How to Train Your Avatar: A Data Driven Approach to Gesture Generation

Chung-Cheng Chiu and Stacy Marsella

University of Southern California
Institute for Creative Technologies
12015 Waterfront Drive
Playa Vista, CA 90094
{chiu,marsella}ict.usc.edu

Abstract. The ability to gesture is key to realizing virtual characters that can engage in face-to-face interaction with people. Many applications take an approach of predefining possible utterances of a virtual character and building all the gesture animations needed for those utterances. We can save effort on building a virtual human if we can construct a general gesture controller that will generate behavior for novel utterances. Because the dynamics of human gestures are related to the prosody of speech, in this work we propose a model to generate gestures based on prosody. We then assess the naturalness of the animations by comparing them against human gestures. The evaluation results were promising, human judgments show no significant difference between our generated gestures and human gestures and the generated gestures were judged as significantly better than real human gestures from a different utterance.

1 Introduction

A virtual human's non-verbal behavior is one of the main criterion that enriches the human-agent interaction. Users are sensitive to whether the gestures of a virtual human are consistent with its speech [9], and therefore a conversational virtual human should be animated based on its speech. One approach to achieve this is by using pre-defined human speech and creating specific motions for each sentence. Often in this case virtual human systems use hand-crafted animations or animations generated by capture technology [1,2]. However, neither of these methods scale well with the length of the dialogue and the effort required to generate new animations becomes significant.

Another approach is to use the text of the speech and construct mappings between features of the text and gestures. For example, [7] and [15] use the syntactic and semantic structure of the speech text along with additional domain knowledge and map them to various gestures. There is also work [18] that uses semi-automated data-driven approach that applies machine learning techniques on textual features and domain knowledge.

However, the aforementioned approaches do not consider the prosodic features in the verbal speech. In human conversations, the same speech spoken in different

H. Högni Vilhjálmsson et al. (Eds.): IVA 2011, LNAI 6895, pp. 127–140, 2011.

manners can express different meanings and much of this difference is conveyed through prosody. In addition, studies show that kinematic features of a speaker's gestures such as speed and acceleration are usually correlated with the prosody of the speech [23].

The goal of this work is to present an automated gesture generation process that maps features of speech to gestures, including prosodic features. There has been previous work that uses prosody information to generate gestures [6,19,17,16]. The approach taken in [17,16] selects animation segments from motion database based on audio input, and synthesizes these selected animations into the gesture animation. Since the approach uses existing motions, the gestures it produces are constrained by those that existed in the motion database.

In this work, we propose a gesture generator that produces conversation animations for virtual humans conditioned on prosody. The generator is built based on hierarchical factored conditional restricted Boltzmann machines (HFCRBMs) [8] with some modification. The model derives features describing human gesture and constrains animation generations to be within the gesture feature space. The method defines the role of prosody as a motion transition controller and learns this relation from the training data. The training data contains audio and motion capture data of people having conversations, and the model learns detailed dynamics of the gesture motions. After training, the gesture generator can be applied to generate animations with recorded speech for a virtual human. Our generator is not designed to learn all kinds of gestures but rather motions related to prosody like rhythmic movements. Gestures tied to semantic information like iconic gestures, pantomimes, deictic, and emblematic gestures are not considered in this work.

An evaluation of our approach with human subjects showed that the rating of animations generated by our learned generator from the audio of an utterance is similar to the original motion capture data and their difference is not statistically significant. Both cases are significantly better than using the motion capture data from a different utterance.

The contribution of this work is three-fold.

- We propose a model that learns speech-to-gesture generation.
- The model provides a way to derive features describing human gestures which helps gesture generations.
- Our gesture generator suggests that prosody provides information about motion movement that makes a prosody-based approach feasible for generating a subclass of arm-gestures.

The remainder of the paper is organized as follows. Section 2 contains a review of related works, Section 3 explains our gesture generator, and Section 4 presents the experimental results. The conclusion is summarized in Section 5.

2 Related Work

To generate gestures, BEAT [7] analyzes the syntactic relation between the surface text and gestures. The input text is parsed into a tree structure containing information such as clauses, themes/rhemes, objects, and actions. Using this information and a knowledge base containing additional information about the world, BEAT then maps them to a set of gestures. The Nonverbal Behavior Generator (NVBG) [15] extends this framework by making a clearer distinction between the communicative intent embedded in the surface text (e.g. affirmation, intensification, negation, etc.) and the realization of the gestures. This design allows NVBG to generate gestures even without a well-defined knowledge base. Stone et al. [20] proposed a framework to extract utterances and gesture motions from recorded human data then generate animations by synthesizing these utterances and motion segments. This framework includes an authoring mechanism to segment utterances and gesture motions then a selection mechanism to compose utterances and gestures. Neff et al. [18] made a comprehensive list of mappings between gesture types and their relations with semantic tags and derived the transition probability of motions from the sample data. The framework captures the details of human motion and preserves the gesture style of each performer, which can be generalized to generate gestures with various forms of input.

One major drawback of the text-based generation approaches [7,15] lies in the limited expressiveness. With this approach, it is not easy to represent detailed information of human motions with meta-description, especially with respect to the dynamics of joint movement. The same dialogue can be spoken with different speed and prosody for which the gesture motions have to be synchronized well with to make the behavior natural.

Audio-based motion generations has been addressed for manipulating facial expression [5], and similar approach has been extended to generate gestures [17]. A common idea of previous works is to collect a set of animation segments, define an objective function to model the relation between utterances and motions and the sequential relation among motions, and synthesize gesture animation via optimizing the objective function. The Hidden Markov models (HMM) fit this design, and it has been applied to generate head motion [6,19] and arm gestures [17]. The HMM-based approach directly associates motions with prosodic features and tends to overfit for learning arm gestures [17]. Thus, [16] proposed to combine conditional random fields (CRFs) with HMMs. HMMs first model the transitions of motions, and CRFs then learns the relation between utterances and hidden states of HMMs. The framework maps prosody to animations through CRFs and HMMs, and applies Markov Decision Processes to synthesize animations. Previous works generate gestures via synthesizing animation segments, so the generated animations are limited to animation segments in the motion database. Our system explicitly generates animation frame-by-frame and does not have this limitation.

3 Gesture Generator

The gesture generator takes past motion frames and generates the next motion frame conditioned on pitch, intensity and correlation audio features. The

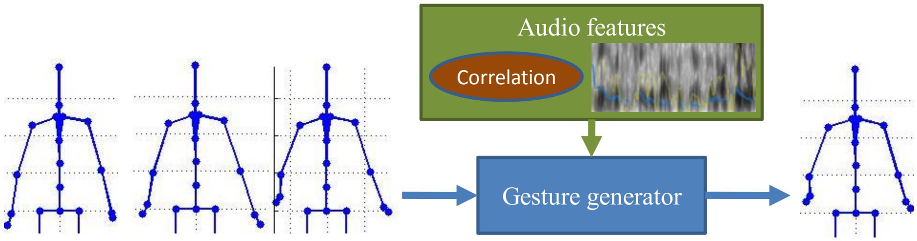


Fig. 1. The architecture of the generation process. Gesture generators take past motion frames and generate next motion frame conditioned on pitch, intensity, and correlation values.

prosodic features of pitch and intensity influence how the generator creates animation, and the correlation parameters indicate how strong that influence is and how much the gesture motion should be correlated with those prosody features. The correlation parameters provide a handle for users to tune the motion of virtual human: the higher the correlation parameters, the more the velocity and acceleration of motions will be correlated with pitch and intensity. On generating the next motion frame, gesture generator not only takes into account the audio feature of current time but also audio features of previous and future time frames. The architecture of entire framework is shown in Fig. 1.

Our framework does not incorporate semantic or syntactic information of the accompanying speech. As we noted in the introduction, semantic content as well as prosody of the utterance correlates with human gestures. Gestures like iconic or deictic gestures carry specific semantic content of the dialogue like abstract depiction of actions, object, or orientations. The space of these kinds of semantic content is large, and therefore a gesture generator requires a rich set of knowledge to map general utterances to gestures. However, our current dataset is small comparing to the entire space of semantic content, and the knowledge for mapping these kinds of semantic content to gestures is sparse. Thus, in our current work we excluded the mapping between semantic content and gestures but limited our focus to prosody-based gesture generation, in which the gestures we address is similar to the idea of motor gesture [13]. Prosody and motion correspond to emphasis, and both of them can exhibit the emotional state of the speaker. We explore the capability of prosody for gesture generation in this work and take semantic content of utterances as an important channel for future extension.

We use motion capture and audio data of human conversations to learn the gesture generator's mapping of prosody to gesture animations. This data must first be processed before it can be used for training. Specifically, we defined a criterion to extract motion segments containing gestures. We analyzed motion data to identify gesture motions and non-gesture motions, and determine what y-coordinate value of wrists best separates these two sets. Motion frames having at least one wrist's y-coordinate higher than this value are defined as gestures. This rule is then applied to extract gesture motions. Among valid motion frames, only the animation segments with length longer than 2 seconds are kept. After the valid motion frames are identified, the corresponding audio features are extracted. The time constraint on data selection will exclude gestures with short period of time. The rationale for defining this constraint is that in our data analysis most of gestures performed in less than 2 seconds are often either iconic gestures or gestures unrelated to utterances, and neither cases are gestures we want to learn. In the motion capture data most of gestures stay for a long period of time.

The extraction process identifies the data containing gestures, and there are two cases that we need to manually exclude from the training data. The first case is the semantic-related gestures. Our current gesture generators use prosody features for motion generation, and prosody features do not preserve the semantic content of the dialogue. Therefore, any semantic-based gestures will mislead the model training and have to be excluded from the data. The second case is non-gesture motion data. Sometimes actors are adjusting their motion capture suit, scratching, or performing an initialization posture which is required for motion capture calibration. We analyzed the extracted motions and excluded these two cases to get final data set.

3.1 Requirements for Building Gesture Generators

A common approach of previous work is to generate gesture motions via synthesizing existing motion segments [17,16]. The gesture generator we are building is a generative function that takes previous motion frames and audio features as input and then outputs motion frames. In other words, the gesture generator learns the relation between previous motion frames, audio features, and the current motion frame, and uses this relation to generate animations. The benefit of this design is that the gesture generator can generalize better to novel speech and create new gestures, in contrast to the previous approach which is limited to existing motion segments. The potential problem of this design is that it runs the risk of generating unnatural gestures. The domain of motion frames is *joint rotation*, and if the generative function learns the motion generation with this domain, then the output of the function is too unconstrained – it can be any joint rotation.

Thus, on building a generative function for gesture generator, a key challenge is to prevent the generation of unnatural gestures. Human gestures move only within certain space and contain certain patterns, so instead of learning the function within an unconstrained space of *joint rotation*, it will be more effective to learn gesture generation in the constrained *gesture motion*. For this reason, our gesture generator detects features of gesture motion, represents gestures in terms of motion features, and learns gesture generation with this new representation system. With the domain constrained in gesture feature space, gesture generators have a better chance of producing natural motions.

Another challenge comes from the mapping between audio features and motion frames. Both an audio feature vector and a motion frame are real value vectors with high dimension, and the space of possible values is large. A gesture generation is a mapping between two sequences of these vectors, and the underlying relation is complex. This brings a requirement for a gesture generator to learn a function that captures this complex relation.

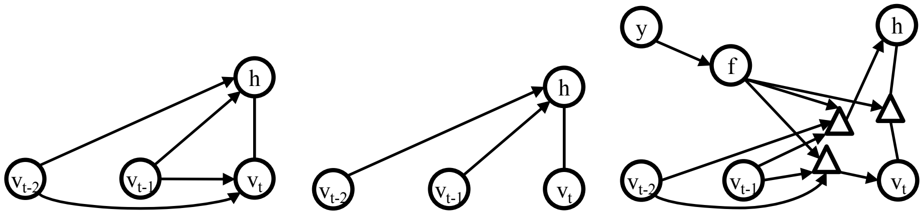
In sum, there are three things a gesture generator has to be capable of learning: gesture motion features, temporal relation of gesture motions, and the relation between audio features and gesture motions. Hierarchical factored conditional restricted Boltzmann machines (HFCRBMs) [8] is a learning model that matches this criterion. It is an extension of deep belief networks (DBNs) [11] that can deal with sequential data. We applied some modification to HFCRBMs to build the gesture generator. Following sections introduce DBNs and components of modified HFCRBMs.

3.2 Background of Modified HFCRBM

The DBN stacks multiple layers of Restricted Boltzmann Machines (RBMs) to construct a multi-level neural network. A multi-level neural network is known to be able to represent a complex function, but it usually suffers from long convergence time and getting trapped into local optimum easily. The way DBN builds neural networks can significantly reduce the convergence time and improve the performance.

The major philosophy behind the design of DBN is to learn a better feature representation for the task. In most of cases, the original representation of the data is not the best way to describe the data and we will prefer to define some features to represent the data. For example, in the object detection task we may prefer to represent an image with edge features than a pixel vector. The DBN applies an unsupervised learning algorithm to initialize its network connection, and the algorithm performs a feature detection process. With the unsupervised learning process, each layer of a DBN acts as a feature detector, and the DBN uses this hierarchical structure to learn features that can model the input-output relation of the given task. The DBN was proposed to learn static data in which the sequential information is not considered in the model. The HFCRBM extends the DBN to consider sequential information. Following sections describe components related to HFCRBMs and modified HFCRBMs: CRBMs, RCRBMs, and FCRBMs.

CRBMs and RCRBMs. The conditional Restricted Boltzmann Machine (CRBM) [22], as shown in Fig. 2a, is a artificial neural network with a binary hidden layer and multiple visible layers taking time-series data as input. The network between the hidden layer h and the visible layers v is a complete bipartite graph in which links between h and v_t is similar to a Hopfield net [12]. The structure differs from Hopfield net in that there are no links between visible nodes and they are connected indirectly through hidden nodes. Both h and v_t have directed links from visible layers for past visible data $v_{t-1}, v_{t-2}, \ldots, v_{t-n}$ where n denotes the order of the model. This network takes past data $x_{t-n} \ldots x_{t-1}$ as input for $v_{t-n} \ldots v_{t-1}$ and output x_t at v_t, and uses the output error to update the connection weights. After training connection weights with time-series data,



(a) CRBM (b) RCRBM (c) Factored CRBM

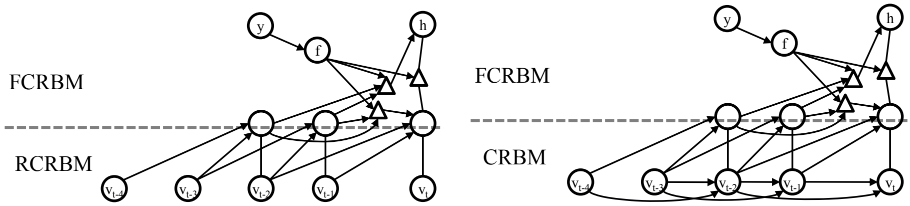
Fig. 2. (a) A CRBM with order 2 where vs denote visible layers, t represent index in time, and h is the hidden layer. (b) A RCRBM with order 2. (c) A FCRBM with order 2 where f denotes feature layer, y represent labels, and triangles represent factored multiplications.

the model can predict future data with given data sequence. We can take current output data x_t to form a new input sequence $x_{t-n+1} \ldots x_t$ for the model and generate next data x_{t+1}. By doing so, the model can iteratively generate a long sequence of data based on a short initial sequence. Reduced CRBMs (RCRBMs), as shown in Fig. 2b, are CRBMs without the lateral links between visible layers. RCRBMs generate data sequence with the same process of CRBMs, but since there are no lateral links between visible layers the output of v_t depends only on the links with the hidden layer.

Factored Conditional Restricted Boltzmann Machines. CRBMs capture the transition dynamic of the time series data in an unsupervised way. They generate data sequence based on only the information of past visible data. In some applications, we would like to use annotation information to help recognition and generation. Taylor & Hinton proposed factored conditional restricted Boltzmann machine with contextual multiplicative interaction (we will simply call it FCRBM in the following text for clarity) which extends CRBMs to output data conditioned on annotated information [21]. The architecture of the FCRBM is shown in Fig. 2c. The FCRBM preserves the original structure of the CRBM, and adds additional input layer for annotated information, the label layer. One major difference in structure is that there are no direct links between layers, and they connect indirectly through factor nodes. All factor nodes have directed connections from the label layer, and through the label layer the annotated information play the role of gating values propagating within network. In this design, FCRBMs output current data based on given data sequence conditioned on annotated information, and update connection weights based on output error.

3.3 Modified HFCRBM

The modified HFCRBMs stacks FCRBMs on top of CRBMs to formulate a temporal model based on features identified by CRBMs. This is different from the original HFCRBM which stacks FCRBMs on top of RCRBMs. The architecture of the two models are shown in Fig. 3. The original HFCRBM was applied to



(a) HFCRBM

(b) Modified HFCRBM

Fig. 3. The difference between the original HFCRBM and our modification lies in the bottom level model

learn walking motions with different styles [8]. The labels used in that work are style labels. To generate a motion with specific style the generation process has to be manipulated using style labels, so RCRBMs are necessary. On the other hand, in our case the gesture generation does not need to be completely manipulated with respect to labels (audio features) and can depend more on past visible data, namely previous gesture motions. Thus, the lateral links in CRBMs are beneficial for gesture generators, and we replace RCRBMs with CRBMs in HFCRBMs.

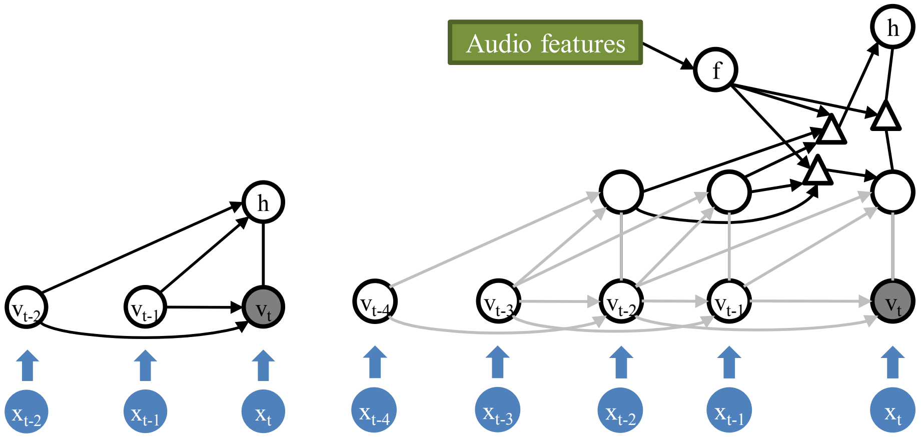
The modified HFCRBM satisfies our three requirements for gesture generation in that the bottom CRBM learns the features of gesture motions, and the top FCRBM learns the temporal relation of gestures and its relation between audio features. In the following paragraphs we will call the modified HFCRBMs as HFCRBMs for simplicity.

We have also added a sparse coding criterion to the unsupervised learning (training CRBMs) step because in our initial investigations it further improves the accuracy of gesture generation. Different from [14] as they only update the bias term of the hidden layer for encouraging sparsity, we also update the connection weight of CRBMs. The objective function regarding the sparsity term is expressed with cross entropy between the desired and actual distributions of the hidden layer as described in [10].

3.4 Training and Generation

The training process of our gesture generator is shown in Fig 4. The gesture generator first performs unsupervised learning on motion data to identify features which better represent the temporal pattern of human motion. After these features are identified with CRBMs, HFCRBMs represent motion data using these features, and take data sequences represented with new features to train top-layer FCRBMs. FCRBMs take data sequence as input and learn to generate data at the next time frame conditioned on audio features. After training FCRBMs, HFCRBMs learn to generate gesture motion based on audio features.

The HFCRBM-based gesture generator requires two input: an initial gesture motion and a sequence of audio features. The initial gesture motion is a required input for the visible layer v of HFCRBMs. A designer can specify what the initial



(a) Training CRBMs.

(b) Training modified HFCRBMs.

Fig. 4. The training process of modified HFCRBMs for gesture generators. (a) The CRBMs of the modified HFCRBM is trained with motion sequence x to find feature representation. The gray-filled node represents output node of the model, and the connection weight is updated based on the output error compared with the training data. (b) After training CRBMs, the data goes bottom-up to train top layer FCRBMs. The network generate output conditioned on audio features, and update connection weights based on prediction error. In this step the links of CRBMs as shown in light gray color are fixed, and only the connection weights of FCRBMs are updated.

gesture of an avatar is preferred through this motion sequence, or can set them to all zero vectors for simplicity. The sequence of audio features are data extracted from given utterance, and is the input to the label layer of HFCRBMs. The HFCRBM takes its output as part of the input data of next generation step, and the entire generation process will output a sequence of motion with the same length as the audio data. The resulting data is the gesture motion for given utterances.

3.5 Smoothing

The motion sequence generated by HFCRBMs can contain some noise and the difference between frames may be greater than natural gestures. Although each motion frame is still a natural gesture and this kind of noise is rare in the output, users are sensitive to the discontinuity of the animation and a short unnatural motion can ruin the entire animation. Therefore, after gesture motions are generated, an additional smoothing process is performed on the output result.

The smoothing process computes the wrist position of each generated frame, and calculate the acceleration of wrist movement. If the wrist acceleration of one motion frame exceeds some threshold, we reduce the acceleration via modifying the joint rotation velocity of that motion frame to be closer to the velocity of previous joint rotations. The new motion frame at time t is computed by:

```
$r = 0.2$
$x'_t = x_t$
**while** (wrist acceleration of $x'_t$) $>$ threshold and $r < 1$ **do**
    $x' = (1 - r)(x_t - 2x'_{t-1} + x'_{t-2}) + 2x'_{t-1} - x'_{t-2}$
    $r+ = 0.1$
**end while**
```

where x' is smoothed motion frame, and x is original output motion frame. The threshold is chosen as the maximum wrist acceleration value of the human gesturing motion observed from the motion capture data. The equation inside the while-loop is adjusting the velocity of current frame to an interpolation between the original velocity and the velocity of its previous frame, and x'_t is the resulting new motion frame corresponding to the smoothed velocity. The smoothness criterion, wrist acceleration, is computed based on the *translation* of the wrist joints, while the values x within update equation are values of joint *rotations*. For a motion frame at time t that does not exceed the acceleration threshold, it is smoothed as:

$$x'_t = 0.8 \cdot x_t + (x'_{t-1} + x_{t+1})/15 + (x'_{t-2} + x_{t+2})/30$$

4 Experiments

We evaluated the quality of our generated gestures by comparing the gestures it generates for utterances with the original motion capture for those utterances as well as using motion capture from different utterances. In the experiment, our data is the dataset used for the study of human sensitivity for conversational virtual human [9]. The dataset contains audio and motion of groups of people having conversations. There are two groups in the dataset, male and female group, and each group has three people. There are two types of conversations, debate and dominant speaker, and each type has five topics. We used the debate conversation data of male group for experiments. The motion capture data contains the skeleton of subjects and the recorded joints movement are a vector with 69 degree of freedom. Since this work focus mainly on arm gestures, we removed leg, spine, and head movement from the data. Elements containing all zeros in the joint rotation vectors are also removed. After removing these elements, the resulting joint rotation vector has 21 degree of freedom.

We extracted pitch and intensity values from audio using Praat [4]. The values of pitch and intensity ranged from zero to hundreds. To normalize pitch and intensity values to help model learning, the pitch values are adjusted via taking $log(x + 1) - 4$ and setting negative values to zero, and the intensity values are adjusted via taking $log(x) - 3$. The new pitch values range 0 to 2.4, and the new intensity values range 0 to 1.4. The log normalization process also correspond to human's log perception property.

In the training of modified HFCRBMs, both the hidden layer of CRBMs and FCRBMs have 300 nodes. The correlation parameters of each time frame is computed as the correlation of prosody sequence and motion sequence with a

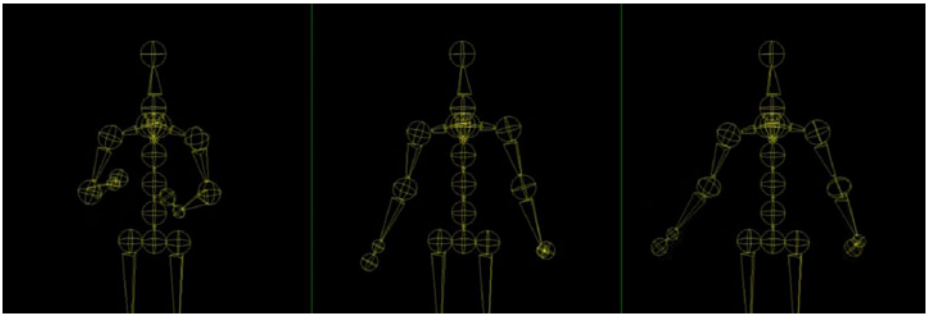


Fig. 5. The video we used for evaluation. We use a simple skeleton to demonstrate the gesture instead of mapping the animation to a virtual human to prevent other factors that can distract participants.

window of ±1/6 seconds. The audio features for gesture generators at each time frame also has a window of ±1/6 seconds. We trained the model with the audio and motion capture data of one actor, and use the other actor's speech as a test case to generate a set of gesture animations. We applied the criterion described in Section 3 to extract training and testing data, and there are total 1140 frames (38 seconds) of training data and 1591 frames (53 seconds) of testing data. Since testing data does not have correlation values, we sample correlation parameters from training data to simulate a pseudo-random values. We applied the same criterion as for training data to extract prosody and pseudo-random correlation to compose audio features for testing data.

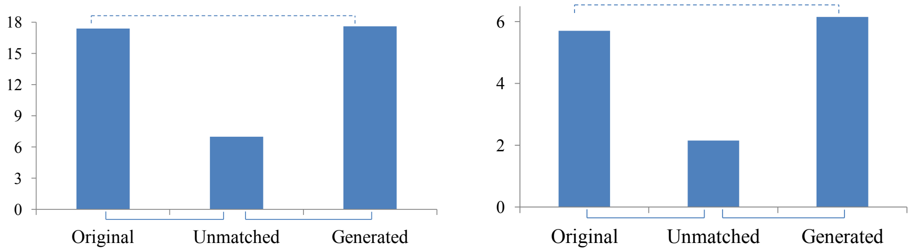
4.1 Evaluation

We used our gesture generator to generate gesture animations with testing data, and to evaluate the quality of generated animations, we compare them with two gesture animations:

- The original motion capture data of the testing data.
- The motion capture data of the test case actor with respect to other utterance.

For the second case, we used the same extraction techniques described in Section 3 to derive motion capture and audio data. We hypothesize that the gesture animations generated by our model will be significantly better than the motion capture data from different utterances, and the difference between generated animation and actual human gestures will not be significant.

We displayed the three animations side-by-side, segmented them to the same length, and rendered them into videos accompanied with original audio. One example frame of our video is shown in Fig. 5. There are total of 14 clips with length 2 to 7 seconds. The relative horizontal position of Original, Generated, and Unmatched cases is different and balanced between clips (e.g. Original is on the left 5 times, middle 5 times, and right 4 times). The presentation of



(a) Total score

(b) Rated first

Fig. 6. Average rating for gesture animations. Dashed lines indicate two sets are not significantly different, solid lines indicate two sets are significantly different.

clips to participants was randomized. An example video can be found in [3]. In this example video, the left one is the motion capture data with respect to different utterance, the middle one is the original gesture, and the right one is the generated gestures. Since the motion capture data with respect to different utterances is real human motion, participants can not tell the difference simply based on whether the motion is natural; they have to match the motion with speech to do the evaluation. We recruited 20 participants with ages ranging from around 25 to 55. All participants are familiar with computer animations, and some of them are animators for virtual human or experts on human gestures. We asked participants to rank which gesture animation in the video best matches the speech.

We performed balanced one-way ANOVA on the ranking results and the analysis result suggests that at least one sample is significantly different than the other two. We then applied Student t-test to test our specific hypotheses. The evaluation results are shown in Fig. 6. On the number of being ranked first, the difference between the original gesture motion and the generated gesture motion is not significant, and both them are significantly better than the unmatched gesture motion. We applied another study via assigning 2 point for ranked-first cases and 1 point for ranked-second cases, and calculated the overall score of each motion. Hypothesis testing show that the generated motion is not different from the original motion, and they are both significantly better than the unmatched gestures. This result implies that the movement of generated gesture animations are natural, and the dynamics of motions are consistent with utterances.

5 Conclusions

We have proposed a method for learning prosody-based motion generators for virtual human with recorded speech. Specifically, we modified HFCRBMs to build a model that learns the temporal relation of human gesture and the relation between prosody and motion dynamics. The model is trained with motion capture and audio data of human conversations to formulate an audio-based gesture generator. Gesture generators learned to generate motion frames based on previous gesture motions and prosody information, and are applied to produce gesture animations using another set of speech audio. Evaluation results showed that the produced gestures were significantly better than using human gestures corresponding to different utterances, and there was no significant difference between produced animation and actual human gestures used in the conversations. This work lays a foundation toward building a comprehensive gesture generator. The next step is to explore speech information other than prosody and include other categories of gestures like iconic and deictic gestures to improve the gesture generator.

Acknowledgements. This research was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM) Simulation Training and Technology Center (STTC). The content or information presented does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

1. `http://ict.usc.edu/projects/gunslinger`
2. `http://ict.usc.edu/projects/responsive_virtual_human_museum_guides/`
3. `http://www.youtube.com/watch?v=OsZ0RI9JH60`
4. Boersma, P.: Praat, a system for doing phonetics by computer. Glot International 5, 341–345 (2001)
5. Brand, M.: Voice puppetry. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999, pp. 21–28. ACM Press, New York (1999)
6. Busso, C., Deng, Z., Grimm, M., Neumann, U., Narayanan, S.: Rigid head motion in expressive speech animation: Analysis and synthesis. IEEE Transactions on Audio, Speech, and Language Processing 15(3), 1075–1086 (2007)
7. Cassell, J., Vilhjálmsson, H.H., Bickmore, T.: Beat: the behavior expression animation toolkit. In: SIGGRAPH 2001: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 477–486. ACM, New York (2001)
8. Chiu, C.C., Marsella, S.: A style controller for generating virtual human behaviors. In: Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2011, vol. 1 (2011)
9. Ennis, C., McDonnell, R., O'Sullivan, C.: Seeing is believing: body motion dominates in multisensory conversations. In: ACM SIGGRAPH 2010 papers, SIGGRAPH 2010, pp. 91:1–91:9. ACM, New York (2010)
10. Hinton, G.: A practical guide to training restricted boltzmann machines. UTML TR 2010003, Department of Computer Science, University of Toronto (August 2010)
11. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Comput. 18(7), 1527–1554 (2006)
12. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79(8), 2554–2558 (1982)

13. Krauss, R.M., Chen, Y., Gottesman, R.F.: Lexical gestures and lexical access: a process model. In: McNeill, D. (ed.) Language and Gesture. Cambridge University Press, Cambridge (2000)
14. Lee, H., Ekanadham, C., Ng, A.: Sparse deep belief net model for visual area v2. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in Neural Information Processing Systems, vol. 20, pp. 873–880. MIT Press, Cambridge (2008)
15. Lee, J., Marsella, S.C.: Nonverbal behavior generator for embodied conversational agents. In: Gratch, J., Young, M., Aylett, R.S., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 243–255. Springer, Heidelberg (2006)
16. Levine, S., Krähenbühl, P., Thrun, S., Koltun, V.: Gesture controllers. In: ACM SIGGRAPH 2010 papers, SIGGRAPH 2010, pp. 124:1–124:11. ACM, New York (2010)
17. Levine, S., Theobalt, C., Koltun, V.: Real-time prosody-driven synthesis of body language. ACM Trans. Graph 28, 172:1–172:10 (2009), `http://doi.acm.org/10.1145/1618452.1618518`
18. Neff, M., Kipp, M., Albrecht, I., Seidel, H.-P.: Gesture modeling and animation based on a probabilistic re-creation of speaker style. ACM Trans. Graph 27(1), 1–24 (2008)
19. Sargin, M.E., Yemez, Y., Erzin, E., Tekalp, A.M.: 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 (2008)
20. Stone, M., DeCarlo, D., Oh, I., Rodriguez, C., Stere, A., Lees, A., Bregler, C.: Speaking with hands: creating animated conversational characters from recordings of human performance. In: SIGGRAPH 2004: ACM SIGGRAPH 2004 Papers, pp. 506–513. ACM, New York (2004)
21. Taylor, G., Hinton, G.: Factored conditional restricted Boltzmann machines for modeling motion style. In: Bottou, L., Littman, M. (eds.) Proceedings of the 26th International Conference on Machine Learning, pp. 1025–1032. Omnipress, Montreal (2009)
22. Taylor, G.W., Hinton, G.E., Roweis, S.T.: Modeling human motion using binary latent variables. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems, vol. 19, pp. 1345–1352. MIT Press, Cambridge (2007)
23. Valbonesi, L., Ansari, R., McNeill, D., Quek, F., Duncan, S., McCullough, K.E., Bryll, R.: Multimodal signal analysis of prosody and hand motion: Temporal correlation of speech and gestures. In: Proc. of the European Signal Processing Conference, EUSIPCO 2002, pp. 75–78 (2002)