Gesture Generation with Low-Dimensional Embeddings

Chung-Cheng Chiu
USC Institute for Creative Technologies
12015 Waterfront Drive
Playa Vista, CA 90094
chiu@ict.usc.edu

Stacy Marsella
Northeastern University
360 Huntington Ave
Boston, MA 02115
stacymarsella@gmail.com

ABSTRACT

There is a growing demand for embodied agents capable of engaging in face-to-face dialog using the same verbal and nonverbal behavior that people use. The focus of our work is generating coverbal hand gestures for these agents, gestures coupled to the content and timing of speech. A common approach to achieve this is to use motion capture of an actor or hand-crafted animations for each utterance. An alternative machine learning approach that saves development effort is to learn a general gesture controller that can generate behavior for novel utterances. However learning a direct mapping from speech to gesture movement faces the complexity of inferring the relation between the two time series of speech and gesture motion. We present a novel machine learning approach that decomposes the overall learning problem into learning two mappings: from speech to a gestural annotation and from gestural annotation to gesture motion. The combined model learns to synthesize natural gesture animation from speech audio. We assess the quality of generated animations by comparing them with the result generated by a previous approach that learns a direct mapping. Results from a human subject study show that our framework is perceived to be significantly better.

Categories and Subject Descriptors

I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*

Keywords

Gesture Controller, Gaussian Process Latent Variable Model, Virtual Agent, Animation, Motion Capture

1. INTRODUCTION

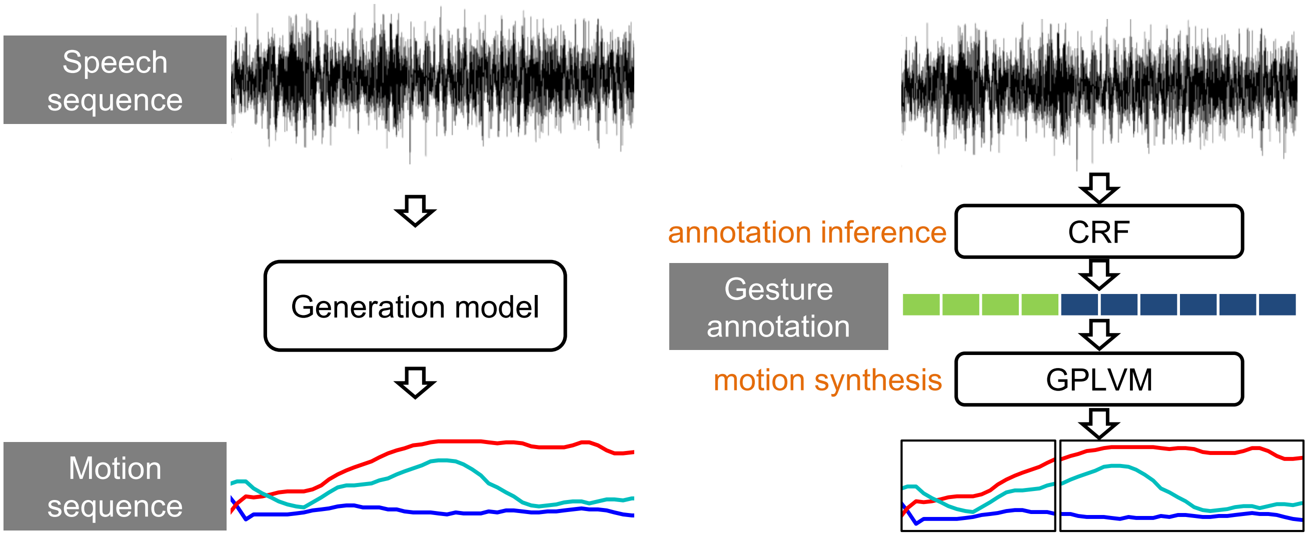
There is a growing demand for embodied agents and animated characters capable of simulating face-to-face dialog interaction using the same verbal and nonverbal behavior that people use. The focus of the work presented here is on providing characters with a capacity to use coverbal hand gestures, gestures performed in close synchrony with the content and timing of the speech.

This capability can be achieved by using motion capture for each utterance or recording speech audio and manually crafting gesture motions for each sentence the character utters. However, those approaches are time-consuming/costly, do not scale well for projects with large numbers of utterances, lead to results applicable just to a single project, and cannot even be used in projects that use open-ended dialog generation techniques. An alternative, general way is to build a gesture generator that can compose gestures for novel utterances.

One way to build such a gesture generator is to learn a model for composing gestures based on speech from human conversation data. A learning approach saves effort on manually determining the model. Additionally it also can derive detailed correlations between speech and motion useful for capturing personal styles or motion dynamics that are challenging to specify manually. One approach to this is to learn a direct mapping between the two time series, speech and gestures [4]. However, the resulting learning problem is fundamentally difficult because of (a) the overall complexity of learning a mapping from the speech signal to the high dimensional gesture motion, (b) the many-to-many nature of the mapping whereby motions with quite different dynamics can convey similar meaning, while motions with similar dynamics can have different meanings and (c) the two time series of speech and gesture do not necessary correlate directly but rather tend to correlate through their respective meanings.

To make the task feasible, we develop a novel approach that decomposes the learning problem into processes of mapping from speech to annotations and then synthesizing gesture motion for the given annotations. Specifically, the speech-annotation mapping is learned using conditional random fields (CRFs) [12] while the motion synthesis uses Gaussian process latent variable models (GPLVMs) [13] to learn a low-dimensional space (manifold) that encodes natural gesture motion. An example of an annotation is the communicative role of the gesture (what it serves to convey). The segmentation and annotation processes represent the original complex and continuous motion data with a small set of discrete annotations thus reducing the complexity of the learning problem. The structural difference between the direct mapping approach and our proposed framework is shown in Figure 1. An additional benefit of this decomposed approach is that we can use different data sets for the two learning tasks. For example, we might use an across subject data set to learn a more universal model of the mapping from speech to annotation while data from a particular individual may be used for

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*



(a) Structure of a direct mapping approach

(b) Structure of our decomposition approach

Figure 1: Decomposing the mapping process. (a) The direct mapping approach learns a complex model to generate gesture animations directly from speech sequences. (b) Our approach decomposes the task into speech-annotation inference (with CRFs) and motion synthesis (with GPLVMs). CRFs infer gesture annotations from input speech signals, and GPLVMs synthesize gesture animations with the inferred gesture annotations.

the motion synthesis process to learned the specific motion style of that person's gestures.

The decomposition reduces the complexity of the learning problem, but there is still an issue of smoothly transitioning between gestures, namely, gesture co-articulations. One approach to gesture co-articulations restricts it to only combine pre-existing motion segments that can transition smoothly [11]. This greatly reduces the quality of the gesture animation, as generating animations for novel utterances in general requires novel sequential compositions of gestures, and this often involves connecting two motion segments with very distinct motion dynamics. Not supporting such novel combinations restricts the expressivity of the character, while supporting it requires some means to realize smooth transitions between various motion segments.

We address the co-articulation issue by applying GPLVMs to learn a low-dimensional space which captures human motion dynamics, thereby providing a means to generate natural, novel transitions between gesture motions. Specifically, we apply GPLVMs with an additional dynamic term in the objective function (which is also known as GPDMs, Gaussian process dynamic models [25]) to derive the dynamic constrain of human motion and the respective manifold. The process of gesture generation embeds gesture motions into the learned manifold and determines the trajectory for the transition between the gestures, and the trajectory is then mapped back to the original high dimensional motion space to generate the composed gesture animations. Details are discussed in subsequent sections.

Although our framework is designed to learn to generate gesture animations from general speech features such as linguistic and acoustic features, in current work we assess our framework by using only acoustic features. Using only acoustic features allows our framework to run in real-time and also helps focus the assessment on the co-articulation capability. Acoustic features are known to be coupled with the dynamics of the arm movement and to correlate with a subset of gestures called beat gestures [24], which are the majority gestures in our conversation data. Thus, we assess our framework by focusing on realizing the relation between acoustic features and beat gestures.

We compared the result of our framework with another learning approach [4] which frames the learning problem as a direct mapping from speech to gesture motion. The evaluation asked participants, using Mechanical Turk, to vote which animation matched the speech best. Our framework was judged overwhelmingly to be superior to the direct mapping approach.

2. RELATED WORK

There have been a range of work work focusing on analyzing the relation between prosodic features of the speech and motion movement [16, 15, 4]. Inspired by speech synthesis work, [16] applied Hidden Markov models (HMMs) to realize the relation. The same idea has also been exploited in terms of modeling head movement [2, 19]. However, [15] pointed out that applying HMMs to directly associate arm movement with prosodic features tends to overfit and therefore they proposed to combine conditional random fields (CRFs) with HMMs. Both works [16, 15] synthesize gestures by only considering gesture motions that can smoothly connect with prior motion, and therefore can run the risk of failing to choose gesture motion that match the speech better. [4] proposed a model which is extended from deep belief nets [9, 22] to learn the mapping from prosody features to motion frames explicitly, and [5] suggested a way to optimize that model with human subjective opinion. The approach resolves the smoothness constraint, but learning a direct mapping from prosodic features to frame-by-frame motion is challenging. In addition to focusing on the relation between prosodic features of the speech and motion movement, there are also data-driven approaches [21, 10] that take an approach in

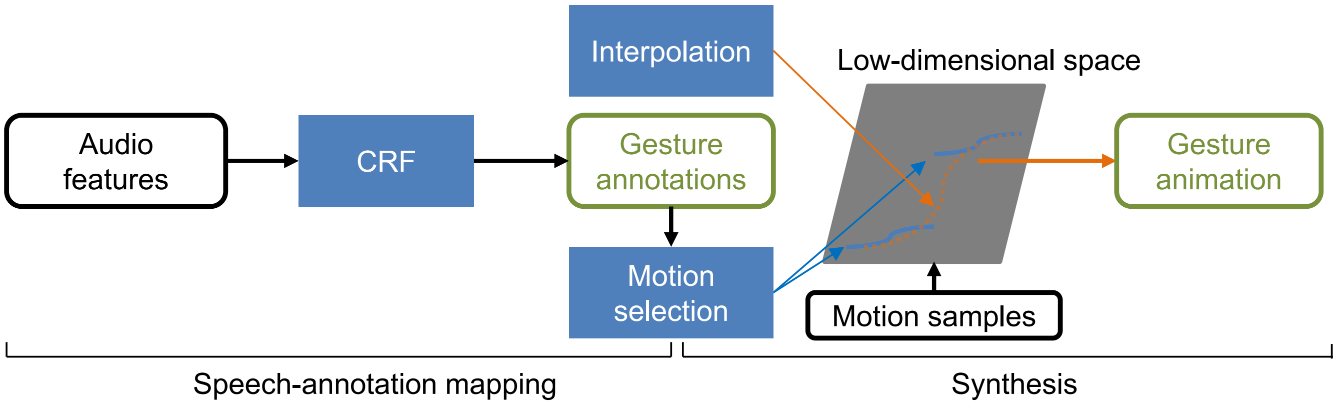


Figure 2: The flow of the gesture generation framework. The CRF determines a sequence of gesture annotations from audio features, and the synthesis process selects corresponding gesture motion from the low-dimensional space. The resulting sequences are in general discontinuous, and the interpolation algorithm infers a trajectory to integrate these sequences. The resulting trajectory is then mapped to gesture animations. The low-dimensional space is derived by embedding motion samples.

which speech and gestures are co-generated.

The other approach to gesture generation is to adopt rule-based systems which exploit the domain knowledge about gestures and meanings. For example, [3] selects gestures based on the linguistic features of the surface text. The Nonverbal Behavior Generator (NVBG) [14] takes a similar approach and extends the framework by addressing more of the communicative functions of the dialog. [18] takes a further step by profiling the gesture style of individual speakers, using the result to determine gestures from speech. This approach allows the system to generate gestures highly correlated with the content of the dialogue. Instead of regarding it as an alternative approach, there is a potential to integrate these text-gesture mapping processes into our framework to enrich the speech-gesture mapping process. Since our synthesis process is designed to be compatible with general gesture annotations, the same synthesis algorithm will be applicable for the new integrated system.

GPLVMs have shown success in modeling various human motion such as walking, golf swings, punching, and kicking [7, 13, 25, 17]. Our work is the first to realizing human gestures and co-articulations.

3. GESTURE GENERATION

A gesture is composed of a sequence of continuous movement, and its correlation with speech is hard to realize when looking at individual frames. A plausible perspective is to analyze motion sequences to determine the correlation. Our proposed framework adopts this perspective in which it first infers gestures from speech signals, and then synthesizes motion with the specified gesture annotations. The framework decomposes the original task into two processes which allows better inference quality from speech to gestures but also introduce a new task: the capability of gesture co-articulation.

The framework addresses this problem by including an interpolation process that allows transitions between discontiguous motion segments. The approach interprets the problem of gesture transition as an interpolation task, and improves the effectiveness of the interpolation by projecting the original data space onto a low-dimensional space which better represents the gesture motion. A detailed flow for the generation process is shown in Figure 2. Specifically, the framework applies Gaussian process latent variable models (GPLVMs) to learn the low-dimensional space (manifold). We give more detailed explanation about this design in the following sections.

3.1 Interpolation as motion transition

One of the central challenges in transitions between gestures is that gestures are in a high dimensional space which can impede conventional motion transition approaches. Algorithms used by conventional motion transition approaches commonly blend using weighted average of motion frames drawn from the two motion segments. Viewing each motion frame as a data point and a motion segment is a sequence of data points, the motion transitioning between the two motion segments can then be understood as interpolating a trajectory that connects the two sequences. There are two issues in this approach that can lead the interpolation algorithm to generate animations that are not natural human motion. First, different pairs of gesture sequences require different lengths of transition motions to retain natural movement, and it is crucial for an interpolation algorithm to be able to infer the appropriate length from the spatial distance. In other words, the space in which an interpolation algorithm infers transition trajectory needs to reflect the transition distance in terms of the spatial distance for any pairs of gesture sequences. Closeness in the original data space, however, does not necessary guarantee a smooth transition. Second, the interpolation algorithm generates the trajectory without considering the constraint of human motion dynamics so the resulting trajectories can lead the animation to have unnatural movement.

The two issues can be resolved by learning a manifold which better represents the similarity of postures and motion dynamics with the spatial distance. An interpolation algorithm finds a smooth trajectory connecting the two specified sequences, and as the manifold realizes the relation among gestures and motion dynamics in terms of the spatial relation, a smooth trajectory in the manifold can correspond to a natural gesture motion. Thus, instead of performing

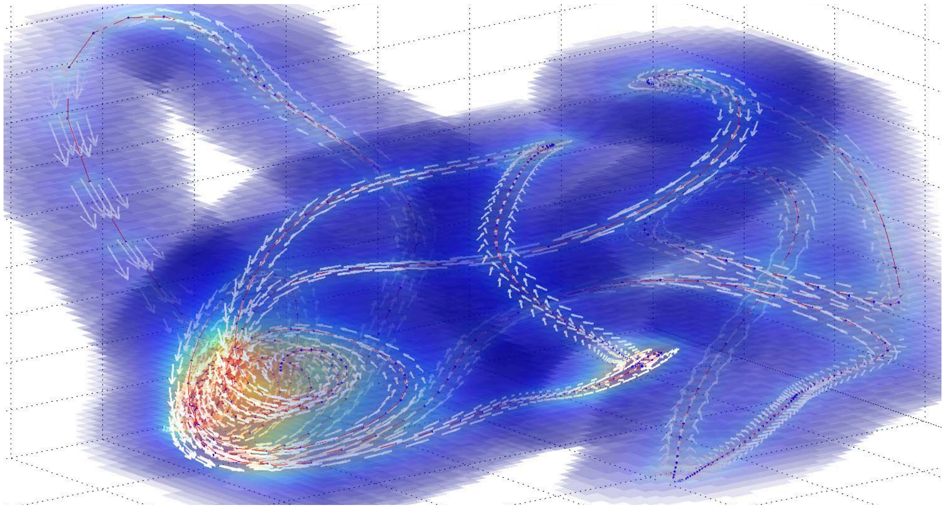


Figure 3: The manifold derived by GPLVMs projected onto an example 3-dimensional space. Each point represents a motion frame and lines indicate their sequential relation. Most of the data points within the dense area correspond to motions with arms resting, while trajectories far away from the dense area correspond to long sequences of gesture motion. Motion dynamics derived by GPLVMs is illustrated with white arrows in which the length of each arrow indicates the transition velocity around that region of the manifold. Our interpolation algorithm utilizes this transition velocity to help determine the transition trajectory. GPLVMs derive the dynamics for the entire space, and here we only show a fraction of it to make the figure clearer.

the interpolation in the original motion space, our framework derives a manifold with respect to gesture and motion dynamic to facilitate the interpolation.

3.2 Learning the manifold with GPLVMs

Among existing algorithms on projecting data into lower-dimension, GPLVMs exhibit the best capability on modeling human motion and respective dynamics. We explain this capability by first giving a brief description about GPLVMs.

3.2.1 Gaussian Process Latent Variable Models

The GPLVM is a dimension reduction approach which determines a low-dimensional space that better represents the given data, and the core idea is based on the Gaussian process. To help explain the idea, we first briefly describe the Gaussian process. A Gaussian process is a stochastic process which models the distribution of the predicting variable y as Gaussian in which the mean is set to 0 in general and the covariance is a function of the input variable x. Specifically, the covariance function is represented as a kernel function K where $K_{i,j} = k(x_i, x_j)$. Its log likelihood function is:

$$\ln p(t) = -\frac{1}{2}\ln|K| - \frac{1}{2}y^T K^{-1} y - \frac{N}{2}\ln(2\pi)$$

where N is the number of data points. Here we omitted some parameters of the Gaussian process to make the equation uncluttered. The GPLVM is an unsupervised learning algorithm in which the original predicting variable y in the Gaussian process is now given while x becomes the parameter to be determined, and the goal of the learning algorithm is to infer x with respect to y and a corresponding Gaussian process that jointly maximize the likelihood of $p(y|x,\theta)$, where θ denotes the parameters of the Gaussian process. GPLVMs find a low-dimensional projection x for y while preserving the similarity relation among the original data y. The data points that are far away from each other in the original space will also be apart from each other in the low-dimensional manifold.

An extension of GPLVMs called Gaussian process dynamical model (GPDM) [25] has been proposed to include dynamics in terms of determining low-dimensional projection. GPDMs contain the same process as GPLVMs in determining low-dimensional projection but with an extra autoregressive likelihood function $p(x_t|x_{t-1})$ to maximize. With this additional function, the optimization process for determining the low-dimensional projection has an extra objective for maximizing the likelihood of $p(x_t|x_{t-1})$. In other words, GPDMs need to allocate x in a way that when two points x_t and $x_{t'}$ are close to each other, their consecutive points x_{t+1} and $x_{t'+1}$ also have to be close to each other. As a result, the projection x in GPDMs reflect also the dynamics of the time series data.

Both GPLVMS and GPDMs have shown success in synthesizing human motion with the derived manifold. In our framework we apply GPDMs to learn the manifold. The dynamic term added to the objective function allows the manifold to incorporate motion dynamics. An example of the modeled dynamic relation along with the manifold is shown in Figure 3. As the GPDM is a GPLVM with an additional objective function that maintains the dynamic relation, we follow the convention and refer it as GPLVMs in the rest of the article.

After deriving the manifold with GPLVMs, each point in the manifold corresponds to a gesture frame, and sampling a trajectory and mapping it back to the original dimension through GPLVMs results in a gesture animation. On transitioning between two motion segments, the framework finds a trajectory connecting the two segments in the manifold, and the resulting trajectory is mapped back to the original space to generate the animation between gestures. The process is shown in Figure 4.

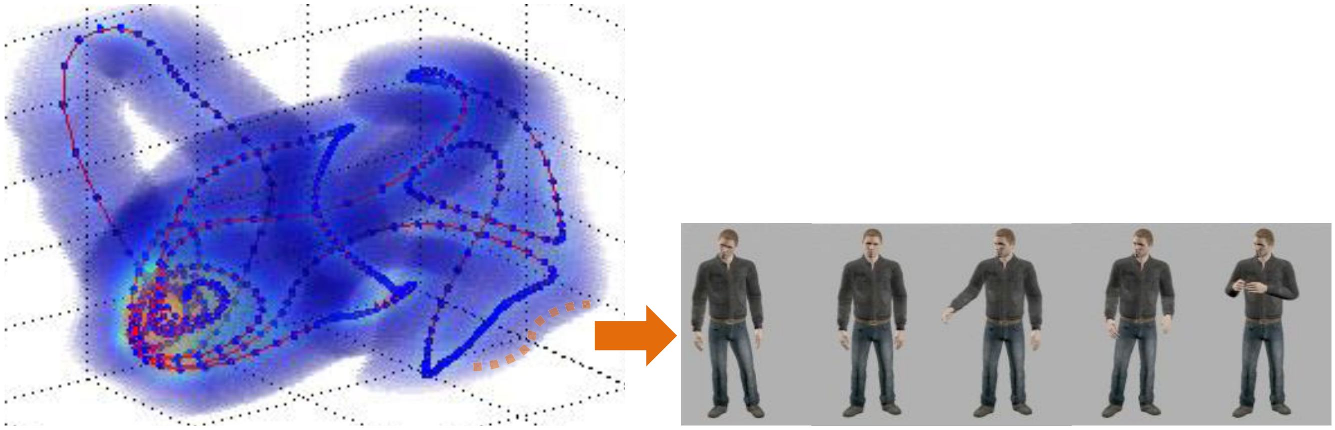


Figure 4: GPLVMs derive a low-dimensional space from the given motion samples. After deriving this space with GPLVMs, we can sample a trajectory in the space and map the trajectory back to the original data dimension and result a gesture animation. Colors of the space indicate the density of data points where the warmer correspond to the denser area.

3.3 Interpolation algorithm

The interpolation algorithm uses a forward inference function to determine a path following the existing motion and a backward inference function to determine a path connecting the consecutive motion. The interpolation of the two results the transition trajectory. Both inference functions generate paths in the manifold derived by GPLVMs. Since GPLVMs derive the manifold with an optimization term which maximizes the likelihood function of $p(x_t|x_{t-1})$ and the likelihood function is formulated as a Gaussian process, the manifold preserves the temporal relation of embedded data points with the criterion of being able to be inferred with a Gaussian process. Our inference algorithm exploits this essential relation and models both inference functions as Gaussian processes. Specifically, the forward inference function has the form of $x_t = f(x_{t-2}, x_{t-1})$ and the backward inference function has the form of $x_t = g(x_{t+2}, x_{t+1})$. Both inference functions take two consecutive points as input instead of one to allow robust predictions. On generating an interpolation trajectory of length M from point at t to $t+M-1$, our algorithm first infers a forward trajectory and a backward trajectory and then combines the two trajectories by performing a linear interpolation:

```
$path[1] \leftarrow x_t$
$path[2] \leftarrow x_{t+1}$
$path[M-1] \leftarrow x_{t+M-2}$
$path[M] \leftarrow x_{t+M-1}$
for $i = 3$ to $M-2$ do
    $forward[i] = f(path[i-2], path[i-1])$
end for
for $i = M-2$ to $3$ do
    $backward[i] = g(path[i+2], path[i+1])$
end for
for $i = 3$ to $M-2$ do
    $path[i] = (forward[i] * (M-1-i) + backward[i] * (i-2))/(M-3)$
end for
```

where $x_t, x_{t+1}, x_{t+M-2}, x_{t+M-1}$ are known data points and *path* contains the inferred trajectory of length M.

3.3.1 Generating interpolation trajectories

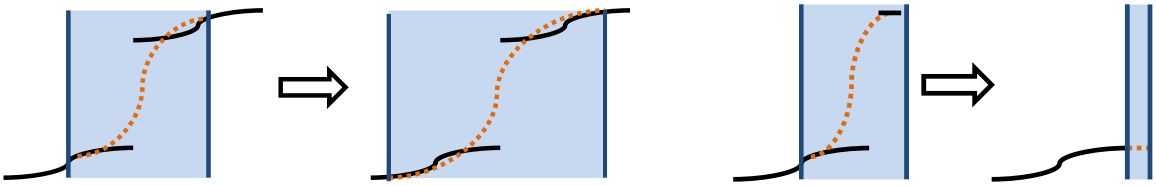
On transitioning between the two motion segments, the process needs to decide the length of the interpolation. The transition process finds a trajectory for connecting two motion segments, and the length of the interpolation decides the length of this trajectory. Different pairs of motion segments require different transition length, and therefore we apply an adaptive approach to determine the length of the trajectory for each interpolation. The transition process is shown in Figure 5.

The transition process starts with setting a short length n for the trajectory, and then applies our Gaussian process interpolation algorithm to infer a trajectory. After a trajectory is generated, the interpolation process checks whether the mean-square difference along the trajectory is smaller than a pre-defined threshold. If the value exceeds the threshold, the interpolation process increases the length and repeats the process until an admissible trajectory is derived or the length exceeds certain limit. The starting length and the length limit are predefined by the users.

In some situations the transition process may not find a trajectory that satisfies the criterion mentioned above. Often times this happens when the consecutive segment is very short, say only one frame. In that case, that short segment usually will be far away from its previous segment and its consecutive segment, and interpolating a trajectory that traverses these segments may result a motion that is not natural. On the other hand, these short segments can be considered as noise in the generation process as they in general are too short to present meaningful gestures. Thus, the transition process will ignore the consecutive motion segment when it fails to determine a feasible trajectory and infers a trajectory with prediction function f based only on the previous motion segment.

3.4 Speech-gesture inference

Our framework decomposes the gesture generator into two processes to make the inference from speech to gestures feasible. The training data is then composed of three categories: audio features of speech, gesture annotations, and corresponding motion segments. Gesture annotations can be defined based on specific project, and in this work we use a simple automatic gesture/non-gesture annotation scheme to assess the framework. Gesture motions are classified and



(a) Interpolation process

(b) Inference process

Figure 5: The transition process adaptively determines a trajectory in the low-dimensional space through interpolation. (a) The transition process decides an interval to interpolate a trajectory for transiting the two segments. If the interpolation process fails to find a smooth transition within the interval, the transition process expands the interval and performs interpolation again to find a smooth trajectory. (b) In the case that the process fails to find a smooth trajectory within the length limits, it will ignore the consecutive segment and infers a trajectory based on only the previous motion segment.

segmented based on the height of the wrist. Specifically, gestures with at least one wrist higher than certain threshold were classified as the same group. This definition roughly classifies gesture segments into "gesture" and "non-gesture" (motions in which both arms are resting at the side while the torso or the head can still be moving) types. We chose this simple annotation definition because the audio features applied in this experiment lack semantic content. Learning to map audio features to gestures involving semantic content would at best be difficult. On the other hand, audio features may more likely reveal information that helps to determine gesture and non-gesture behaviors.

The process of mapping speech to gesture annotations is realized by treating it as the problem of learning the mapping from one time series to another. Gestures are related to not only speech but also the previous and consecutive motions, and as each gesture is a long sequence of motions, the inference process needs to be capable of modeling long-term temporal relation to determine matching gestures for the speech signals. Our framework applies the conditional random field (CRF) with linear-chain structure to learn the inference task. CRFs are a type of graphical models which encapsulate the likelihood between labels conditioned on the input signals. The likelihood between labels allows CRFs to infer the label at each time frame with the criterion that the resulting label sequence as a whole is the most likely sequence. The capability matches the property of our task since each gesture is composed of a sequence of motion frames. In this learning task, the observable data is the audio features of the speech and the learning target is the annotations for the respective gestures. The CRF learns the transition between gestures conditioned on the audio features.

4. EXPERIMENTS

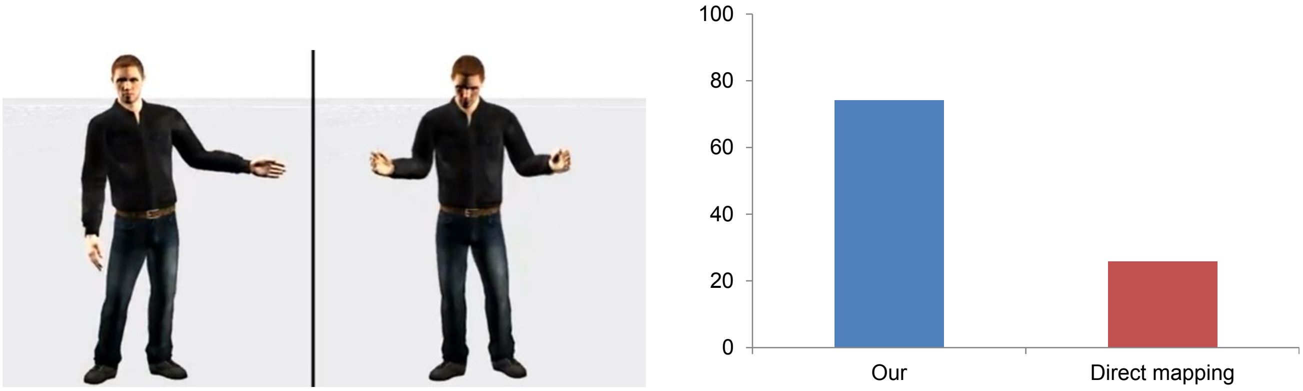
We evaluated the quality of the generated animations by comparing them with animations generated by the approach taken by [4], that uses an approach of direct modeling between speech and gestures. We used a dataset created for examining how audio and body motion affect the perception of virtual conversations [8]. The dataset contains speech audio and motion capture of three people having conversations. We chose the records where each person gives a long speech without being interrupted. We chose the data of male number 1 as training data and the data of male number 3 as testing data. There are total 193 seconds of training data and 238 seconds of testing data. The motion capture data contains the skeleton of subjects and the recorded joints movement is a vector with 69 degrees of freedom, and we trained GPLVMs to project the data onto a 9-dimensional space. The original motion capture data has 120 frame rate, and we down-sampled it to 15 frame rate.

For speech input, our experiment extracts the following audio features: normalized amplitude quotient (NAQ), peak slope, fundamental frequency (f0), energy, energy slope, spectral stationarity. We also apply an automatic approach to determine the tenseness of the voice at each time frame which gives the probability of being at low, medium, or high tenseness [20]. The extraction process also determines whether the speaker is speaking based on f0, and for the periods in the speech that identified as not speaking all audio features are set to zero. The resulting audio features have 9 dimensions.

As described in section 3, we applied an automatic approach to segment and classify gestures into gesture/non-gesture classes. This automatic classification scheme results 52 non-gesture and 47 gesture motion segments from the training data, where the length of each segment ranges from 1/15 seconds to 21.8 seconds.

We trained the framework of [4] with the training data mentioned above and follow the same configuration reported in that experiment. The test data are 14 speech audio clips with length ranging from 10 to 21 seconds. We animate the generated motion on a virtual character with SmartBody [23]. Since the motion capture data does not include finger nor lip information, to make the virtual character more natural, we applied SmartBody's mechanism to automatically generate lip movement synchronized with the speech audio, and put an idle motion to animate fingers. For the evaluation, we recruited 48 participants on Mechanical Turk and asked them to make pairwise comparisons to vote which animation best matches the speech audio. The evaluation video displays animations generated by both algorithms side-by-side as shown in Figure 6a, and there are total 14 videos. The evaluation result, illustrated in Figure 6b, shows that our framework is better than the previous work where 74.1% of evaluations choose the animations generated by our framework. Pearson's Chi-square test shows that the difference is statistically significant.

While animations generated by both frameworks appear natural and related to the speech, gesture animations generated by our framework exhibit more active movement. This is due to that [4] learns a direct mapping from speech to motion which is a challenging task and as a result the de-



(a) Evaluation video

(b) Vote percentage

Figure 6: Comparing animations generated by our method and the previous work. (a) Each video displays a pair of gesture animations generated for the same speech audio by different approaches. (b) The percentage of animations being voted as best matching the speech, and the difference is statistically significant.

rived model tends to generate less active movement since performing dramatic motion can lead to higher error when mismatching the target gestures than performing smooth and less active movement.

The evaluation experiments assess the overall performance of our framework, and to gain further insight about our model we also evaluated the performance of the inference process on mapping speech signals to gesture annotations. The evaluation experiment compares our inference model (using CRFs) with support vector machines (SVMs) [6] and random forests [1]. Unlike CRFs, neither SVMs nor random forests utilize the temporal relation between the data sequences to make predictions, and therefore to make a more concrete comparison our experiment includes the speech signals at the previous and the next time frame as input signals for both models. The experiment uses the same training and testing data mentioned above. SVMs achieve highest accuracy with RBF kernels and window width 3 for the input data (include signals at the previous 1 time step and the future 1 time step), and their accuracy results are shown in Figure 7. As both the performance of SVMs and random forests are close to random, the results indicate that the inference problem is extremely difficult and it is crucial to include temporal information. Although our inference model is better than other state-of-the-art algorithms, there is still a room for improvement.

We did not compare our framework with [15] that also models the generation task with two processes, because their approach only selects motion segments that can seamlessly connect to the last frame of current motion. This requires a much larger dataset that makes building a gesture controller with our limited training dataset infeasible. On the other hand, the experiment shows that our approach can learn a gesture controller from a small set of data that can generate natural gestures for novel utterances.

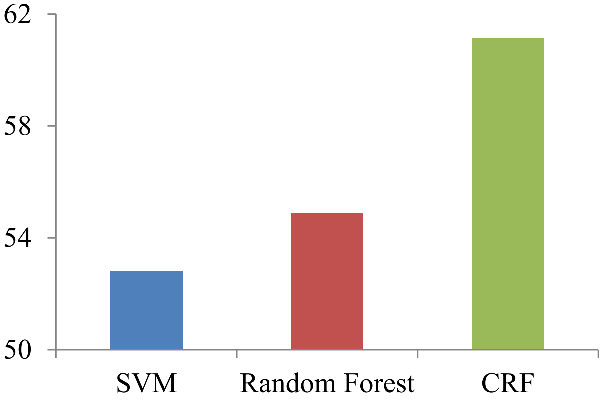


Figure 7: Our inference model (CRFs) outperform SVMs and random forests on predicting gestures from speech signals. y-axis represent the percentage value of the prediction accuracy.

5. CONCLUSIONS

We have proposed a framework to map speech features to gesture annotations and synthesize gesture motion for the annotations. The model for mapping speech features to annotations is derived using CRFs. To synthesize gesture motion we apply GPLVMs to learn a low-dimensional representation of gesture motions, select motion segments in the low-dimensional space based on annotations, and use an interpolation algorithm based on the Gaussian process to determine a trajectory that allow transitions between motion segments. The evaluation result indicates that the generated gestures match the speech significantly better than those generated with the direct mapping approach. As the second study demonstrated, there is still a room to improve the inference quality of speech signals to gesture annotations.

This work lays a preliminary foundation toward building a comprehensive gesture controller. The critical next step is to increase the expressiveness of the gesture controller so that the mapping learned by the speech-annotation mapping process can realize expressive gestures more tightly coupled to the uttered content. Achieving this goal requires including linguistic features of speech such as content and syntactic structure, and as well annotating motion gestures with a richer set of annotations.

6. ACKNOWLEDGEMENTS

The projects or effort described here has been sponsored by the U.S. Army. Any opinions, content or information presented does not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

7. REFERENCES

[1] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[2] C. Busso, Z. Deng, M. Grimm, U. Neumann, and S. Narayanan. Rigid head motion in expressive speech animation: Analysis and synthesis. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(3):1075 –1086, 2007.

[3] J. Cassell, H. H. Vilhjálmsson, and T. Bickmore. Beat: the behavior expression animation toolkit. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 477–486, New York, NY, USA, 2001. ACM.

[4] C.-C. Chiu and S. Marsella. How to train your avatar: A data driven approach to gesture generation. In *11th Conference on Intelligent Virtual Agents*, pages 127–140, 2011.

[5] C.-C. Chiu and S. Marsella. Subjective optimization. In *12th Conference on Intelligent Virtual Agents*, pages 204–211, 2012.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] A. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 478–485. IEEE Computer Society, 2004.

[8] C. Ennis, R. McDonnell, and C. O'Sullivan. Seeing is believing: body motion dominates in multisensory conversations. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 91:1–91:9, New York, NY, USA, 2010. ACM.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, 2006.

[10] S. Kopp and K. Bergmann. Individualized gesture production in embodied conversational agents. In M. Zacarias and J. V. Oliveira, editors, *Human-Computer Interaction: The Agency Perspective*, volume 396 of *Studies in Computational Intelligence*, pages 287–301. Springer Berlin Heidelberg, 2012.

[11] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 473–482, New York, NY, USA, 2002. ACM.

[12] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[13] N. D. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.

[14] J. Lee and S. Marsella. Nonverbal behavior generator for embodied conversational agents. In *6th Conference on Intelligent Virtual Agents*, volume 4133 of *Lecture Notes in Computer Science*, pages 243–255, 2006.

[15] S. Levine, P. Krähenbühl, S. Thrun, and V. Koltun. Gesture controllers. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 124:1–124:11, New York, NY, USA, 2010. ACM.

[16] S. Levine, C. Theobalt, and V. Koltun. Real-time prosody-driven synthesis of body language. *ACM Trans. Graph.*, 28:172:1–172:10, December 2009.

[17] S. Levine, J. M. Wang, A. Haraux, Z. Popović, and V. Koltun. Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics*, 31(4):28, 2012.

[18] M. Neff, M. Kipp, I. Albrecht, and H.-P. Seidel. Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Trans. Graph.*, 27(1):1–24, 2008.

[19] M. Sargin, Y. Yemez, E. Erzin, and A. Tekalp. Analysis of head gesture and prosody patterns for prosody-driven head-gesture animation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(8):1330–1345, Aug. 2008.

[20] S. Scherer, J. Kane, C. Gobl, and F. Schwenker. Investigating fuzzy-input fuzzy-output support vector machines for robust voice quality classification. *Computer Speech and Language*, 27(1):263–287, 2013.

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

[22] G. Taylor and G. Hinton. Factored conditional restricted Boltzmann machines for modeling motion style. In L. Bottou and M. Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 1025–1032, Montreal, June 2009. Omnipress.

[23] M. Thiebaux, S. Marsella, A. N. Marshall, and M. Kallmann. Smartbody: behavior realization for embodied conversational agents. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 1*, AAMAS '08, pages 151–158, 2008.

[24] L. Valbonesi, R. Ansari, D. McNeill, F. Quek, S. Duncan, K. E. McCullough, and R. Bryll. Multimodal signal analysis of prosody and hand motion: Temporal correlation of speech and gestures. In *Proc. of the European Signal Processing Conference 2002*, EUSIPCOŠ02, pages 75–78, 2002.

[25] J. Wang, D. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):283–298, feb. 2008.