Gesture Controllers

Sergey Levine Philipp Krähenbühl Sebastian Thrun Vladlen Koltun

Stanford University *

Abstract

We introduce *gesture controllers*, a method for animating the body language of avatars engaged in live spoken conversation. A gesture controller is an optimal-policy controller that schedules gesture animations in real time based on acoustic features in the user's speech. The controller consists of an inference layer, which infers a distribution over a set of hidden states from the speech signal, and a control layer, which selects the optimal motion based on the inferred state distribution. The inference layer, consisting of a specialized conditional random field, learns the hidden structure in body language style and associates it with acoustic features in speech. The control layer uses reinforcement learning to construct an optimal policy for selecting motion clips from a distribution over the learned hidden states. The modularity of the proposed method allows customization of a character's gesture repertoire, animation of non-human characters, and the use of additional inputs such as speech recognition or direct user control.

CR Categories: I.3.6 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: human animation, data-driven animation, optimal control, nonverbal behavior generation, gesture synthesis

1 Introduction

Body language forms a crucial component of face-to-face conversation. However, it has been conspicuously missing from interactions in networked virtual environments. While voice communication is quickly gaining popularity in virtual worlds, non-verbal communication is usually performed with manual keyboard control or not at all. Automatic generation of body language for avatars in virtual worlds would make nonverbal communication a natural part of social interaction in these environments.

In this paper, we introduce *gesture controllers*, a new method for automatically generating body language animations from live speech. Gesture controllers schedule motion segments online, in real time, using a pre-computed optimal policy. In order to use optimal-policy control driven directly by speech, gesture controllers must rely on features that can be reliably extracted online from a live speech stream. The controller is thus driven by acoustic features, known as prosody, which can be computed robustly with no advance knowledge of the utterance.

*e-mail:{svlevine,philkr,thrun,vladlen}@cs.stanford.edu

Figure 1: *Gesture controllers animate two avatars in conversation.*

Prior speech-based gesture synthesis techniques directly associate animation segments with prosody features [Levine et al. 2009]. Such methods are sensitive to the amount and quality of training data, and suffer heavily from extraneous, accidental associations, known as overfitting. In contrast, gesture controllers decouple the kinematic properties of gesture, such as velocity and spatial extent, from its shape. This key distinction is informed by a body of previous work in linguistics and psychology which suggests that prosody is an informative indicator of the kinematic properties of gestures, but is generally insufficient to infer their precise shape.

Gesture controllers infer gesture kinematics from speech using a conditional random field that analyzes acoustic features in the input and infers a distribution over a set of hidden states. These hidden states model the hidden structure in gesture kinematics and are agnostic to the precise shape of gesture, thus reducing the number of false correlations and alleviating overfitting. The hidden states are used as an input to a Markov decision process, which uses an optimal policy for selecting motion segments, driven by a distribution over the hidden states. The use of reinforcement learning to compute the motion selection policy allows gesture controllers to employ transition motions and generate higher-quality animations.

The development of gesture controllers was inspired by the recent introduction of online locomotion controllers that assemble motion clips from motion capture to produce an animation that is smooth, natural, and satisfies a set of constraints [Treuille et al. 2007; McCann and Pollard 2007]. While locomotion controllers are driven by direct high-level commands (such as desired movement direction), no such clear control signal is available for body language. Therefore, we augment the optimal-policy controller with a dedicated inference layer that learns how to extract the control signal, in the form of a discrete set of hidden states, from ambiguous, multi-dimensional input.

Gesture controllers enable a number of novel applications. The motion library used to synthesize animations need not be the same as the motion capture corpus used to train the inference layer. Thus the motion library that a gesture controller draws from to schedule gestures can come from a variety of sources, need not be accompanied by corresponding speech, and need not be large enough for training a probabilistic model. This enables easy creation of gesture con-

trollers for characters with idiosyncratic gestures, characters whose gestures are affected by their environment, and even non-human characters.

2 Related Work

Previous work on body language animation can be divided into rule-based and data-driven approaches, with the former most often used to animate arm gestures, and the latter generally being applied to facial animation, with a few notable exceptions. Early approaches to gesture synthesis used manual annotation and behavior scripts to specify the placement and shape of gestures [Cassell et al. 1994; Perlin and Goldberg 1996]. In fact, new annotation schemes for manually specifying gestures are still being developed [Hartmann et al. 2002; Kopp and Wachsmuth 2004; Kipp et al. 2007]. Automatic methods for gesture synthesis have focused almost exclusively on natural language parsing techniques, and therefore generally operate on text input. Cassell et al. [2001] proposed a rule-based method utilizing NLP techniques to place gestures for simultaneous speech and animation synthesis. More recently, Neff et al. [2008] used a probabilistic technique to learn gesture placement patterns from text and transcribed video (with gestures annotated in the transcript). Stone et al. [2004] used gesture animations obtained from motion capture of real speakers, but could only synthesize gesture animations for utterances that consisted of re-arranged parts of pre-recorded phrases, and therefore could not operate on arbitrary input.

Fully data-driven, speech-based methods have been employed more heavily for synthesis of facial animations, often with a focus on lip synchronization. Bregler et al. [1997] proposed a video-based method that reordered frames in a video sequence to correspond to a stream of phonemes extracted from speech. This method was further extended by Brand [1999] by retargeting the animation onto a new model and adopting a sophisticated hidden Markov model for synthesis. Hidden Markov models and other probabilistic temporal models are now commonly used to model the relationship between speech and facial expression [Li and Shum 2006; Xue et al. 2006; Englebienne et al. 2007]. Since lip motion dominates facial animation and is physiologically determined by speech, most speech-driven facial animation synthesis methods process the input to extract either phonemes or lower-level features that are used for phoneme recognition [Deng and Neumann 2007].

In contrast, several methods have been proposed that employ vocal prosody features, such as pitch and intensity, to generate facial expressions and head motion [Albrecht et al. 2002; Chuang and Bregler 2005]. Recently the authors have initiated the study of full-body gesture synthesis from prosody features in live speech [Levine et al. 2009]. This prior work directly associates animation segments with prosody features and suffers from overfitting. No prior method has been proposed that attempts to model the connection between prosody and full-body gesture as a hidden process that associates prosody and gesture kinematics.

3 Prosody and Gesture Kinematics

In order to capture true correspondences between gesture and prosody and avoid extraneous, false correlations, we learn a relationship between prosody features and a set of explicitly defined kinematic parameters that are orthogonal to gestural form. This choice is motivated by a body of previous work in linguistics and psychology that suggests that gestural kinematics and rhythm, as well as the style of human motion in general, correspond well to prosody [Feyereisen and de Lannoy 1991; de Meijer 1989; Shröder 2009], while gestural form is more strongly tied to the semantic meaning of the accompanying utterance [McNeill 1992].

Explicit study of the connection between gesture and prosody dates back to Dobrogaev's experiments [1931] and Birdwhistell's pioneering work on kinesics [1952]. Decades of observation and experimentation established strong links between prosody and the timing of gestures. For example, the phonological synchrony rule states that the stroke of the gesture precedes or ends at, but does not follow, the phonological peak of the utterance [McNeill 1992]. More generally, speech and gesture have an intricate rhythmic relationship [Loehr 2007].

There is a general consensus in linguistics that gesture form is tied closely with the semantic meaning of the accompanying speech [McNeill 1992; Kendon 2004], which suggests that prosody alone is insufficient to infer the precise shape of a gesture. However, prosody and motion are known to correspond to emphasis [Feyereisen and de Lannoy 1991; Valbonesi et al. 2002] and were individually found to correlate with emotional state [de Meijer 1989; Shröder 2009]. Aspects of motion that correlate with emotion, such as force, directness, and velocity [de Meijer 1989], are better described as dynamics or kinematics than form, since a gesture can present the same shape at a variety of velocities, with varying levels of directness, and with different amounts of force. In addition, beat gestures, which are used to mark emphasized words or phrases and form the majority of extemporaneous gestures, are "formless" [McNeill 1992], and are differentiated only by their kinematic style. We conclude that prosody is an informative indicator of the kinematics and overall activation of gestures and can be used to generate a stream of form-invariant kinematic parameters from live speech.

Prosody can be described with well-understood features such as pitch, syllable length, and intensity. However, there is no universally accepted set of parameters for describing the kinematic style of human motion. A widely used classification is Laban Movement Analysis [Newlove 1993]. LMA factors are descriptive and are generally extracted manually by a trained specialist.

Our choice of kinematic parameters is based on observation and prior work in both computer graphics and linguistics. The proposed method makes use of six parameters to capture the kinematics of a gesture: temporal and spatial extent, velocity, acceleration, curvature, and the height at which the hands are held. In one of the earliest quantitative analyses of gesture, Efron classified gestures by their kinematics according to descriptors such as "sinusoidal" and "angular," referring to curvature, the distance from the body attained by the hand during the stroke, referring to spatial extent, and the number of gestures executed in a specific time span, referring to the inverse of temporal extent [Efron 1972]. More recently, Hartmann et al. [2005] showed that expressive gestures can be generated from a kinematic description consisting of temporal and spatial extent, a measure of overall activation, and a measure of power, consisting largely of acceleration magnitudes. Zhao and Badler [2005] showed that LMA factors can be inferred with neural networks using temporal and spatial extent, velocity, acceleration, and curvature. The hand height parameter provides a quantitative measure of "overall activation," since hand height is higher when the hands are raised into the active space, and lower when the speaker is not gesturing. It is also inspired by McNeill's partitioning of the gesture space [1992].

4 Overview

Gesture controllers consist of two layers: an inference layer, which analyzes vocal prosody and produces a distribution over a set of learned hidden states, and a control layer, which uses the inferred hidden state distribution and other available inputs to select the most appropriate gesture segments from a library of motion data

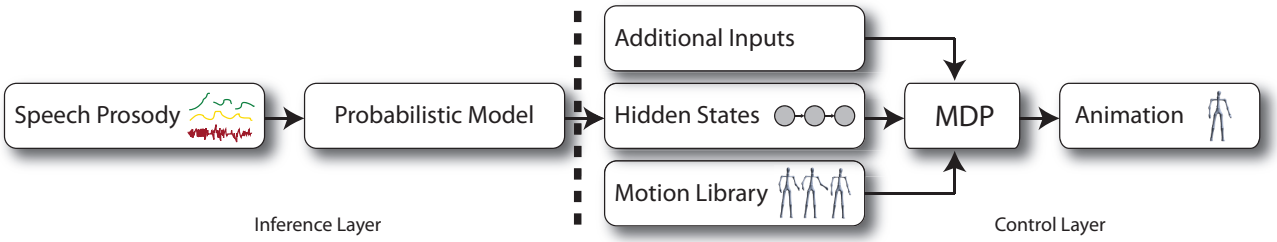


Figure 2: *Gesture controllers consist of two layers: the inference layer, which infers a distribution over a set of intermediate hidden states, and the control layer, which selects appropriate gesture segments from a motion library.*

using a pre-computed optimal policy. The full synthesis process is summarized in Figure 2.

Gesture controllers are inspired by the recent introduction of locomotion controllers that assemble a sequence of motion capture clips to navigate through an environment under continuous control from the user [Treuille et al. 2007; McCann and Pollard 2007]. However, while locomotion controllers are driven by direct high-level commands, speech does not directly provide a clear control signal for body language. To overcome this challenge, we introduce a dedicated inference layer to the controller. This layer infers a distribution over a learned hidden-state representation of the underlying gesture style – the "glue" that, in our method, connects vocal prosody with a concrete gestural display. The optimal-policy gesture selection is then driven by this low-dimensional hidden representation.

The inference layer, described in Section 6, consists of a conditional random field that is trained on a corpus of motion capture data with accompanying audio using a specialized training algorithm that infers hidden structure in the motion data. To ensure that the learned structure reflects those aspects of motion that can reasonably be inferred from vocal prosody, we constrain the CRF to use only the kinematics of the motion, rather than its precise form. This is accomplished by representing the motion data in terms of its kinematic parameters, which are formally defined in the next section. The parameters are analyzed to learn a hidden-state representation using expectation maximization (EM), and the CRF is then trained to infer a belief distribution over these hidden states from a sequence of multi-dimensional prosody features.

The control layer, described in Section 7, uses the inferred belief distribution to select the gesture segment that minimizes a cost function, which includes a cost for animation smoothness, deviation from the inferred belief, and other terms that can be used to account for additional input channels, such as semantic interpretation. We model gesture selection as a Markov decision process (MDP), and use reinforcement learning to train an optimal policy that is able to quickly select the most appropriate motion at runtime. Gesture segments are selected from a gesture library, which need not contain the same data as that used to train the inference layer.

The separation of the inference and control layers allows the control layer to accept additional sources of input that modulate gesture kinematics or inform the selection of specific motion segments. We demonstrate two possibilities for such additional input in Section 8.

Decoupling the gesture library from the inference layer training set also allows the same inference layer to be used with any gesture library. This enables fast and easy creation of new gesture controllers by reusing the same probabilistic model, allowing customization of a character's gesture repertoire, the use of gestures that do not have accompanying audio, and even the use of gestures created by an artist for a non-human character. In Section 9.2, we demonstrate a variety of gesture controllers created with the same probabilistic model and different gesture libraries. These controllers can animate characters constrained by environmental factors, characters holding objects, and even characters with non-humanoid morphologies.

5 Data Processing and Representation

The inference layer is trained on a corpus of motion capture data with accompanying speech, and the control layer uses a gesture library that is produced from motion capture or artist-created animation. The motion data is segmented and processed to extract kinematic parameters, and the speech data is processed to extract prosody features. In this section, we formally define the kinematic parameters and prosody features used in our method, and describe how they can be automatically computed from the motion capture and audio data.

To generate the training set for the inference layer, motion capture and audio were obtained from unscripted conversations. The recorded subject was instructed to avoid extraneous movement, such as scratching, and asked to minimize iconic gestures [McNeill 1992], but no other instructions were given. The training set for the probabilistic model used throughout this paper consists of a 12-minute excerpt from a conversation on video games. Unless stated otherwise, the gesture library for the control layer is produced from the same motion data.

5.1 Kinematic Parameters

The motion capture data, consisting of joint orientations for a 14-joint skeleton recorded at 60 frames per second, is processed to determine gesture phase boundaries and extract kinematic parameters. Gesture units consist of active phases, such as *strokes* and *retractions*, as well as passive phases, such as *holds* [McNeill 1992]. The boundaries between these phases are identified by examining the velocities of the hands and placing boundaries when the mean velocity crosses a pre-determined threshold.

Kinematic parameters are computed for each segment. All parameters are expressed on a log scale to reflect the perceptual similarity between motions that exhibit high parameter values, since the same absolute difference in length between two long segments is less noticeable than between two short ones. Let t_1 and t_n denote the starting and ending times of a segment, let p_i, v_i, and a_i denote the position, velocity, and acceleration of the two hands, expressed as 6-dimensional vectors at frame i within the segment, and let h_i denote the sum of the heights of the hands. The parameters corresponding to temporal extent $\bar{T}$, spatial extent $\bar{D}$, velocity $\bar{V}$,

acceleration $\bar{A}$, curvature $\bar{C}$, and height $\bar{H}$ are then given as

$$\bar{T} = W_T \log B_T(t_n - t_1)$$

$$\bar{D} = W_D \log \left|\left| \frac{B_D}{\bar{T}} \sum_{i=1}^{n} p_i - p_1 \right|\right|$$

$$\bar{V} = W_V \log \left|\left| \frac{B_V}{\bar{T}} \sum_{i=1}^{n} v_i \right|\right|$$

$$\bar{A} = W_A \log \left|\left| \frac{B_A}{\bar{T}} \sum_{i=1}^{n} a_i \right|\right|$$

$$\bar{C} = W_C \log \left|\left| \frac{B_C}{\bar{T}} \sum_{i=1}^{n} v_i \times a_i \right|\right|$$

$$\bar{H} = W_H \log \frac{B_H}{\bar{T}} \sum_{i=1}^{n} h_i$$

The weights B_* and W_* normalize the parameters to have approximately the same range.

5.2 Prosody Features

The audio signal is processed to extract pitch F_0 and intensity I, as well as syllable boundaries, which are identified by detecting troughs separated by peaks in the intensity curve. F_0 and I are averaged over each syllable to reduce the effects of local phonetics on their values, producing $\bar{F_0}$ and $\bar{I}$. The final set of prosody features consists of $\bar{F_0}$, $\bar{I}$, and L, the length of the syllable segment. The resulting stepwise functions represent the inputs to the model. Since F_0 is undefined for unvoiced syllables, such as consonants and periods of silence, this feature is represented as a uniformly distributed random variable for unvoiced segments.

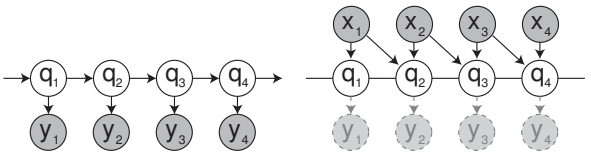
6 The Inference Layer

The inference layer takes as input a temporally discretized sequence of prosody features and infers a distribution over a set of hidden states at each time step. The hidden-state representation of gesture motion is learned from the kinematic parameters, which constrain the learning process to only the stylistic content of the motion. In this section, we consider the specific challenges associated with the ambiguous and complex relationship between prosody and body language and present a method for training a conditional random field with latent variables to perform the inference task.

6.1 Probabilistic Temporal Models

Let the time series $X = (x_1, x_2, \ldots, x_n)$ denote the input sequence of prosody features, where $x_t = (\bar{F_0}, \bar{I}, L)^T$. Let $Y = (y_1, y_2, \ldots, y_n)$ denote the output sequence, where $y_t = (\bar{T}, \bar{D}, \bar{V}, \bar{A}, \bar{C}, \bar{H})^T$. The inference layer must learn a discrete hidden state space Q such that a sequence of hidden states $(q_1, q_2, \ldots, q_n)$ can be inferred from X and then used to reconstruct Y. The use of a hidden process both reduces the dimensionality of the signal to make the use of an optimal-policy controller tractable in the control layer, and allows the inference layer to model the complexities of the ambiguous relationship between prosody and gestural style.

Previous work on associating facial poses and head orientation with speech has employed hidden Markov models [Brand 1999; Sargin et al. 2008]. However, there are indications that bodily motion and vocal prosody cues may not always align in time. For example, McNeill suggests that gestures may anticipate the corresponding phrase [McNeill 1992]. Therefore, the kinematic parameters may at times correlate with prosody features at different points in time more strongly than with cotemporal ones. To account for this, we must consider not just the current input at a specific time step, but a temporal window over nearby inputs. In an HMM, such relationships creates dependencies between hidden states across time steps, which the standard HMM does not handle. Since a discriminative model does not need to take dependencies within the input into account, a conditional random field is more appropriate for handling a temporal window [Lafferty et al. 2001].



(a) HMM trained on output (b) CRF trained on hidden states

Figure 3: *Training a probabilistic model that captures hidden structure in the kinematic parameters. (a) An HMM is trained on the kinematic parameters with EM to obtain a distribution over a sequence of hidden states. (b) This distribution is fixed and a CRF is trained with ML to infer the hidden states from the input prosody features.*

The standard CRF is trained directly by maximizing $P(Y|X)$. However, such a model is fully observed and cannot capture hidden structure in the output. Several methods have been proposed to augment the CRF with latent variables, but these methods rely on a discrete set of output labels to keep the training process tractable [Morency et al. 2007; Wang et al. 2006]. In general, training a CRF with hidden states in the same manner as an HMM requires multiple steps of expectation maximization (EM), with maximum likelihood estimation (ML) at each step [Rabiner 1989]. The partition function couples the parameters of the CRF, so ML requires an iterative procedure such as gradient ascent, and therefore is much more time consuming than ML in an HMM [Lafferty et al. 2001]. This makes EM, which performs ML at each step, intractable in the general case. Instead, we train the CRF in two stages, by first capturing the hidden structure in the output signal using EM, and then maximizing the conditional likelihood of the hidden structure given the input signal with standard ML parameter estimation. This method is similar to the remapped HMM proposed by Brand [1999].

6.2 The Remapped Conditional Random Field

First, a hidden Markov model Λ_Y is trained only on the output signal Y with EM to obtain a distribution over a sequence of hidden states $Q = (q_1, q_2, \ldots, q_n)$. This allows the model to capture the hidden structure in the output signal. We then fix the distribution over Q and train the CRF with ML to maximize the probability of the hidden state distribution given the input signal, $P(Q|X)$. The stages of the training algorithm are illustrated in Figure 3. The CRF uses transition features of the form $1_{\{q_{t-1}=i, q_t=j\}}$ and observation features of the form $1_{\{q_t=i\}} P(x_t \in C_k)$. $P(x_t \in C_k)$ gives the probability of input vector x_t belonging to one of K clusters identified in the input signal with EM clustering. Each cluster is modeled with a Gaussian distribution. The distribution over kinematic parameters $P(Y)$ can be readily reconstructed from a distribution of hidden states $P(Q)$ as $P(Y) = P(Y|Q)P(Q)$, where $P(Y|Q)$ is readily available from the observation distributions of the HMM Λ_Y. The distribution $P(Q)$ itself can be inferred online using the forward algorithm at each time step [Rabiner 1989].

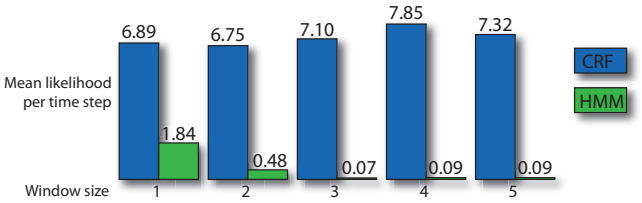


Figure 4: *The CRF produced higher likelihood scores on a novel motion-captured utterance from a novel speaker than the HMM, and a slight improvement was observed with increase in input window size. Scores are given as likelihood per time step.*

6.3 Comparison with a Hidden Markov Model

To evaluate the performance of the remapped CRF, we compared it to an HMM trained to perform the same inference task. The HMM was constructed by remapping Λ_Y to another HMM, as proposed by Brand [1999]. We compared the likelihood scores produced by the two models for a novel input-output pair, consisting of a motion-captured utterance from a novel speaker. These likelihood scores reflect the degree to which each model is able to explain the novel data, and indicate how well the model generalizes. The results of the comparison are shown in Figure 4. Scores were computed for each model with five different temporal windows, ranging from 1 time step to 5. As expected, the performance of the HMM dropped sharply with the size of the input window, due to unhandled dependencies in the observations. The CRF consistently outperformed the HMM, which indicates that the CRF is a more powerful model and is better suited for inference of kinematic parameters from speech. The likelihood scores of the CRF increased slightly with larger input windows, peaking at size 4. The CRF with an input window of size 4 is used in all examples in this paper and the accompanying video.

7 The Control Layer

The inference layer presented in the preceding section produces a belief distribution over a set of hidden states. Given this belief, appropriate gestures are selected from a gesture library using the control layer described in this section. The library can be generated automatically from the same motion data that is used to train the models or, as discussed in Section 9.2, a different motion library. The synthesized animation is constructed by selecting segments from the library that follow each other smoothly and are consistent with the inferred belief distribution. Other sources of input can also be integrated, as described in Section 8. The current section introduces optimal gesture selection by describing how the animation can be synthesized offline with dynamic programming. We then describe our online control layer, which uses a Markov decision process to synthesize an animation stream using an optimal gesture selection policy.

7.1 Offline Animation with Dynamic Programming

We formulate the animation task as follows: let Y_t be a random variable representing the kinematic parameters at time step t, and let y_a be the parameter values associated with a segment a. We define a *successor function* $S(a_i, f_i, a_j, f_j)$ that returns the cost, in terms of loss of animation quality, of transitioning from frame f_i of segment a_i to frame f_j of segment a_j. Our goal is to obtain the lowest-cost animation sequence $A = \big((a_1, f_1), (a_2, f_2), \ldots, (a_T, f_T)\big)$. While the definition below explicitly contains the kinematic parameters y_a, they will later be factored out when we consider how the expected difference of the parameters is computed:

$$A = \arg\min_A \sum_{t=1}^{T} \big[S(a_{t-1}, f_{t-1}, a_t, f_t) + E(||y_{a_t} - Y_t||)\big].$$

A variety of successor functions S may be used, but to reduce the state space of the problem, we will only allow transitions into the beginning of a segment, and will only allow a segment a to end at its last frame n_a, or at one of a fixed number of intermediate *interruption* frames R_a, rather than at every frame in the segment. In our implementation, 3 interruption frames are selected at uniform intervals for every segment. We define S as

$$S(a_i, f_i, a_j, f_j) = \begin{cases} 0 & \text{if } a_i = a_j \text{ and } f_j = f_{i+1} \\ D_{\text{int}}(a_i, f_i, a_j) & \text{if } f_i \in R_{a_i} \text{ and } f_j = 0 \\ D_{\text{s-p}}(a_i, a_j) & \text{if } f_i = n_{a_i} \text{ and } f_j = 0 \\ \infty & \text{otherwise} \end{cases}$$

The function $D_{\text{s-p}}$ is the successor-predecessor distance function given by $D_{\text{s-p}}(a_i, a_j) = D_{\text{seg}}(a_{i+1}, a_j) + D_{\text{seg}}(a_i, a_{j-1})$, where a_{i+1} is the segment that follows a_i in the training data, and a_{i-1} is the segment that precedes it. D_{seg} gives the perceptual distance between two segments. This distance takes into account segment length, initial pose, and a signature of the dynamics of the segment that consists of the difference in the log of the segments' velocities and displacements. The candidate segment a_j is compared against a_{i+1} the true successor of a_i, and a_i is in turn compared against the true predecessor of a_j. This provides a good measure of the perceptual discontinuity of substituting in a_j as a new successor.

The function $D_{\text{int}}(a_i, f_i, a_j)$ gives the perceptual distance between frame f_i of a_i and the first frame of a_j, along with an additional penalty term to discourage unnecessary interruptions. Allowing transitions out of interruption frames improves the flexibility of the controller in cases when no one segment provides a good fit for the predicted parameter distribution Y, but should be discouraged with a penalty term to ensure that most gestures play to completion.

As mentioned previously, the kinematic parameters y_{a_t} can be factored out of the cost function. Let γ_t be the distribution over the CRF hidden states at each time step, given by

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)},$$

where α_t and β_t are the forward and backward probabilities at time t [Rabiner 1989; Lafferty et al. 2001]. Since the output distribution for each state i is a Gaussian with mean μ_i and covariance Σ_i, the expected difference in parameter values is given by

$$E(||Y_t - y_{a_t}||) = \sum_i \gamma_t(i) \int_y ||y - y_{a_t}|| \mathfrak{N}[y, \mu_i, \Sigma_i],$$

where $\mathfrak{N}[y, \mu_i, \Sigma_i]$ is the density of the Gaussian with mean μ_i and covariance Σ_i at y.

The integrals for all states can be precomputed for each segment a into a vector δ_a, allowing the expected distance to be simplified to $E(||y_{a_t} - Y_t||) = \gamma_t \cdot \delta_a$. We can thus construct an efficient cost function at each time step, given by

$$Q_t(a_j, f_j) = \gamma_t \cdot \delta_{a_j} + \min_{a_i, f_i} \left[S(a_i, f_i, a_j, f_j) + Q_{t-1}(a_i, f_i) \right].$$

The optimal sequence A can then be found using a dynamic programming algorithm that constructs, at each time step, a table containing the lowest-cost path to each frame in each segment. Note that the kinematic parameters are no longer a part of the cost function: they are subsumed by the hidden state distribution γ_t, which now serves to connect the control layer to the inference layer.

7.2 Online Animation as a Markov Decision Process

When generating an animation stream from live speech, the dynamic programming algorithm in Section 7.1 cannot be used, since the controller must decide which motion segment to show before the entire utterance is known. Instead, we model gesture selection as a Markov decision process and use value iteration to find the optimal gesture selection policy [Bertsekas 2007].

During online synthesis, the belief over the model's states is given by the forward probabilities α_t, since the backward probabilities used in the computation of γ_t are not available. The cost of displaying frame f' from segment a' at time step t, given the previously displayed frame f from segment a, is then given by

$$C(a', f', a, f, \alpha_t) = \alpha_t \cdot \delta_{a'} + S(a, f, a', f').$$

The value function that optimizes this cost over an infinite time horizon with a discount factor η is given by

$$V_c(a', f', a, f, \alpha_t) = C(a', f', a, f, \alpha_t) + \eta \min_{a'', f''} V_c(a'', f'', a', f', \alpha_{t+1}).$$

However, since V_c is linear in α_t, we have $V_c(a', f', a, f, \alpha_t) = \sum_i \alpha_t(i) V_c(a', f', a, f, e_i)$, where e_i is the i^{th} canonical vector. In the absence of future observations, the forward probabilities at the next time step are given by $\alpha_{t+1}(i) = \sum_j P(s_i|s_j)\alpha_t(j)$, where s_i represents the i^{th} state of the CRF. These properties allow us to transform the problem into a fully discrete MDP with value functions given by

$$V_d(a', f', a, f, s_i) = C(a', f', a, f, e_i) + \eta \min_{a'', f''} \sum_{s_j} P(s_j|s_i) V_d(a'', f'', a', f', s_j).$$

At run-time, the optimal frame and segment at time t is given by

$$(a^\star, f^\star) = \arg\min_{a', f'} \sum_i \alpha_t(i) V_d(a', f', a, f, s_i).$$

Unfortunately, this formulation requires one entry in the value function table for each pair of frames in the gesture library, which is not tractable for large libraries. However, our choice of successor function ensures that a segment a always begins at the first frame and only ends on completion or at one of a small set of interruption frames R_a. This allows us to remove from the table all entries for which $f' \neq 0$, or f is not the last frame in a and $f \notin R_a$. The formal definition of the resulting value function recursion is presented in Appendix A.

The value function is computed using the value iteration algorithm. We then cull all entries $V'_d(a', f', a, f, s_i)$ for which there exists $(a^\star, f^\star)$ such that, for all s_i, $V'_d(a^\star, f^\star, a, f, s_i) < V'_d(a', f', a, f, s_i)$ — that is, $(a^\star, f^\star)$ is a better choice than (a', f') *regardless* of the current belief over the kinematic parameters, which is often the case when (a, f) and (a', f') are very different and do not follow each other smoothly. In practice, this pruning scheme usually removes about 98% of the entries in the table.

8 Additional Inputs

The separation of the inference and control layers allows the control layer to accept additional sources of input that modulate gesture kinematics or inform the selection of specific motion segments. First, we show how the input to the control layer can be augmented with rudimentary word recognition to produce semantically meaningful gestures for live speech. Since speech recognizers can only interpret words after they are spoken, and since gestures generally co-occur with their associated word, online generation of semantically meaningful gestures for live speech poses a significant challenge. Our method is able to ameliorate this difficulty with a modification to the MDP cost function, which enables an ongoing gesture to be interrupted and replaced with a semantically meaningful one, resulting in a minimal delay between a word and its associated gesture. Second, we show how the presence of an intermediate gesture style representation enables the user to directly manipulate gesture style, in order to achieve a desired communicative effect. Such manipulation is known as Transformed Social Interaction and is one of the key advantages of communication via networked virtual environments [Bailenson et al. 2004].

8.1 Speech Recognition

To demonstrate the ability of gesture controllers to incorporate additional inputs, we show how the MDP formulation can be modified to handle a set of *semantic labels* w generated by a speech recognizer. These labels can be produced by a keyword spotter, as discussed in Section 9.2, or by more sophisticated NLP techniques. The value function is augmented to take the semantic label w into account by adding a semantic cost term $C_{\text{sem}}(a', w)$. The semantic cost is set to a large value for gestures which do not match the semantic label, and 0 for gestures that do. The semantic cost can also vary continuously for gestures that resemble the label to varying degrees, but we found that a binary cost worked well.

Unfortunately, real semantically meaningful gestures often co-occur with the associated word, but speech recognizers do not recognize a word until after it is spoken. We make up for this deficiency by allowing the MDP to transition into the active stroke of a semantic gesture, which effectively transforms the currently running gesture into the semantically meaningful one. In practice, this results in the character performing the gesture almost in synchrony with the associated word.

To ensure that the resulting value function remains compact, we still only allow transitions into one frame in a semantic segment, but this frame is no longer constrained to be the first frame. Instead, for each frame f at which a segment a is allowed to end, we select the optimal frame $f'_{\text{start}}(a, f)$ in each semantic segment a', using the frame-to-frame distance $D_{\text{int}}(a, f, a', f')$ from Section 7.1. To avoid transitions into the very end of a semantic gesture, which would not produce the desired visual effect, we constrain $f'_{\text{start}}(a, f)$ to be below a fixed fraction of the total length of a'. The formulation of the resulting value function is presented at the end of Appendix A.

Augmented in this way, the control layer produces gestures that

carry the style, emphasis, and rhythm inferred by the probabilistic model from prosody and, when a semantic label is generated, the form of a correct semantically meaningful gesture, also selected to best match the inferred style.

8.2 Style Manipulation

To illustrate the flexibility of gesture controllers, we provide the user with direct control over the style of the generated body language. This lets the user modulate his or her avatar's behavior to achieve a desired communicative effect. For example, a teacher in a virtual classroom might choose to adjust the body language of his or her avatar to generate more energetic animations for students who are easily bored, and more subdued animations for students who are easily distracted. Other uses for such manipulation, commonly termed Transformed Social Interaction, are discussed by Bailenson et al. [2004].

Recall that the interface between the inference and control layers consists of a set of pre-computed vectors δ_a for each motion segment a, which are determined by the kinematic parameters of the segment and the parameter distributions associated with the hidden states. By transforming the parameters of a segment a before computing δ_a, we can transform the style of the synthesized animation. We estimate a style transformation T_m for each desired tone. By fitting a Guassian to the set of parameter vectors observed in a motion capture sequence exhibiting a particular tone and comparing it with a Guassian fitted to parameters in a neutral data set, we can compute the transform T_m that maps the neutral distribution to the one with the desired tone. For each transform T_m and each segment a, we then compute a transformed vector δ_a^m, defined as

$$\delta_a^m = E(||T_m Y_i - y_a||) = \int_y ||T_m y - y_a|| \mathfrak{N}[y, \mu_i, \Sigma_i].$$

The user can continuously control the degree to which their avatar exhibits each tone by specifying, at run-time, a unit vector d. The m^{th} entry of d indicates the degree to which the m^{th} tone should be expressed. Recall that the cost function for the MDP described in Section 7.2 is linear in the CRF state distribution. With transformed kinematic parameters, the cost function is also linear in d, and is defined as

$$C(a', f', a, f, \alpha_t, d) = \sum_m d(m)\alpha_t \cdot \delta_{a'}^m + S(a, f, a', f').$$

Since the MDP transition model predicts no change in d, the resulting value functions are separable and can be trained independently, resulting in a set of discrete value functions. At runtime, these value functions are simply combined linearly to produce the value function for the desired emotional vector d as $\sum_m d(m) V_d^m$.

9 Results

9.1 Experimental Evaluation

We conducted two studies to evaluate the quality of the generated animations by comparing them to the original motion capture for an utterance and to animations synthesized by the approach of Levine et al. [2009]. In the first study, the utterance and accompanying motion capture were taken from an actor with similar speaking skills as the actor used to train the gesture controller. In the second study, the utterance and motion capture were taken from an individual with no special training, in order to determine if the gesture controller could transplant the more compelling body language of the training set actor onto the novel speaker's speech. 35 participants were recruited from the student body of an English-speaking university in the United States, using local e-mail lists that spanned a variety of departments and degree programs. Each participant received a $10 Amazon gift certificate.

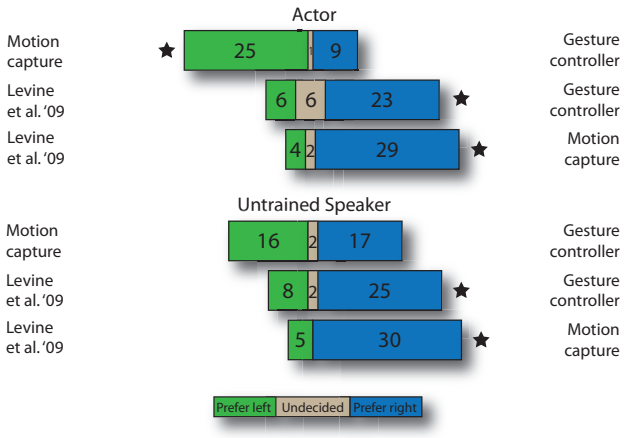


Figure 5: *The gesture controller was compared against original motion capture data and the previous approach of Levine et al. [2009]. 35 responders compared sets of animations generated for speech from an untrained speaker and for speech from a skilled actor. Results marked with $\star$ are statistically significant ($p < 0.01$) according to a two-tailed independent single sample t-test.*

Each participant was shown side-by-side pairwise comparisons and asked to select the one they believed was most realistic in each pair (or select "undecided"). The videos were shown on a computer screen in a controlled environment, to ensure that each participant could watch and listen to the videos without distraction. Each comparison consisted of two 15-25 second animations for the same utterance generated by two of the three methods – gesture controllers, Levine et al. [2009], and the original motion capture. Both the order of the pairwise comparisons and the left-right order of the two videos were randomized. Each comparison used one of ten randomly selected excerpts from its utterance ("actor" or "normal speaker"). The ten 15-20 second clips were excerpted at random from the "actor" and "normal speaker" utterances, which were 3 and 7 minutes in length, respectively.

Figure 5 presents the results of the studies. The participants were 40% female, 60% male, with ages ranging from 20 to 60 (mean age 23). All responders spoke fluent English and reported completing at least four years of an English-language primary or secondary school. All reported having no difficulty understanding English-language television or film without subtitles.

Gesture controllers consistently outperformed the approach of Levine et al. [2009], which trains an HMM that directly associates gesture segments with prosody features. In the comparison against the trained actor, the original motion capture outperformed gesture controllers. However, the comparison against the motion capture sequence from the untrained speaker showed that gesture controllers compared well with the original motion capture. This suggests that our technique was able to successfully transplant the more effective body language of the training set actor onto the novel utterance.

Motion Library	Library Size	Training Time
Standard library	11 m 46 s	68 m 30 s
Sitting on stool	3 m 10 s	5 m 07 s
Holding phone	4 m 00 s	6 m 12 s
Sitting with mug	2 m 40 s	3 m 50 s
Holding pitchfork	5 m 52 s	12 m 05 s
Octopus	51 s	7 s

Table 1: *New gesture controllers can be generated quickly by reusing the same probabilistic model. We show training times for five different gesture controllers, as well as the gesture controller for the standard library comprised of the same motion data as was used to train the probabilistic model.*

9.2 Examples

In the accompanying video, we present examples of animations generated using a variety of gesture controllers. All controllers use an inference layer consisting of a CRF with 25 hidden states, trained on a 12-minute excerpt from a conversation on video games. The training set for the control layer varies between examples – Table 1 shows the lengths of the various libraries.

To demonstrate the ability of gesture controllers to handle additional inputs, we use the publicly available Sphinx 3 speech recognizer [The CMU Sphinx Group 2007] to spot a set of keywords in the utterance. Each keyword is manually associated with a set of gestures in the gesture library that carry the appropriate meaning, and the word recognizer produces semantic labels when it detects the keywords. Semantic parsing of speech is not the subject of this work, and therefore we use a simple method for associating gestures with semantic meaning. Semantic analysis of *live* speech in particular is a difficult and under-explored topic. A more in-depth analysis of the association between semantics and gesture can be found in works by Cassell et al. [2001] and Neff et al. [2008]. Instead, this application demonstrates that gesture controllers can smoothly integrate additional information into the decision process when it is available.

User control of body language tone is demonstrated with animations generated for the same utterance using three different tones: a subdued tone trained on a sad speaker, an energetic tone trained on an excited speaker, and the neutral tone used in the standard library. The accompanying video shows the three tones side-by-side for comparison.

We also demonstrate the modularity of gesture controllers with examples of controllers where the gesture library does not come from the same motion corpus as the one used to train the inference layer. The decoupling of the gesture library from the inference layer's training set enables the same probabilistic model to be reused with different gesture libraries to quickly construct new, customized gesture controllers. The new libraries do not need to be accompanied by speech, since the motion segments are associated with hidden states using only their kinematic parameters. They also need not be large enough for training a probabilistic model, but must merely contain enough variation to produce an interesting animation stream. In Table 1, we show the time needed to train new customized gesture controllers with the same probabilistic model. Performance was measured on an Intel Core i7 2.66 GHz.

As shown in Figure 6, we can use small libraries of motions in particular situations – such as holding a phone or sitting with a mug – to animate avatars engaged in such activities. The controllers are driven by the same probabilistic model, but the gestures are selected from different motion libraries that contain gestures appropriate for the respective environmental conditions. The situation-specific libraries range from 3 to 6 minutes in length, providing enough variation for a convincing animation, but not enough motion-speech associations to train an effective CRF.

In addition, the motion data used in the library need not come from motion capture, which enables us to animate even non-humanoid avatars. Figure 6 also shows frames from an animation of an octopus character, which uses the same probabilistic model as the other examples. The gesture library for the octopus was created by a professional animator. The octopus character uses the same kinematic parameters as the human characters, with all values averaged over eight tentacles instead of two arms. Since the library need not contain accompanying audio, the animator did not need to exert additional effort synchronizing and fitting the character's gestures to speech. Since animators already create gesture animations for characters in animated films, this technique could be used to export these characters as animated, gesturing avatars for use in networked virtual environments. The accompanying video presents examples of each of these gesture controllers, preceded by an excerpt from the motion library with which they were generated.

10 Limitations

Synthesis of body language from live speech carries several inherent limitations. Most importantly, although gesture has been observed to anticipate the co-occuring word [McNeill 1992], a method that is driven by live speech can only follow the utterance. However, a number of other limitations that are present in earlier works can be addressed more easily with the proposed approach. For instance, gestures often carry additional information that is not reflected in speech [McNeill 1992]. While gesture controllers are mainly speech-driven, and therefore would not be able to add significant additional information, the capacity of the method to handle additional input channels suggests that such additional channels can be used to inform the synthesized animation. For example, a user can use keyboard commands to seamlessly insert meaningful gestures into a stream of speech-driven body language.

Gesture controllers carry several additional limitations due to their utilization of a pre-computed optimal policy. While new gesture controllers can be generated quickly for small gesture libraries, this still requires a new value function to be learned (see Table 1). Gestures therefore cannot be swapped in and out of the controller interactively, and new gestures cannot be inserted into an existing controller without retraining the MDP. This limits a controller's gesture repertoire during online synthesis to only those gestures found in its library. In addition to requiring all gestures to be present in the library when the control layer is trained, the training set for the inference layer must be extensive enough to contain a representative sample of significant prosody cues and prosody-gesture associations. While this set need not be as large as for methods that employ a direct gesture-speech association, a significant amount of motion capture is still required.

11 Conclusion

We presented gesture controllers, a new method for generating body language animations from live speech. The proposed technique uses dedicated inference and control layers that decompose the control problem by first inferring a distribution over a set of hidden states that correspond to gestural style, and then using this distribution to select motion segments according to a pre-computed optimal policy. Inference is driven by vocal prosody, which can be reliably extracted from live microphone input, enabling online animation of avatars in spoken conversation.

The modularity of gesture controllers makes them particularly suit-

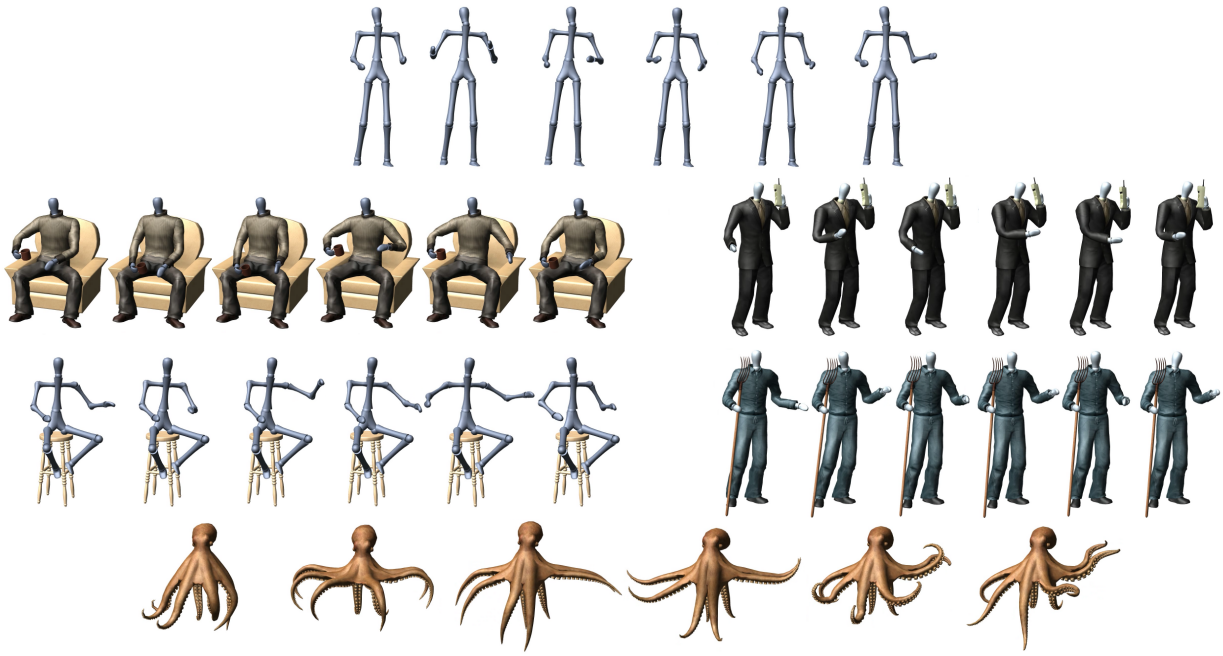


Figure 6: *New gesture controllers can be quickly created using different gesture libraries. Here, different controllers animate avatars whose gestures are affected by their environment, by the objects they are holding, or by their non-human morphology. All characters are animated for the same utterance.*

able for further extension in future work. In particular, the speech recognition module can be improved to extract more complex labels from speech through sophisticated natural language processing. Additional user input and environmental knowledge can be incorporated to provide additional context and help select the most appropriate gestures. The control layer can also be further developed to allow animation of gesturing characters engaged in other activities, with the MDP serving to arbitrate competing control signals to strike a balance between gesturing and carrying out other actions.

The versatility of the proposed method also affords considerable customization. Users can swap in gesture libraries without the need for training new probabilistic models, and even non-humanoid, artist-created avatars can be animated using models trained on human body language. By allowing the gesture library to be modified or swapped out entirely, the method allows the user to customize which gestures their avatar should utilize. This makes the method particularly appealing for networked virtual worlds and games, where customization and user choice are important for providing an experience that the user can actively engage with.

12 Acknowledgments

We thank Milton Garcia, Tracy McSheery, and others at PhaseSpace Inc. for generously allowing us to use their motion capture facilities and assisting us with acquisition and processing of motion data. Juan Batiz-Benet was our motion capture actor, Chris Platz created the character and scene art, Adam Watkins created the octopus gesture library, and numerous people lent their voices for testing and evaluation. We also thank all of our survey participants. This work was supported by NSF grants SES-0835601 and CCF-0641402.

References

ALBRECHT, I., HABER, J., AND PETER SEIDEL, H. 2002. Automatic generation of non-verbal facial expressions from speech. In *Computer Graphics International*, 283–293.

BAILENSON, J. N., BEALL, A. C., LOOMIS, J., BLASCOVICH, J., AND TURK, M. 2004. Transformed social interaction: Decoupling representation from behavior and form in collaborative virtual environments. *Presence: Teleoperators and Virtual Environments 13*, 4, 428–441.

BERTSEKAS, D. 2007. *Dynamic Programming and Optimal Control*, third ed. Athena Scientific.

BIRDWHISTELL, R. 1952. *Introduction to Kinesics*. Department of State Foreign Service Institute, Washington, DC.

BRAND, M. 1999. Voice puppetry. In *SIGGRAPH '99: ACM SIGGRAPH 1999 papers*, ACM, New York, NY, USA, 21–28.

BREGLER, C., COVELL, M., AND SLANEY, M. 1997. Video rewrite: driving visual speech with audio. In *SIGGRAPH '97: ACM SIGGRAPH 1997 Papers*, ACM, New York, NY, USA, 353–360.

CASSELL, J., PELACHAUD, C., BADLER, N., STEEDMAN, M., ACHORN, B., BECKET, T., DOUVILLE, B., PREVOST, S., AND STONE, M. 1994. Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *SIGGRAPH '94: ACM SIGGRAPH 1994 Papers*, ACM, New York, NY, USA, 413–420.

CASSELL, J., VILHJÁLMSSON, H. H., AND BICKMORE, T. 2001. Beat: the behavior expression animation toolkit. In *SIGGRAPH '01: ACM SIGGRAPH 2001 papers*, ACM, New York, NY, USA, 477–486.

CHUANG, E., AND BREGLER, C. 2005. Mood swings: expressive speech animation. *ACM Transactions on Graphics 24*, 2, 331–347.

DE MEIJER, M. 1989. The contribution of general features of body movement to the attribution of emotions. *Journal of Nonverbal Behavior 13*, 4, 247–268.

DENG, Z., AND NEUMANN, U. 2007. *Data-Driven 3D Facial Animation*. Springer-Verlag Press.

DOBROGAEV, S. M. 1931. Ucenie o reflekse v problemakh jazykovedenija. [Observations on reflex in aspects of language study.]. *Jazykovedenie i Materializm 2*, 105–173.

EFRON, D. 1972. *Gesture, Race and Culture*. The Hague: Mouton.

ENGLEBIENNE, G., COOTES, T., AND RATTRAY, M. 2007. A probabilistic model for generating realistic lip movements from speech. In *Neural Information Processing Systems (NIPS) 19*, MIT Press.

FEYEREISEN, P., AND DE LANNOY, J.-D. 1991. *Gestures and Speech: Psychological Investigations*. Cambridge University Press.

HARTMANN, B., MANCINI, M., AND PELACHAUD, C. 2002. Formational parameters and adaptive prototype instantiation for mpeg-4 compliant gesture synthesis. In *Proceedings on Computer Animation*, IEEE Computer Society, Washington, DC, USA, 111.

HARTMANN, B., MANCINI, M., AND PELACHAUD, C. 2005. Implementing expressive gesture synthesis for embodied conversational agents. In *Gesture Workshop*, Springer, 188–199.

KENDON, A. 2004. *Gesture – Visible Action as Utterance*. Cambridge University Press, New York, NY, USA.

KIPP, M., NEFF, M., AND ALBRECHT, I. 2007. An annotation scheme for conversational gestures: How to economically capture timing and form. *Language Resources and Evaluation 41*, 3-4, 325–339.

KOPP, S., AND WACHSMUTH, I. 2004. Synthesizing multimodal utterances for conversational agents: Research articles. *Computer Animation and Virtual Worlds 15*, 1, 39–52.

LAFFERTY, J. D., MCCALLUM, A., AND PEREIRA, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the 18th International Conference on Machine Learning*, Morgan Kaufmann Inc., 282–289.

LEVINE, S., THEOBALT, C., AND KOLTUN, V. 2009. Real-time prosody-driven synthesis of body language. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, ACM, New York, NY, USA.

LI, Y., AND SHUM, H.-Y. 2006. Learning dynamic audio-visual mapping with input-output hidden Markov models. *IEEE Transactions on Multimedia 8*, 3, 542–549.

LOEHR, D. 2007. Aspects of rhythm in gesture and speech. *Gesture 7*, 2, 179–214.

MCCANN, J., AND POLLARD, N. 2007. Responsive characters from motion fragments. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, ACM, New York, NY, USA.

MCNEILL, D. 1992. *Hand and Mind: What Gestures Reveal About Thought*. University Of Chicago Press.

MORENCY, L.-P., QUATTONI, A., AND DARRELL, T. 2007. Latent-dynamic discriminative models for continuous gesture recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 1–8.

NEFF, M., KIPP, M., ALBRECHT, I., AND SEIDEL, H.-P. 2008. Gesture modeling and animation based on a probabilistic recreation of speaker style. *ACM Transactions on Graphics 27*, 1, 1–24.

NEWLOVE, J. 1993. *Laban for Actors and Dancers*. Routledge Nick Hern Books, New York, NY, USA.

PERLIN, K., AND GOLDBERG, A. 1996. Improv: a system for scripting interactive actors in virtual worlds. In *SIGGRAPH '96: ACM SIGGRAPH 1996 Papers*, ACM, 205–216.

RABINER, L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 2, 257–286.

SARGIN, M. E., YEMEZ, Y., ERZIN, E., AND TEKALP, A. M. 2008. Analysis of head gesture and prosody patterns for prosody-driven head-gesture animation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 8, 1330–1345.

SHRÖDER, M. 2009. Expressive speech synthesis: Past, present, and possible futures. *Affective Information Processing*, 111–126.

STONE, M., DECARLO, D., OH, I., RODRIGUEZ, C., STERE, A., LEES, A., AND BREGLER, C. 2004. Speaking with hands: creating animated conversational characters from recordings of human performance. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, 506–513.

THE CMU SPHINX GROUP, 2007. Open source speech recognition engines.

TREUILLE, A., LEE, Y., AND POPOVIĆ, Z. 2007. Near-optimal character animation with continuous control. In *SIGGRAPH '07: ACM SIGGRAPH 2007 Papers*, ACM, New York, NY, USA.

VALBONESI, L., ANSARI, R., MCNEILL, D., QUEK, F., S. DUNCAN, K. E. M., AND BRYLL, R. 2002. Multimodal signal analysis of prosody and hand motion: Temporal correlation of speech and gestures. In *EUSIPCO '02*, vol. 1, 75–78.

WANG, S. B., QUATTONI, A., MORENCY, L.-P., DEMIRDJIAN, D., AND DARRELL, T. 2006. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition*, 1521–1527.

XUE, J., BORGSTROM, J., JIANG, J., BERNSTEIN, L., AND ALWAN, A. 2006. Acoustically-driven talking face synthesis using dynamic Bayesian networks. *IEEE International Conference on Multimedia and Expo*, 1165–1168.

ZHAO, L., AND BADLER, N. I. 2005. Acquiring and validating motion qualities from live limb gestures. *Graphical Models 67*, 1, 1–16.

Appendix A

This appendix presents the formulation of the value function recurrence used to train the Markov decision process described in Section 7.2. Recall that the MDP is only allowed to transition into a new segment (a', f') from (a, f) when $f' = 0$ and $f \in R_a \cup \{n_a\}$, where R_a is a small set of interruption points and n_a is the last frame of the segment a. Let $\ell_a(k)$ be the length of the k^{th} subsegment of a such that, if $R_a \cup \{n_a\} = \{f_1, f_2, ..., f_r\}$, we have

$\ell_a(1) = f_1, \ell_a(2) = f_2 - f_1, ..., \ell_a(r) = f_r - f_{r-1}$. Let the distribution $P_n(s_j|s_i)$ give the probability of arriving in state s_j from state s_i after n time steps. This distribution can be obtaining from the i^{th} column of T^n, the n^{th} power of the normalized state transition matrix of the CRF. The semantic cost $C_{\text{sem}}(a', w)$ of a candidate segment is discussed Section 8.1. For any $f_k \in R_a \cup \{n_a\}$, the simplified value function in the case that $a' \neq a$ or $f_k = n_a$ is

$$\begin{aligned} V_d(a', a, f_k, s_i, w) &= C(a', 0, a, f_k, e_i) + C_{\text{sem}}(a', w) \\ &+ \sum_{g=1}^{\ell_{a'}(1)} \sum_{s_j} \eta^g P_g(s_j|s_i)\delta_{a'}(j) \\ &+ \eta^{\ell_{a'}(1)} \min_{a''} \sum_{s_j} P_{\ell_{a'}(1)}(s_j|s_i) V_d(a'', a', f'_1, s_j, w), \end{aligned}$$

and in the case that $a' = a$ and $f_k \neq n_a$, $C(a, f_k{+}1, a, f_k, e_i) = 0$ and the value function reduces to

$$\begin{aligned} V_d(a, a, f_k, s_i, w) = &\sum_{g=1}^{\ell_a(k+1)} \sum_{s_j} \eta^g P_g(s_j|s_i)\delta_a(j) + C_{\text{sem}}(a, w) \\ &+ \eta^{\ell_a(k+1)} \min_{a''} \sum_{s_j} P_{\ell_a(k+1)}(s_j|s_i) V_d(a'', a, f_{k+1}, s_j, w). \end{aligned}$$

In the special case when a' is a semantically meaningful gesture, let f'_h be the next interruption frame in a' after $f'_{\text{start}}(a, f_k)$. The value function is then given by

$$\begin{aligned} V_d(a', a, f_k, s_i, w) &= C(a', f'_{\text{start}}(a, f_k), a, f_k, e_i) + C_s(a', w) \\ &+ \sum_{g=1}^{\ell_{a'}(h)} \sum_{s_j} \eta^g P_g(s_j|s_i)\delta_{a'}(j) \\ &+ \eta^{\ell_{a'}(h)} \min_{a''} \sum_{s_j} P_{\ell_{a'}(h)}(s_j|s_i) V_d(a'', a', f'_h, s_j, w'), \end{aligned}$$

where w' is the semantic label indicating the absence of a key word if a' satisfies the label w, and w otherwise. This causes the semantic label to be reset once a gesture is selected to satisfy it.