A style controller for generating virtual human behaviors

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

Stacy Marsella
USC Institute for Creative Technologies
12015 Waterfront Drive
Playa Vista, CA 90094
marsella@ict.usc.edu

ABSTRACT

Creating a virtual character that exhibits realistic physical behaviors requires a rich set of animations. To mimic the variety as well as the subtlety of human behavior, we may need to animate not only a wide range of behaviors but also variations of the same type of behavior influenced by the environment and the state of the character, including the emotional and physiological state. A general approach to this challenge is to gather a set of animations produced by artists or motion capture. However, this approach can be extremely costly in time and effort. In this work, we propose a model that can learn styled motion generation and an algorithm that produce new styles of motions via style interpolation. The model takes a set of styled motions as training samples and creates new motions that are the generalization among the given styles. Our style interpolation algorithm can blend together motions with distinct styles, and improves on the performance of previous work. We verify our algorithm using walking motions of different styles, and the experimental results show that our method is significantly better than previous work.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Style-Content Separation, Restricted Boltzmann Machines, Virtual Agent, Animation, Motion Capture

1. INTRODUCTION

In the short film *Luxo Jr.* by Pixar Animation Studios, the two Anglepoise desk lamps demonstrate a simple and entertaining story. Without the aid of verbal and facial expressions, the desk lamps successfully express their character and emotional states through motions. Human sensitivity to information conveyed through such expression breathes life into these virtual characters. In fact, we can perceive identity [20] and gender [12] of walkers simply based on the motion of lights points attached to their joints. Thus, motion is one of the main criterion for building realistic virtual characters.

Humans have many different kinds of behaviors, and each behavior is composed of many different motions. Even for a single motion, there can be various ways to perform it. The variation can be due to different mental states, physical properties, personality, etc. To exhibit this resemblance to reality, the virtual character requires a large set of animations, and it is not always obvious how to determine the subtle dynamics expressing these characteristics. One common approach to creating a virtual character's behaviors is employing animators. Another approach is to apply motion capture. The motion capture technique can record the temporal difference of each motion and the subtle variance within different styles. However, recording every possible kind of motion is very time consuming. Moreover, when a human performs the same motion, each will show some variation. It is not practical to collect a huge set of animations for each motion for either approach, and replaying the same animation every time reduces the resemblance to reality of the virtual character.

To generate realistic motion animations and save animators' efforts, one approach is to generalize motion from examples. There are many ways to approach this generalization. One that has been widely applied is synthesizing motion from a motion library [7, 9]. Segmenting motion clips and combining them is an easy way to make a general use of existing motion, but animations are limited to the finite set of clips. Another approach to generate new animation is to learn a style translator and translate a given motion to a specific style [6, 4]. We can increase the amount of virtual human behaviors via converting some motions to new styles with such a translator.

This approach becomes more powerful if we can infer what parameters determine the style of motion. The style parameter of the virtual character gives control over motion generation, and we can adjust it to express appropriate signals like emotional states in different situation. Thus, style-content separation is an appealing approach to generate new motions. There have been several works to explore the separation of style and content of motion data [18, 3, 15, 2, 21, 16]. After separating the style parameters from the motion, we can generate new motions via interpolation or extrapolation in the style space [14, 19].

Previous work showed success in synthesizing new motions

Cite as: A style controller for generating virtual human behaviors, Chung-Cheng Chiu and Stacy Marsella, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. XXX-XXX.

with analogy among samples, but they suffer from overfitting and usually will fail on synthesizing new styles. These works followed the design of bilinear models [18] that represent styles as a separate parameter, use different style values to learn the motion generation, and then generate new motions by changing the style value. When using this design, the model is assumed to capture the style space so that adjusting the style value leads to style interpolation or extrapolation. However, to satisfy this assumption, we need a sufficient amount of data distributed throughout the style space so that the model can comprehend the structure of the style space. This is because the style space can be a nonlinear manifold [3], and it requires a lot of data for the model to identify this structure, unless the members of the data set is already close to each other. This condition leads to the requirement of either collecting a large set of data or requiring all motions to have similar styles.

In designing a virtual character behavior controller, we would like to have the capability of generalization among style space while minimizing the required effort to collect training samples. However, overfitting is an inevitable problem when the styles of motions are quite different and the training samples are insufficient, and therefore generating new motions via interpolation with style parameters will simply produce implausible results. To design a robust method that can generate new motions with a limited set of training samples, we need to abandon the assumption that the general structure of the style space can be identified accurately from the training data. Instead, the key issue to address is *how to do style interpolation when the model is overfitted*.

In this work, we propose a learning model and a style interpolation algorithm that can generate new motions via style interpolation when given a few training samples with distinct styles. Our model, called the hierarchical factored conditional Restricted Boltzmann Machine (HFCRBM), is a modification of the factored conditional Restricted Boltzmann Machine (FCRBM) [16] that has additional hierarchical structure. The HFCRBM includes a middle hidden layer for a new form of style interpolation. Our style interpolation algorithm, called the multi-path model, performs the style interpolation using the middle hidden layer.

To verify the effectiveness of our approach, we apply our algorithm to learn and generate walking motions with different styles. The walking motion samples are from the CMU mocap database. We evaluate the performance of our algorithm against motion generation of previous works, and compare different style interpolation approaches. The experiment results show that (1) the HFCRBM has better performance than the FCRBM [16], (2) the multi-path model generates new motions much more successfully than conventional style label interpolation, and (3) the multi-path model is also applicable to the FCRBM [16] and improves its performance.

The contribution of this work is three-fold.

- We propose a model and a style interpolation algorithm that can generate new styles of motions with given a limited set of training samples.
- Our style interpolation algorithm improves the performance of the previous work on blending different styles.
- To the best of our knowledge, our work is the first to answer the question of how to do style interpolation when the general structure of the style space cannot be identified accurately from the training data.

2. RELATED WORK

One idea as to how to automatically generate human motion is to learn a motion generation function, such as learning the parameters of muscle control for the motion [11], identifying dynamics of motion transition with a linear dynamic system for further synthesis [13, 10, 1], or learning the transition between each frame with a Dynamic Bayesian Network and generating new motions via adding noise to the function [8]. Another idea is to convert existing motions to new motions with the same content but different styles, and to achieve this by learning a style translation function [6, 4]. A style translation function can produce new motion in a specific style with given animations, but it will be even more powerful if the factors that influence the style of motion can be determined. In this case, we need to separate these properties from the content, learn the functional space of the properties, and add variations within this function.

The problem of determining the properties that influence the content is called *style-content separation*, and was introduced by Tenenbuam & Freeman [18]. They proposed a bilinear model that represents the training data as the product of content, style, and interaction matrices. Elgammal & Lee [3] extended the idea by representing content on a nonlinear manifold. When the manifold is constructed, the model learns nonlinear mappings from the embedding space to the training data, and derives interactions (called content bases in their paper) and style matrices from coefficients. When given a new data, with fixed content bases, the style (projection vector) and content (manifold coordinates) are calculated with an EM-like iterative procedure. Shapiro et al. [15] proposed to apply Independent Component Analysis to decompose motion sequences into several components (also motion sequences), and have users select representative components. The new motion with a specific style is generated via merging corresponding components.

These methods take regression-like approaches that treat the motion data as trajectories, and do not model the transitions between frames. Brand & Hertzmann [2] designed a model to learn this kind of transition relation. They extended hidden Markov models (HMMs) with an additional style variable to model different motion sequences. While hidden states capture the "mean" of the motion (the content) the additional style variable models the deviation between different motion (the style). The HMM can have only a few discrete states, so the representation capability for poses is limited. Wang et al. [21] proposed to use the Gaussian Process Latent Variable Model to learn a function that predicts the subsequent frames of the sequence from the previous frame and specified information. The mapping function explicitly includes the *identity* and *style* factors, and learns *identity*, *style*, and *content* from motion data performed by different skeletons for various styles. The method showed the synthesis of new motions via interpolation between similar motions. Taylor & Hinton [16] proposed factored Conditional Restricted Boltzmann Machines (FCRBMs) to model the transition between frames while gated by style parameters. This method can learn motions with quite different styles, but for synthesizing new styles, it requires sufficient samples to learn generalization.

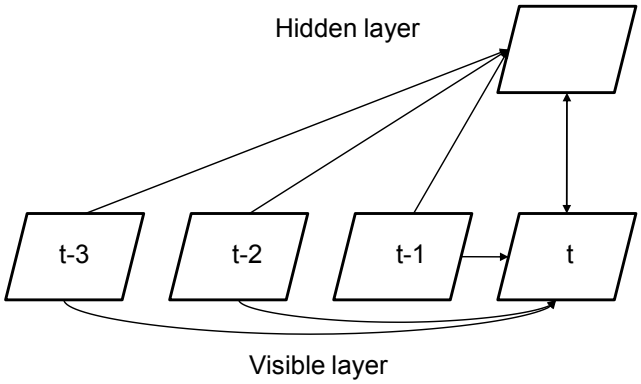


Figure 1: The architecture of a CRBM of order 3.

3. ALGORITHM BACKGROUND

The conditional Restricted Boltzmann Machine (CRBM) [17], as shown in Fig. 1, is a model for learning transitions within time series data. The CRBM adds directed links from the past visible layers to send previous observed values to the current visible and hidden layers. The new structure includes the information from the past, and can learn the temporal relation of the time series data. A CRBM treats the messages sent from the past as biases, or *dynamic biases* to be more specific. When given a sequence of data, the CRBM adds these values to the current prediction through directed links as biases and uses alternating Gibbs sampling (sending information iteratively between the visible layer and the hidden layer) to construct the next piece of data. The energy function of a CRBM for real-valued visible data (assuming unit variance) is:

$$E(\mathbf{v}_t, \mathbf{h}_t|\mathbf{v}_{<t}) = \frac{1}{2}\sum_i (v_{i,t} - \hat{a}_{i,t})^2 - \sum_{ij} W_{ij} v_{i,t} h_{j,t} - \sum_j \hat{b}_{j,t} h_{j,t}$$

where $\mathbf{v}_t$ and $\mathbf{h}_t$ are current visible nodes and hidden nodes, $v_{<t}$ denotes past visible nodes, W represents undirected connections between visible and hidden layers, and $\hat{a}_{i,t}$ and $\hat{b}_{j,t}$ are dynamic biases such that $\hat{a}_{i,t} = a_i + \sum_k A_{ki} v_{k,<t}$ and $\hat{b}_{j,t} = b_j + \sum_k B_{kj} v_{k,<t}$, where A and B represent directed connections from the past visible nodes to the current visible and hidden layers, and a_i and b_j denote the bias of visible and hidden layers.

CRBMs capture the transition dynamic of the time series data in an unsupervised way. In some applications, we would like to use annotation information to help recognition and generation. For example, for motion generation style annotation can improve the training process of learning various forms of motions. The ancestor of CRBMs, the Restricted Boltzmann Machine (RBM), can be stacked into a multi-layer model to construct deep belief networks [5] for supervised learning. As its successor, the CRBM can also be stacked into multiple layers, so it is straightforward to stack multiple CRBMs to build similar deep networks for supervised learning on time series data. However, the strategy is no more effective. The limitation comes from the dynamic biases. The values from the past observations $\mathbf{v}_{<t}$ are too strong and will dominate the values from the label parameters. Thus, the generation process relies mainly on the past observed values [16].

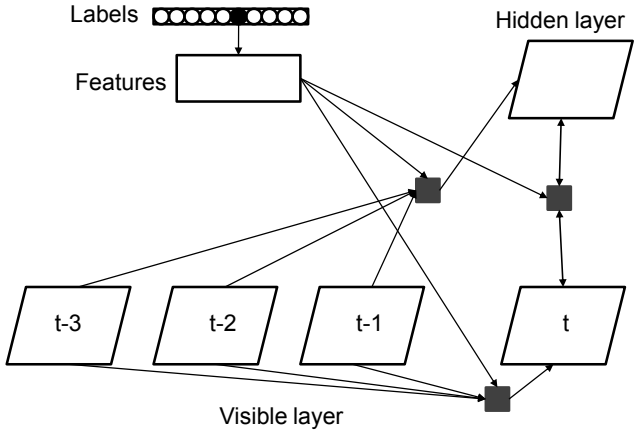


Figure 2: The architecture of a FCRBM with contextual multiplicative interactions.

Instead of defining labels as part of the inputs to the hidden nodes, we can model the labels as gates for controlling other inputs. In this way, the label information has a strong influence on the CRBM. To construct these gating capabilities for the label units, each set of connections is expanded with an additional "label" dimension. The new weight matrix of the connections between the visible and hidden layers is a three-way weight tensor W_{ijk} connecting visible, hidden, and label nodes. With this new form of weight matrix, label nodes then can comprise the transition between visible and hidden layers.

Assigning label nodes as a manipulator for the original model can allow it to learn complex data, but this design also makes the resulting model parametrically cubic. In fact, much real world data, including mocap, has some form of regularity, and the structure can be captured with a more contiguous model. Taylor & Hinton proposed Factored CRBM (FCRBM) with contextual multiplicative interaction (we will simply call it FCRBM in the following text for clarity) to model this property [16]. The FCRBM contains the structure of the CRBM, and it applies additional label information to change the information transition within the original CRBM model in a factored form, as shown in Fig. 2. The energy function of the FCRBM is:

$$E(\mathbf{v}_t, \mathbf{h}_t|\mathbf{v}_{<t}) = \frac{1}{2}\sum_i (v_{i,t} - \hat{a}_{i,t})^2 - \sum_f \sum_{ijl} W^v_{if} W^h_{jf} W^z_{lf} v_{i,t} h_{j,t} z_{l,t} - \sum_j \hat{b}_{j,t} h_{j,t}$$

Readers can refer to [16] for further details.

4. HIERARCHICAL FCRBM

We extended the FCRBM to construct the hierarchical FCRBM. The hierarchical structure is crucial for style interpolation, because the structure provides a new form of style interpolation, and the new approach produces much better results than conventional style interpolation. We begin our explanation by discussing the problems of previous approaches.

Previous approaches perform well at reproducing given examples, but to generate new motions and avoid overfit-

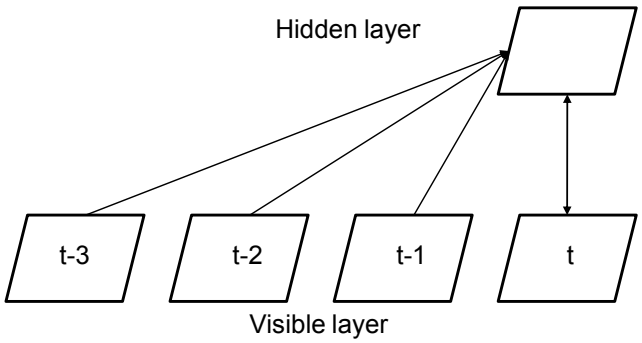


Figure 3: The architecture of a reduced CRBM of order 3.

ting, the model needs sufficient training samples with the same content and different style throughout the style space for which we want to generalize. For example, previous work [16] applied the model to learn the generalization of style parameters *speed* and *stride length* of walking motion. They recorded nine sequences of walking motions which correspond to the crossproduct of (*slow*, *normal*, *fast*) for speed and (*short*, *normal*, *long*) for stride length, and fed these samples to FCRBMs for training. The model shows good generalization across speed and stride length. However, when building a realistic virtual character, the character needs to have a rich set of behaviors. A great number of training samples will be required to complete its style table, which makes the style-content separation approach less practical. To make the generation function useful in practice, the model needs the capability of learning from a limited set of animations in which the style generalization is not demonstrated explicitly.

Conventional style-content separation approaches accomplish style interpolation via adjusting the values of the style label to indicate the ratio of style interpolation. The labels can be real-valued or binary. In the real-valued representation, it is assumed that the style space is contiguous, the label values provide the correct position of the style in the style space, and the model can formulate the style space. If the label values are assigned correctly, then this way of labeling helps the learning process. However, the success of this approach depends on whether the prior knowledge of these motions is sufficient to provide an accurate annotation. It also limits the variety of the motion style. In the binary representation each label corresponds to a feature vector since the label layer connects to a feature layer. The feature vector not only represents the vector generating a specific style, it also corresponds to a way of generating motions, the content. Interpolating two vectors in the Euclidean space does not correspond to interpolating two styles in the style space, and the new vectors can easily fall out of the appropriate space for motion generation. Thus, a vector resulting from this approach will rarely map the generation to the appropriate style, and the function may be no longer appropriate for generating the correct content.

We propose to perform style interpolation with the hidden layer instead of with the label parameter directly. To formulate the hidden layer, we construct a hierarchical model with the FCRBM. Instead of learning kinematics parameters directly, our model first extracts the patterns of the motion samples and represents them as binary variables. The model

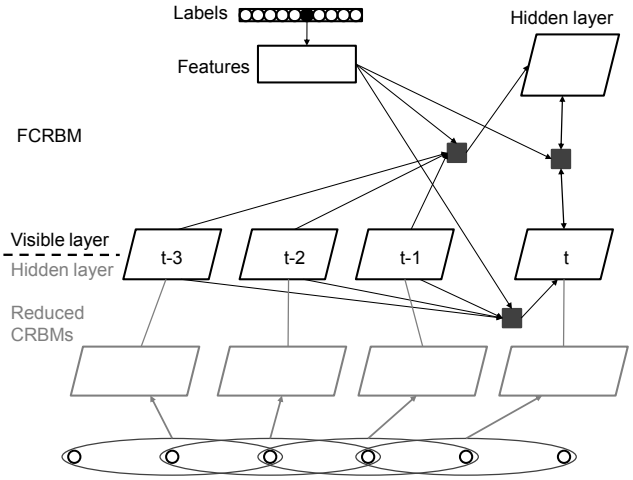


Figure 4: The architecture of the entire model. The reduced CRBM at the bottom layer is trained first, and the FCRBM then takes the approximate filtering distribution from the bottom layer as input to train its connections. There is a feature layer linked to the label nodes that propagates the label information to the model.

for performing such a step is called *reduced CRBM*.

4.1 Reduced CRBM

We modify the CRBM in order to construct the hierarchical structure. The new model is a CRBM without the directed links from past visible layers to the current visible layers. This *reduced CRBM* includes the past observed information, and the activation of hidden nodes conveys the appearance of certain motion patterns. Without the lateral links from the past visible layers, the generation depends completely on top-down information. Therefore, the upper layers have full control of the motion generation. The reduced CRBM is shown in Fig. 3. Its energy function is:

$$E(\mathbf{v}_t, \mathbf{h}_t|\mathbf{v}_{<t}, \theta) = \frac{1}{2}\sum_i (v_i - a_i)^2 - \sum_{ij} W_{ij} v_{i,t} h_{j,t} - \sum_j \hat{b}_{j,t} h_{j,t}$$

where all the terms are the same as for the CRBM, except the bias of the visible layer is static bias instead of dynamic bias.

The reduced CRBM can be trained with a very efficient approximate learning algorithm called contrastive divergence [5]. Given the training motion samples, the reduced CRBM learns the reconstruction of the data x_t based on the sequence x_{t-1} to x_{t-n} (for an order n model), where the hidden layers receive x_{t-1} to x_{t-n} through connection B as the dynamic bias.

4.2 Hierarchical FCRBM

Our model stacks a FCRBM on top of the reduced CRBM to learn motion generation with label information. After training the reduced CRBM, the connection within this layer is fixed. To train the FCRBM, the training data goes bottom-up through the reduced CRBM to the FCRBM. The motion sequence is then converted into the approximate filtering distribution, and the FCRBM learns the

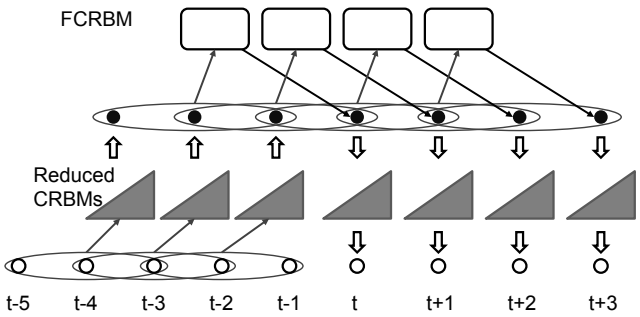


Figure 5: The generation process. A short motion sequence is input to the reduced CRBM, and the motion data is converted into the seed sequence of the FCRBM. Starting from this seed sequence, the FCRBM generates new data and uses it as new seed for further generations. The reduced CRBM takes the output of the FCRBM to construct motion data.

generation based on the sequence of the distribution. The visible layer of the top layer FCRBM is binary-valued, and we tied feature-factor parameters in our model as it further reduced the complexity of the model while maintaining good performance [16]. The architecture of the entire model and the training process is shown in Fig. 4. Each node in the label vector corresponds to each category of motion sample, and only one node is active when training a motion sample.

The model takes a short sequence of motion as a seed to generate future motions with the specified style parameters. After each generation step, the model concatenates its output to the seed sequence, drops the first data, and uses the new sequence as a seed to generate the next data. Via this recurrent-like structure, the generation process can perform multiple steps of prediction that allow it to generate a motion sequence of any length. In this multi-layer model, the seed sequence is sent bottom-up to the top layer to generate the next data. However, in the self-concatenation step, instead of using the output real-valued data at the bottom and sending it all the way up to the top layer as the new seed, the top layer model uses the generated data at its visible layer directly as input to generate the succeeding sequence. The data generated by the top layer model then goes down to the bottom layer to construct the motion vector. We demonstrate the generation process in Fig. 5.

5. STYLE INTERPOLATION

The style controller is a prediction function which takes the form:

$$x_t = f(x_{i<t}, \theta)$$

where x_t denotes current motion data, $x_{i<t}$ denotes past motion data, and θ represents the style vector. Using the current output data as one of the inputs for the next generation, the function can iteratively produce a data sequence with a specified length. We use one-hot encoding for the style vector since it does not require prior knowledge for assigning values as real-valued representation does. The style vector has the same length as the number of styles provided for training. Each element of the vector corresponds to a category of the sample motion. A vector with value 1 at the ith element and 0 elsewhere will make the generation function reconstruct a motion with the style of the ith training

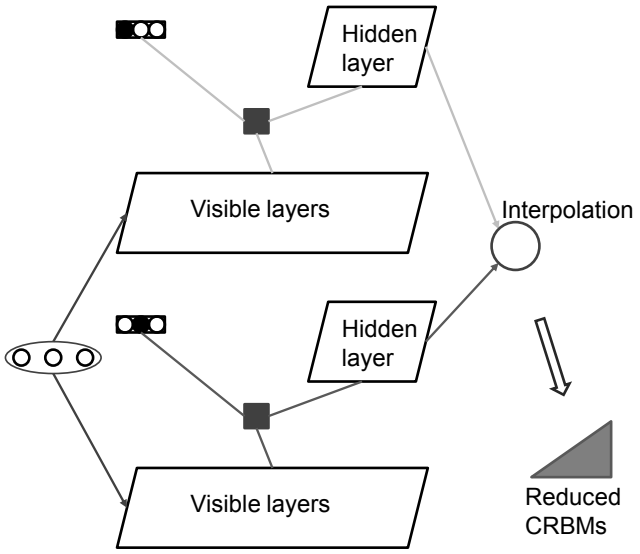


Figure 6: A two-motion blending example of the multi-path model. The multi-path process is executed at top layer FCRBMs. The interpolated result is then sent to the hidden layer of reduced CRBM to convert to the distributions of the hidden nodes.

sample. To synthesize a new style, previous work uses the values of style vector to represent the fractional weights of styles we want to generate. In this case, a style vector with 0.5 at the ith and jth elements corresponds to a style that is an average of the two respective categories. When assigning different fractions to different elements for the style vector, the generation function will create new styles of motions which are the blending of different styles based on the fractional weights.

We do not follow the original method but propose a new style interpolation approach called the multi-path model. For each style element with positive values, the multi-path model creates a FCRBM instance to generate the motion independently with only the corresponding style label being active. After the visible data of each style is generated, an average of the data weighted according to the respective fractional values is sent to the hidden layer of the bottom reduced CRBM. For example, when given a style vector $[0.6, 0.4, 0]$, the model creates a FCRBM instance with style vector $[1, 0, 0]$ and a FCRBM instance with style vector $[0, 1, 0]$ to do the generation separately. The connection weights of both are the same, and the output is interpolated with $0.6 \times x_1 + 0.4 \times x_2$ where x_1, x_2 denotes the respective output. The architecture of the multi-path model is shown in Fig. 6.

The style interpolation across the hidden layer is a new form of style interpolation. Hidden layer interpolations result in a motion vector which is the interpolation of two motion styles and can be different from all the motion samples. Since the generation result will feed back to the model for the next prediction, the new motion frame can lead the model to generate a new sequence of motion. On the other hand, it may result in unfamiliar input for the model and lead the function to be unable to predict the next frame. Thus, it is possible that this approach will fail on some

style interpolations. Although the multi-path model cannot guarantee a complete generalization, it is much more robust than interpolation among style parameters. This is because the overfitting of the motion generation function attributed more to style vector θ than past motion data $x_{i<t}$. In the multi-path model, the style label parameters assigned to each instance of the FCRBM are familiar to the prediction function. Thus, there is only one uncertain factor, the input data $x_{i<t}$. On the other hand, an explicit style interpolation with style label parameters can result in a style label parameter unfamiliar for the prediction function. All the conditional parameters of the prediction function are then uncertain in this approach. In this way, performing style interpolation with the hidden layer is more robust.

Overall, there are four ways to do style interpolation:

1. **Animation blending.** Two motions with the same content but different styles can be combined with interpolation among motion vectors. In this approach, each motion is viewed as a high dimensional trajectory, and motions can be combined after time warping and corresponding points are assigned. Animation blending is the most popular way to combine two motions. It does not suffer from the risk of generating inadmissible motions that prediction-based methods do. On the other hand, it lacks the generalization capability of those methods, such as creating new motions through analogy, and its performance depends on the correctness of time warping and matching correspondent frames. Moreover, it is also known to average out the styles of motions on combination, while style-content separation approach can preserve more significant styles [15].

2. **Style label interpolation.** The conventional approach to blend different styles together is to apply a linear interpolation of the label parameters.

3. **Visible layer interpolation.** Our multi-path model can also be applied to a single layer FCRBM. The only difference is that the output of the FCRBM is then a motion vector, and the resulting motion data is the direct interpolation across these vectors.

4. **Hidden layer interpolation.** In the hierarchical FCRBM, the multi-path model does the interpolation at the hidden layer. As shown in Fig. 6, the interpolation process works on the hidden node distributions of the reduced CRBMs. In this way, the style interpolation blends motions implicitly instead of modifying motion vectors explicitly.

Due to the limitations of conventional animation blending with respect to style-content separation, we did not include animation blending in the experiment and only compare style interpolation approaches.

6. EXPERIMENTS

Our motion samples are derived from the CMU Graphics Lab Motion Capture Database. The skeleton of the CMU motion capture data contains 38 nodes, and the total degree of freedom of all joints is a vector with 96 dimensions. There is a root node containing the global information of translation and rotation, and every other node maps to a part of the body that contains the local rotation information. The rotation of each node is represented as exponential maps with three dimensions. To learn a motion generator that focuses on the dynamics and interaction of body parts, we remove the global translation from the motion vector. We selected eight walking motions with different styles from database subject #105.

In this experiment, we applied a previous approach [16], which uses FCRBM with style label interpolation, as a baseline for comparison. The FCRBM program is derived from Taylor's website[1]. To evaluate the performance of the HFCRBM and the multi-path model, we evaluated two approaches: the HFCRBM model with conventional style label interpolation and the HFCRBM model with the multi-path model. To test whether our multi-path model can also improve the performance of the FCRBM, we evaluated the performance of the combination of the FCRBM and the multi-path model.

To sum up, we compared the performance of (1) FCRBM with style label interpolation, (2) FCRBM with multi-path model, (3) HFCRBM with style label interpolation, and (4) HFCRBM with multi-path model. The performance is evaluated with pairwise blending of two motions. In style interpolation, the generation process succeeds more easily when the ratio is weighted more toward one style; for example, a 80%/20% blending. It is more challenging when the ratio is close to one. In our experiment, we chose the most difficult option, the 50%/50% blending, for every case. For FCRBM-based models, the prediction function has two sets of input, the style label and the past data sequence. When blending two motions per a given ratio, using the partial sequence of one motion as initial input is considered a different case than using the other motion for initialization. Thus, there are two configurations for blending two motions, and total 64 configurations of pairwise blending for 8 motion sequences. We used a FCRBM with 600 hidden nodes and a HFCRBM with 360 nodes at the first hidden layer and 360 nodes at the second hidden layer.

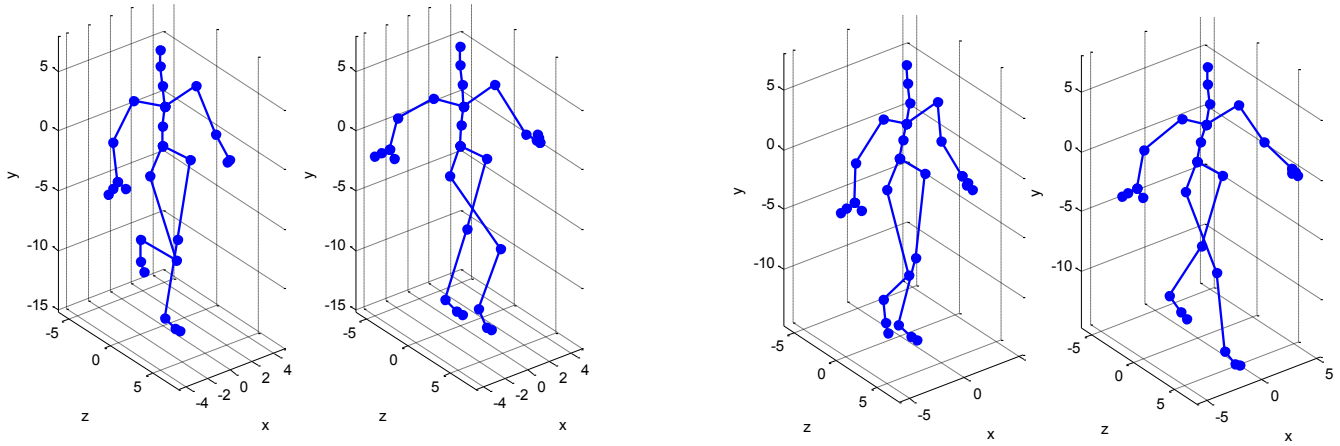
In our experiment, we recruited 8 participants and asked them to evaluate the results of motion generation based on the following criteria:

- The movement must respect the range of motion for each joint.
- The movement must not significantly violate physical law. For example, it is unacceptable to see the skeleton swimming in the air.
- It must be walking, and the pace must be close to one of the motions or lie in between the two.
- The resulting motion must contain some of the style of each sample. It is permissible if the style is not as significant as in the original samples as long as the related style cues are observable.

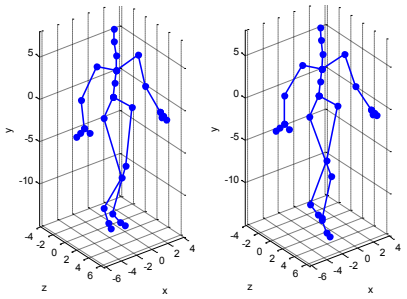
If a motion satisfies these four criteria, then we consider the motion generation successful. The evaluation results of four approaches are as follows.

FCRBM with style label interpolation. There are some generated motions that are acceptable, but most of them have two problems: (1) Most of the motions synthesized shake in an unnatural way. (2) The styles are averaged.

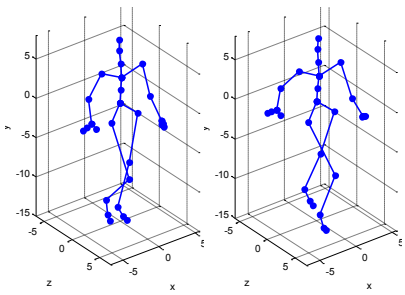
[1] `http://cs.nyu.edu/~gwtaylor/publications/icml2009/code/index.html`



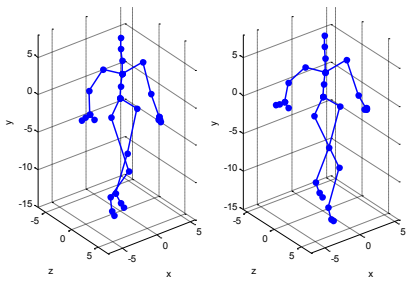
(a) March and Quick walk



(b) FCRBM with style label interpolation.



(c) HFCRBM with style label interpolation.



(d) HFCRBM with multi-path model.

Figure 7: (a) Representative frames of motions *March* and *QuickWalk*. (b)–(d) Motions generated via 50/50 interpolation between *QuickWalk* and *March*. The FCRBM with visible layer interpolation cannot blend two styles appropriately and therefore is not shown in the figure. As we can observe from (c) and (d), both approaches based on the HFCRBM catch the leg movements of *March*, and hand movements of *QuickWalk*, which are the most significant style features of the two motions. Subfigure (b) shows that the motion generated by the FCRBM with style label interpolation has a vague style from both samples.

In other words, those motions (ignoring the fact that many of them are shaking) may acceptably be considered "walking", and it is evident that they contain the styles from both motions, but the styles are quite vague. Some of them look similar to the motions generated from animation blending, as they both exhibit the phenomenon of averaging out the styles. Overall, ignoring the shaking properties and weakened styles, the approach has a 8.3% success rate for motion generation.

FCRBM with multi-path model. Applying the multi-path algorithm at the visible layer of the FCRBM, we achieved a success rate close to 32.8%. Characteristic of the resulting motions is that we usually can observe one style significantly while the other style is vague. In other words, the style blending of this approach is more like a competition than an averaging. Thus, when doing style interpolation, it has a success rate higher than 32.8% for generating an admissible walking motion, but some of them are evaluated as having failed because they did not successfully blend two styles.

HFCRBM with style label interpolation. The overall success rate for motion generation using this approach is 36.7%. Among its more successful results are that none of the motion shake, and the style from both component motions usually appear significant on the blended motions.

HFCRBM with multi-path model. The style quality of this approach is similar to some of the results of the HFCRBM with style label interpolation in that the styles of both motions are more apparent than in approaches based on the FCRBM. The overall success rate of this approach is 55%.

The experimental results show that the HFCRBM with hidden layer interpolation has a success rate 6.6 times higher than the previous work, and the blended style quality is the same as or better than the results of those approaches. Examples of motions generated by these approaches are plotted in Fig. 7.

7. CONCLUSIONS

We have proposed a method for style-content separation and motion style interpolation. Specifically, we developed the HFCRBM which learns style-based motion generation, and the multi-path model which performs style interpolation with the hidden layer. The approach produced motions with a success rate judged to be 6.6 times better than that in previous work using the FCRBM. We also demonstrated

that the multi-path model improves the FCRBM. The hierarchical structure provides the capability of hidden layer interpolation, which is the major improvement for the style interpolation approaches.

Although our algorithm yields better performance than the previous work, it still needs further improvement on the success rate for practical use. In part, this is due to the small training set comprised of highly different styles. It is also due to the model being trained without assigning any constraints. Walking is a complex behavior that must obey many biomechanical and physical constraints. To learn a good model for various walking motions, without using any constraints or domain knowledge, presents a considerable challenge. This suggests that adding domain knowledge to improve learning is a plausible way to improve the model without increasing the amount of training samples.

8. 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.

9. REFERENCES

[1] A. Bissacco. Modeling and learning contact dynamics in human motion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 421–428. IEEE Computer Society, 2005.

[2] M. Brand and A. Hertzmann. Style machines. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183–192, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[3] 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.

[4] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 522–531, New York, NY, USA, 2004. ACM.

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

[6] E. Hsu, K. Pulli, and J. Popović. Style translation for human motion. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1082–1089, New York, NY, USA, 2005. ACM.

[7] 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.

[8] M. Lau, Z. Bar-Joseph, and J. Kuffner. Modeling spatial and temporal variation in motion data. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, pages 1–10, New York, NY, USA, 2009. ACM.

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

[10] Y. Li, T. Wang, and H.-Y. Shum. Motion texture: a two-level statistical model for character motion synthesis. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 465–472, New York, NY, USA, 2002. ACM.

[11] C. K. Liu, A. Hertzmann, and Z. Popović. Learning physics-based motion style with nonlinear inverse optimization. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1071–1081, New York, NY, USA, 2005. ACM.

[12] G. Mather and L. Murdoch. Gender discrimination in biological motion displays based on dynamic cues. *Royal Society of London Proceedings Series B*, 258:273–279, 1994.

[13] V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 981–987. MIT Press, Cambridge, MA, 2001.

[14] C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph. Appl.*, 18(5):32–40, 1998.

[15] A. Shapiro, Y. Cao, and P. Faloutsos. Style components. In *GI '06: Proceedings of Graphics Interface 2006*, pages 33–39, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.

[16] 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.

[17] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1345–1352. MIT Press, Cambridge, MA, 2007.

[18] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[19] L. Torresani, P. Hackney, and C. Bregler. Learning motion style synthesis from perceptual observations. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1393–1400. MIT Press, Cambridge, MA, 2007.

[20] N. F. Troje, C. Westhoff, and M. Lavrov. Person identification from biological motion: Effects of structural and kinematic cues. *Perception & Psychophysics*, 67(4):667–675, May 2005.

[21] J. Wang, D. Fleet, and A. Hertzmann. Multifactor Gaussian process models for style-content separation. In Z. Ghahramani, editor, *Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007)*, pages 975–982. Omnipress, 2007.