Dance Motion Transfer

Farnoosh Javadi
University of British Columbia
fjavadi@cs.ubc.ca

Mona Fadaviardakani
University of British Columbia
mfadavi@cs.ubc.ca

Abstract

In our project, given a source video of a person dancing, we want to transfer the dance motion to an amateur target performing some standard moves. There are lots of application for this work in advertisement or film industry. We approach this problem as video to video translation using pose as an intermediate representation. To transfer the motion, we extract poses from the source subject and apply the learned pose to appearance mapping to generate the target subject. For the background, as opposed to the baseline model [6] we use the target video, due to having more realistic features. The output gif of our results can be seen in https://drive.google.com/open?id=1OaKLNmlrS5zXpGSkXlHl00YxOuOPHe5_.

1. Introduction

Consider the two videos of Figure 1. We want to give the top row as the input to our model and transfer its motion to an amateur subject not performing dance moves, as shown in the second and third row [6]. The baseline model [6] uses target video's background (as you can see in the third row), however we find out that generally source video's background is more realistic (second row). The interesting point is that the target subject knows nothing about dancing and has a different gender, body shape,clothing and height from the target. But the motion is well correspond with him. To transfer motion between two video subjects in a frame-by-frame manner, we must learn a mapping between images of the two individuals. Our goal is, therefore, to discover an image-to-image translation [14] between the source and target subjects. We don't have correspondences between the source and target to supervisedly train a network. Hence, we use keypoint-based pose as an intermediate representation because it preserves the motion over time, and is a good representative for that. We therefore use pose stick figures obtained from Openpose [4], and use them to learn an image to image translation model between the source and target subject. To transfer motion from source to target, we input the poses the source into the trained model to obtain images of the target subject in the same pose as the source. The baseline paper [6] gets all the appearance both for the background and foreground, from the target video, as you can see in 1. But we notice that targets' backgrounds are usually very simple not having many features, however source's background are more realistic and having some features like shadows, and lights that correspond with the dance move. Therefore we decided to divide the appearance into foreground and background and have different sources for each of them. In our project, we get the background from the source video and the foreground's appearance comes from the target video.

2. Related Work

2.1. Pose Detection

Modern pose detection systems including OpenPose [4] and DensePose [10] allow for reliable and fast pose extraction in a variety of scenarios. In our project, we use Openpose which is a pre-trained pose detector and can accurately estimate all of the subject's multiple joint coordinators. The model consists of a convolution neural network with a specific series of matrix operations which is optimized for pose estimation.

2.2. Motion Transfer

Motion Transfer is a hot and interesting topic in computer vision and computer graphics and there has been lots of research in this area. Early methods focused on creating new content by manipulating existing video. For example, Video Rewrite [3] creates videos of a subject saying a phrase they did not originally say by finding frames where the mouth position matches the desired speech. [8] uses optical flow as a descriptor to match different subjects performing similar actions allowing "Do as I do" and "Do as I say" retargeting.
Some approaches rely on calibrated multi-camera setups to scan a target actor and manipulate their motions in a new video through a fitted 3D model of the target. To obtain 3D information, [7] propose a multi-view system to calibrate a personalized kinematic model, obtain 3D joint esti-

Figure 1. Given a video of a dancer (top), we want to transfer the motion to an amateur subject. The second row is the results of our model using source video's background. The third row is the results using target video's background as proposed in [6]

mations, and render images of a human subject performing new motions. [20] uses multi-view captures of a target subject performing simple motions to create a database of images and transfer motion through a fitted 3D skeleton and corresponding mesh for the target. [5] uses 4D video textures to store a texture representation of a target person and use their temporally coherent mesh and data representation to generate video of the target subject performing new motions. In contrast, in our project we explore motion transfer between 2D subjects and avoid data calibration and mapping to the 3D space.

Many approaches rely on deep learning for the task. In [15], given synthetic renderings, a face model, and a gaze map as input, they transfer head position and facial expressions between human subjects. Our project is similar to them except we retarget full body motion, and the inputs to our model is 2D pose as opposed to 3D representations. Similarly, [16] applies neural re-rendering to enhance rendering of human motion capture for VR/AR purposes. The primary focus of this work is to render realistic humans in real time and uses a deep network to synthesize their final result, but does not address motion transfer between subjects.

Some Recent methods focus on disentangling motion from appearance and synthesizing videos with novel motion. MoCoGAN [19] employs unsupervised adversarial training to learn this separation and generates videos of subjects performing novel motions or facial expressions. This theme is continued in Dynamics Transfer GAN [1] which transfers facial expressions from a source subject in a video onto a target person given in a static image. In this project, we also apply our representation of motion to different target subjects to generate new motions.

Many approaches have shown success in generating detailed single images of human subjects in new poses, however they are not designed specifically for motion transfer.

2.3. Image to Image Translation

Image to image translation involves the controlled modification of an image and synthesize an image that does not exist. CycleGAN [21] is a technique for training unsupervised image translation models via the generative adversarial network (GAN) architecture using unpaired collections of images from two different domains. Pix2PixHD [13] translated semantic label maps into photo-realistic images for synthesizing portraits from face label maps. In our project we used the NVIDIA implementation of pix2pixHD.

2.4. Background Detection

Background subtraction is a popular method for isolating the moving parts of a scene by segmenting it into background and foreground. The shape of the human silhouette plays a very important role in recognizing human actions, and it can be extracted from background subtracted blobs. Several methods based on global, boundary, and skeletal descriptors have been proposed to quantify the shape of the silhouette. Global methods consider the entire shape region to compute the shape descriptor. Boundary methods, on the other hand, consider only the shape contour as the defining characteristic of the shape. Such methods include chain codes [9] and landmark-based shape descriptors [12]. Skeletal methods represent a complex shape as a set of 1D skeletal curves, for example, the medial axis transform [2]. These methods have found applications in shape-based modeling of the human silhouette.

In our project, for detecting the background we use the proposed method of [17]. After computing the background we get the segmentation mask for each frame by the traditional background subtraction method.

3. Method

Given a video of a source person and another of a target person, our goal is to generate a new video of the target

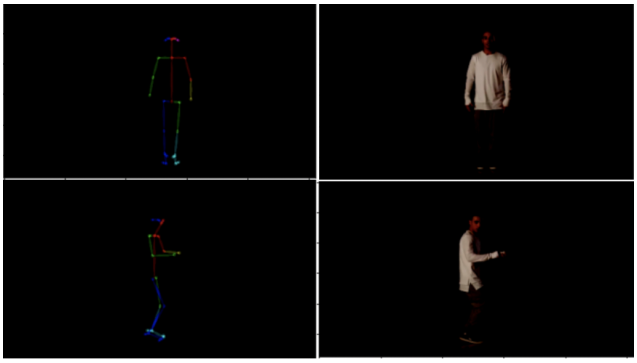

Figure 2. The examples of pose figures and the corresponding images

performing the same dance motions of the source person with the source background. In other words, our goal is to switch the source subject with the target one in the source dancing video. We summarize our steps for accomplishing this task in the following sections.

3.1. Pose Encoding

The pre-trained weight files of OpenPose are used to encode the pose of humans. Using OpenPose able us to take the pose coordinates and draw a representation of resulting pose stick figures. The key points and the lines between connected joints are plotted. These pose figures are extracted for every image frames of the video resulting to create a rich dataset of dancing poses. Figure 2 shows some examples of pose figures with the corresponding image figures.

3.2. Pose to Video Translation

In order to transfer the motion from the source video to the target subject, we should learn the mapping between the images of the two individuals in the frame-by-frame manner. Since we do not have the corresponding subjects performing the same motion, the labeled data cannot be used and it should be done in the completely unsupervised way. The adversarial training can be used to learn the mapping from the pose stick figures to images of the target person. However, the Conditional GAN (CGAN) structure from pix2pix is not suitable for this task because it cannot capture the fine details of the human motions and the temporal coherency is violated.

3.2.1 The baseline Training Stage

The pose stick figures are used as an intermediate representation for this transfer. Therefore, the target's image and corresponding pose figures are fed as the input to GAN to generate a new video conditioned on the source pose with the target appearance.

The GAN is consisting of two components including the generator and discriminator. The U-Net architecture is used for the generator which directly connected the encoder layers to decoder layers using skip connections. Instead of generating an individual frame, the generator is modified to generate two consecutive frames. The first output is conditioned on its corresponding pose stick figure and the generated frame at the previous time step. The second output is conditioned on its corresponding pose stick figures and the first output. The discriminator uses pose images and compares them with the input images regarding both temporal and realism's coherency to distinguish fake and real sequences. The top row of the figure 3 shows these steps. As illustrated in this figure, the labeled images are denoted by y_t and y_{t+1}, and the related poses by x_t and x_{t+1}. G_t and G_{t+1} indicate the generated images. In order to consider the temporal coherency, smooth loss ($L_{smooth}\ (G, D)$) is defined in equation 1 and the GAN objective is updated base on that in the equation 2.

$$\begin{aligned} L_{smooth}(G, D) = E_{(x,y)}[logD(x_t, x_{t+1}, y_t, y_{t+1})] \\ +E_x[log(1 - D(x_t, x_{t+1}, G(x_t), G(x_{t+1}))]) \end{aligned} \tag{1}$$

$$\begin{aligned} \min_G((\max_{D_i} \sum_{k_i} L_{smooth}(G, D_k)) + \lambda_{FM} \sum_{k_i} L_{FM}(G, D_k) \\ +\lambda_P(L_P(G(x_{t-1}), y_{t-1}) + L_P(G(x_t), y_t))) \end{aligned} \tag{2}$$

A multi-scale discriminator is used $D = (D1, D2, D3)$ so $i = 1, 2, 3$. The $L_{FM}(G, D)$ is the discriminator feature-matching loss presented in pix2pixHD [13]. This loss is defined to minimize the statistical difference between the features of the real images and the generated images.
The $L_P(G(x), y)$ is the perceptual reconstruction loss. It uses the pre-trained VGGNet [18] features for extracting the features of generated images and compare them with original feature images. This loss is computed at different layers of the network to improve the generation capability. The λ terms used control the importance of the loss terms.
The baseline paper proposed a separate pipeline to generate more realistic face synthesis called Face GAN. The full image GAN explained is optimized separately from the face GAN. We did not implement this step in our project and this is why the faces in our output synthesized results do not look very realistic.

3.2.2 The Baseline Testing Stage

Once both the generator and discriminator are trained, the pose key points of the source person are transformed in a way that they appear in accordance with the target person's body shape and location. This step called global pose normalization with the goal to adjust the differences between source and target subjects. This step is illustrated in figure's

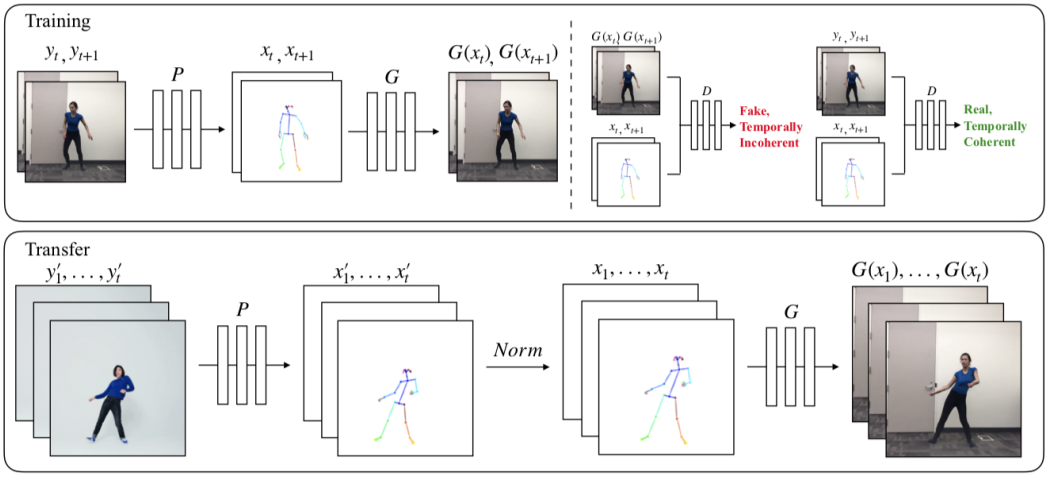

Figure 3. The training and testing stages of the baseline Model

3 bottom row. These adjustments try to match the two subjects in both scale and location. It usually works when two subjects have different limb proportions or stand closer or farther to the camera than one another. After these transformations applied on of all pose key points of all given frame of the source subject, the results are fed as an input to the generator and the synthesized target images are generated performing the same motion of the source subject. It is important to note that our group did not implement the normalization step in our project and directly pass the source poses as the input to the generator. Figure 4 shows some of our synthesized outputs with the corresponding source poses.

3.3. Background Detection

We notice that in our dataset, the source video's background is more realistic in comparison with the target video in which an amateur person is doing some standard moves. That motivated us to get the background from the source video as opposed to the baseline model. For this end, at first we need to detect the background of source video. For background detection, we use the proposed method in [17], in which they estimate the background by taking the median pixel value across all frames. We also tested computing the background by taking mean of pixel values across all frames, but median worked better and could eliminate noises better. The method is very simple however it assumes the background is static which is a strong constraint.

3.4. Segmentation Mask Detection

Another component that we need to blend the background with foreground is the segmentation mask which we need to have it for every frame of our final video. To

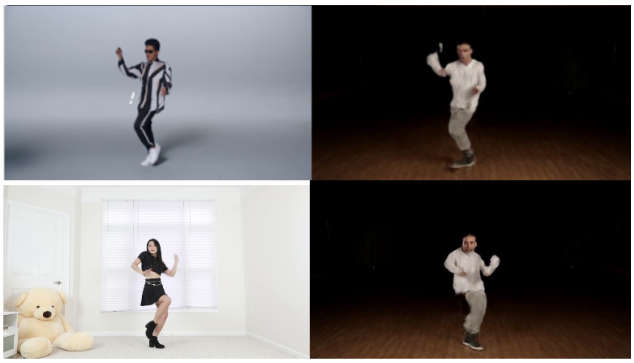

Figure 4. Some examples of synthesized images, first column is the source image and second one is the target image.

compute the segmentation mask S for each frame, at first, we compute the background of the target video as the explained way in previous section. Then we subtract every synthesized frame at time stem t from background and map the result to 0-1 range to have the segmentation masks S_t . Finally, we blend the source video's background B_s and generated frames' $\hat{I}$ foreground by the formula 3 to synthesize the final video frames F_t.

$$F_t = S\hat{I}_t + (1 - S)B_s \tag{3}$$

4. Experiments

We used the youtube-dl python package to download and cut the first 20 seconds of the single-dancer videos as our source and target videos, and then performed OpenPose to get the resulting pose videos. The videos we used mainly have 4 to 5 minutes duration with 1920 1080 resolution. Next, the pictures from these videos were generated as our source and target dataset. We used different videos for our source to test synthesize the target subject on different mo-

tions, and our target video including 5000 images. Some of our target video frames can be seen in figure 2. The network

Figure 5. The trends for the losses

architecture follows the pix2pixHD [13] architecture. We used 20 epochs to train the generator and discriminator. For hyperparameters we used $\lambda_P = 5$ and $\lambda_F M = 10$. It took about one day and half for training over 5000 images. The figure 5 pictures our losses trends. We validate our training on 275 images and get the G-smooth:0.416, G-FM:0.237, G-P:0.698, D-fake:0.1678 and D-real:0.172.

5. Results

We tested our model qualitatively and quantitatively.

5.1. Qualitative results

You can see some of our results for background detection part in figure 6. On the left column we have a random video frame, and on the right we see the estimated background for that video. Our background is not perfect for some parts like the curtain for the first video and the floor for the second video, because our background detection method is very sensitive to subject's movements. If the subject doesn't move uniformly from side to side the model can't capture some background pixels correctly, and we will end up with some of the foreground pixel values in the background as you can see in figure 6.

For background detection part we had issues with memory, and didn't have enough space for loading all video frames in memory and computing the median. Hence, in practice we choose one frame from every k subsequent frames and take the median of selected frames for computing the background. That is another reason that we don't have perfect backgrounds. The more frames we have, the more accurate background we expect our method to give. Transfer results for multiple source and target subjects can be seen in Figure 7. We have shown our intermediate and final pose and background transfer results for two source subjects and one target subject in three different frames. In the

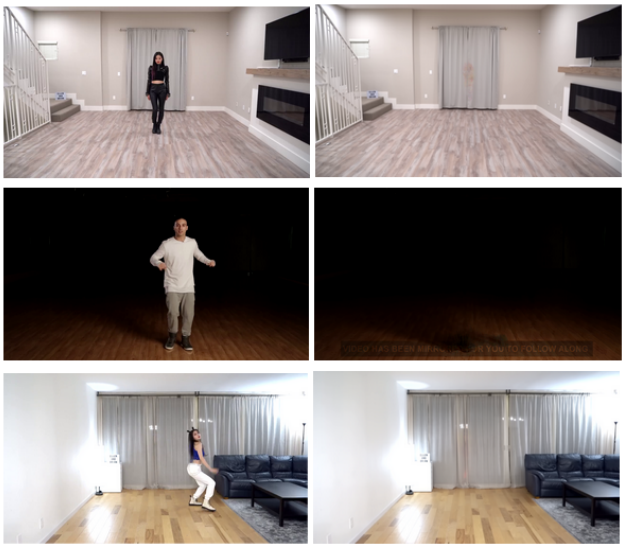

Figure 6. Detected backgrounds from the input video frames.

final synthesized images shown in the last column of Figure 7, you can see that the pose and background are coming from the source video and the subject's appearance comes from the target video. As you can see in the picture 7 some body parts such as face is not generated with a high quality since the face is very detailed and hard to synthesize. In [6] they propose a separate pipeline for generating faces that yields more realistic results with higher quality, however we didn't take use of that due to our limited time. Besides, we can see that feet of the person is not synthesized accurately for the sample target subject. The reason is that sometimes the model ends up keeping some of the foreground pixel values in the background particularly when the subject doesn't move much in diverse directions, that causes having poor segmentation masks for the target subject. Therefore, the final blending and synthesizing wouldn't be perfect.

5.2. Quantitative results

We evaluated our model using Mean Per Joint Position Error (MPJPE) metric and compare the results with the baseline that you can see in table 1. In our framework, this metric calculates the Euclidean distance between the source frame pose and the pose extracted from the synthesized frame, that can measure how accurate our pose transfer results are. As you can see in the table 1, our error is larger than the baseline model, however we are using the same method as the baseline paper for generating a person condition on the pose. We guess that the difference is due to having an extra step in our model for blending the foreground and background, as our masks are not very accurate in some small and detailed parts like feet.

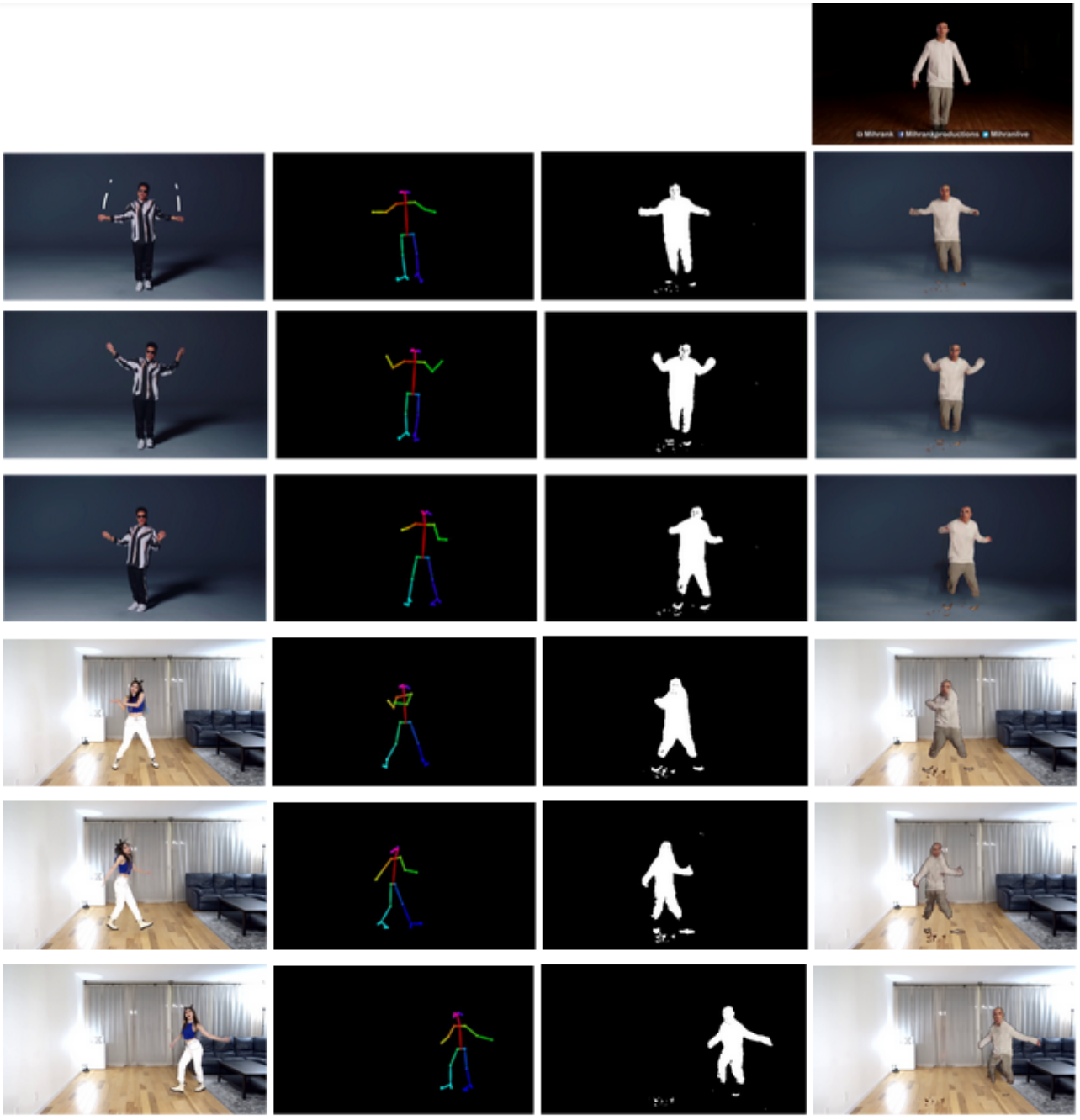

Figure 7. Transfer results. The top image is one random target frame, from which we get the foreground appearance. The first column is source video frames, the second column is extracted pose stick figures from the source frames with Openpose, the third column is segmentation masks corresponds with each frame, and the fourth column is our final background and foreground blending results.

6. Discussion and Conclusion

In general, we could generate arbitrarily long, and good-quality videos of a target person dancing given the movements of a source dancer. To transfer the motion, we extract poses from the source subject with pre-trained Openpose model and apply the pose-to-appearance mapping to generate the target subject, and finally we blend the target's foreground with the source video's background with respect to the segmentation mask. The framework is very simple, however, it suffers from several limitations.

As mentioned before, highly detailed parts such as faces

Method	MPJPE
Baseline	0.0073
Ours	0.0102

Table 1. Results of MPJPE metric for our model compared and the baseline model in m.

are not generated with a good quality. In future, a separate pipeline like FaceGan [6] can be used for generating higly-detailed regions. Another limitation is that the assumption of having a static background and a foreground that moves diversely in different directions for our background detection method is very strong. In practice, there are not much videos which satisfies this constraint, that could affect the quality of the background and segmentation mask negatively. Hence, in future, another techniques for background detection and segmentation such as Mask R-CNN [11] could be tested. Nevertheless, the model is able to generalize to new motions fairly well from the training data. The source subject could perform any arbitrary dance moves and the target subject also does not need to perform similar motions to any source. The model itself generalizes well to a wide range of source and target motions. However the model sometimes struggles to extrapolate to different poses. For example, artifacts can occur if the source motion contains poses such as handstands if the target training data did not contain such upside-down poses. Future work could focus on the training data, i.e. what poses and how many are needed to learn an effective model or which training examples are most influential.

References

[1] Wissam J. Baddar, Geonmo Gu, Sangmin Lee, and Yong Man Ro. Dynamics transfer GAN: generating video by transferring arbitrary temporal dynamics from a source video to a single target image. *CoRR*, abs/1712.03534, 2017. 2

[2] Harry Blum. A Transformation for Extracting New Descriptors of Shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967. 2

[3] Christoph Bregler, Michele Covell, and Malcolm Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, page 353–360, USA, 1997. ACM Press/Addison-Wesley Publishing Co. 1

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1812.08008, 2018. 1

[5] Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 4d video textures for interactive character appearance. *Comput. Graph. Forum*, 33(2):371–380, May 2014. 2

[6] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5933–5942, 2019. 1, 2, 5, 7

[7] German K. M. Cheung, Simon Baker, Jessica Hodgins, and Takeo Kanade. Markerless human motion transfer. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, 3DPVT '04, page 373–378, USA, 2004. IEEE Computer Society. 1

[8] Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, pages 726–733, Nice, France, 2003. 1

[9] T. V. Gopal and V. Kamakshi Prasad. A novel approach to shape based image retrieval integrating adapted fourier descriptors and freeman code. 2008. 2

[10] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. *CoRR*, abs/1802.00434, 2018. 1

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 7

[12] B. Ibragimov, B. Likar, F. Pernuš, and T. Vrtovec. Shape representation for efficient landmark-based segmentation in 3-d. *IEEE Transactions on Medical Imaging*, 33(4):861–874, 2014. 2

[13] P. Isola, J. Zhu, T. Zhou, and A. Efros. Image-to-image translation with conditional adversarial networks. *CVRP*, 2017. 2, 3, 5

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 1

[15] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Trans. Graph.*, 37(4), July 2018. 2

[16] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello. Lookingood: Enhancing performance capture with real-time neural re-rendering. *ACM Trans. Graph.*, 37(6), Dec. 2018. 2

[17] Helge Rhodin, Victor Constantin, Isinsu Katircioglu, Mathieu Salzmann, and Pascal Fua. Neural scene decomposition for multi-person motion capture. *CoRR*, abs/1903.05684, 2019. 2, 4

[18] K. Simonyan and A. Zisserman. Very deep convolutions networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[19] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *CoRR*, abs/1707.04993, 2017. 2

[20] Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. Video-based characters: Creating new human performances from a multi-view video database. *ACM Trans. Graph.*, 30(4), July 2011. 2

[21] J. Zhu, T. Park, F. Isola, and A. Efros. Unpaired image-to-image translation using cycle- consistent adversarial networks. *In Computer Vision (ICCV), 2017 IEEE International Conferenc*, 2017. 2