prepare a set of gesture commands, we collected gestural vocabularies from two sources: psychological literature on gestures and popular books on presentation and public speaking.

In either source, we recognize that our gesticulation approach has its limitations. The interpretation might be both culturally oriented and individually biased; personality and social context may constrict or amplify the motions. But in general, we seek to set a baseline of gesticulatory behavior that we can then parameterize and modify by other means.

We implemented the gestures described here as commands. The presenter performs the corresponding gestures at the inserted commands' positions in the input texts.

Gesticulation: psychological perspective

Delsarte described a set of stereotypical arm, hand, and head gestures.[9] For arm gestures, different inclinations indicate different degrees of affirmation—from 0 (straight down) to 45 degrees indicates neutral, timid, and cold; from 45 to 90 degrees, expansive and warm; and from 90 to 180 degrees, enthusiastic. We implemented this series of arm gestures as a representative (metaphorical) mapping from affirmation concepts to gestures such that the presenter can correlate them with the degrees of affirmation in a speech.

For hand gestures, a small set of hand gestures correlate with grasping, indicating, pointing, and reaching. We implemented all these gestures. The presenter can perform them with either the left or right hand, with preference for the right hand under default circumstances.

For head gestures, Delsarte gave nine head positions, or attitudes, combined with different eye expressions or movements. We also implemented these gestures, which help express abstract concepts.

Gesticulation: practical perspective

We also collected gestural vocabularies from books on presentation and public speaking. For instance, we implemented four traditional speech gestures mentioned by Rozakis[10]: giving and taking, rejecting, warning, and pointing. In the giving-and-taking gesture, the hand is placed out with the palm turned upward.

Pointing gestures make quick visual references to visual aids and interesting objects being presented. From pointing examples in real-world presentations, we modeled and implemented four types of pointing gestures: `\point_idxf{}`, `\point_back{}`, `\point_down{}`, and `\point_move{}`. To avoid crossing the arm over the body and to keep the body posture open, our presenter uses the hand nearer to the pointed location.[11]

If the presenter cannot easily point to the next referenced location from the current body position, he is designed to move before his speech reaches the pointing command in the input text. Such anticipation gives more realism to the virtual human's presentation. Locomotion itself, which we discuss later, makes for more active staging, which is more interesting to watch.[12]

If the presenter needs to move, choosing the left or right hand for the next pointing gesture is also important. Particularly for presentation with visual aids, the hand choice determines the extent of the visual aid (virtual board) blocked to the audience's view.[8,11] The hand for pointing is thus determined by a heuristic that minimizes both visual aid occlusion and the distance from the current body position to the next one.

Other presentation skills

We modeled other presentation skills in our virtual presenter, based on guidelines taken from books on presentation and public speaking, as follows.

Posture. Posture is a highly visual presentation element. Presenters should stand up straight with both feet slightly apart and firmly planted on the floor.[11,12] Even for real human presenters, arms and hands are difficult to position when not in use. Brody advises us to let our
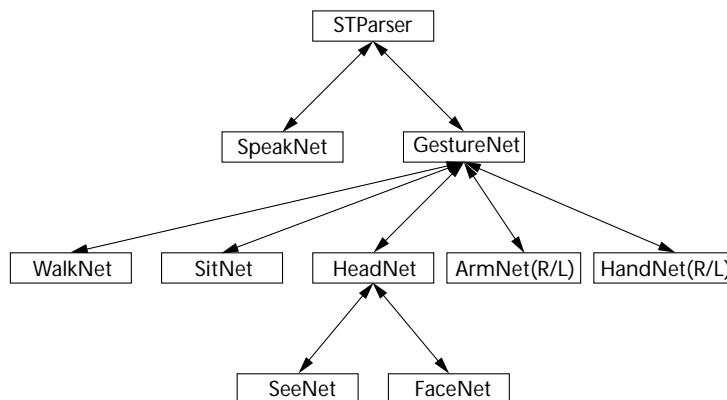

Figure 3. The PaT-Nets structure in the virtual presenter.

arms hang down naturally at our sides;[11] our virtual presenter follows these rules by default. Presenters' shoulders should be oriented to the audience,[11] which is interpreted as the viewpoint (virtual camera). For quickly pointing to visual aids and other objects, our system allows the presenter to angle slightly away from the audience. These neutral and slanted body orientations can be manually specified by the posture commands `\posture_neutral` and `\posture_slant`.

Eye contact. Many authors emphasize the importance of eye contact with the audience.[8,10-12] In real public presentation, presenters should vary the person at whom they look.[11] Our virtual presenter, however, is designed to talk to a fixed viewpoint (the TV camera), which means that he talks to each person in the audience directly, eye to eye. Although eye contact is important, the presenter must not constantly look at the audience. When pointing to a location on visual aids or to other objects, the presenter should glance at those to direct the audience's attention.[13] After pointing, he needs to look back immediately at the audience and then maintain eye contact.

Except for these "meaningful" gestures and motions, our virtual presenter will move as little as possible so that meaningless motions don't draw attention to themselves. A more detailed discussion on presentation skills appear in the literature.[4]

Presenter control

A set of Parallel Transition Networks (PaT-Nets)[1] controls the virtual presenter. These networks work from parsing input texts to animating individual joints in an integrated fashion.

Control via PaT-Nets

The PaT-Nets simultaneously execute finite-state automata. Every clock tick, the networks call for action and conditionally make state transitions. In our PaT-Nets, a single net can have multiple states at the same time to represent simple parallel execution of actions, and it can send/receive messages to/from other nets.

As currently implemented, 12 PaT-Nets run in parallel in our virtual presenter to animate a single agent. Two ArmNets and two HandNets are used to control the right and left arms and hands individually. Figure 3 shows the nets' structure. Each arrow represents message passing between the nets.

4 Representation of a step.
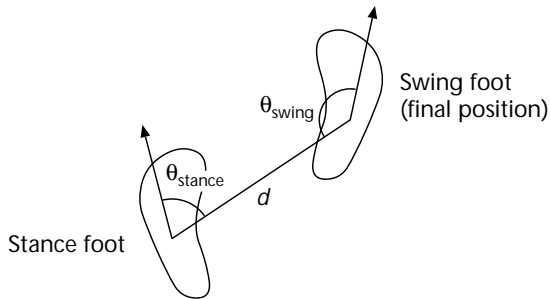
5 Presentation with visual aids.
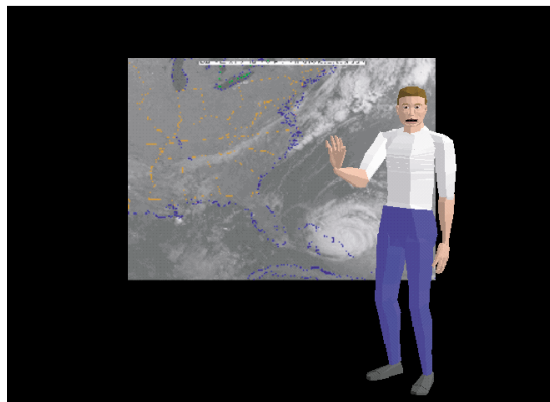
PaT-Nets as body parts

In most cases, animated agents should perform various types of motions, and the transitions between motions should be smooth. A simple approach for smoothness is to have the virtual presenter begin and end every motion in the same standard posture. While this approach offers smooth continuous transitions, beginning and ending each motion in the same still posture is unnatural.

An awkward, combinationally expensive approach is to define transitions between every pair of possible motions. New York University's Improv project[14] uses a technique called motion blending to automatically generate smooth transitions between isolated motions. This approach surely succeeds in avoiding the presenter's return to a required "neutral" pose, but it does not necessarily guarantee natural and rational transitions.

Rose et al.[15] use a combination of spacetime constraints and inverse kinematic constraints to generate dynamically plausible transitions between motion segments. The approach requires a very fast recursive dynamics formulation, which makes it impossible to use spacetime constraints on systems with many degrees of freedom, such as human figures.

To solve the motion-blending problem, we assigned groups of body parts to individual PaT-Nets: WalkNet, SitNet, SeeNet, FaceNet, ArmNet, HandNet, and SpeakNet. These lower level nets in the PaT-Net hierarchy (Figure 3) assign joint angles to their own body parts, depending on the messages sent from the higher level nets such as STParser and GestureNet. For example, an ArmNet manages clavicle, shoulder, elbow, and wrist joints of a single arm. To move the arm, the higher level nets have only to send message parameters to the ArmNet. The ArmNet moves the joints depending on messages such as "pointing at a particular position on a virtual board" (via inverse kinematics) or "taking a particular arm posture." Since the higher level motion generators (for example, STParser) aren't involved in directly assigning joint angles, the motion coordination of all the body parts is localized in each body part PaT-Net. Each net can then use simple interpolations to ensure continuity.

Parsing on a PaT-Net

As stated earlier, PaT-Nets are finite automata. Treating commands and lists of words in the input texts as tokens, we parse the inputs by the highest-level PaT-Net (called STParser in Figure 3) and make it control the other PaT-Nets.

The whole STParser has 66 nodes and makes transitions depending on the input tokens. It can parse the inputs sequentially in real time. The output animation throughput is independent of the input length.

Locomotion

Researchers have studied numerous approaches to animating human locomotion, most of which generate forward locomotion along straight or curved paths, such as that described by Bruderlin.[16] But presenters often need to step laterally or backward, and a more broadly capable locomotion engine is thus desirable for our virtual presenter.

Ko and Cremer[17] proposed the 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. But their engine wasn't applicable to our system for two reasons: It requires streams of body center positions and facing directions as inputs; and continuous locomotion in VRLoco isn't appropriate for locomotion in presentation, which is normally within a few steps.

We thus developed yet another locomotion engine, whose distinctive feature covers forward, lateral, and backward stepping and turning around in a unified fashion. Locomotion is generally considered a sequence of steps. In a step, the final arrangement of a swing foot can be represented by a triplet (θ_{stance}, θ_{swing}, d), as Figure 4 shows. Variations in (θ_{stance}, θ_{swing}, d) lead to variations in locomotion types. To animate a step for an

arbitrary (θ_{stance}, θ_{swing}, d), we first obtain the joint angles for the final frame of the step by interpolating samples prepared in advance. We then generate in-between frames by interpolating between the current/starting and final frames in the step.

To generate a sequence of steps, we choose the (θ_{stance}, θ_{swing}, d) among the possible steps in θ_{stance}, θ_{swing}, d-space so that the virtual presenter's body approaches the goal as much as possible. For the step to be definite, the following rules determine the next step:

1. If the next swing foot can be placed on the goal position (and preferably in the goal direction), then do so.
2. Otherwise, if the next swing foot is not in the direction of the goal body position, then change the direction into it as much as possible, and if still permitted, move the body center to the goal body position as much as possible.
3. Otherwise, move the body center to the goal body position as much as possible.

We modeled the above rules in a PaT-Net called WalkNet. It has eight nodes, each of which corresponds to a single step. Continuous transitions on the WalkNet thus generate continuous steps integrating various types of locomotion.

For presentation with 3D environments, the WalkNet has a list of objects for the presenter to avoid during locomotion. The net makes the presenter walk in a way that avoids these obstacles.

Implementation and results

We implemented our presenter system on an SGI Onyx/RealityEngine. The system animates an articulated human figure model in Jack. By inputting the speech texts with commands such as those in Figure 2, the system can generate presentation animations in real time (30 frames per second). Exceptions include time-consuming operations such as changing an image on the virtual board.

For voice output, we used an Entropic Research Laboratory's TrueTalk TTS (Text-To-Speech) system running on an SGI Indigo2. A PaT-Net called SpeakNet controls the TrueTalk via a TCP/IP socket. The SpeakNet also mimics the mouth motion by moving the jaw joint randomly during the presenter's speech. The synchronization between animation output and voice output was satisfactory, though it could be improved easily by applying known lip movement techniques.[18]

Our experiments show three uses of our virtual presenter: First, it works as yet another presentation system. While conventional presentation systems, such as Microsoft's PowerPoint, display only (mostly 2D) presentation materials as visual aids, our system can make explanations by itself with visual aids, as Figure 5 shows. Furthermore, in 3D environment presentations, the virtual human walks around the 3D virtual worlds, as Figure 6 shows.

Second, the system can serve as a programming toolkit for an animated interface agent. In addition to file
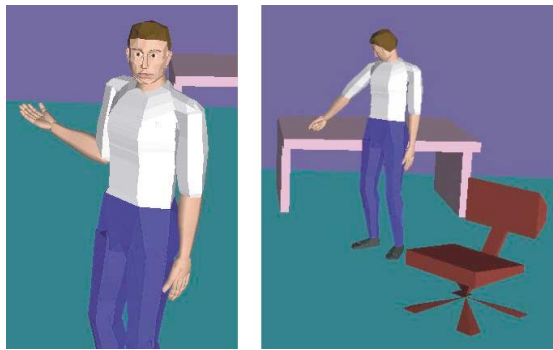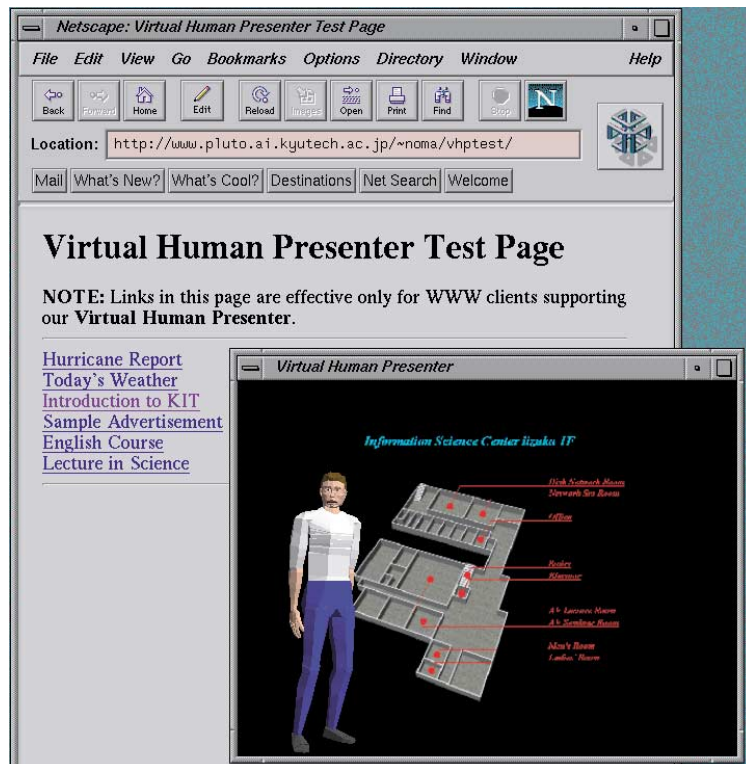

6 Presentation with 3D environments.

```
main()
{
    …
    choice = getchoicefrommenu();
      /*  returns a user's choice on the menu */
    while(choice  != quit)  {
        switch(choice)  {
        case choice1:
            processing for choice1;
            jvp_sendstrwait(speechtext_for_choice1, fp);
            /*  speechtext_for_choice1 is an input text
                for choice1  */
            /* fp is a FILE pointer for the TCP/IP socket  */
            break;
        case choice2:
            processing for choice2;
            jvp_sendstrwait(speechtext_for_choice2, fp);
            break;
        …
        }
        choice = getchoicefrommenu();
    }
    …
}
```

7 Template for a menu-based interactive system.

inputs, our virtual presenter can be controlled by inputs via a TCP/IP socket. Controlling the presenter is simple. For example, Figure 7 shows a template for a menu-based interactive system with the virtual human presenter. It works as a client program of our presenter system. In this program, the function `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. Even for inputs via socket, the system generated the agent's performance in real time. Programmers are thus freed from manipulating the virtual human body itself and can concentrate on scenario control with speech text generation and gesture specification. In fact, as part of this work we implemented a menu-based interactive weather reporter. The client controlled the virtual presenter interactively.

Third, the system can be a Web application. URLs are valid arguments for the board command `\board{}`. The system can thus obtain image and VBM files from Web servers. To use our system with the Web, administrators of Web servers must register a media type for the presenter on their servers, and users must register the same media type on their browsers. These settings on both sides enable our system to handle its files appropriately on the Web. Presentation by a virtual human

8 Presentation on the Web.

speech at the level of individual words or syllables. This capability is indispensable to support many features of human speech, such as the use of gestures, head nods, and eye gaze for emphasizing words.[1] To generate such lively gestures, our system should support the synchronization of motion and speech at the level of words or syllables in the near future.

Digitally coded video comparison

A typical approach for making presentations on the Web is transferring and replaying digitally coded video files such as MPEG, AVI, and QuickTime. Compared with using these compressed video files, our presenter system requires much less data transfer, since it needs only speech texts and image files for the virtual board.

At the same time, our approach has flexibility and extensibility. Systems using compressed videos replay animations just as their producers expected. In our system, the speech texts are transformed into presentation animation determined by local computers on the viewers' side. In this sense, our system is a first step to the "smart TV" that Negroponte predicted.[7]

Position mapping

In visual aid presentations on our system, the VBM file maps position names, specified as arguments of pointing commands, to the image coordinate. Users must thus prepare VBM files per image in advance.

A straightforward way to simplify VBM file preparation would be through an interactive tool that lets users specify positions and their names, and then output VBM files. In addition, for routinely supplied images on Web servers (such as weather maps), VBM files can often be reused for images of the same scale.

A more challenging solution would be automatically identifying interesting and/or important features on the image, possibly through XML tags. We've left this for future work.

agent then starts by clicking a link to our input speech file anywhere on the Web, as Figure 8 shows.

At present, 3D environments are implicitly expressed as figures in Jack[6] and thus inapplicable to the Web, as the Web currently permits only visual-aid presentations. In the near future, however, changing the environment descriptions to VRML would allow 3D-environment presentations on the Web. For sample Web movies of our virtual human presenter, see http://www.pluto.ai.kyutech.ac.jp/~noma/vpre-e.html.

Discussion

Several design issues, including areas for future work, deserve comment.

2D versus 3D animation

An alternative design of our presenter would use sequences of 2D video images such as video widgets and video actors.[19] In such 2D animation, however, individual images must be prepared per body posture and per viewpoint, thus more images are required as presentations and human motions become more complex. On the other hand, in 3D animation, images for any body posture and viewpoint can be rendered once a 3D body model is given. 3D animation thus offers superior flexibility, generalized control, and future compatibility with VRML. The advantages increase as applications and computer platforms improve.

Synchronization of nonverbal movements

Most current animated characters including ours are incapable of synchronizing nonverbal movements with

Conclusions

Viewing our system as a specialized animation system, command-embedded speech text is its animation language. It has no explicit timing information, which often appears in conventional animation languages.

Our virtual presenter system still has much room for improvement: To enhance the presenter's believability, it would be desirable to synchronize the lip movement with the voice and to improve prosody control. Compared to the state of the art, such features are not now perfectly realized. Generated motions should also be smoother and more natural.[20] In addition, automatic insertion of gesture-commands into the text offers an interesting issue for the gesture selection problem.

Our virtual human presenter has many potential applications. For example, the agent can make sales presentations depending on consumers' preferences. Such applications may have a great impact on merchandise marketing. For such purposes, the TCP/IP socket interface as well as the applicability to the Web will prove of considerable use. ∎

References

1. J. Cassell et al., "Animated Conversation: Rule-based Generation of Facial Expression, Gesture and Spoken Intonation for Multiple Conversational Agents," *Proc. Siggraph 94*, ACM Press, New York, 1994, pp. 413-420.

2. N. Magnenat-Thalmann and P. Kalra, "The Simulation of a Virtual TV Presenter," *Proc. Pacific Graphics 95*, World Scientific, Singapore, 1995, pp. 9-21.

3. E. André, T. Rist, and J. Müller, "WebPersona: A Lifelike Presentation Agent for the World-Wide Web," *Knowledge-based Systems*, Vol. 11, No. 1, Sept. 1998, pp. 25-36.

4. T. Noma and N.I. Badler, "A Virtual Human Presenter," *Proc. IJCAI-97 Workshop on Animated Interface Agents*, 1997, pp. 45-51; http://www.pluto.ai.kyutech.ac.jp /~noma/doc/ijcai97ws.ps.

5. L. Zhao and N.I. Badler, "Gesticulation Behaviors for Virtual Humans," *Proc. Pacific Graphics 98*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1998, pp. 161–168.

6. N.I. Badler, C.B. Phillips, and B.L. Webber, *Simulating Humans: Computer Graphics Animation and Control*, Oxford University Press, New York, 1993.

7. N. Negroponte, *Being Digital*, Knopf, New York, 1995.

8. T. Leech, *How to Prepare, Stage, and Deliver Winning Presentations*, 2nd ed., Amacom, New York, 1993.

9. T. Shawn, *Every Little Movement—A Book about Delsarte*, M. Witmark & Sons, Chicago, 1954.

10. L.E. Rozakis, *The Complete Idiot's Guide to Speaking in Public with Confidence*, Alpha Books, New York, 1995.

11. M. Brody and S. Kent, *Power Presentations*, John Wiley & Sons, New York, 1993.

12. M. Kushner, *Successful Presentations for Dummies*, IDG Books Worldwide, Foster City, Calif., 1996.

13. W. Hendricks et al., *Secrets of Power Presentations*, Career Press, Franklin Lakes, N.J., 1996.

14. K. Perlin and A. Goldberg, "Improv: A System for Scripting Interactive Actors in Virtual Worlds," *Proc. Siggraph 96*, ACM Press, New York, 1996, pp. 205–216.

15. C. Rose et al., "Efficient Generation of Motion Transitions Using Spacetime Constraints," *Proc. Siggraph 96*, ACM Press, New York, 1996, pp. 147–154.

16. A. Bruderlin, C.G. Teo, and T. Calvert, "Procedural Movement for Articulated Figure Animation," *Computers and Graphics*, Vol. 18, No. 4, July/Aug. 1994, pp. 453–461.

17. H. Ko and J. Cremer, "VRLoco: Real-Time Human Locomotion from Positional Input Streams," *Presence*, Vol. 5, No. 4, Fall 1996, pp. 367–380.

18. M.M. Cohen and D.W. Massaro, "Modeling Coarticulation in Synthetic Visual Speech," in *Models and Techniques in Computer Animation*, N.M. Thalmann and D. Thalmann, eds., Springer-Verlag, Tokyo, 1993, pp. 139–156.

19. S. Gibbs et al., "Video Widgets and Video Actors," *Proc. of 1993 ACM Symp. on User Interface Software and Technology (UIST 93)*, ACM Press, New York, 1993, pp. 179–185.

20. N.I. Badler, D.M. Chi, and S. Chopra, "Virtual Human Animation Based on Movement Observation and Cognitive Behavior Models," *Proc. Computer Animation 99*, IEEE Computer Society Press, Los Alamitos, Calif., 1999, pp. 128–137.

Tsukasa Noma *is an associate professor in the Department of Artificial Intelligence at Kyushu Institute of Technology. His research interests include computer graphics, animation, and vision. He received the BSc degree in mathematics from Waseda University, and the MSc and the DSc in information science from the University of Tokyo. He is on the editorial board of* The Visual Computer *and is a member of the ACM, the IEEE, the IPSJ, the CGS, and the JSAI.*

Liwei Zhao *is a graduate student of computer and information science at the University of Pennsylvania. His research interests include real-time 3D graphics, human figure animation, and game AI. Zhao received a BS and an MS in computer science from Beijing University, and an MS in computer science from Ohio State University. He is a member of the ACM.*

Norman I. Badler *is a professor of computer and information science at the University of Pennsylvania, and he directs the Center for Human Modeling and Simulation. His research interests include real-time 3D graphics, animation systems, intuitive user interfaces, and connections between language and action. Badler received a BA in creative studies mathematics from the University of California, Santa Barbara, and an MS in mathematics and a PhD in computer science from the University of Toronto. He is co-editor of the* Graphical Models *journal.*

Readers may contact Noma at the Kyushu Institute of Technology, Dept. of Artificial Intelligence, 680-4 Kawazu, Iizuka Fukuoka 820-8502, Japan, e-mail noma@ai.kyutech.ac.jp.