

Audio to Body Dynamics

Eli Shlizerman^{1,3}, Lucio Dery², Hayden Schoen¹, and Ira Kemelmacher-Shlizerman^{1,3}

¹Facebook Inc.

²Stanford University

³University of Washington

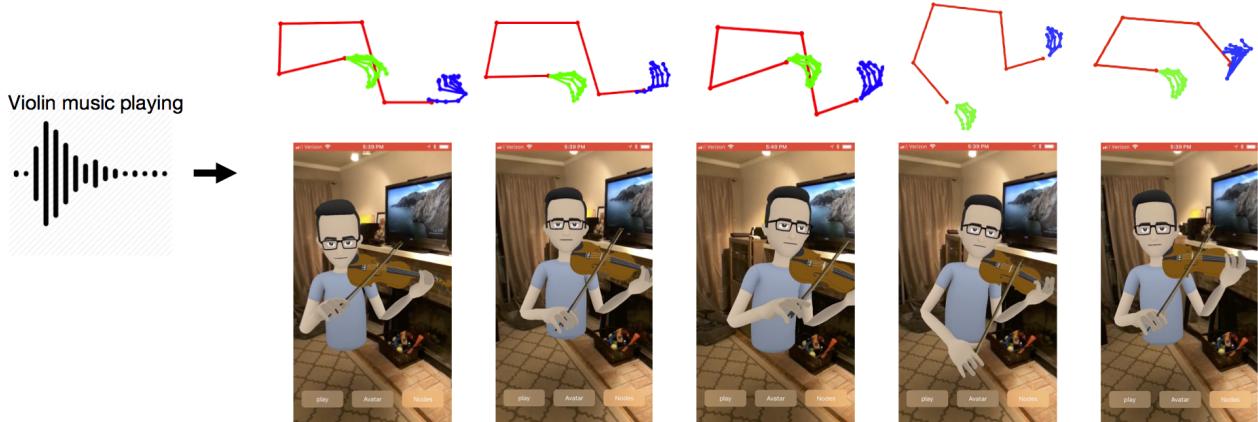


Figure 1. Given as input a music of violin (audio signal), our method (1) predicts body skeleton and (2) uses the skeleton to animate an avatar. See videos in the supplementary material (with audio on).

Abstract

1. Introduction

All the same it is being said everywhere that I played too softly, or rather, too delicately for people used to the piano-pounding of the artists here.

Frederic Chopin

We present a method that gets as input an audio of violin or piano playing, and outputs a video of skeleton predictions which are further used to animate an avatar. The key idea is to create an animation of an avatar that moves their hands similarly to how a pianist or violinist would do, just from audio. Aiming for a fully detailed correct arms and fingers motion is a goal, however, it's not clear if body movement can be predicted from music at all. In this paper, we present the first result that shows that natural body dynamics can be predicted at all. We built an LSTM network that is trained on violin and piano recital videos uploaded to the Internet. The predicted points are applied onto a rigged avatar to create the animation.

When pianists play a musical piece on a piano, their body reacts to the music. Their fingers strike piano keys to create music. They move their arms to play on different octaves. Violin players draw the bow with one hand across the strings and touch lightly or pluck the strings with the other hand's fingers. Faster bowing produces faster music pace.

An interesting question is: can body movement be predicted computationally from a music signal? This is a highly challenging computational problem. We need to have a good training set of videos, we need to be able to accurately predict body poses in those videos, and build an algorithm that is able to find correlation between music and body, to further predict movement.

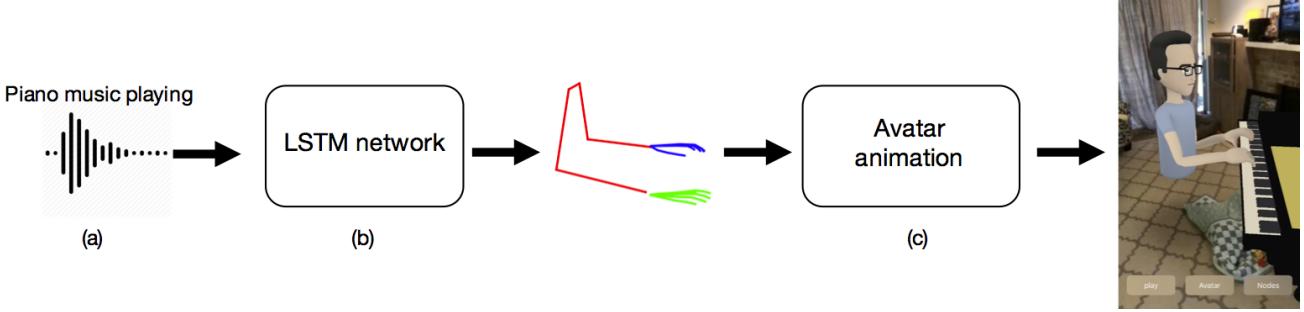


Figure 2. Method overview: (a) Our method gets as input an audio signal, e.g., piano music, (b) that is fed into our LSTM network to predict body movement points, (c) which in turn are used to animate an avatar and show it playing the input music on a piano (the avatar and piano are models while the rest is real apartment background).

Human body dynamics is complex, particularly given the quality needed to learn correlation to audio. Traditionally, state of the art prediction of natural body movement from video sequences (not audio) used laboratory captured motion capture sequences. E.g., in our scenario we would need to bring a pianist to a laboratory and have them play several hours with sensors attached to their fingers and body joints. Such approach is hard to execute in practice and not easily generalizable. If we could leverage the publicly available videos of highly skilled people playing online we potentially allow higher degree of diversity in data. Until recently, though estimating accurate body pose from videos was not possible. This year, several methods appeared that may allow us to learn from data “in the wild”.

In parallel, a number of methods [39, 41, 26] also showed remarkable results for lip sync from speech. I.e., given an audio of a person saying a sentence they showed that it is possible to predict how that person’s mouth landmarks would move while saying the words.

These two advancements inspired us to tackle the challenging ideas of predicting body and fingers movement just from music. The goal of this paper is to explore if it’s possible at all, and if we can create natural and logical **body** dynamics from audio. Note that we do not use information like midi files from which we can potentially learn the correlation between exact piano keys and music. We focus on creating an avatar that moves their hands and fingers, like a pianist would.

An interesting complementary direction would be also to learn the correlation between music and piano keys by training on midi files, and then combine with the method described in this paper.

We consider two sets of data, piano and violin recitals (Fig. 3). We collected videos for each of the two categories, and processed the videos by detecting upper body, and fingers in each frame of each video (given the points are visible). Total of 50 points per frame, where 21 points represent the fingers in each hand, 8 points for upper body.

In addition to predicting the points, one of our goals is

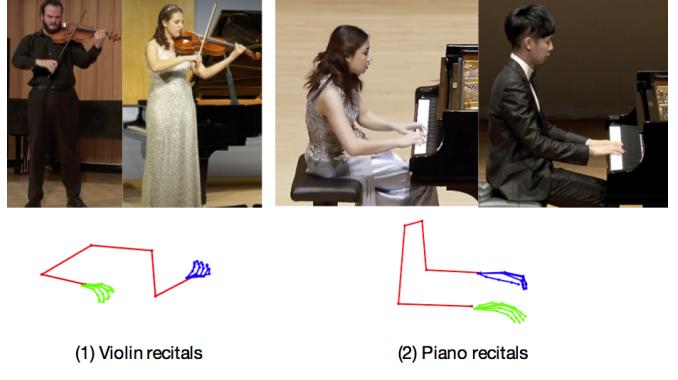


Figure 3. Training data: Example frames from violin and piano sets and their corresponding example keypoints.

to visualize the points via animation of an avatar that moves naturally according to the given audio input. We solve this problem in two steps. First, we build a Long-Short-Term-Memory (LSTM) network that learns the correlation between audio features and body skeleton landmarks. In the second part, we automatically animate an avatar using predicted landmarks. The final output is an avatar that moves according to the audio input.

For each set we trained a separate neural network. I.e., separate network for violin and separate network for piano. The output skeletons are promising, and produce interesting body dynamics. We encourage the reader to watch the supplementary videos with audio turned on, to experience the results.

2. Related Work

The correlation between speech and facial movements was researched extensively beginning with classical works by [6] and [7] and most recently showing remarkable results of generating high quality videos of a face talking just from audio [39] and animating avatars using speech [41, 26]. The three later papers have an extensive state of the art summary on facial animation from speech.

Animating body pose just from music was not explored as far as we are aware. There is, however, a large body of work in three related areas: behavioral studies of how people’s bodies react to music and sounds, computer vision, learning and graphics research in creating and predicting natural body pose changes, for example to learn walking and dancing styles from videos (no audio), and multi-modal studies of combining audio and video input to improve recognition of facial expressions and body poses. We will describe the related works below.

Multi-modal studies [24] have shown that combining audio and visual inputs produces higher accuracy results than just each of the modalities alone [47]. For example, facial expression recognition may benefit from also getting voice input since the emotion/pitch/loudness of the message may help recognition [10, 36]. Wang et al. showed that body pose estimation [33] works better with audio input [45]. Multi-speaker tracking [1], and combination of modalities to identify intent to interact [35] are additional examples.

In dyadic communication, the relationship between speech and body rhythms was investigated, e.g., [13, 43] with interesting results showing that certain speech characteristics were correlated with body movement frequencies [5]. Different emotional scenarios had different types of movements and interactions, e.g., regular social interaction [14] vs. interview [5]. [18] and [42] researched if there is correspondence between the way pianists perform particular music, and found that there is consistency in the way we perceive music-movement correspondences. They also demonstrate that correspondences emerge flexibly, i.e. that the same musical excerpt may correspond to different variants of movement. We find these studies an inspiration to our idea that body movement can be possibly predicted from music and speech.

There is a large body of work of the use of LSTMs to predict and edit future poses given a short video of movement or even a single photo, e.g., [15, 31, 17] and [21, 27, 44, 9]. State of the art body pose estimation and recognition techniques typically use CNNs, e.g., [23, 2, 22, 46] and [19]. Earlier works showed that it is possible to learn and predict motion style [29] from videos and further animate the characters in real-time [16]. These works did not use audio as their input.

A complementary area of research is to predict audio from video. This is the inverse of our goal of predicting video from audio. Some examples include estimating speech from face videos with the goal of lip reading [11], or predicting which sounds an object make from video [12, 34] or photo [38]. Finally, [28] explored creating audio driven video montages.

While related works do not talk directly about animating body pose just from speech or music, it provides sufficient evidence that some correlation exists. This encourages us

to explore estimation of such correlation. We do not expect to reproduce exact body motion, and in fact assume that the transformation from speech/music to body motion is not unique but we do aim to create natural looking body pose that makes sense for the music or speech.

3. Body Dynamics Representation

There are variety of classes of videos where the correlation between audio and video is interesting. E.g., videos of dance, people giving lectures, playing musical instruments, or stand-up comedy to name a few. We chose to experiment with several of those. Below we describe which datasets we use, how we process the audio and video signals, and how the training data is prepared and represented. The training and testing algorithm is described in Section 4.

3.1. Data sets

We have experimented with data from violin recitals, and piano recitals. The full list of URLs is available in the supplementary material and example URLs in the footnote below¹. Figure 3 shows example frames from each of the sets and their corresponding keypoints.

All the videos used were downloaded from the Internet and are videos “in the wild”. We made sure to select videos which are favorable to the processing algorithms. Specifically, the intuition behind our choice of video was to have clear high quality music sound, no background noise, no accompanying instruments, solo performance. On the video quality side, we searched for videos of high resolution, stable fixed camera, and bright lighting. We preferred longer videos for continuity. Our selected videos range from 3 min to 50min. We found that using recitals or single person shows are optimal sets that satisfy the above goals. Total of 3.6 hours of violin recitals was collected and 4.4 hours of piano recitals.

3.2. Audio Processing

MFCC features were shown to be successful in previous art to identify and classify different musical instruments. E.g., [30] showed that a network trained on PCA of MFCC coefficients of flute, piano and violin can recognize which instruments are used. It was also shown that MFCCs perform well for capturing the variation in speech [48]. We follow the optimal process for computing the features described in [39] with several modification to adjust to our datasets frame rate. Specifically, we compute the features on stereo 44.1Khz sample rate audio, perform RMS normalization to 0 db using FFmpeg [3], and choose the window length as $1000/\text{video}_{\text{fps}}$ with $\text{fps} = 24$, i.e., 41.66ms. We

¹Violin recital: <https://www.youtube.com/watch?v=BP65MiYNh50>, and Piano recital: <https://www.youtube.com/watch?v=SD1nhx9qYH0>

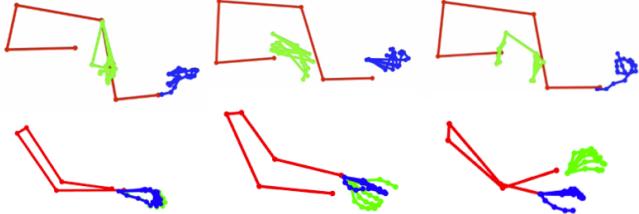


Figure 4. Example failure cases of keypoints detectors (row 1 for violin set, and row 2 for piano set) which were eliminated automatically in our preprocessing step.

synchronized all videos to have the same audio and video rates. The final audio feature size is 28-D which includes the 13-D feature vector, their temporal derivative, and log mean energy for volume.

3.3. Keypoints Estimation

We are interested in two types of keypoints: body, and hand fingers. Typically, estimation of keypoints on videos in the wild is challenging due to the large variation in camera, lighting, and fast movement which can divert from type of benchmark videos on which the algorithms evaluate. Recently, however a number of methods appeared that can handle in the wild videos much better. We have came up with a process that allows us to get sufficiently accurate keypoints, as described below.

We begin by running a video through three libraries: OpenPose that provides face, body, and hands keypoints [37, 46, 8], MaskRCNN [19], and DeepFace face recognition algorithm [40]. Those three libraries perform well on benchmarks, but on our videos in the wild each fails on some frames. We have noticed, however, that the failure and success are somewhat complementary across the algorithms.

Thus, per video we select a single frame and a detection box that includes the person of interest, as our reference frame. The person's face in the box gets a signature from the face recognition algorithm. Each consequent frame's box is automatically eliminated if it's 1) too far from the location of the box in the reference frame, 2) face signature doesn't match, or 3) the L_2 distance between points in consecutive frames is too big. See thresholds in Experiments section. Given the box we use the hands points from OpenPose that appear in that box. We further choose among MaskRCNN or OpenPose which body points to use based on the confidence and existence of points (in case part of the points are not detected due to occlusions, or mis-detection). In case part of points are still missing we exclude that frame from training. See Figure 4 for example frames that were eliminated from training.

3.4. Keypoints Motion Factorization

Consider a video with keypoints estimation per frame, resulting in a timeseries of points. The motion in keypoints across frames is a product of several key components:

$$V_{\text{frame}} = M_{\text{camera}}(M_{s,T,R}(V_{\text{person}} + V_{\text{audio}})) \quad (1)$$

where M_{camera} is motion due to camera, e.g., zoom in-out and viewpoint change, $M_{s,T,R}$ is the rigid transformation of the person, V_{audio} is the person body transformation due to audio, e.g., bow drawing, and striking piano keys, and V_{person} is non-rigid transformation of the person's body which is not correlated to audio, e.g., person pacing on stage while playing violin. We assume fixed stable camera, and we solve for scale, translation and rotation in 2D by choosing a reference points configuration. We assume that V_{person} is zero, for simplicity. Our goal is then to predict V_{audio} . To reduce the dimensionality and noise of the data we compute PCA coefficients of the keypoints as follows.

PCA on keypoints: Given aligned points per frame in each video, we collect all points to a single matrix of size $2p \times f$, where p is the number of points per frame, each point has 2 dimensions, and f is the total number of frames in a dataset. Each set of points is reshaped to a 1-D vector of length $2p$. We compute PCA on the final matrix across all frames, and reduce the dimensionality to capture 90% of the data. Each frame is then represented using the PCA coefficients. This allows us to reduce noise, as well as reduce dimensionality. As a final step we upsample the PCA coefficients linearly to $4 \times \text{video}_{\text{fps}}$. Exact numbers per set and implementation details are also presented in Section 6.

The resulted PCA coefficients (represent body motion) and MFCC coefficients (represent audio) are used in the next steps.

4. Audio to Body Keypoints Prediction

Our goal is to learn a correlation between audio features and body movements. For this, we built an LSTM (Long-Short Term Memory) network [20]. It was shown recently that LSTMs can be used successfully to predict lip sync, e.g., [39] (this paper also has a nice explanation of LSTM vs. RNN vs. HMM models, and the use of LSTM with time-delay). Our architecture is visualized in Fig. 5. We chose to use a unidirectional single layer LSTM with time delay. x_i is an audio MFCC in a particular time instance i , y_i are PCA coefficients of body keypoints, and m is the memory. We also add a fully connected layer 'fc' which we found to increase performance. We have experimented with other variations of the architecture which did not show improvement in results. Those include: using H norm for loss, global time delay, multi-layer LSTM, training on keypoints directly rather than PCA components.

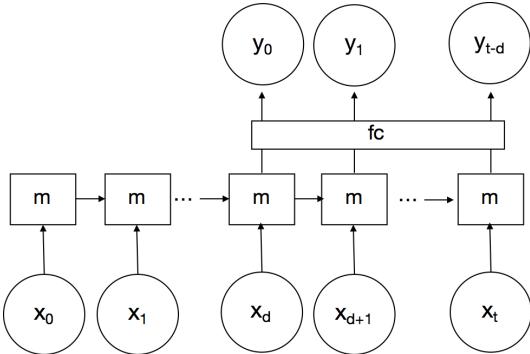


Figure 5. Architecture of our points prediction LSTM. x_i represent the audio features, and y_j represent the corresponding keypoints.

The parameters that we used are hidden state of 200 (the length of m), trained with truncated back propagation with time steps of 400, time delay of 5, dropout of 0.4, learning rate of $5e - 3$. The number of PCA components we typically use is 10. We ran the training for 300 epochs. The network is implemented in Caffe2 [25], and use ADAM optimizer. Both input and output are normalized by subtracting the mean and dividing the variance. In Fig. 6 we show the improvement in PCA coefficients prediction as a function of epochs for the piano set.

5. Body Keypoints to Avatar

Once the keypoints are estimated we animate an avatar using the points. We have built an Augmented reality application using ARkit which runs on the phone in real time. Given a sequence of 2D predicted points, and a body avatar, the movements are applied onto the avatar. Below we describe the specific details.

The avatars we have used are 3D body models with a body bone rig. We first initialize the rig by aligning the predicted points to 3D world coordinates. We do it by estimating average left and right shoulder points across all frames, and calculate rigid transformation to the avatar. Then we consider each of the body, arms, head, and fingers separately. If some of the body, arms, head or fingers are not provided only the provided parts are animated. Finally, we apply root rotation offset to match pose angle of piano. The z depth for the piano is calculated off of the wrists x position, the further left it is the closer it gets to camera. There are details below on how we rig the violin.

Body: IK chain was created where the root node defined as average between the left and right hip, and connected to the average of left and right shoulder. This defines the spine. We then estimate the average spine length across the frames, and scale the avatar spine accordingly.

Arms: We defined an IK chain, where the reference point was used as the wrist point. Next we calculate an offset that defines how much the arms point forward (out of

fixed plane). The length of the forearm determines the offset. First, the maximum forearm length is calculated across all frames, then if the forearm appeared in its maximum length, then the reference is on the original 2D plane and arm is straight, otherwise if length is zero, then the arm is perpendicular to the source plane.

Hands/Fingers: Hand rotation was determined via root joint of the pinkie finger and the root joint of the pointer finger. E.g., if on the right hand the pointer joint is to stage left of the pinkie joint, then the hand must be rotated to the avatar’s right with the palm facing up. The palm would be facing down if the opposite were true. Additionally, for each finger, the angle between the reference point and root was calculated and applied.

Rigging the violin: Rigging the violin was done with four points used as constraint references. The violins position was constrained to a point attached to the head and the rotation was determined using a lookat constraint the was attached to the left hand. The bow position is constrained to a point on the right hand and the rotation uses a lookat constraint attached to the bridge of the violin.

6. Experiments

In this section, we discuss our experiments, implementation details, comparisons, and evaluations.

Running times and hardware: Our preprocessing, train and test runs are done on a GPU server of 8 NVIDIA M40 GPUs, two 12 cores CPUs, 256GB RAM. The running time for 200 epochs of training of 3.6 hours of violin set took 1.5 hours (30s per epoch). Piano dataset is 4.4 hours. The running time of the test set and animation of the avatar is real time. The processing time to calculate keypoints on the training set is 10fps for OpenPose and 2fps for MaskRCNN.

Data: We have experimented with Violin recitals and Piano recitals. All videos were synchronized to 24fps. We have randomly selected 20% of each of the datasets for validation and 80% for training. Each of the sets was trained and tested completely independently, i.e., we have a violin net, and a piano net. We have removed approximately 10% of the frames in the videos due to not accurately predicted keypoints (based on the procedure described in Sec. 3), the total number of frames in Violin set is about 324,000, and 414,000 for piano recitals. The threshold for removing frames are if points are farther from previous frame more than 10% of the width of frame (typically 20 pixels). Test audio was not part of the training or validation sets.

Evaluations: We have experimented with different parameters choices in our network and provide comparisons in Tables 1 and 2. To find the optimal parameters specified in Sec. 4 we ran a hyper-parameters search. The errors in the tables are presented in pixels, where low is better. To achieve good results it is important to filter out all bad frames in the training data (wrong skeleton, wrong per-

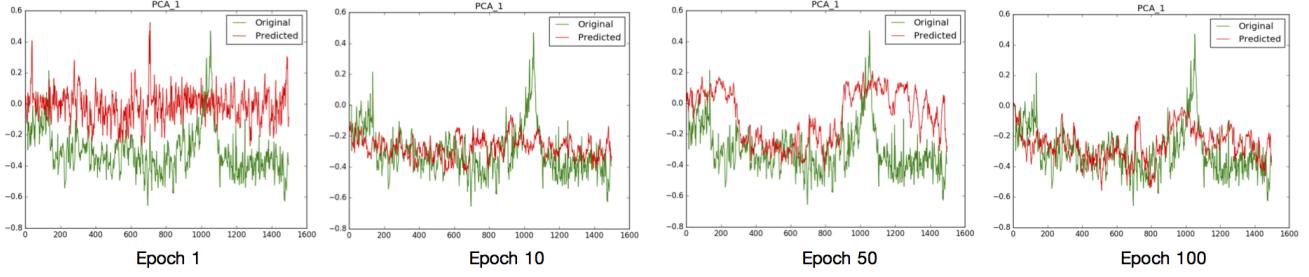


Figure 6. Evolution of testing of first PCA mode (piano) as function of epochs.

Method	Train	Valid	Test
Frames not dropped	12.44	12.09	18.31
50% training data	7.22	11.18	9.38
PCA coeff = 5	6.86	7.04	9.35
Dropout = 0.6	7.25	7.51	9.31
Seq. length = 500	7.40	6.95	9.24
No time delay	7.32	7.76	9.19
Upsample 6x	7.07	6.95	9.18
PCA coeff = 30	7.39	7.12	9.13
Seq. length = 200	6.88	6.75	9.13
Dropout = 0	7.32	7.71	9.03
Upsample 2x	7.33	9.09	8.66
Final ours	7.35	6.58	8.31

Table 1. Violin net: Comparison of errors in training, validation, and testing with different parameter choices. Errors presented in pixels (low is better). Ours is: PCA coeff = 15, Hidden variables = 200, Seq. length = 60, Batch size = 100, Learning rate 1-e03, time delay=24ms, dropout = 0.15, upsample = 4x.

son detection, wrong person recognition), we see that errors drop significantly just by filtering out the data. In case we use half of the training data we improve the training error but test error is larger. By using less PCA coefficients we can fit better to the training data but test error is larger than using more coefficients. Using dropout doesn't improve results in our case. Time delay helps improve results. We have experimented (Fig. 7) with giving a random sequence as test audio to predict points with both nets, the test showed no correlation and no movement was predicted, as expected.

Results: In Figure 8 and 9 we present representative results. We show the predicted keypoints for different body poses, as well as the original frame for context. For the keypoints we show them overlaid on the ground truth points for visual comparison. Note that we don't expect the points to be exactly the same, but the fingers and hands to produce a similar satisfactory movement which was the goal of this paper. The ground truth in our case is result of 2D body pose detector which can be mistaken. Finally, we show failure cases in Fig. 12, row 1 shows piano, and row 2 violin. These show limitations of our system: currently our system is trained on 2D poses, while actual poses in training videos

Method	Train	Valid	Test
Frames not dropped	17.15	14.64	9.49
50% training data	6.94	6.65	8.87
Interpolated missing data	7.05	6.41	8.14
Upsample 2x	7.18	6.95	7.98
No time delay	6.83	6.04	7.48
PCA coeff = 30	6.66	6.79	7.42
Dropout = 0.6	6.76	5.97	7.38
Dropout = 0	6.77	6.07	7.28
Seq. length = 200	6.76	6.92	7.06
PCA coeff = 5	6.27	7.24	6.94
Final ours	6.52	6.06	6.84

Table 2. Piano net: Comparison of errors in training, validation, and testing with different parameter choices. Errors presented in pixels (low is better). Ours is: PCA coeff = 15, Hidden variables = 200, Seq. length = 60, Batch size = 100, Learning rate 1-e03, time delay=24ms, dropout = 0.1, upsample = 4x.

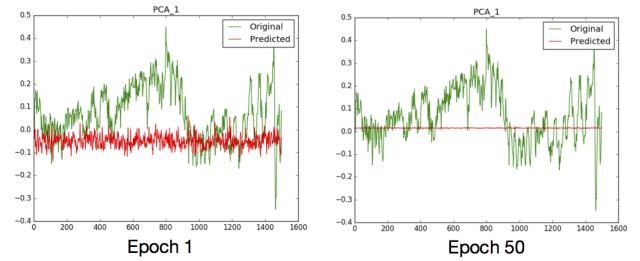


Figure 7. Predicting a random sequence with violin net. We see that the prediction is stationary, means there is no correlation between the signal and violin net.

are 3D. Consequently, occlusions and invisible points are not predicted well. In high pace and frequency parts of the videos, the body pose detectors may create mistakes, similarly in case of motion blur. This causes the network to learn behave in high frequency audio accordingly.

Figures 10 and 11 show screenshots of our avatar animation based on predicted points. The piano, violin, and avatar are synthetic objects placed in a real scene (augmented with ARKit).

Videos: We recommend watching the supplementary video with audio on. In addition to showing the results we

present how an animation may look like when music and keypoints don't match, you will see in the video that it's look very unnatural thus the importance of the network.

7. Discussion and Limitations

We have proposed a new hypothesis that body gestures can be predicted from audio signal, and showed promising initial results. We believe the correlation between audio to human body is very promising for a variety of applications in VR/AR and recognition. It was shown previously that mouth animation can be done just from audio, and in this paper we show initial results on body animation. We hope it will open up further research. There are a number of limitations and many extensions that would be interesting to explore.

One direction is to enable 3D movement, currently we use OpenPose and MaskRCNN to estimate keypoints. These are both 2D points estimators. Provided a 3D body keypoints estimator, e.g., Vnect [32] or SMPL [4] our approach can potentially be extended to handle 3D motion as well, and allow more diverse human modeling.

Another direction is to predict occluded keypoints. Currently, we can only predict the visible points and if training frames do not have all the points we ignore that frame. It would be interesting to see if the keypoints from audio network can be extended to predict occlusions, e.g., in piano the left hand usually occludes the right hand but the audio includes music played by both hands.

We have used only YouTube videos as training data. In the future, to increase realism and accuracy we could consider complementing the training data with sensor information or midi files.

Finally, getting good training data per class of activity is not straight forward. Note the constraints we required in Sec. 3. It would be interesting to explore a general network that can handle variety of poses, without the need to classify the type of action a priori. Alternatively, it would be interesting to incorporate video activity recognition into our learning framework.

References

- [1] Y. Ban, L. Girin, X. Alameda-Pineda, and R. Horaud. Exploiting the complementarity of audio and visual data in multi-speaker tracking. In *ICCV Workshop on Computer Vision for Audio-Visual Media*, 2017. 3
- [2] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 468–475. IEEE, 2017. 3
- [3] F. Bellard, M. Niedermayer, et al. FFmpeg. Available from: <http://ffmpeg.org>, 2012. 3
- [4] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016. 7
- [5] D. S. Boomer and A. T. Dittmann. Speech rate, filled pause, and body movement in interviews. *The Journal of nervous and mental disease*, 139(4):324–327, 1964. 3
- [6] M. Brand. Voice puppetry. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 21–28. ACM Press/Addison-Wesley Publishing Co., 1999. 2
- [7] C. Bregler, M. Covell, and M. Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353–360. ACM Press/Addison-Wesley Publishing Co., 1997. 2
- [8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 4
- [9] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng. Forecasting human dynamics from static images. *arXiv preprint arXiv:1704.03432*, 2017. 3
- [10] L. S.-H. Chen and T. S. Huang. *Joint processing of audio-visual information for the recognition of emotional expressions in human-computer interaction*. University of Illinois at Urbana-Champaign, 2000. 3
- [11] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip reading sentences in the wild. *arXiv preprint arXiv:1611.05358*, 2016. 3
- [12] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman. The visual microphone: Passive recovery of sound from video. 2014. 3
- [13] A. T. Dittmann. The body movement-speech rhythm relationship as a cue to speech encoding. *Studies in dyadic communication*, pages 135–152, 1972. 3
- [14] A. T. Dittmann and L. G. Llewellyn. Body movement and speech rhythm in social conversation. *Journal of personality and social psychology*, 11(2):98, 1969. 3
- [15] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. 3
- [16] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. In *ACM transactions on graphics (TOG)*, volume 23, pages 522–531. ACM, 2004. 3
- [17] I. Habibie, D. Holden, J. Schwarz, J. Yearsley, and T. Komura. A recurrent variational autoencoder for human motion synthesis. 2017. 3
- [18] E. Haga. Correspondences between music and body movement. 2008. 3
- [19] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017. 3, 4
- [20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4
- [21] D. Holden, J. Saito, and T. Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):138, 2016. 3

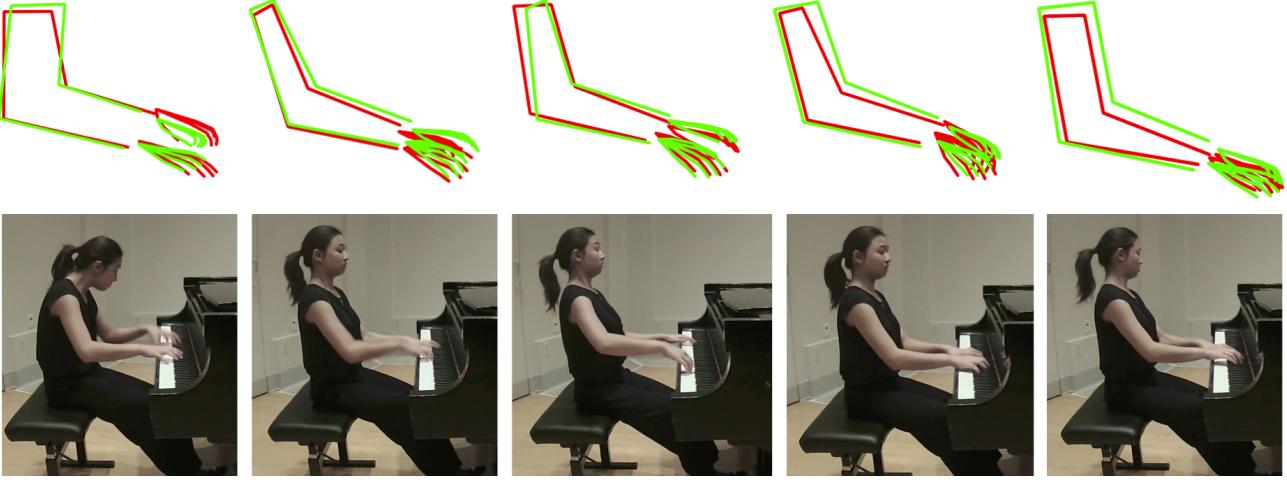


Figure 8. Piano test results. We encourage the reader to watch the supplementary videos (with audio on) too. In row 1 we show predicted points from audio (in green) overlaid on top of ground truth points (red; 2d pose detector). Row 2 shows the corresponding frame for context. Note that we don't expect for them to fit exactly but just aiming for similar hands and fingers configurations.

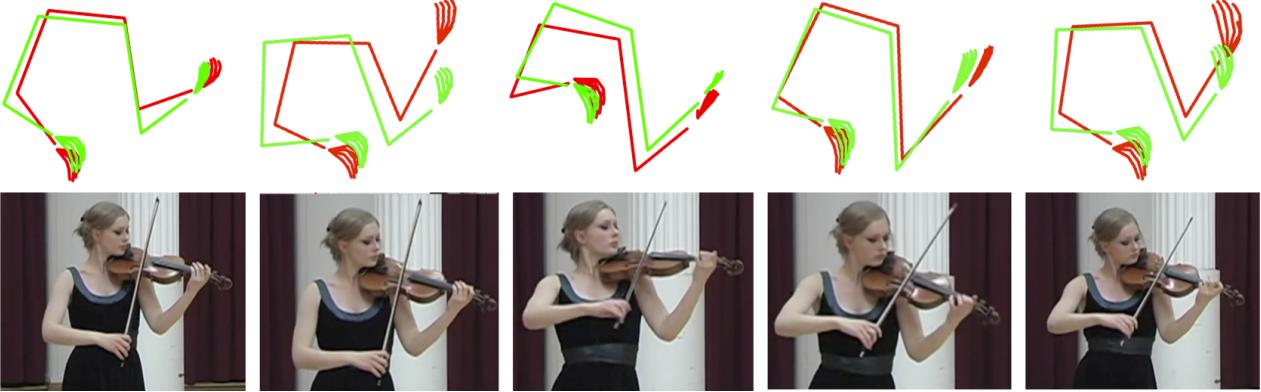


Figure 9. Violin test results. We encourage the reader to watch the supplementary videos (with audio on) too. In row 1 we show predicted points from audio (in green) overlaid on top of ground truth points (red; 2d pose detector). Row 2 shows the corresponding frame for context.

- [22] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepcut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016. [3](#)
- [23] E. Insafutdinov, B. Schiele, and B. Andres. *Dense-CNN: Fully Convolutional Neural Networks for Human Body Pose Estimation*. PhD thesis, Universität des Saarlandes Saarbrücken, 2016. [3](#)
- [24] A. Jaimes and N. Sebe. Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1):116–134, 2007. [3](#)
- [25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. [5](#)
- [26] 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)*, 36(4):94, 2017. [2](#)
- [27] 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*, 2017. [3](#)
- [28] Z. Liao, Y. Yu, B. Gong, and L. Cheng. Audeosynth: music-driven video montage. *ACM Transactions on Graphics (TOG)*, 34(4):68, 2015. [3](#)
- [29] C. K. Liu, A. Hertzmann, and Z. Popovic. Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics (TOG)*, 24(3):1071–1081, 2005. [3](#)
- [30] R. Loughran, J. Walker, M. O'Neill, and M. O'Farrell. The use of mel-frequency cepstral coefficients in musical instrument identification. In *ICMC*, 2008. [3](#)
- [31] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. *arXiv preprint arXiv:1705.02445*, 2017. [3](#)

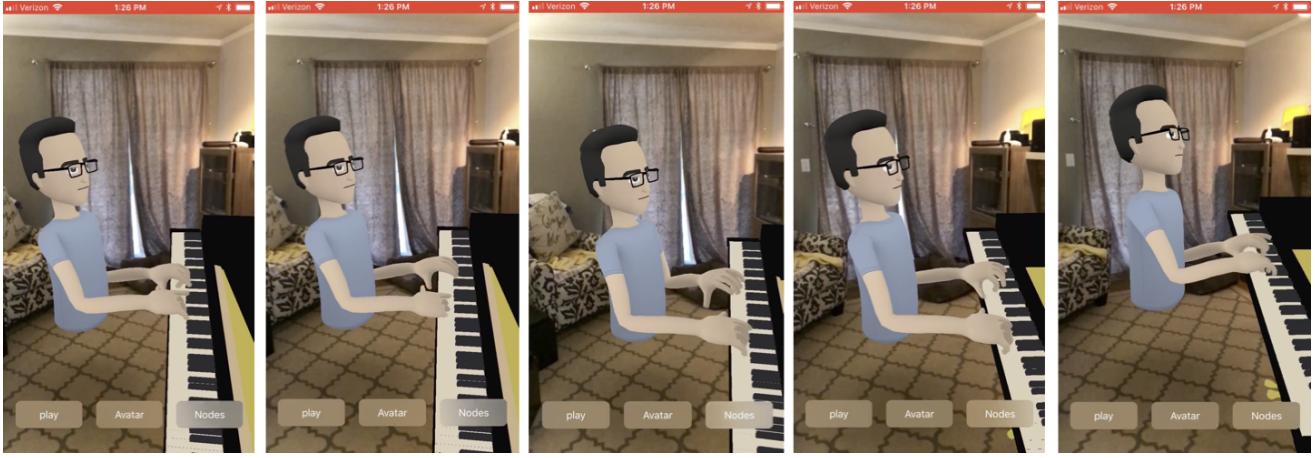


Figure 10. Avatar playing piano: Screen shots of avatar animation result with the predicted skeleton.

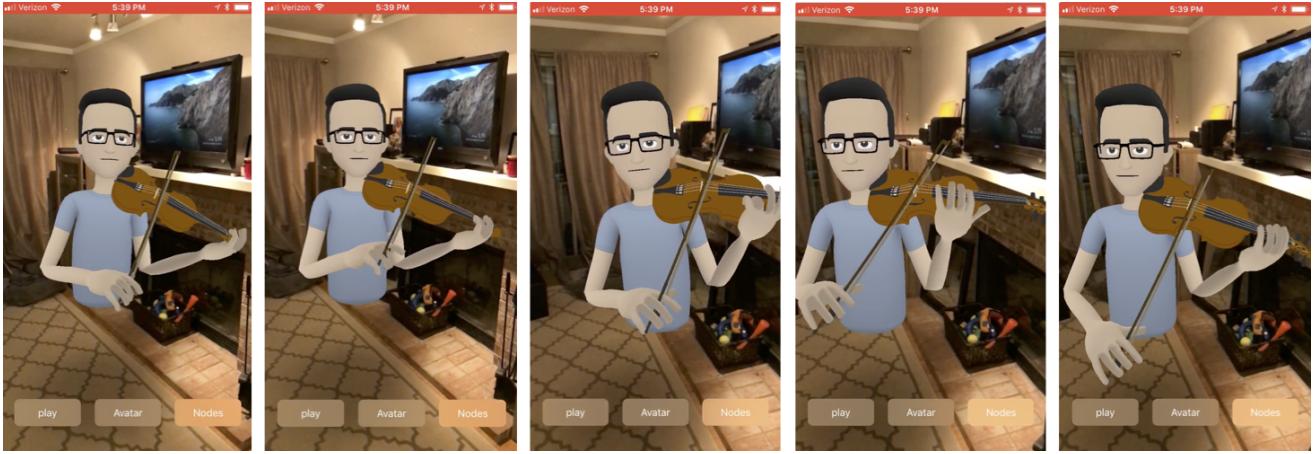


Figure 11. Avatar playing violin: Screen shots of avatar animation result with the predicted skeleton.

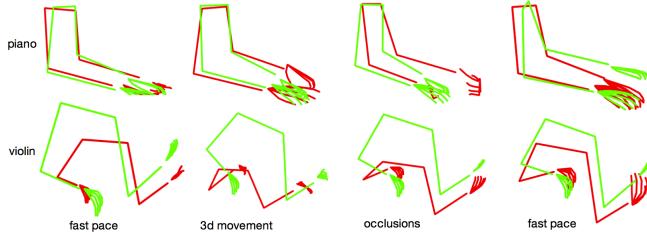


Figure 12. Typical limitations of the method. The method fails with extreme occlusions, or unexpected tones or movements that are not captured well in the training data (fast raising of arms, movement from one side of keyboard to extreme other, or very fast pace). It may also fail since we train based on 2D pose estimators while people in YouTube videos move in 3D. The predicted points (green) are overlaid on top of ground truth (red) for comparison, in those cases also the ground truth points often (estimated by pose detection) are not exact.

Vnect: Real-time 3d human pose estimation with a single rgb camera. *arXiv preprint arXiv:1705.01583*, 2017. 7

- [32] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt.
- [33] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2329–2336, 2014. 3
- [34] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2405–2413, 2016. 3
- [35] J. Schwarz, C. C. Marais, T. Leyvand, S. E. Hudson, and J. Mankoff. Combining body pose, gaze, and gesture to determine intention to interact in vision-based interfaces. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3443–3452. ACM, 2014. 3
- [36] N. Sebe, I. Cohen, T. S. Huang, et al. Multimodal emotion recognition. *Handbook of Pattern Recognition and Computer Vision*, 4:387–419, 2005. 3
- [37] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 4

- [38] M. Soler, J.-C. Bazin, O. Wang, A. Krause, and A. Sorkine-Hornung. Suggesting sounds for images from video collections. In *European Conference on Computer Vision*, pages 900–917. Springer, 2016. [3](#)
- [39] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017. [2, 3, 4](#)
- [40] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. [4](#)
- [41] 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)*, 36(4):93, 2017. [2](#)
- [42] M. R. Thompson and G. Luck. Exploring relationships between pianists body movements, their expressive intentions, and structural elements of the music. *Musicae Scientiae*, 16(1):19–40, 2012. [3](#)
- [43] P. Wagner, Z. Malisz, and S. Kopp. Gesture and speech in interaction: An overview. *Speech Communication*, 57:209–232, 2014. [3](#)
- [44] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. *arXiv preprint arXiv:1705.00053*, 2017. [3](#)
- [45] S. B. Wang and D. Demirdjian. Inferring body pose using speech content. In *Proceedings of the 7th international conference on Multimodal interfaces*, pages 53–60. ACM, 2005. [3](#)
- [46] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016. [3, 4](#)
- [47] M. Wöllmer, M. Kaiser, F. Eyben, B. Schuller, and G. Rigoll. Lstm-modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 31(2):153–163, 2013. [3](#)
- [48] F. Zheng, G. Zhang, and Z. Song. Comparison of different implementations of mfcc. *Journal of Computer Science and Technology*, 16(6):582–589, 2001. [3](#)