AUTOMATIC CHOREOGRAPHY GENERATION WITH CONVOLUTIONAL ENCODER-DECODER NETWORK

Juheon Lee **Seohyun Kim** **Kyogu Lee**
Music & Audio Research Group, Seoul National University , Korea
{juheon2, shzkim, kglee}@snu.ac.kr

ABSTRACT

Automatic choreography generation is a challenging task because it often requires an understanding of two abstract concepts - music and dance - which are realized in the two different modalities, namely audio and video, respectively. In this paper, we propose a music-driven choreography generation system using an auto-regressive encoder-decoder network. To this end, we first collected a set of multimedia clips that include both music and corresponding dance motion. We then extract the joint coordinates of the dancer from video and the mel-spectrogram of music from audio and train our network using music-choreography pairs as input. Finally, a novel dance motion is generated at the inference time when only music is given as an input. We performed a user study for a qualitative evaluation of the proposed method, and the results show that the proposed model is able to generate musically meaningful and natural dance movements given an unheard song. We also revealed through quantitative evaluation that the network has created a movement that correlates with the beat of music.

1. INTRODUCTION

Choreography is a kind of art that designs a series of movements. In particular, in performing art, choreography extends to the use of human bodies to express movements, and these are often performed with music. The choreography suitable for music has significance in that it is not only an artwork itself, but also maximizes the expression of music [4, 7]. For this reason, choreography has become an essential element in many pop music works in recent years. Therefore, the process of creating choreography for music is also considered to be important, and research on a system capable of automatically generating choreography is actively conducted. However, automatic choreography generation is a challenging task because both music and dance are abstract art concepts, and the clear relationship between the two concepts is also not defined by established rules.

In this paper, we proposed a music-driven choreography generation system. In order to model the relationship between music and movement, two concepts in different domains, we firstly designed an autoregressive sequence to sequence model based on neural network that has been actively studied recently. Sequence used as input is time series data with strong correlation between adjacent timestep. Therefore, we designed a network of causal-dilation convolutional layers to fully reflect the information in the adjacent frame. We also applied local conditioning methods to the network to ensure that information related to music is effectively conditioned in the process of creating choreography movements. To evaluate whether a trained network actually produces a dance motion that matches music, we conducted a user study that evaluated naturalness by comparing video that matched random choreography with music and video generated by the proposed network. We also proposed a comparison of the two sequences' auto-correlations to analyze whether the choreography actually reflects music. As a result, we confirmed that the proposed network produced choreography that better reflected music than randomly matched videos, and that the joint movements of the generated choreography had a periodicity similar to the tempo of the music.

The contribution of this paper is as follows: First, we designed a music driven choreography generation network trained by an end-to-end method. Second, to generate choreography reflecting music, we successfully applied a local conditioning method used in speech synthesis field. Third, for the task of creating a choreography that is relatively difficult to assess quantitatively, we proposed the evaluation method using auto-correlation and user evaluation.

The rest of the paper is organized as follows. Studies related to this paper are introduced in Section 2. In Section 3, we explain in detail our proposed method for choreography generation based on the encoder-decoder network. We describe the dataset for experiments and the training process in Section 4. The evaluation scheme and the results are presented in Section 5, followed by conclusions and directions for future work in Section 6.

2. RELATED WORK

Recent advances in machine learning and deep learning techniques have led to a variety of attempts to study the relationship between dance and music. Lee et al. proposed

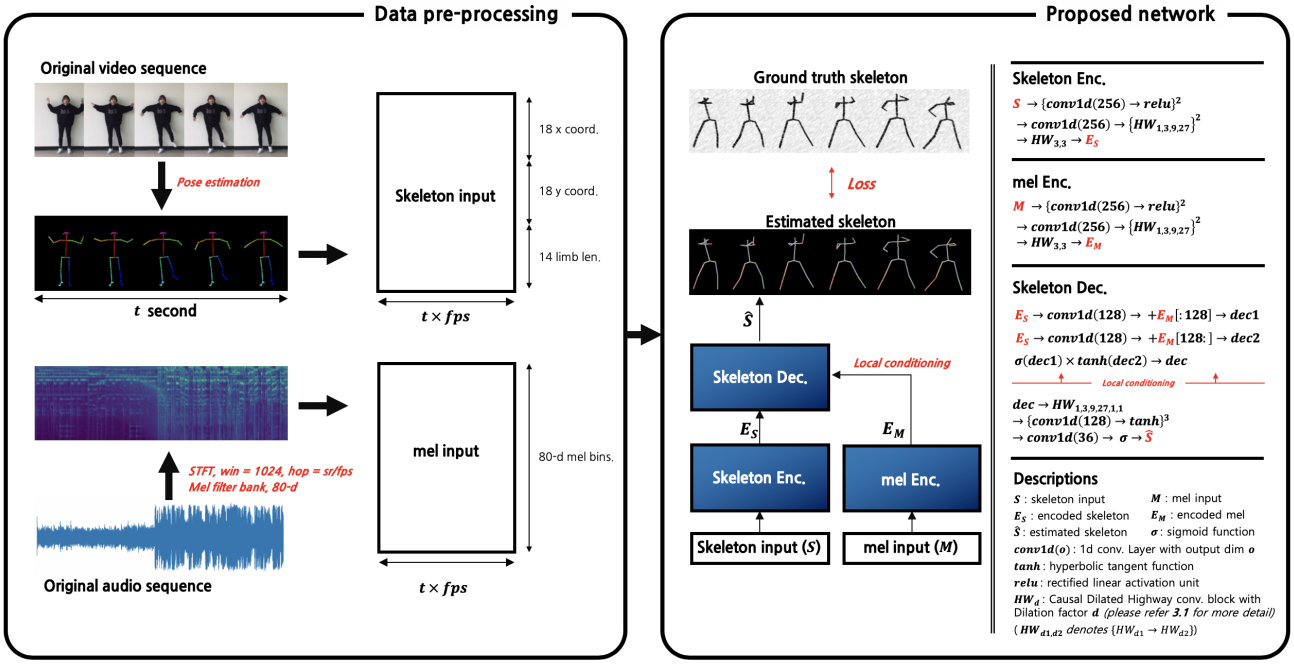


Figure 1. A schematic diagram of the proposed music-driven choreography generation system.

a choreography generation algorithm that retrieves the motions corresponding to the most similar pieces of music in the predefined motion-music-paired database for a given new music segment. [8]. This method selects dance motion from a predefined database, so choreography retrieved with high correlation with music is guaranteed. However, it has limitations in that it can not create novel dance movements that are not included in the database. Ofil et al. proposed a HMM-based model that categorizes the genre of music based on the Mel-Frequency Cepstrum Coefficients (MFCC) [10] feature and generates matching choreography based on the results [11]. But since the choreography is determined by the categorical value obtained through the genre classifier, there is a limit to generate a novel choreography. Omid et al. proposed a music-driven choreography model named Groovenet [1]. They used pairs of music and three-dimensional motion data to train the Factored Conditional Restricted Boltzmann Machines (FCRBM) [14]. They attempted to directly train the relationship between music and dance by using the mel-spectrogram in the training process. However, they reported that their model created awkward dance moves for unheard song, so they conclude that the model was overfitted and the dance moves according to music were not generalized enough.

Lee et al.'s and Ofil et al.'s studies have a limitation in that they can not create novel choreography because the former synthesizes motion by reusing the choreographic samples in a predefined database, and the latter creates choreography only for music input categorized by its genre. Omid et al. did succeed to create novel dance motions, but failed to yield good results mainly due to insufficient training data of merely 23 minutes.

In this study, we proposed a music-driven choreography generation system that can produce novel and natural choreography. [1] In order to secure the novelty of choreography, we used the method of creating choreography with frame by frame generation, not the method of retrieve in the pre-defined dataset. Also, to train the network with sufficient data, we also proposed a way to use the choreography-music data pairs that can be easily obtained from online video sharing community as training data. Finally, in order to conduct effective conditioning of music information, we have applied the methods used in other conditional sequence generation tasks effectively to our task.

3. PROPOSED APPROACH

In this chapter we explain the detailed structure of the proposed network. An overview of the proposed system is illustrated in Figure 1.

In order to learn the relationship between the time-series data of two different modalities, i.e., music and dance, we need a model that performs multi-modal sequence-to-sequence transformations. Also, since the choreographic movement at a certain time-step has a strong correlation with the information at the previous time-step, we should consider a system that provides sufficient reference to the information at the adjacent time-step. From this point of view, we have noted a text-to-speech system that shows reliable performance in a similar environment to these conditions, and then designed our system, inspired by the DCTTS [13] model, which is known to be capable of efficient text-to-speech training.

Our proposed model takes skeleton input S and mel input M as input, to predict skeleton $\hat{S}$ in the next time step:

[1] The generated result can be found at: listentodance.strikingly.com.

$$\hat{S}_{1:T} = CG(S_{0:T-1}, M_{1:T}) \quad (1)$$

where CG denotes our proposed model, choreography generator. For this purpose, each input is encoded via two encoder first. Then the encoded skeleton E_s passes through the decoder and predicts the next time step's skeleton $\hat{S}$, which utilizes the encoded E_M as conditioned information.

3.1 Causal Dilated Highway Conv. Block

In this section we explain the Causal Dilated Highway Convolution Block, one of the core structures of the proposed network. The choreography, which is basically the object that we should create, has a strong correlation with the information of the adjacent time-step. Therefore, in order to predict movement in the next time-step, information from previous time-steps should be fully consulted. Also, for choreography, a wide range of historical information should be referred to because it has relatively long-term dependency. To this end, we use the Causal Dilated Convolution. *Causal* means that only the input data from time 0 to $t-1$ can be referred to when calculating the output at time t. We used a causal convolution layer because our network must be an auto-regressive model to generate the next frame that is not yet known from the preceding frames. In addition, we used the *dilated convolution* proposed in the Wavenet [16] to ensure that the model has a wider receptive field. Finally, to enable efficient training even in deep model structures, we used a *highway network* architecture [12] where gated function could be trained. That is, the output of the CDHC block is calculated as:

$$\textbf{output} = tanh(\text{H1}) \cdot relu(\text{H2}) + (1 - tanh(\text{H1})) \cdot \textbf{input} \quad (2)$$

where [H1, H2] is the tensor calculated through the causal dilated convolution layer of the input tensor. The output channel of this convolution layer is twice the input channel, and the kernel size is 3.

3.2 Encoder & Decoder structure

To predict the next time-step skeleton information from the given input information, we used a method of effectively encoding input information and then combining them to decode. To this end, we designed two encoder and one decoder with CDHC block. Both the skeleton encoders and the audio encoders all consist of three convolution layers and 10 CDHC blocks. The first convolution layer of each encoder increases the input channel to 256 dimensions, and the other two layers perform 1x1 convolution. Thereafter, the output values from last convolutional layer are connected in sequence to 10 CDHC blocks with a dilation factor of (1,3,9,27,1,3,9,27,3,3), and the corresponding operations result in audio and skeleton data are encoded to have a sufficiently wide receptive field to reflect sufficient past information.

A decoder is a network that generates skeleton data for the next frame from an encoded skeleton and an encoded audio. To do this, the encoded skeleton input to the decoder is combined with the encoded audio in the following:

$$Dec1 = conv1d(E_S) + E_M[:128] \quad (3)$$

$$Dec2 = conv1d(E_S) + E_M[128:] \quad (4)$$

$$Dec = \sigma(Dec1) \times tanh(Dec2) \quad (5)$$

Where E_S and E_M refer to the encoded skeleton and encoded audio, respectively, and $conv1d$ means the convolution layer with an output channel of 128 and a kernel size of 1. The combined Dec tensor then goes through six CDHC blocks with a dilation factor of (1,3,9,27,3,3) and then through three 128-channel convolutional layers with a $tanh$ activation function. Finally, after passing through a convolution layer with the same output channel as the dimension of the target, the final decoder output is obtained via $sigmoid$ activation.

3.3 Proposed network

This network receives skeleton and music data from time 0 to $t-1$ as input. Both data are encoded via encoders and combined at the beginning of the decoder. The final output of the decoder is compared with the ground truth motion data at time 1 to t and we used it as a $L1$ loss. Since all convolution operations included in the network are with kernel size 1 or causal operations, the k-th value of output refers to only the 0 to $k-1$ time step of the input during the operation. Therefore, the model satisfies the causal condition.

4. EXPERIMENT

4.1 Data

We have collected 100 YouTube choreography videos and corresponding audios. The genre was selected mainly for K-pop dance, and the total length of collected data was 6.26 hours. We divided 85 songs into train sets, 5 songs into valid sets, and 10 other songs into test sets, to train and evaluate the proposed network.

4.1.1 Skeleton data

We extracted the x, y coordinates of 15 human body joints from each frame using the Openpose algorithm [3] from the collected video as shown in Fig. 2. Next, we min-max normalize the extracted coordinate values for each video, and use the linear-interpolation for the unrecognized coordinate values.

Since we can not measure the exact 3d angle between the human body limbs using the 2d joint coordinate, we used the absolute coordinates values of each point as the training target. However, in this case, the length of each limb in the projected skeleton can vary, and awkward motion can be generated if the model learns it incorrectly. So we additionally calculated the lengths of the 14 main limbs together and added a loss to compare with the limb length of the skeleton that the model generated. Therefore, the x, y coordinates of the total 15 joints, and the total of 14 main limb length are used as skeleton data.

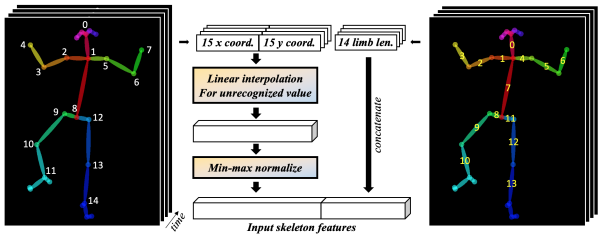


Figure 2. The process of extracting skeleton data from video frames.

4.1.2 Music data

We separated the audio contained in the collected video and used it as music data. The mel-spectrogram was extracted from the audio waveform with the window size of 1024 samples, and 80 mel-frequency bins. Because we need time-aligned audio-video pairs for training, we adjusted the hop size when extracting the mel-spectrogram so that audio and video data end up with the same frame rate.

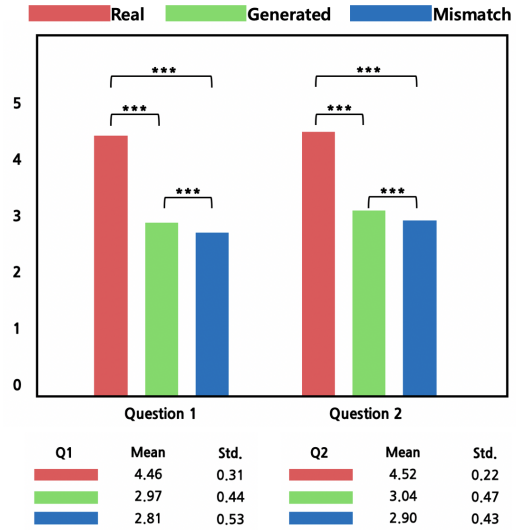
4.2 Training

We have trained the proposed network that creates the next skeleton coordinate for a given previous skeleton sequence and music sequence. To do this, we first input skeleton data and music data from 0 to t-1 frames. Then, the output of the network is compared with the ground truth choreographic data corresponding to 1 to t frame by use L1 loss as a cost function. In addition, we calculated the length of each limb from the skeleton data of the generated frame, and compared with the actual ground truth length through the L1 loss.

We used the adam optimizer [6], with $\beta_1 = 0.5$, $\beta_2 = 0.9$, for training and set the learning rate to 0.0002. At every iteration, we used a video-audio pair that was cut in 500 frames for training, and it contains about 20 seconds of choreography and music information. The length of the input sample was set to 500 frame because we decided that the sequence of lengths, which fully reflected meaningful levels of behavior in the choreography, should be used for training. We set the batch size to 16, and then we finished the training after proceeding with a total of 30,000 iterations. We trained our network with one GEFORCE GTX 1080 ti GPU for three days.

4.3 Inference

The choreography inference process is performed in an auto-regressive manner different from training. That is, the initial position of each joint is given as an input skeleton frame, and at the same time, the first frame of mel-spectrogram is input to the trained model. When inference is performed once, estimated skeleton at $t = 1$ is output. Then we concatenate skeleton at $t = 0$ and $t = 1$, then input them back into the model with mel-spectrogram at $t = 0$ and $t = 1$. After than, we get estimated skeleton at $t = 1$ and $t = 2$. Therefore, we can generate the choreography by repeating the above process for the length of music input, and used it to evaluate the generated choreography.



Q1	Mean	Std.	Q2	Mean	Std.
	4.46	0.31		4.52	0.22
	2.97	0.44		3.04	0.47
	2.81	0.53		2.90	0.43

Figure 3. Average Likert-scale user scores on two questions (Q1: Is choreography natural? / Q2: Does choreography fit well with music?). The table below the graph indicates the mean and variance of responses by model for each question. The p-values for pairwise comparisons between the groups are also shown at the top. ***: $p < 0.001$; **: $p < 0.01$.

5. EVALUATION & RESULTS

5.1 User study

We conducted a user study to evaluate whether the generated choreography was natural and whether it was produced in accordance with the music. First, we generated 20 videos for each of the three groups: *Real*, *Generated*, and *Mismatch*. Group *Real* consists of music A_i and actual choreography for music A_i. Group *Generated* consists of music B_i and novel choreography generated by our model given music B_i. Finally, the group *Mismatch* consists of music C_i and novel choreography generated by our model but with randomly selected music rather than C_i. Music A_i, B_i, and C_i were randomly selected among the songs included in the validation dataset that was not used in training, and the length of each audio/video was 16 seconds.

After mixing the three groups of videos in a random order, we asked the participants whether each video's choreography is natural (Question 1) and whether it fits well with music (Question 2), and to give a score in a Likert scale [2]. After collecting the responses, we performed isoquantity and normality tests using data averaging 20 responses from each group, to see if there was a difference in the mean of the responses of the groups. After evaluating significance through repeated-measure ANOVA test , further post-hoc paired t-test analysis was performed to calculate the p-value, and the difference between the groups was examined [5].

A total of 33 participants answered the questionnaire

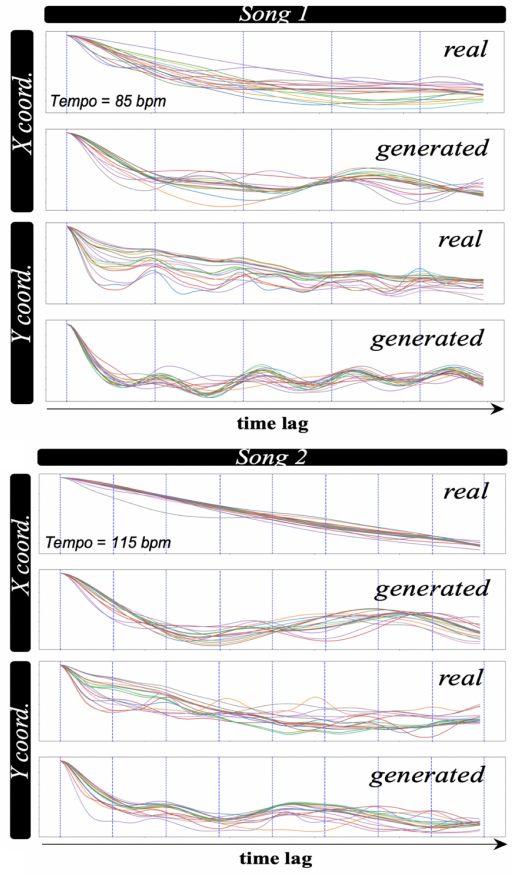


Figure 4. Autocorrelation of the x,y coordinates of each joint from real and generated choreography for two songs. The x-axis of each graph represents the time lag, y-axis of each graph represents the autocorrelation, and the blue vertical lines represent the beat positions of each song.

and the results are shown in the Figure 3. The results of the statistical tests confirmed that the mean scores between the three groups were significantly different for both questions. Average user score for both questions were highest in *Real* group and lowest in *Mismatch* group. It is clear that the *Real* group score is the highest, because it is made up of the choreography created by the human. The average score of the *Generated* group surpassed the *Mismatch* group in both questions. If the proposed model generates choreography that is not associated with music, participants will have a similar response, regardless of what music is played with the generated choreography. However, from the fact that the video received a significantly higher score when played with the music used in choreography generation, we judged that the proposed model produced choreography that listen and reflects the music.

5.2 Autocorrelation Analysis

We also performed an autocorrelation analysis to further investigate the differences between the generated choreography and the actual choreography. Autocorrelation is a correlation between a given sequence with itself, reflecting the periodic properties of the sequence. We can identify the periodic component of a given sequence through the location of the peaks observed in the autocorrelation results. Using this, we analyzed the motion by calculating the autocorrelation on the x, y coordinates of the choreography movement and compared it with the tempo of corresponding music. Our hypothesis was that if the model can produce dance by listening to the music, the autocorrelation peak position of the motion will appear at the same point as the beat of the music.

Fig. 4 shows the autocorrelation results of two choreography samples along with the tempo of corresponding music. In actual choreography, a clear peak is observed in y-direction movement, but not in x-direction movement. This tendency is also observed in the generated choreography. From this we can determine that the proposed network has learned the periodic tendency of the real choreography used in training. Also, In actual choreography, the first or second peak of the y-direction auto-correlation appears at the same position as the music beat. This means that music and choreography have similar periodic properties. This tendency can be confirmed also in the case of the generated sample. From this, it is judged that the proposed model has generated the choreography that listen the music and reflects its periodic nature.

6. CONCLUSION

In this study, we proposed an auto-regressive encoder-decoder network that generates matching choreography for a given music input. We used audio-video pairs data obtained from YouTube for training. As a result, it was found that motions matching with the music were generated through comparison of user study and autocorrelation analysis. This study has a significance in that it shows a significant performance in the area of learning-based choreography generation, in which sufficient performance has not been secured yet. Also, it is meaningful not only to learn the movement of dance but also to use the relationship with music together for generation.

Although we found in this study that the choreography generated compared to the mismatch group has a higher correlation with music at a significant level, we still have the limitation of having a large difference score from the real group. To overcome this, we will further model the correlation between movement and music more elaborately and carry out follow-up studies that reflect it in the network architecture. Also, this research has limitations that generated choreography reflects only the periodicity among various properties of music. Ultimately, it is necessary to create appropriate choreography according to various genres, moods, and contexts of music as well as periodicity. In order to do this, we plan to establish data sets that satisfy various conditions and carry out further research. In addition, we use 2-d skeleton position for training, and it is difficult to use this type of data in case of needing actual implementation such as a robot. Therefore, the extension of the model to 3-d choreography generation using the improved 3-d pose estimation algorithm [9, 15, 17] is also a future research topic.

7. ACKNOWLEDGEMENTS

This work was supported partly by LG Electronics and partly by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2017M3C4A7078548).

8. REFERENCES

[1] Omid Alemi, Jules Françoise, and Philippe Pasquier. Groovenet: Real-time music-driven dance movement generation using artificial neural networks. *networks*, 8(17):26, 2017.

[2] I Elaine Allen and Christopher A Seaman. Likert scales and data analyses. *Quality progress*, 40(7):64–65, 2007.

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[4] Sommer Gentry and Eric Feron. Modeling musically meaningful choreography. In *Systems, man and cybernetics, 2004 IEEE international conference on*, volume 4, pages 3880–3885. IEEE, 2004.

[5] Ellen R Girden. *ANOVA: Repeated measures*. Number 84. Sage, 1992.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Carol L Krumhansl and Diana Lynn Schenck. Can dance reflect the structural and expressive qualities of music? a perceptual experiment on balanchine's choreography of mozart's divertimento no. 15. *Musicae Scientiae*, 1(1):63–85, 1997.

[8] Minho Lee, Kyogu Lee, and Jaeheung Park. Music similarity-based approach to generating dance motion sequence. *Multimedia tools and applications*, 62(3):895–912, 2013.

[9] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018.

[10] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[11] Ferda Ofli, Yasemin Demir, Yücel Yemez, Engin Erzin, A Murat Tekalp, Koray Balcı, İdil Kızoğlu, Lale Akarun, Cristian Canton-Ferrer, Joëlle Tilmanne, et al. An audio-driven dancing avatar. *Journal on Multimodal User Interfaces*, 2(2):93–103, 2008.

[12] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[13] Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. *arXiv preprint arXiv:1710.08969*, 2017.

[14] Graham W Taylor and Geoffrey E Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th annual international conference on machine learning*, pages 1025–1032. ACM, 2009.

[15] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2500–2509, 2017.

[16] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, page 125, 2016.

[17] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5255–5264, 2018.