Piavca: A Framework for Heterogeneous Interactions with Virtual Characters

Marco Gillies[1] Xueni Pan[2]
Mel Slater[3]
[1]Goldsmiths, University of London Department of Computing,
London, UK m.gillies@gold.ac.uk
[2]University College London, Department of Computer Science,
London UK
[3]ICREA-Universitat de Barcelona, Barcelona, Spain.

October 1, 2009

Abstract

This paper presents a virtual character animation system for real time multimodal interaction in an immersive virtual reality setting. Human to human interaction is highly multimodal, involving features such as verbal language, tone of voice, facial expression, gestures and gaze. This multimodality means that, in order to simulate social interaction, our characters must be able to handle many different types of interaction, and many different types of animation, simultaneously. Our system is based on a model of animation that represents different types of animations as instantiations of an abstract function representation. This makes it easy to combine different types of animation. It also encourages the creation of behavior out of basic building blocks. making it easy to create and configure new beahviors for novel situations. The model has been implemented in Piavca, an open source character animation system.

1 Introduction

Animated virtual humans are a vital part of many virtual environments today. Of particular interest are virtual human that we can interact with in some approximation of social interaction. However, creating characters that we can interact with believably is an extremely complex problem. Part of this difficulty is that human interaction is highly multimodal. The most obvious modality is speech, but even this can be divided into the verbal content and non-verbal aspects of speech, such as tone of voice, whose function can be very subtle and complex. When we take into account bodily modalities we also have to deal with facial expression, gaze, gesture (both accompanying speech and giving feedback while listening), posture, body movements and touch. While non-immersive environments can ignore some of these modalities of interactions, the full power of an immersive virtual reality interaction with a character can only be achieved by modelling all (of most) of the modalities.

This multimodality implies that there will be a great variety of ways of interacting with a virtual character. A human participant can use these different ways of interacting with a character:

- Verbal interaction, characters that can engage in conversation using a dialogue engine.
- Non-verbal aspects of speech, picking up features of the participants voice.
- What Slater[29] calls “Body Centered Interaction”. The participants movements are tracked, and their normal movements used to interact with a character. For example, the character may track the participants movements with their gaze and maintain a normal conversational distance to them.
- Control, the character might be an avatar which is being controlled by a participant, either through one of the above modalities or a more conventional user interface such as a mouse and keyboard or a joystick.
- Watching, not all of a characters behavior will be interactive, some will simply play back and the participant can observe it.

The true complexity of the behavior of an interactive character is that most of these types of interactions are likely to be happening simultaneously. For example, a participant might be engaging in conversation with a character that is controlled by a chat-bot. The character nods in rhythm with variations in the participants speech amplitude and its gaze follows the participants position, while it walks around following the joystick movements made by another participant. Finally, its posture shifts occasionally, following a non-interactive algorithm. (Admittedly, this example is contrived, if the character is an avatar under the control of a participant, then its speech will almost always be controlled by that participant. However, even given this constraint, many modalities of interaction are likely to appear simultaneously). This paper presents a method of creating characters that combine these very diverse forms of interaction.

The diverse styles of interaction also imply diverse methods of generating behavior. This paper is mostly restricted to animation, but there are still many different styles. Some animation can be played back from pre-existing data, whether it is from motion capture or hand animation. Some types of animation, such as gaze or lip synchronization are best generated on the fly, algorithmically, a process called procedural animation. Finally, most interactive animation is generated by transforming and combing clips of pre-existing animation data, to produce new animations.

Another source of diversity is the different time scales of interactivity. A character's gaze has to respond instantly to changes in the humans position, while other body centered interactions have more variable time constraints. Speech interaction tends to be turn based, with long periods of non-interaction, interrupted by a change of speaker. Other modalities have no time constraints for interaction, or are not interactive at all. This flexibility in level of interactivity is also important because of the great difference in quality that exists between behavior that is generated on the fly and pre-recorded behavior. This is particularly true with speech where it can be difficult to produce coherent speech for long periods with all but the best existing dialogue systems, and where synthesized speech falls far short of recorded speech quality. However, it is also true of animation, motion captured or hand animated animation is generally better than what can be produced in real time, even by transforming motion capture data. However, completely pre-recorded behavior reduces interactivity. The sense that we are truly interacting with a character is likely to be a strong contributor to presence and thus is vital. However, the illusion of interactivity can be maintained even if the behavior is not entirely interactive, as long as some elements are. For this reason, we believe it is important to balance the quality of pre-recorded elements. Behavior that consists of some pre-recorded clips, of audio and animation, should be used to ensure quality of the output. However, the sense of interactivity can be maintained to a degree if other aspects of behavior such as gaze and feedback, remain highly interactive. In turn this sense of interactivity can maintain a high level of presence.

This flexibility also implies that the characters will need to be used in different ways and that different styles of interaction will need to be combined differently in different situations. Different types of behavior need to be interacted with in different ways. Sometimes facial expressions need to be under the control of a human, at other times they must respond to the participants speech and at other times they can be random or scripted. The same factors apply to all the behavior types. Therefore it needs to be simple to create new interactive behaviors and combine them with others in a variety of different ways.

This paper describes how to create a character capable of very heterogeneous forms of interaction. We first describe related work in this area. We then describe a functional model of animation that is designed for handling very diverse styles of animation, and its implementation with the Piavca open source character animation system. Finally we describe how it is used to create our heterogeneous character.

2 Related Work

This work builds on a long tradition of research on expressive virtual characters[34]. This work has aimed at building animated characters that can autonomously produce the type of expressive non-verbal communication that humans naturally use in day to day interaction. This generally entails both an animation system and a higher level model for determining what behavior to produce in response to stimuli. Numerous general purpose systems have been produced notably the “JACK” system by Badler *et al.*[4]; the “GRETA” system by Pelachaud and colleagues[27]; work of Cassell *et al.*[7], and Guye-Vuilléme *et al.*[15]. Much of this work has concerned expression of emotions (for example the work of Gratch and Marsella[14]), but our work is closer to research that models use of non-verbal communication in face-to-face conversation, for example the work of Vilhjálmsson[32]. Each individual modality of expression is highly complex and many researchers have worked to create models of single modalities. Gaze is often linked to turn taking in conversation, for example the work of Lee, Badler and Badler[23], further developed by Vinayagamoorthy *et al*[33]. Facial expression has been extensively studied in the context of emotional expression, for example the work of Kshirsagar and Magnenat-Thalmann[21] or The Duy Bui [5]. Gestures are very closely related to speech[8] and have been modeled in a number of ways, from the highly data driven methods of Stone *et al.*[30] to the totally procedural methods of Kopp *et al*[19, 18]. Finally, posture is also an important modality of expression that has been used as a means of producing believable idling behaviour[9] or of expressing interpersonal relationships [12].

One of the first systems that allows people to interact multimodally with a character using voice and gestures tracking was Thórisson's Gandalf system[31]. This work was later developed by Cassell's group into an interaction system with a full bodied character capable of complex non-verbal communication[6]. The Max system focus primarily on voice and gesture interaction[17]. Work by Maatman *et al.*[24] focused on the listening behaviour of a character, and like our work uses head tracking and voice input. These systems have demonstrated the power of multimodal face-to-face interaction with virtual characters. This paper shows how it is possible to rapidly create and customise such systems from basic building blocks.

Our work also uses research in the area of data driven animation and motion editing. In particular the Motion Warping formulation of motion editing [35]. Our functional model is an excellent way of combining motion editing techniques, and we have implemented several within our framework. Examples include, interpolation based animation such as the work of Rose *et al.*[28], Principal Component Analysis based animation [1] and the Motion Graph data structure[3, 20, 22].

The work we present is a functional abstraction of character animation and behavior. In this sense it is similar to functional abstraction frameworks used in other domains. In particular there are a number of interesting abstraction frameworks used in virtual reality and graphics for example Figueroa *et al.*'s InTML system[11] or Elliott *et al.*'s TBAG system[10].

3 A Functional Model of Animation

Handling and combining many diverse methods of animation on a single character requires a single representation for all of them. At its most abstract, an animation can be viewed as a function of time to the state of a character:

$$\mathbf{x} = f(t) \tag{1}$$

where $\mathbf{x}$ is some representation of a characters state. The most common representation in body animation would be a skeletal one, in which the root of the character has a vector position p_0 and quaternion orientation q_0 and all of the joints have quaternion orientations (q_i): $\{\mathbf{p}_0, \mathbf{q}_0, \mathbf{q}_1, \dots \mathbf{q}_n\}$. However, we do not restrict animations to this representation, for example joint orientations can also be represented using Grassia's exponential map representation[13] if it is more convenient for certain calculations. For facial animation the state can be represented as a set of weight values for morph targets or as positions or rotations of facial bones. The state can also have more abstract representations, for example, the result of one animation function can be used as the output of another, as we shall see below.

Most animations, will in fact take other parameters making this the general form of an animation function:

$$\mathbf{x} = f(\theta, t) \tag{2}$$

We will now show how different types of animation can easily be represented in this framework.

Keyframe or Motion Captured Animation Both hand animated data and motion captured animation are represented as a list of evenly or unevenly spaced keyframes. This can be represented as the following function:

$$f(\mathbf{k}, t) = \mathit{interpolate}(k_{\tau(t)}, k_{\tau(t)+1}, t)$$

Where $\mathbf{k}$ is the keyframe data and $\tau(t)$ is the keyframe time prior to t. *interpolate* can be any suitable interpolation function, we use cubic spline interpolation.

Procedural Animation Procedural animation is the most general instantiation of the functional model, and can be represented as any function of the form (2). The parameters θ will depend heavily on the type of motion, for example, a gaze motion will have parameters the include the target of gaze and the length of gaze.

Motion Transforms An animation function can be a transformation of another motion function g:

$$\mathbf{x} = f(g, \theta, t)$$

The most general form of transformation is a general motion warp [35], which consists of a timewarp, α, tranforming t and a space warp, β transforming the output of g:

$$f(g, \theta_\alpha, \theta_\beta, t) = \beta(\theta_\beta, g(\alpha(\theta_\alpha, t))$$

Simple examples include, spatial scaling[1]: $f(g,s,t) = sg(t)$; temporal scaling: $f(g,s,t) = g(st)$, and looping $f(g,t) = g(t \bmod |g|)$, where $|g|$ is the length of g (in practice a more complex looping function is used to ensure

[1] for quaternion animations multiplication is replaced by scaling the rotation angle by s

smooth transitions). A paricularly useful transform is a mask, which is used to select certain joint or morph targets to which to apply an animation, based on a mask, **m**:

$$f(g,\mathbf{m},t)_i = \begin{cases} g(t)_i & \text{if } \mathbf{m}_i = 1, \\ 0 & \text{otherwise} \end{cases}$$

As mentioned above the parameters of a motion transform can themselves be other motion functions. For example, a general motion warp can be created by using a motion function, h as the warping parameter: $f(g,h,t) = g(h(t))$.

Combining Animations Animations can also be combined together by functions of two other animations, which have the general form:

$$f(g_1, g2, \theta_\beta, \theta_{\alpha 1}, \theta_{\alpha 2}, t) = \beta(\theta_\beta, g_1(\alpha_1(\theta_{\alpha 1}, t)), g_1(\alpha_2(\theta_{\alpha 2}, t)))$$

Examples include addition[2]: $f(g_1, g2, t) = g_1(t) + g2(t)$; blending between animations: $f(g_1, g2, \lambda, t) = \lambda g_1(t) + (1-\lambda) g2(t)$ and sequencing motions:

$$f(g_1, g2, t) = \begin{cases} g_1(t) & \text{if } t < |g_1|, \\ g_2(t - |g_1|) & \text{otherwise} \end{cases}$$

As with looped motions, in practice we would used a smoothed version of this function. There are also functions for combining multiple animations, for example blending between several animations; finite state machines that choose different animations based on their state; animations based on a Principal Component Analysis of other animations [1], and motion graphs described below.

The power of the functional model of animation comes from the fact that all types of animation have the same form. The animation functions can be curried, pre-applying all parameters apart from t to get a new function of the form (1). Once in this form the user does not need to know anything about the type of animation. In particular the functions for transforming and combining animations have the same form as their inputs, making it possible to compose them. As we shall see arbitrary composition of functions for transforming and combining animations can be a powerful tool for creating complex transformations. This leads to a methodolgy of decomposing transformations into their most basic elements, which can then be reused by composing them in a number of different ways. This in turn makes it easy to author new behaviours, and variants of existing behaviours.

[2] for quaternions, quaternion multiplication is used instead of addition

3.1 Implementation

Before discussing how the functional model is used, a quick note on how it is implemented. The model has been implemented as part of Piavca (the Platform Independent Architecture for Virtual Characters and Avatars), an open source software framework for creating virtual characters (available at http://piavca.sourceforge.net/). Piavcs is a generic API built in C++ that can be combined with different graphics engine. The current implementation uses Cal3d (http://home.gna.org/cal3d/) as a low level animation system providing functionality such as smooth skinning and morph targets while Piavca overrides the motion blending features. The renderer is based on Cal3d's OpenGL renderer. This system can be used with numerous graphics and virtual reality systems. Currently it is implemented on Dive (http://www.sics.se/dive/), XVR (http://www.vrmedia.it/Xvr.htm) and we are working on an implementation for OpenSG (http://www.opensg.org/).

Our implementation (in the C++ and Python languages) is object oriented. All of the functions are in fact function objects, and so can contain state. They all inherit from a single base class, Motion, which acts as an abstract representation of an animation function. Currying the functions is achieved by providing all parameters except t at initialization time, and making the function objects callable with t as a parameter.

Users can configure and combine motion functions using a Python based scripting interface to Piavca, for maximum flexibility. Alternatively there is an XML based behavior definition language that allows users to create character behavior models without requiring programming skills. The behavior definition language exactly mirrors the functional model with tags for each possible function.

4 A Heterogeneous character system

Figure 1 shows the type of heterogeneous interaction that is possible with our characters. The human participant's behavior is input with typical sensors for an immersive VR system, a microphone and head tracker. However, these two inputs are used in a variety of different ways by different behaviors. The head tracker is used to obtain the position of the participant in order to maintain an appropriate conversational distance (maintaining distance in conversation is called proxemics in the non-verbal communication literature). It is also used to detect when the participant shifts posture. The character's posture shifts are then synchronized with those of the participant, which is know to build rapport[16]. The position of the participant is

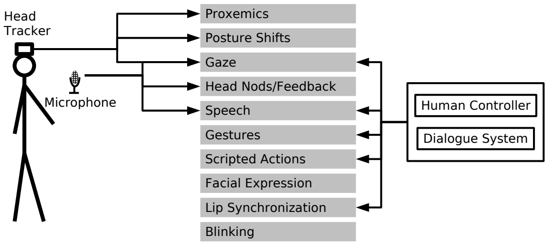

Figure 1: Heterogeneous Interaction.

also used by a gaze behavior to ensure the character is looking in the right place. The audio from the microphone is used to detect when the participant is talking. This is used by the gaze behavior so the character can look at the participant more when listening to him or her. The character also gives head nods and other feedback signals when the participant is speaking. All these behaviors happen automatically in response to the sensor input, using real time algorithms.

The microphone audio is also used for speech interaction. Speech interaction is either controlled by a human controller, if the character is an avatar, or by a dialogue engine. When the character speaks a number of other behaviors are triggered. The character's lip movements will be synchronized to the speech and the character will gesture. The gaze behavior is also altered to take account of the fact that the character is speaking[2]. As well as triggering speech the character's controller can also trigger certain scripted actions and gestures. Apart from lip synchronization and gaze, the character's facial expression is independent of the participants behavior, consisting of occasional smiling and blinking. The character therefore has a wide range of styles of interaction, all happening simultaneously. These contain many different animation processes, both facial and bodily, that must be combined to create a single coherent animation.

Figure 2 shows how this type of character can be implemented using our functional model. The example we give is of a human controlled character that is used in Wizard of Oz style experiments. A human operator controls certain aspects of the behavior, while others are automatic. The character's speech is controlled by a human being (the controller) selecting speech sequences from a library of possible utterances, while the character interacts with another person (the participant). This system has 3 inputs: the position of the participant, an audio signal of the voice of the participant and input

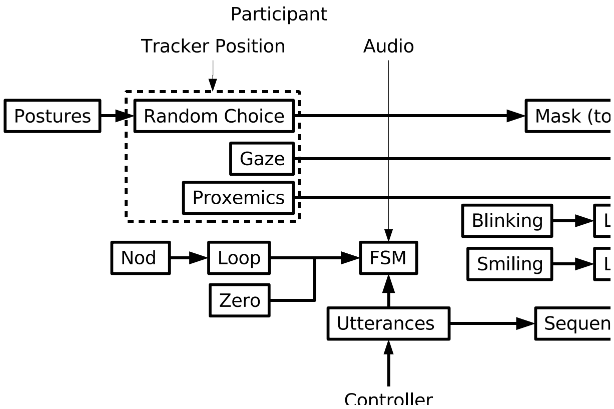

Figure 2: Implementing a Heterogeneous Interaction.

from the person controlling the character, specifying speech utterances. The position input is a 3-vector whose value is obtained every frame from a head tracker on the participant. The voice signal is obtained from an ordinary microphone. For this application we simply threshold the audio value to detect whether the participant is speaking. The controller has a user interface with a number of buttons used to trigger speech utterances.

Some behaviors of the character are not influenced by any of the inputs. For example, the character has a simple blinking behavior. This is a loop containing a blinking animation sequenced with a zero animation. A zero animation is simply an animation function that returns zero for all joint or facial expression values and in this case it is used to model the inter-blinking period. The length of the zero motion is varied every time around the loop to ensure the timings are not too repetitive. A facial animation loop, in which the character smiles occasionally, is implemented similarly. The head tracker input is used in a number of ways. The first is for posture shifts. During conversation people tend to synchronize their movements, particularly movement such as posture shifts[16]. This synchronization is a strong sign of rapport between individuals. In order to simulate this we detect posture shifts by finding large changes in position. We then trigger a posture shifts. The characters posture is modeled as a finite state machine animation in which each state is a different possible posture. On a posture shift a new state is chosen at random. A finite state machine animation performs a smooth sequence between the animations associated with each state, ensuring smooth posture shifts. The head tracker is also used by the proxemics behavior. Proxmemics is the use of space in social interaction. For our characters this means maintaining a comfortable distance from and orientation to the participant. The relative distance and angle of the participant to the character are calculated from the tracker position. If they are too large or small the character turns to face the participant, or takes a step forward or backward. Again this behavior is modeled as a finite state machine, with a default state being the zero motion and a state for each motion direction. The final use for the position input is to control the gaze behavior. The position gives a target to look at. The audio input is used to detect when the participant is speaking and give feedback behavior. In this implementation the feedback consists of occasional nodding to give encouragement. This is implemented as a loop in the same way as blinking.

The other major input is from the controller, who can issue commands to control the character's speech. The behavior consists of a number of multi-modal utterances that can be triggered using a graphical user interface. Multi-modal utterances are short scripted behaviors that combined speech (in this case audio files) with animation elements. For example, the audio is accompanied by facial animation for lip synchronization and also appropriate gestures. The scripts give the creators of the character very tight control of the character's behavior and potentially high quality behavior can be created. This comes at the possible cost of some interactivity, however, we believe that our methodology of combining more scripted elements with real time interaction can combine be benefits of both.

Figure 3 shows some still frames from an interaction with our virtual character, the accompanying video shows the actual interaction. The type of character set up we have described is only one way of interacting. For example, the character could be total autonomous with utterances triggered from an AI "chat-bot" system, or the character's speech could be directly taken from the controllers own voice. In the second case, many elements such as gestures and lip synchronization would have to be generated auto-

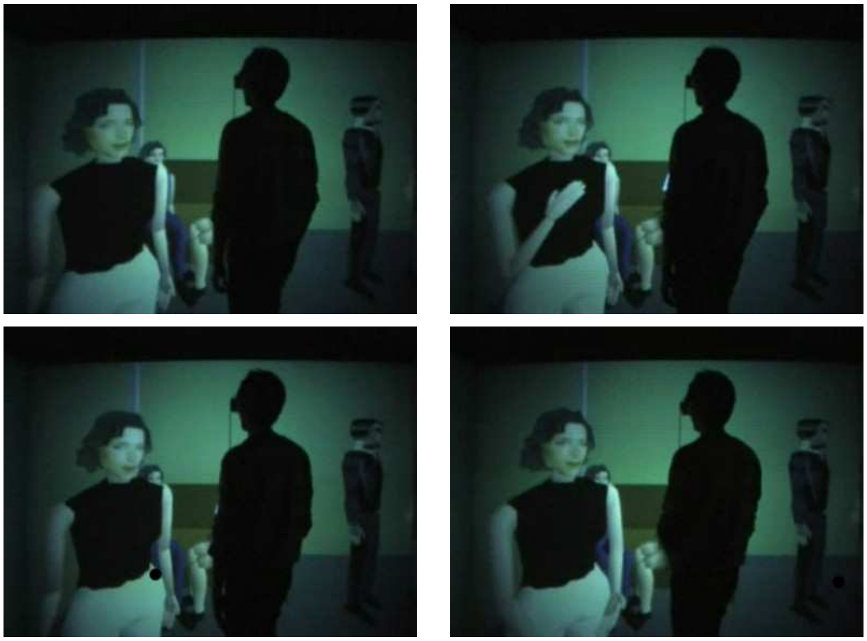

Figure 3: A real and virtual human interacting in an immersive virtual environment

matically to suit the speech. Our frameworks makes it easy to build new styles of interaction from existing components.

5 Conclusion

This paper has presented a software framework for creating interactive virtual characters. The many different types of behavior involved with human social interaction imply a range of different styles of animation and interaction with a character. Our framework allows us to unify and combine these diverse methods using a single abstract function representation. It makes it easy to create new character systems by combining different behavior modules in different ways. The framework has been release as part of the open source project Piavca (http://piavca.sourceforge.net/), we encourage readers to try out the functionality.

This framework has been used for virtual reality experiments at University College London. These have demonstrated that people respond to the characters in some way as if they are human. For example Pan and Slater

[26] conducted an experiment in which socially phobic male participants interacted with a virtual woman that engaged them in conversation that became increasingly intimate. The experimenters measured skin conductance level which demonstrated that the intimate conversation resulted in greater arousal. Interestingly, proxemic behavior played an important role in this. At one point the character's proxemic distance was decreased resulting in her moving closer to the participant. This produced the highest skin conductance levels in the experiment. These quantitative results were supported by the participants subjective reports, with many reporting strong emotions such as anxiety or even guilt at "cheating" on their partner with a virtual woman. An analysis of the body movements of the participants during the scenario [25] showed that they used more social non-verbal cues such as nodding or cocking their heads during the conversation than before it.

Future work on the project will involve increasing the range of functionality and of possible applications. In a modular system such as ours it is easy to add functionality by either added new animation functions or combining existing ones in new ways. As the system is applied to different situations and styles of interaction new requirements will naturally emerge and therefore drive the development of new functionality. We are currently applying the system to more graphically realistic character and making greater use of motion capture, raising the level of realism that is possible. This greater realism will itself bring new requirements to our animation framework. The most important change we are currently planning is the addition of a graphical user interface for combining behavior functions. This will supplement the existing scripting interface and definition language, and provided a more accessible method of creating characters.

acknowledgements

We would like to thank the funders of this work: BT plc, the European Union FET project PRESENCIA (contract number 27731) and the Empathic Avatars project funded by the UK Engineering and Physical Sciences Research Council. We also would like to thank the members of the University College London Department of Computer Science Virtual Environments and Graphics Group.

References

[1] M. Alexa and W. Müller. Representing animations by principal components. *Computer Graphics Forum*, 19(3):411–418, Aug. 2000.

[2] M. Argyle and M. Cook. *Gaze and Mutual Gaze*. Cambridge University Press, 1976.

[3] O. Arikan and D. A. Forsyth. Interactive motion generation from examples. *ACM Transactions on Graphics*, 21(3):483–490, July 2002.

[4] N. Badler, C. Philips, and B. Webber, editors. *Simulating Humans: Computer Graphics, Animation and Control*. Oxford University Press, 1993.

[5] T. Bui, D. Heylen, and A. Nijholt. Combination of facial movements on a 3d talking head. In *Computer Graphics International 2004*, pages 284–290, June 2004.

[6] J. Cassell, T. Bickmore, L. Campbell, K. Chang, H. Vilhjálmsson, and H. Yan. Embodiment in conversational interfaces: Rea. In *ACM SIGCHI*, pages 520–527. ACM Press, 1999.

[7] J. Cassell, T. Bickmore, L. Campbell, K. Chang, H. Vilhjálmsson, and H. Yan. Requirements for an architecture for embodied conversational characters. In *Computer Animation and Simulation '99*, Sept. 1999.

[8] J. Cassell and M. Stone. Living hand to mouth: Psychological theories about speech and gesture in interactive dialogue systems. In *AAI 1999 Fall Symposium on Psychological Models of Communication in Collaborative Systems*, pages 34–42, 1999.

[9] A. Egges, T. Molet, and N. Magnenat-Thalmann. Personalised real-time idle motion synthesis. In *12th Pacific Conference on Computer Graphics and Applications*, pages 121–130, Oct. 2004.

[10] C. Elliott, G. Schechter, R. Yeung, and S. Abi-Ezzi. Tbag: A high level framework for interactive, animated 3d graphics applications. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, pages 421–434, July 1994.

[11] P. Figueroa, M. Green, and H. J. Hoover. Intml: A description language for vr applications. In *Proceedings of Web3D '02*, 2002.

[12] M. Gillies and D. Ballin. A model of interpersonal attitude and posture generation. In T. Rist, R. Aylett, D. Ballin, and J. Rickel, editors, *Fourth Workshop on Intelligent Virtual Agents*, Kloster Irsee, Germany, September 2003.

[13] F. S. Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48, 1998.

[14] J. Gratch and S. Marsella. Tears and fears: modeling emotions and emotional behaviors in synthetic agents. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 278–285, New York, NY, USA, 2001. ACM Press.

[15] A. Guye-Vuilléme, T.K.Capin, I.S.Pandzic, N. Magnenat-Thalmann, and D.Thalmann. Non-verbal communication interface for collaborative virtual environments. *The Virtual Reality Journal*, 4:49–59, 1999.

[16] A. Kendon. Movement coordination in social interaction. *Acta Psychologica*, 32:1–25, 1970.

[17] S. Kopp, B. Jung, N. Lessmann, and I. Wachsmuth. Max–a multimodal assistant in virtual reality construction. *KI-Knstliche Intelligenz*, 3(4):11–17, 2003.

[18] S. Kopp, P. Tepper, and J. Cassell. Towards integrated microplanning of language and iconic gesture for multimodal output. In *International Conference on Multimodal Interfaces (ICMI'04)*, pages 97–104. ACM Press, 2005.

[19] S. Kopp and I. Wachsmuth. Synthesizing multimodal utterances for conversational agents. *The Journal Computer Animation and Virtual Worlds*, 15(1):39–52, 2004.

[20] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482, July 2002.

[21] S. Kshirsagar and N. Magnenat-Thalmann. A multilayer personality model. In *2nd International Symposium on Smart Graphics*, pages 107–115, June 2002.

[22] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics*, 21(3):491–500, July 2002.

[23] S. P. Lee, J. B. Badler, and N. I. Badler. Eyes alive. *ACM Transactions on Graphics*, 21(3):637–644, July 2002.

[24] R. M. Maatman, J. Gratch, and S. Marsella. Natural behavior of a listening agent. In T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, P. Olivier, and T. Rist, editors, *Intelligent Virtual Agents, 5th International Working Conference*, Kos, Greece, September 2005.

[25] X. Pan, M. Gillies, and M. Slater. Male bodily responses during an interaction with a virtual woman. In H. Prendinger, J. C. Lester, and M. Ishizuka, editors, *Intelligent Virtual Agents, 8th International Conference, IVA 2008, Tokyo, Japan, September 1-3, 2008. Proceedings*, volume 5208 of *Lecture Notes in Computer Science*, pages 89–96. Springer, 2008.

[26] X. Pan and M. Slater. A preliminary study of shy males interacting with a virtual female. In *Presence 2007: The 10th Annual International Workshop on Presence*, pages 101–108, 2007.

[27] I. Poggi, C. Pelachaud, F. De Rosis, V. Carofiglio, and B. De Carolis. Greta. a believable embodied conversational agent. In S. O. and Z. M., editors, *Multimodal Intelligent Information Presentation (Text, Speech and Language Technology Vol. 27)*. Kluwer, 2005.

[28] C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics & Applications*, 18(5):32–40, September - October 1998.

[29] M. Slater and M. Usoh. Body centred interaction in immersive virtual environments. In N. M. Thalmann and D. Thalmann, editors, *Artificial Life and Virtual Reality*, pages 125–148. John Wiley and Sons, 1994.

[30] M. Stone, D. DeCarlo, I. Oh, C. Rodriguez, A. Stere, A. Lees, and C. Bregler. Speaking with hands: creating animated conversational characters from recordings of human performance. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 506–513, New York, NY, USA, 2004. ACM Press.

[31] K. Thórisson. Real-time decision making in multimodal face-to-face communication. In *second ACM international conference on autonomous agents*, pages 16–23, 1998.

[32] H. H. Vilhjálmsson. Augmenting online conversation through automated discourse tagging. In *6th annual minitrack on Persistent Conversation at the 38th Hawaii International Conference on System Sciences*, 2005.

[33] V. Vinayagamoorthy, M. Garau, A. Steed, and M. Slater. An eye gaze model for dyadic interaction in an immersive virtual environment: Practice and experience. *Computer Graphics Forum*, 23(1):1–12, Mar. 2004.

[34] V. Vinayagamoorthy, M. Gillies, A. Steed, E. Tanguy, X. Pan, C. Loscos, and M. Slater. Building expression into virtual characters. In *Eurographics Conference State of the Art Reports*, 2006.

[35] A. Witkin and Z. Popović. Motion warping. In *ACM SIGGRAPH*, pages 105–108, 1995.