Follow the Music: Dance Motion Synthesis Corresponding to Arbitrary Music

Yuan Yao
Department of Computer Science
University of British Columbia
Vancouver, Canada V6T1Z4
rozentil@cs.ubc.ca

Ruochen Wen
Department of Electronic and Computer Engineering
University of British Columbia
Vancouver, Canada V6T1Z4
ruochenwen@alumni.ubc.ca

Abstract

Dance choreography for an arbitrary music is an ill-posed problem and it's very challenging as there is no deterministic solution. Recently, due to the large effort made by researchers in Computer Vision, it's easy for people to extract both audio and visual features from video and directly predict body dynamics from music (1) or synthesize some simple motions corresponding to the music (2). But it is still a hard task for two reasons: 1) As choreography is an ill-posed problem, it is hard to determine a single sequence of dance motion for a given music. 2) The beats and rhythm from the two modalities should be aligned well to make the final results reasonable and natural. Inspired by these observation, we propose a two-step architecture for dance synthesis from a given music. We firstly generate a specified professional dance sequence using *Style Module*. Then we align the rhythm and beats from both music and dance sequences to create the final video in *Rhythm Module*. This approach is more robust and convenient as we can simply replace the style module for synthesizing different types of dance. It also has good generalization as the quality of final video is independent to the input music but only relies on the first step.

1 Introduction

In real life, choreography is widely used in a variety of fields, including musical theater, cheerleading, cinematography and etc. However it takes some time even for a professional dancer to choreograph for a new music. Besides, due to their expertise, they also can only design limited style of dance for a music. So here comes an interesting question, can we have a model which can do choreographing on arbitrary music using different styles of dance?

With the development of deep neural network, it is possible to construct a mapping from music or audio to visual representation. Researchers have proved that lip motion has large relationship with the speech audio and can be synthesized realistically (3; 4; 5; 6). Besides, body dynamics can also be inferred for a given music (1; 7) by training a multi-modal sequence model like long short-term memory network(LSTM). All of these have demonstrated that we can build a connection between audio and video and synthesize visual motion corresponding to an audio. But there are two main problems: 1) the synthesized motion is limited since most video data only has one audio corresponding to one visual sequence. 2) most of these methods do not care about the rhythm and beat which makes the results not realistic. Our approach proposed in this work mainly tries to solve these two problems using a two-step motion synthesis architecture.

Our main contributions in this work are:

- We propose a two-step architecture which consists of style module and rhythm module. Style module is used for synthesizing a specified dance motion using LSTM and VAE(variational auto encoder). Rhythm module is to warp the motion to match the rhythm of the input music.

- Our approach can easily synthesize different styles of dance motion for arbitrary music which no state-of-the-art method can achieve. To do this, we just need to train the style module on different styles of dance motion respectively and use different trained weights.
- Our method also has a great generalization as we have no limitation for the input music since our second step is to warp the pose sequence to match the audio. This means the input music won't affect the quality of the result.

2 Related Work

In this section, we will briefly discuss about the related works in audio-visual learning, motion synthesis and dance retargeting.

Audio-Visual Learning Multi-modal learning is always a hot topic in artificial intelligence as people are interested in finding the relationship between two different modalities like audio and video. Recently, there has been a large progress in talking face generation with respect to the input audio. Some works (6) regress the face expression using standard sequence model like LSTM, while others (5) use convolutional neural network to do this. Most recently, (4) generates the talking face by learning the disentangled audio-visual representation which can achieve more generalizable results.

Furthermore, people also tried to regress body dynamics using such technique. (1) also trains a LSTM for predicting the hand motion for violin or piano playing audio. Similarly (2) first use CNN for audio feature extraction, and then regress the pose sequence using LSTM. In addition, to make the results more realistic, (7) integrate rhythm information during the training.

Motion Synthesis Many people observe that recurrent neural network is good at generating a new sequence of samples from the training dataset. Based on this, (8) proposes an auto-conditioned LSTM for 3D motion synthesis which trains the network with both ground truth data and predicted results. To encode human motion into a more semantic space for learning, (9) presents a feed forward network on top of an autoencoder which synthesize the motion as a latent vector.

Dance Retargeting Most recently, there are many video retargeting works. (10) proposes a general framework for translating source video to a target style. More specific, (11; 12) try to transfer only the poses. However, there is few work care about the music-oriented retargeting as they are totally different modalities. A very new work (13) presents the observation on rhythm from the video, which can generate very decent results on retargeting the video to any audio.

3 Approach

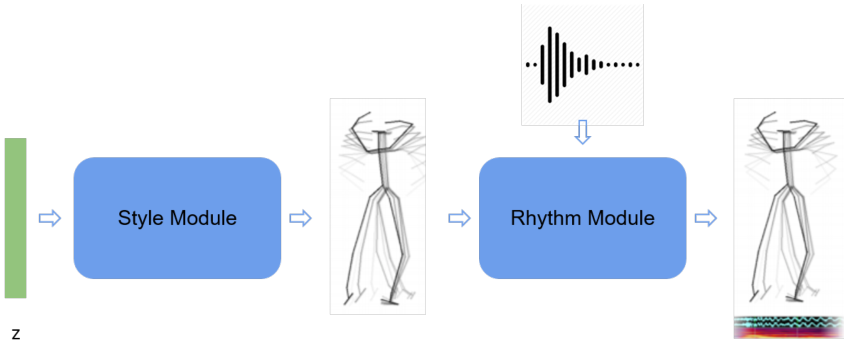


Figure 1: Overview of our approach. It is separated into two modules: **Style** and **Rhythm**. The **Style** module is only responsible for generating the specified style pose, while the **Rhythm** module is to warp the rhythm and beats of pose sequence to align with the music.

In this section, we mainly describe our proposed approach. We design a two-step architecture for music-oriented dance pose synthesis as shown in figure 1. Style module can generate a sequence of dance motions in a specified style. It consists of variational autoencoder(VAE) and long short-term memory network(LSTM) and is trained on a single genre of dancing data. When doing the testing, we can just randomly sample a noise **z** from latent space and send it to the module to generate a sequence of dance pose. In rhythm module, it allows input both the dance sequence and music sequence and output an aligned video. It extract the visual beats and rhythm and warp them with respect to the music. In the rest of this section, we introduce our two modules in detail.

3.1 Style Module

In this module, we aims to generate a specific style dance pose. We combine a three layer LSTM with a Variational Auto-Encoder. The architecture of this modules is shown in 2.

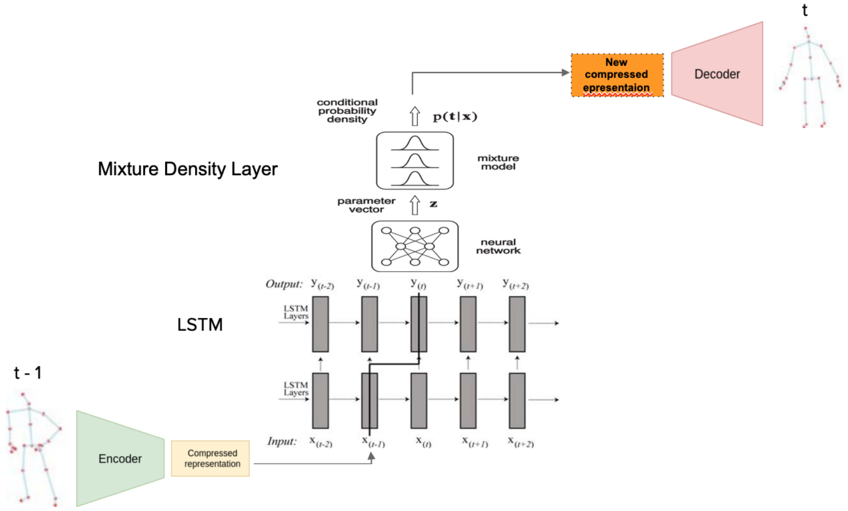

t
New compressed epresentaion
Decoder
conditional probability density
p(t|x)
Mixture Density Layer
mixture model
parameter vector
z
neural network
Output: y(t-2) y(t-1) y(t) y(t+1) y(t+2)
LSTM Layers
LSTM
t - 1
LSTM Layers
Input: x(t-2) x(t-1) x(t) x(t+1) x(t+2)
Encoder
Compressed representation

Figure 2: Architecture of Style Module

Long Short-Term Memory RNN has been proved successful on the sequence to sequence task. LSTM is more stable than vanilla RNN over long training runs and can be stacked into deeper network without loss of stability. Since the dance pose data is a multidimensional time series, we consider to use a three-layer LSTM model, input a series of a specific style of continuous dance frames, and output the same style of dance pose sequences.

Mixture Density Network Regarding to the problem of LSTM whose output would converge to an average pose eventually using a mean square error metric, we attach a Mixture Density Network(MDN) on the top of the vanilla LSTM. We construct an MDN by creating a neural network to parameterize a mixture model consisting of Gaussian distributions. The linear combination of K Gaussian distributions enable MDN to output a probability distribution for each dimension in the pose vector rather than a single value. Formally, the we can represent the conditional probability $p(y \mid x)$ as

$$p(y \mid x) = \sum_k \pi_k(x) N(y \mid \mu_k(x), I\sigma^2(x)) \tag{1}$$

Where $\pi(x)$ is a normalized vector of k mixing coefficients as a function of x, here x is the output of regular LSTM. $N(y \mid \mu_k(x), I\sigma^2(x))$ is a Gaussion distribution conditioned on x with corresponding mean $\mu_k(x)$ and varience $\sigma^2(x)$. We can construct loss function of MDN using negative log-likelihood and optimize it with parameter w

$$L(w) = \frac{-1}{N} \sum_{n=1}^{N} log(\sum_k \pi_k(x_n, w) N(y_n \mid \mu_k(x_n, w), I\sigma^2(x_n, w))) \tag{2}$$

Where N is batch size for training.

Therefore, we are able learn a more complex structure to the data and avoid the problem of sticking pose.

Variational Auto-Encoder To make it easier for LSTM to train, we use a Variational Auto-Encoder to represent each frame in a compressed space instead of input the whole image. To be concrete, the input of encoder x_i is a 120 by 208-pixel image. The encoder encodes the information of frame $\imath t - 1 \jmath$ which is a 24960-dimensional vector into its compressed representation whose dimensional is only 128. Then, the compressed vector is fed into the LSTM to generate a new compressed vector. Essentially, the decoder reconstructs the same style dance pose as input at frame $\imath t \jmath$ from the output of the LSTM.

The loss function of VAE consists of 2 parts, a reconstruction loss with a regularizer. The loss function of one data point x_i can be represented as

$$l_i(\theta, \phi) = E_{z \sim q_\theta(z|x_i)}[logp_\phi(x_i \mid z)] + KL(q_\theta(z \mid x_i) \parallel p(z)) \tag{3}$$

The first term is a negative log-likelihood of $i - th$ data point representing reconstruction loss. z is the hidden represent and $q_\theta(z \mid x_i)$ represents the encoder with biases θ. $p_\phi(x_i \mid z)$ represents the decoder. The expectation is taken with respect to the encoder's distribution over the representations. This term encourages the decoder to learn to reconstruct the input. The second term is Kullback-Leibler divergence between the encoder's distribution $q_\theta(z \mid x)$ and $p(z)$.

The total loss is then summed up as

$$L = \sum_{i=1}^{N} l_i \tag{4}$$

for N total data points. Our ultimate goal is to minimize L for VAE.

3.2 Rhythm Module

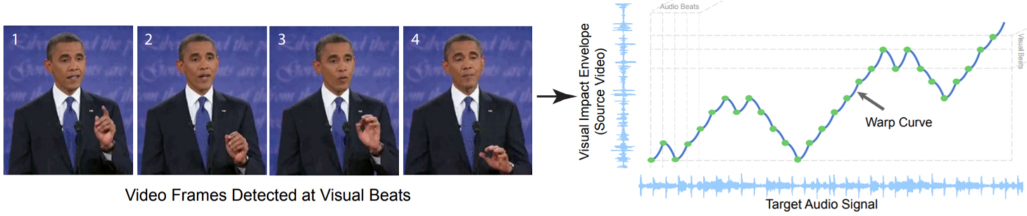


Figure 3: Visual beats description from (13). These visual beats lie at the high and low points of a repetitive up-and-down hand gesture. On the right is a warp curve showing the process of unfolding, which synthesizes dance video corresponding to a random walk through the visual beats of a source segment.

In rhythm module, it mainly focuses on retargeting the input pose sequence to match the rhythm of the input music. (13) observes that there is a way to figure out the rhythm and beats from video which is called visual beats. In many videos which contain human motion, the repetitive up-and-down gesture can easily be treated as visual beats. To factor the motion into different angles, they use directogram which is computed using optical flow. The method then extract the impact envelope from directogram and onset envelop from spectrogram of the music. Finally they do the warping to match the visual beats the the music as figure 4 shows.

4 Experiment Setup

This section describes the experiment of our methods. We first describe the datasets and relative preprocessing we used, then the details of our models.

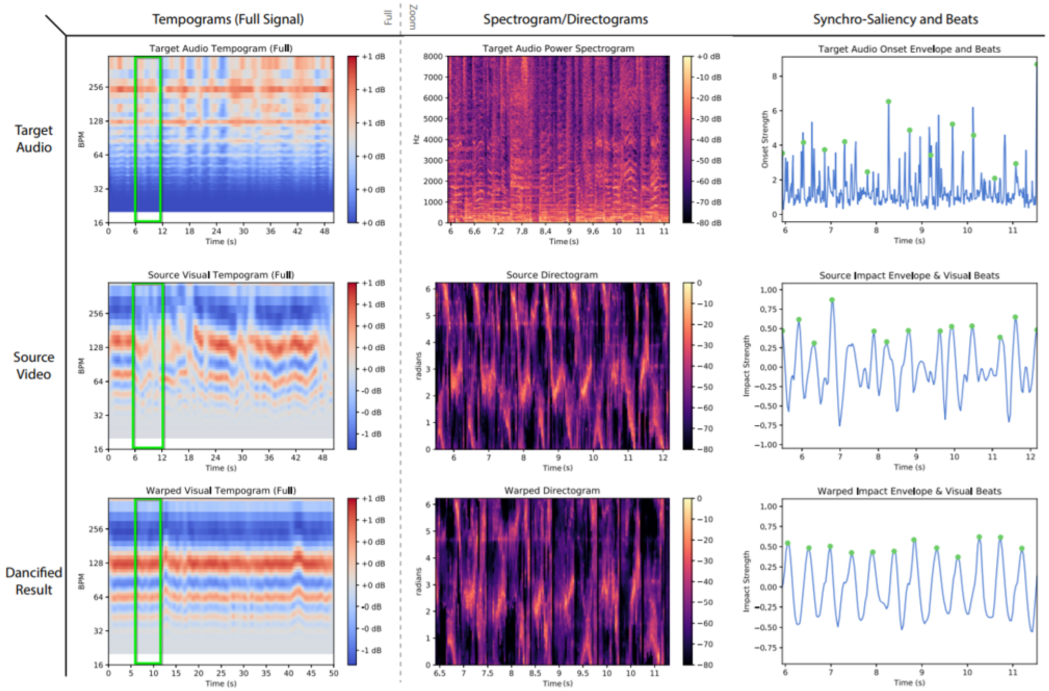


Figure 4: Dancification in the **Rhythm** module — The top row visualizes a tempogram (left), power spectrogram (middle), and an onset envelope (right). The spectrogram and onset envelope correspond to the boxed region of the tempogram, and green dots on the onset envelope show the locations of musical beats. The middle row visualizes the corresponding visual complements for a video. The bottom row shows what those same complements look like after retargeting the video to Canned Heat. From the left column we can see that the music has a mostly constant tempo around 128bpm, while the visual tempo of our source video varies quite a bit over time. After dancification, visual tempo is more constant and has been shifted to match that of the target audio. In the right column we can see how dancification shifts visual beats that are irregularly distributed in the source video into alignment with the tempo of the target audio.

4.1 Datasets

Video Dataset For now, only one type of dance, Popping, is used for training and evaluating our model. All the videos are downloaded from YouTube from two dancers "Poppin John SBK" and "MARQUESE SCOTT". Those videos all meet the requirements of background without other participants, barely non-moving camera, single dancer, body and feet are always in the field of view, and their video duration can reach 1.5h or more respectively. In total, we have more than three hours Popping dance video. We reset all videos to 24fps. We use OpenPose to detect keypoint representation in the video, and extract each 3 frames in each video as an image, resulting in 13772 images. Then we resize each image to size of 120 * 208 and normalize them to (0,1). Our raw data is a series of label-less videos. We use the pose image at time t-1 as input, and pose at time t as ground truth.

Audio Dataset We download 5 different music to align with our generated pose video. Although they can all be used as accompaniment songs for popping, there are still differences existing in whether there are singers, rhythms, and melody to some extent. We take a 20-second clip from each song to match our generated videos.

4.2 Implementation Detail

LSTM For training LSTM, we use Adam optimization with a learning rate of 0.0005. The batch size is 200, time step is 5. A dropout with the rate of 0.4 is applied to each layer of LSTM. Mixture components set to 24 for MDN.

See details for experiments we have conducted in 1.

Extracting Per/frame	TimeStep	BatchSize	MixComponent
1	1	1024	24
1	1	1024	50
3	5	200	24
1	10	100	24
3	20	50	24

Table 1: LSTM Settings

Variational Auto-Encoder For VAE, the latent dimension for compressed vector is set to 128. we use RMSprop with a learning rate as 0.001. The batch size is 100. The network was trained for 100, 200, 500 epochs seperately.

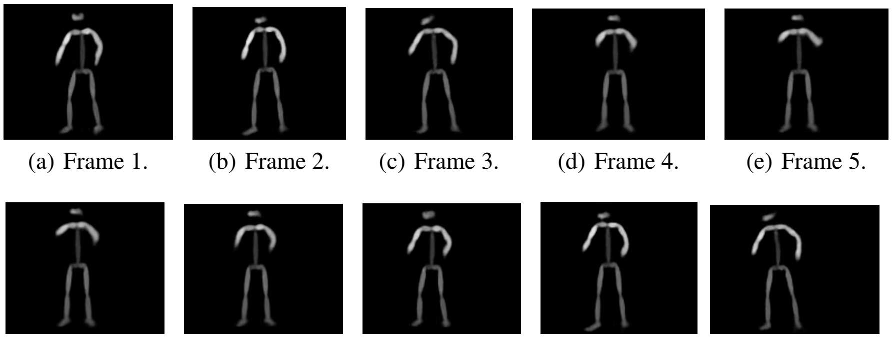
Alignment The visual beats of the generated video are calculated, the impact envelope are extracted and then wrapped together for video-music alignment.

5 Result and Discussion

5.1 Result

The main result of this work is the following and as be seen in 5. For LSTM, the setting of extracting image per 3 frames, time step with 5, batch size with 200, 24 mixture components and 20000 epoch performs better than others. LSTM plus MDN, extracting image per 3 frames rather than 1 frame imporves the problem of some static poses during the videos, We assume that because Popping has several fixed-point actions, leading to the situation that same poses are possible to be learned repeatedly. Besides it takes longer time to train the model with extracting image per 3 frames. Changing number of mixture components and time step regarding to our settings in the last section do not have a significant effect on our result. For VAE, the setting of 200 epoch with GREY image performs better. With 100 epoch, the edges of poses are still fuzzy; with 500 epoch the poses appear some irregular changes which might be overfitting.

The generated video then match the 5 arbitrary music. The music with stronger beats aligns better with our Popping dance video and performs more reasonable. It makes sense since the movements of Popping dance mostly change with the beats.



(a) Frame 1. (b) Frame 2. (c) Frame 3. (d) Frame 4. (e) Frame 5.

(f) Frame 6. (g) Frame 7. (h) Frame 8. (i) Frame 9. (j) Frame 10.

Figure 5: Output Frames.

User Study We also conduct user study to verify the visual quality of the network. 15 users are performed a simple task answering 2 questions: In the generated video, whether the generated dancing pose are reasonable, whether the generated dancing pose are similar to the ground truth. Our model would be evaluated according to their ratings.

5.2 Discussion

Overall our model is able to create reasonable and arbitrarily dance videos given keypoint image input. Although our setup can produce plausible results in many cases, occasionally our results suffer from several issues. The generated poses persistent jitter during moving, this might be due to the fact that we did not add the temporal smoothing constraint during training. Besides, in some cases, the pose would still converge to fixed pose, the model may still be in underfitting, and we will try more epochs. Or try to improve the structure of the existing LSMT plus MDN. Then we would also try to input audio feature combining with poses into our LSTM to generate new poses under the condition of designated style music. There are also some artifects appearing sometimes, cause the input consists of multiple videos, And we haven't handled the sudden changes of pose among videos yet.

6 Conclusion

In this work, we showed that the combination of our style module and rhythm module successfully synthesis a specific style of dance poses and align well with arbitrary music. There are a number of interesting paths to explore for future work. This includes trying other types of dance like Ballet, generating realist video, combining audio feature with poses as the input for LSTM.

7 Work

In this project, we have two people in our team, and we did all the literature review and experiments together.

References

[1] E. Shlizerman, L. Dery, H. Schoen, and I. Kemelmacher-Shlizerman, "Audio to body dynamics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7574–7583.

[2] N. Yalta, S. Watanabe, K. Nakadai, and T. Ogata, "Weakly supervised deep recurrent neural networks for basic dance step generation," *arXiv preprint arXiv:1807.01126*, 2018.

[3] S. Taylor, T. Kim, Y. Yue, M. Mahler, J. Krahe, A. G. Rodriguez, J. Hodgins, and I. Matthews, "A deep learning approach for generalized speech animation," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 93, 2017.

[4] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang, "Talking face generation by adversarially disentangled audio-visual representation," *arXiv preprint arXiv:1807.07860*, 2018.

[5] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, "Audio-driven facial animation by joint end-to-end learning of pose and emotion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 94, 2017.

[6] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: learning lip sync from audio," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 95, 2017.

[7] T. Tang, J. Jia, and H. Mao, "Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis," in *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 1598–1606.

[8] Z. Li, Y. Zhou, S. Xiao, C. He, and H. Li, "Auto-conditioned lstm network for extended complex human motion synthesis," *arXiv preprint arXiv:1707.05363*, vol. 3, 2017.

[9] D. Holden, J. Saito, and T. Komura, “A deep learning framework for character motion synthesis and editing,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 138, 2016.

[10] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, “Video-to-video synthesis,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[11] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, “Everybody dance now,” *arXiv preprint arXiv:1808.07371*, 2018.

[12] N. Neverova, R. Alp Guler, and I. Kokkinos, “Dense pose transfer,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 123–138.

[13] A. Davis and M. Agrawala, “Visual rhythm and beat,” *ACM Trans. Graph.*, vol. 37, no. 4, pp. 122:1–122:11, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3197517.3201371