Few-Shot Human Motion Transfer by Personalized Geometry and Texture Modeling

Zhichao Huang[1], Xintong Han[2], Jia Xu[2], Tong Zhang[1]
[1]The Hong Kong University of Science and Technology [2]Huya Inc
zhuangbx@connect.ust.hk, {hanxintong, xujia}@huya.com, tongzhang@ust.hk

Abstract

We present a new method for few-shot human motion transfer that achieves realistic human image generation with only a small number of appearance inputs. Despite recent advances in single person motion transfer, prior methods often require a large number of training images and take long training time. One promising direction is to perform few-shot human motion transfer, which only needs a few of source images for appearance transfer. However, it is particularly challenging to obtain satisfactory transfer results. In this paper, we address this issue by rendering a human texture map to a surface geometry (represented as a UV map), which is personalized to the source person. Our geometry generator combines the shape information from source images, and the pose information from 2D keypoints to synthesize the personalized UV map. A texture generator then generates the texture map conditioned on the texture of source images to fill out invisible parts. Furthermore, we may fine-tune the texture map on the manifold of the texture generator from a few source images at the test time, which improves the quality of the texture map without over-fitting or artifacts. Extensive experiments show the proposed method outperforms state-of-the-art methods both qualitatively and quantitatively. Our code is available at https://github.com/HuangZhiChao95/FewShotMotionTransfer.

1. Introduction

Human motion transfer [7, 9, 12, 17, 22, 27, 34, 44, 45] generates videos of one person that takes the same motion as the person in a target video, which has huge potential applications in virtual characters, movie making, *etc*. The rapid growth of generative networks [11] and image translation frameworks [15, 40] enables generating photo-realistic images for human motion transfer. Basically, one would extract the pose sequence of the target video and take the pose as the input to generate the video for a new person.

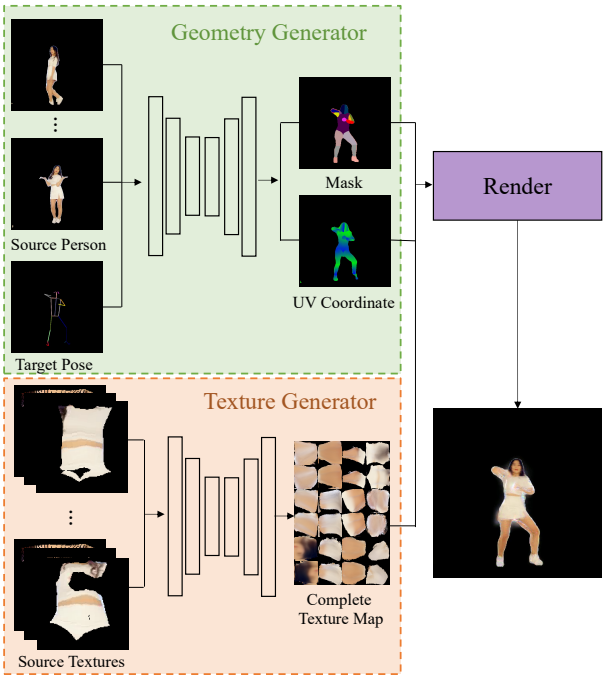


Figure 1. Method overview. Our method contains three key components: a geometry generator for generation of a personalized UV map, a texture generator to fill out incomplete texture, and a neural renderer rendering the human image. Detailed network structures are illustrated in Figure 2.

The appearance of the new person can be provided in two ways. One type of models aim to train an individual model for a specific person. To obtain such a model, one needs to collect a large number of images for the new person and trains a network to translate the pose to the image of the person [7, 39]. Then the appearance information is stored in the weights of the network and the image of new pose can be directly generated by taking the new pose as input. However, such methods need a large amount of training data and training time to obtain a model for the new person, which hinders the applications of these approaches.

For the other type, the information of appearance is pro-

vided by taking a few images of the new person as input. Few-shot human motion transfer requires the network to learn the complicated relationship between human appearance and pose by only looking at few human images. The relationship is very hard to learn and generalize to unseen people. Therefore, directly conditioning the output image on the pose and the appearance leads to poor quality. Some approaches warp the input appearance to the output with optical flows [38] or affine transformations [3, 44] to generate a coarse pose of the new person. However, the mapping is usually inaccurate and fails to recover realistic human from the intermediate warping result. Even if the architecture gets more and more complicated, there are still many artifacts in the images of few-shot human motion transfer.

The appearance of one person at different pose is the same. Therefore, we can directly transfer the pixels from the source pose to the target pose without generating the pixels. DensePose [2] provides the UV map of one person so that the texture can be transferred between different poses to synthesize the human at the new pose. However, directly using the original model to transfer the texture fails to generate realistic human image [2, 28].

The DensePose can be trained to better fit the generation of human and achieves high quality avatars [31]. However, their method is only suitable for single person and cannot be directly used for few-shot synthesis. On the one hand, their generator is not able to synthesize accurate geometry (*i.e*. IUV representation) of different people whose shapes are different. On the other hand, their method cannot produce complete texture map from only few the source human images. We propose a new method that generalizes the algorithm for the few-shot scenario and get better results than previous few-shot approaches. As shown in Figure 1, we use a geometry generator to generate personalized UV map given a target pose and a few source images. Meanwhile, a texture generator merges each incomplete texture map and hallucinates the invisible. Then the texture map is rendered to the UV map to generate an image with target pose and source appearance. The decoupling generation of personalized geometry and texture leads to better quality of motion transfer.

We summarize our contributions as follows:

1. We propose a geometry generator to predict an accurate personalized UV map and a texture generator to generate a complete texture map. These two generators work collaboratively for rendering high quality human images.

2. By training on multiple videos of multiple persons and fine-tuning on a few examples of an unseen person, our method successfully transfers geometry and texture knowledge to the new person.

3. Experiments demonstrate that our method generates better human motion transfer results than state-of-the-art methods both qualitatively and quantitatively.

2. Related Work

Human Motion Transfer. There have been a lines of work about synthesizing a human image in an unseen pose [3, 7, 10, 23, 28, 32]. Most of the methods implement a generator condition on 2D keypoints (or connection of the keypoints) of the pose. One type of the methods train different models for different persons. EDN [7] utilize pix2pixHD [40] framework to translate 2D skeleton to the image of a specific person. Vid2Vid [39] uses more complicated network for generation of the video, which contains foreground-background separation and optical flow warping module. While single person model generates photo-realistic picture of the human, it requires collecting training data for each person and takes long time to train.

Another type of the methods train single model to transfer motion for all persons. As appearance information needs to be obtained from the source images, which makes the task much more complicated, many papers add additional modules for synthesizing intermediate images and use them as one input for later generative networks. The additional modules include affine transformation [3, 44], flow warping [38], DensePose transfer [28] and SMPL transfer [22, 24]. Our method also uses DensePose for modeling the geometry of the pose, but we directly render the texture to the DensePose to produce final outputs instead of using it as the intermediate layer. While our method requires accurate personalized UV map and high quality texture map, we omit directly generating the pixel. Multi-stage methods also generate the coarse intermediate images with neural network generators [26, 32, 45]. In addition, Siarohin *et al*. [34] and Liu *et al*. [24] modify blocks of the network to adapt the task. MonkeyNet [33] does not depend on the 2D pose. Instead, it extracts and maps the keypoints of the source and target image and animates any objects by the motion of these keypoints. Moreover, fine-tuning on a few source images of one person can improve the quality of the output [21]. Fine-tuning is also part of our method. However, we mainly fine-tune the texture map, which seldom overfits to the few source images at test time.

Human Avatars. Full-body human avatars are usually represented by textured animatable 3D human models. There have a large group of works on building 3D model from single-view or multi-view images [1, 4, 14, 20, 25]. Many 3D models of human are based on SMPL [25], a parametrized model describing the shape and pose of human. Lazova *et al*. [20] build fully-textured avatars from a single input image. Starting from SMPL, it uses neural network to model the displacement of geometry and complete partial texture of human. PIFu [30] learns an implicit function to align surface and texture so that human avatar can be reconstructed from single-view or multi-view images. Aliaksandra *et al*. [31] learn the translation from skeleton to UV map from a video. And full texture is generated by di-

rectly optimizing the output image of neural rendering. Our method is close related to [31], but we need to train one model for all persons in a few-shot setting instead of training a single model for each person. So our model should be able to synthesize personalized geometry and hallucinate unseen texture map.

3. Personalized Geometry and Texture Model

Given a target pose P and a few images of the source person $I_1, ..., I_b$, our goal is to learn a model f that synthesizes an image

$$\widehat{I} = f(P, \{I_j\}_{j=1}^{b}), \tag{1}$$

in which the generated person has the pose P and the appearance of source images $I_1, ..., I_b$. In this paper, we assume the source person images share a fixed background, so we can separate the generation of human image $\widetilde{I}$ and background B as:

$$\widehat{I} = (1 - m) \odot B + m \odot \widetilde{I}, \tag{2}$$

where m is the soft mask indicating the human image foreground $\widetilde{I}$, and $\odot$ represents element-wise product. The mask m can be easily obtained with an off-the-shelf person segmentation model like DeepLab V3 [8].

3.1. Neural Human Rendering

$\widetilde{I}$ can be directly generated from the pose P with an image-to-image translation network [3, 7, 15]. However, these methods fail to model complex geometric changes and detailed textures of the person, resulting in low-quality outputs especially when only few training samples of the source person are available. To mitigate this issue, we takes a neural rendering based approach [31, 37] by dividing the image synthesis into two parts: generation of a personalized human geometry (UV map) and generation of personalized human texture (texture map). And the person image can be generated by sampling the texture map according to the UV map as shown in Figure 1. On the one hand, while human geometry varies across different persons, the variation is much smaller than that of person images. Therefore, generating personalized geometry is easier than directly generating the human image. Texture map, on the other hand, is fixed for one person without complex geometric changes. So it can be effectively learned from the source textures.

DensePose [2] descriptors have been widely adopted for disentangling the generation of human geometry and texture [28, 31]. We follow this line of work and subdivide human's body into $n = 24$ non-overlapping parts. The k-th body part ($k = 1, 2..., n$) is parameterized by a 2D coordinate (C^{2k}, C^{2k+1}) (*i.e.*, a UV map) and associated with a texture T^k as shown in Figure 1. Then, we can render the k-th part with bilinear interpolation:

$$R^k[x, y] = T^k[C^{2k}[x, y], C^{2k+1}[x, y]]. \tag{3}$$

For pixel $[x, y]$ of the image, we assign a score $S^k[x, y]$ to the k-th part of the DensePose, representing the probability that the pixel belongs the k-th part. $\sum_{k=1}^{n+1} S^k[x, y] = 1$, where $S^{n+1}[x, y]$ indicates the probability that $[x, y]$ belongs to the background (*i.e.*, $(1 - m)$ in Equation (2)). By summing up the rendered part weighted by its probability, we generate the image of the human as:

$$\widetilde{I} = \sum_{k=1}^{n} S^k[x, y] R^k[x, y]. \tag{4}$$

In this work, we design a geometry generator (Section 3.2) to estimate body part score S^k and its UV map (C^{2k}, C^{2k+1}), as well as a texture generator (Section 3.3) for generating body part texture T^k. These two generators can be trained collaboratively and render the human image with transferred motion using Equations (3) and (4).

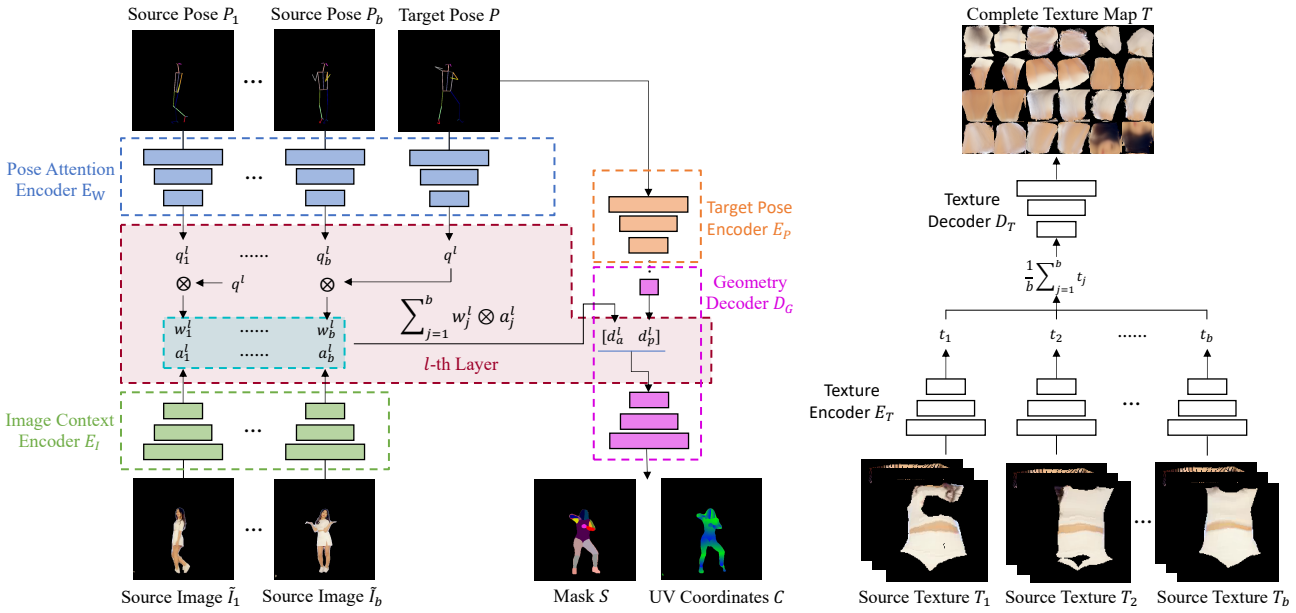
3.2. Geometry Generator

As shown in Figure 2(a), the body UV geometry not only depends on the target pose P but also needs to be personalized, varying across different persons. To this end, our geometry generator G_ϕ takes the pose P as input to generate the geometry with desired pose, and at the same time, it also takes the source human images $\widetilde{I}_1, ..., \widetilde{I}_b$ to model personalized details (*e.g.*, hair style, clothing, body shapes):

$$\begin{aligned} C &= G_\phi^C(P, \{\widetilde{I}_1, \widetilde{I}_2, ..., \widetilde{I}_b\}), \\ S &= G_\phi^S(P, \{\widetilde{I}_1, \widetilde{I}_2, ..., \widetilde{I}_b\}), \end{aligned} \tag{5}$$

where b is the number of source human images provided as input to the geometry generator. We denote C^k for the k-th channel of $C \in \mathbb{R}^{48 \times H \times W}$, and C^{2k}, C^{2k+1} are the U- and V-coordinate for the k-th body part, respectively. $S \in \mathbb{R}^{25 \times H \times W}$, whose k-th channel S^k is the soft assignment mask for k-th part of the DensePose, represents the probability that a pixel belongs to which part of the body. Note that during training, P is different from the poses of $\widetilde{I}_1, ..., \widetilde{I}_b$, so that the network is forced to rely the on the target pose P when extracting geometric information.

As shown in Figure 2(a), the geometry generator contains three encoders and one decoder: an image context encoder E_I, a pose attention encoder E_W, a target pose encoder E_P, and a geometry decoder D_G. The target pose encoder E_P extracts the information of target pose P as input to the geometry decoder D_G to ensure the generated geometry reflects the desired pose. The image context encoder E_I extracts personalized body information from $\widetilde{I}_1, ..., \widetilde{I}_b$. The pose attention encoder E_W then compares the similarity between target pose P and source poses $P_1, ..., P_b$ to determine which source images should be paid more attention to for incorporating personalized shape details in the generated geometry output by D_G.



(a) Geometry Generator (Section 3.2)

(b) Texture Generator (Section 3.3)

Figure 2. Architecture of our proposed geometry generator (Section 3.2) and texture generator (Section 3.3). The generated geometry and texture are finally used to render a reconstructed human image (Section 3.1).

We implement our geometry generator with a U-Net [29] like architecture. And we perform the above feature interaction at each corresponding layer of the three encoders and the decoder (*i.e.*, layers with the same spatial resolution). Consequently, the architecture captures multi-level features from the input images and poses at different resolutions, which helps to generate high quality geometry with accurate personalized shape and pose.

More specifically, the encoder E_I, E_W, E_P and decoder D_G have the same number of layers L. We define E^l as l-th layer of the encoder, and D^l as $(L-l)$-th layer of the decoder. At l-th layer of the encoders, we calculate the feature for $(l+1)$-th layer as:

$$a_j^{l+1} = E_I^l(a_j^l), \quad q_j^{l+1} = E_W^l(q_j^l), \tag{6}$$

$$w_j^l = (q_j^l)^\top \otimes q^l, \quad d_a^l = \textstyle\sum_k w_j^l \otimes a_j^l, \tag{7}$$

$$d_p^{l+1} = D_G^l([d_a^l, d_p^l]), \tag{8}$$

where $j = 1, ..., b$ is the index of the source images, $a_j^0 = \widetilde{I}_j$, $q_j^0 = P_j$, and $d_p^0 = E_P(P)$. $\otimes$ denotes matrix multiplication. In Equation (6), E_I^l and E_W^l encode features from previous layer. Equation (7) shows how we merge the personalized shape information from different source images using an attention mechanism. As different source images carry different shape information, when predicting the geometry C and S, we give different weights to these inputs according to their similarities with the target pose P. For instance, if P describes the side view of the person, it may be hard to infer the detailed shape geometry from front-side images. So, we should give more attention to the images whose pose is similar to the target pose. In Equation (7), we reshape a_j^l and q_j^l into $\mathbb{R}^{C_l \times N_l}$ where C_l is the number of channels and $N_l = H_l \times W_l$ denotes the spatial size of the feature map, then we perform matrix product with $\otimes$. At last, as shown in Equation (8), d_a^l is reshaped back into $C_l \times H_l \times W_l$, concatenated with d_p^l, and fed into D_G^l.

3.3. Texture Generator

Our texture generator is responsible for generating a full human texture map T given the source images. An intuitive approach would be directly extracting a DensePose texture map from each source image I_j and aggregating them (*e.g.*, through spatial average or max pooling) to get a merged texture map. However, such merged texture map is usually incomplete since not all body parts are visible from the given source images. Plus, due to inaccurate DensePose estimation, this texture map is unrealistic and lacks of fine texture details. To solve this problem, we introduce a texture generator H_θ that takes the textures $T_1, ..., T_b$ extracted by DensePose from source images $I_1, ..., I_b$ to synthesize the complete texture T in a learnable fashion:

$$T = H_\theta(T_1, T_2, ..., T_b). \tag{9}$$

The architecture of our texture generator is a vanilla encoder-decoder as shown in Figure 2(b). We reshape the input texture T_j from $24 \times 3 \times H_T \times W_T$ to $72 \times H_T \times W_T$ and feed it to H_θ. The encoder E_T encodes each texture T_j to the bottleneck embedding denoted as t_j. We merge the textures of different source images by taking average of

their embedding features:

$$t = \frac{1}{b}\sum_{j=1}^{b} t_j. \tag{10}$$

Note that the architecture allows different numbers of input textures. And t is then fed into decoder D_T to produce the complete texture map of size $72 \times H_T \times W_T$, which is finally reshaped into $T \in \mathbb{R}^{24\times 3\times H_T \times W_T}$. Our texture generator not only produces the complete texture map T, but also defines a manifold of human body textures. As we will discuss later, we may fine-tune the embedding t at test-time to improve the quality of texture map T.

4. Training

Our training process consists of three stages: an initialization stage, a multi-video training stage, and an optional few-shot fine-tuning stage.

4.1. Initialization

Training our geometry and texture generators with image-level reconstruction loss from scratch is infeasible, as the model does not have any prior information about human body. For example, it is impossible for the geometry generator to learn body semantics and output a human mask S, of which each channel corresponds to a specific body part, without any explicit supervision. Thus, we follow [31] and use the output of an off-the-shelf DensePose extractor [2] to initialize our networks.

Geometry Generator. For a target ground truth image I we aim to reconstruct, we take the pseudo ground truth body part mask S^* and UV-coordinate C^* extracted by the DensePose model as the supervision signal to initialize the geometry generator G_ϕ by minimizing:

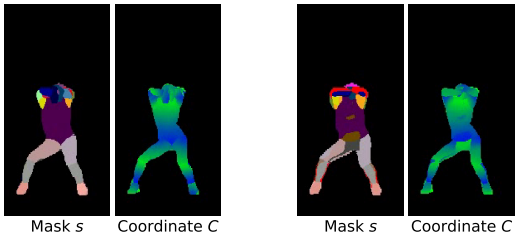
$$\begin{aligned} L_C =& \sum_{k=1}^{24}(\|S^{*k} \odot (C^{2k} - C^{*2k})\|_1 + \\ & \|S^{*k} \odot (C^{2k+1} - C^{*2k+1})\|_1), \end{aligned} \tag{11}$$

$$L_S = L_{CE}(S, S^*), \tag{12}$$

where C and S are the outputs of our geometry generator as in Equation (5). L_C is the L1 norm between C^* and C on the given body part. And L_S is the cross-entropy loss as used in semantic segmentation that guides the generator to predict the same body part masks as S^*.

Texture Generator. The texture generator is initialized by requiring its output to have the same texture on the visible part of its inputs. Suppose σ_j is the binary mask indicating the visible part of the input texture T_j, we optimize H_θ with the following pixel L1 loss:

$$L_T = \sum_{j=1}^{b} \|\sigma_j \odot (T - T_j)\|_1 \tag{13}$$



(a) With Regularization (b) Without Regularization

Figure 3. Impact of adding regularization terms (Equations (16) and (17)) to the loss. The belly and leg regions are messy without regularization.

4.2. Multi-video Training

After the initialization, our model can roughly generate human geometry and texture. However, the generated geometry is similar to the pre-trained DensePose outputs that lack personalized shape details. Also, the texture map is only coarsely rendered, missing important detailed textures that are necessary for generating realistic humans.

To this end, we train our generators on multiple training videos after the initialization. In each training mini-batch, we only sample data from one person so that the texture T and feature d_a can be shared across different target pose P. Besides, training time is saved as T and d_a only need to be generated once for all target pose in this mini-batch. We consider the following losses during training.

Image Loss. Image loss makes the reconstructed human $\widetilde{I}$ to be closer the ground truth human image $m \odot I^*$ in both image space and the feature space of a neural network [16]. Suppose Φ^v is an intermediate feature of a pre-trained VGG-19 network [35] at different layers. The image loss is:

$$L_I = \|\widetilde{I} - m \odot I^*\|_1 + \sum_{v=1}^{N} \|\Phi^v(\widetilde{I}) - \Phi^v(m \odot I^*)\|_1, \tag{14}$$

Mask Loss. The mask loss is a cross-entropy (CE) loss between the generated background mask S^{25} and a pseudo ground truth background mask $1 - m$ output by a SOTA segmentation model [8]:

$$L_M = L_{\text{CE}}(S^{25}, 1 - m). \tag{15}$$

Generator Regularization Loss. The rendering process makes the optimization of geometry and texture ambiguous. There would be infinite combinations of geometry and texture map to render the same human image. As the neural network have high flexibility, we need to constrain dramatic changes of texture, coordinates and mask. Otherwise, the geometry and texture generators are prone to overfitting and cannot generalize to unseen people and poses. As shown in Figure 3, the mask and coordinate gets irregular without the regularization. Thus, we introduce a regularization loss that is similar the losses at the initialization stage to prevent the

network from generating unrealistic results or overfitting. The regularization loss contains a texture term:

$$L_{RT} = \sum_{j=1}^{b} \|\sigma_j \odot (T - T_j)\|_1, \tag{16}$$

a coordinate term:

$$\begin{aligned} L_{RC} = \sum_{k=1}^{24} (&\|S^k \odot (C^{2k} - C^{*2k})\|_1 + \\ &\|S^k \odot (C^{2k+1} - C^{*2k+1})\|_1), \end{aligned} \tag{17}$$

and a body part mask term that ensures S do not deviates too much from S^* output by the DensePose model on the foreground mask:

$$L_{RM} = L_{CE}((1 - S^{*25}) \odot S, (1 - S^{*25}) \odot S^*), \tag{18}$$

The total loss can be expressed as follows:

$$L = \lambda_I L_I + \lambda_M L_M + \lambda_{RT} L_{RT} + \lambda_{RC} L_{RC} + \lambda_{RM} L_{RM}, \tag{19}$$

where λ's are the weights balancing the contribution of each loss term. Note that we do not add adversarial loss during image reconstruction as we find it would add instability to the training of geometry generator.

4.3. Few-shot Fine-tuning

The model trained at our multi-video training stage can be readily used for generating human motion transfer results on unseen people and poses. Fine-tuning on a few source images at test time is an optional step that greatly improves the quality of the synthesized images.

Since we only need to generate one texture map T for one person at test time, fine-tuning of texture map seldom overfits. If the texture map gets closer to the texture of source images, the texture of synthesized human at other pose becomes more photo-realistic.

We fine-tune the embedding t in Equation (10) to produce a smooth and realistic texture map. Compared with directly optimizing texture map T as in [31], fine-tuning embedding causes few artifacts to the texture map and it is able to hallucinate incomplete textures as shown in Figure 7. During test time fine-tuning, we first initialize t by averaging the embedding of source textures $T_1, ..., T_n$. And generate complete image $\widehat{I}$ with Equation (2) and fine-tune t by minimizing:

$$\widehat{L}_I = \|\widehat{I} - I^*\|_1 + \sum_{v=1}^{N} \|\Phi^v(\widehat{I}) - \Phi^v(I^*)\|_1 \tag{20}$$

$$L_{test} = \lambda_I \widehat{L}_I + \lambda_M L_M + \lambda_{RC} L_{RC} + \lambda_{RM} L_{RM}. \tag{21}$$

We do not add L_{RM} as we do not hope to constrain the optimization of texture map at test time. Meanwhile, we also fine-tune the geometry generator G_ϕ and through L_{test} for few steps since the geometry generator can already generalize pretty well to unseen person geometries and fine-tuning for a long period may lead to overfitting. We also fine-tune background B through L_{test} as it can eliminate artifacts for merging backgrounds.

5. Experiments

Dataset. We collect 62 solo dance videos with almost static background from YouTube. The videos contain several subjects with different genders, body shapes, and clothes (examples can be found in Figure Figure 6). Each video is trimmed into a clip lasting about 3 minutes. We further divide them into training and test set with no overlapping subjects. Training set contains 50 videos with 283,334 frames and test set contains 12 videos with 70,240 frames.

Preprocess. For each frame I in the dataset, we crop the center part of the image and extract the 25 body joints with OpenPose [5, 6]. The joints are connected to form a "stickman" image as the input pose P. UV coordinate C^* and mask S^* are extracted with DensePose model [2]. DensePose just gives a coarse segmentation of the human. We use Deeplab V3 [8] to get foreground mask m separate foreground human image $\widetilde{I}$. Source texture $\widetilde{T}$ is produced by warping the image I according to C^*.

Implementation Details. All video frames are resized to 256×256 while the size of body part texture map is set to 128×128. The input pose is a "stickman" image with 26 channels and each channel contains one "stick" of the pose. The encoder and decoder for geometry and texture generators are built on basic convolution-relu-norm blocks, and we include more detailed illustrations in the supplementary material. Geometry generator contains about 60M parameters and that of texture generator is around 34M.

Both generators are training using Adam [19] optimizer with $(\beta_1, \beta_2) = (0.5, 0.999)$. Learning rate starts at 0.0002. The initialization stage lasts for 10 epochs and the learning rate decays half at the 5th epoch. We train the multi-video stage for 15 epochs with learning decaying half at the 5th and the 10th epoch. At test time, we randomly select 20 images from one video as the source images. The number of fine-tuning steps is 40 for geometry generator and is 300 for the texture embedding. We generate the background B by directly merging visible background from source images and fill the left invisible parts with deepfillv2 [42].

Compared Methods. We compare our method with state-of-the-art human motion transfer approaches: Posewarp [3], MonkeyNet [33], FewShotV2V [38] and Liquid Warping GAN (LWG) [24]. We use the source code provides by the authors to train the model. For those providing pre-trained models (Posewarp, LWG), we use it as initialization and train on our dataset for fair comparison. Otherwise, we train the model from scratch. At test time, we fine-tune all these

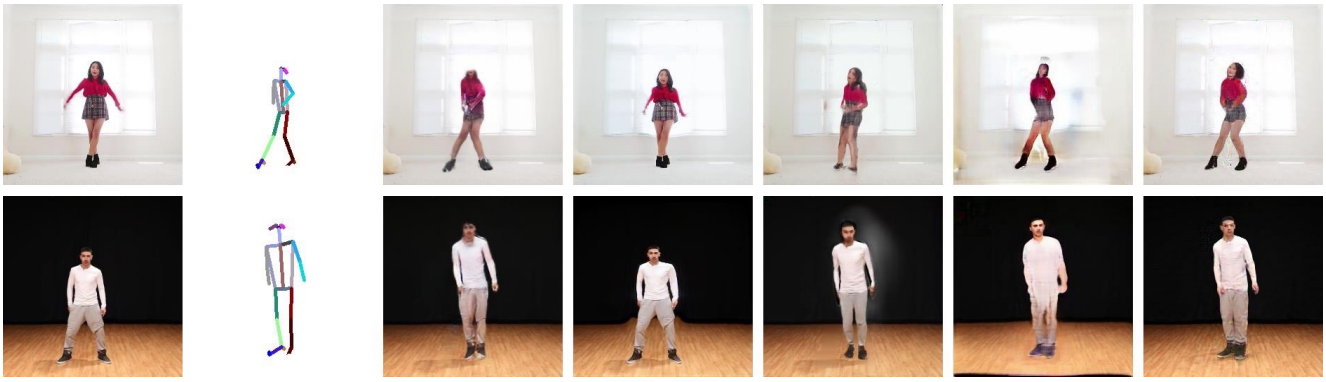


Figure 4. Qualitative comparison of our method to state-of-the-art methods. *More video results can be found in the supplementary material.*

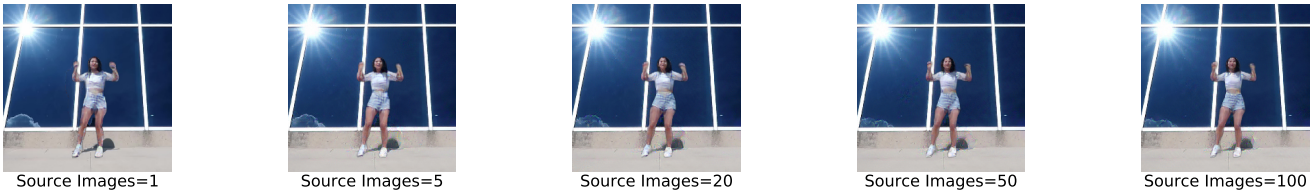


Figure 5. Synthesized human with different number of source images.

Table 1. Quantitative evaluation metrics on our test set. ↑ represents higher is better, and ↓ means lower is better.

Methods	Reconstruction		Motion Transfer	
	SSIM↑	LPIPS↓	FReID↓	PoseError↓
Posewarp [3]	0.808	0.175	5.73	12.46
MonkeyNet [33]	0.763	0.234	13.40	41.49
LWG [24]	0.760	0.238	9.99	14.45
FewShotV2V [38]	0.694	0.332	8.24	10.50
$\text{Ours}_{\text{w/o Fine-tuning}}$	0.837	0.166	4.91	**6.48**
$\text{Ours}_{\text{w/ Fine-tuning}}$	0.861	0.157	3.78	6.58
$\text{Ours}_{\text{Direct Merge}}$	0.843	0.183	4.40	8.09
$\text{Ours}_{\text{Texture Map}}$	**0.881**	**0.151**	**3.23**	7.27

Table 2. User study of human motion transfer. The numbers indicate the percentage of clips that the users prefer our method to each of the competing method. Chance is 50%.

Posewarp	MonkeyNet	LWG	FewShotV2V
98.40%	99.47%	87.50%	90.43%

models on the source images. As only FewShotV2V accepts multiple inputs, for other methods, the source image is chosen to be the first one.

5.1. Quantitative Comparisons

Evaluation Metrics. For each video in the test set, we consider two evaluation settings: reconstruction and motion transfer. For the reconstruction, we set the pose sequence from the video of the source person as the target motion. The generator is asked to reconstruct images to be the same as the ground-truth video. We compare the similarity of the synthesized and ground-truth image using SSIM [41] and Learned Perceptual Similarity (LPIPS) [43]. For motion transfer, we extract the pose sequence from one video in the test set and let a person in other videos imitate the pose sequence. We compare the quality of the images in two aspects: Pose Error [7, 38] and FReID [24]. Pose Error is the average L2 distance (in pixels) of 2D pose keypoints between of synthesized human and target pose, which estimates the accuracy of the transferred motion. FReID is a Fréchet Distance [13] on a pre-trained person-reid model [36], which measures the quality of generated appearance. The results are shown in Table 1. Our method achieves higher similarity in reconstruction and lower discrepancy of pose and appearance in motion transfer.

User Study We run all the methods on 30 randomly chosen 5-second video clips for human motion transfer and ask 8 people to perform user study. In each trial, given the motion transfer results of the five methods on the same video clip, the users are asked to select the method with the best generation quality. The percentage of trials that our method is preferred is shown in Table 2. We can find that our method is favored in most of the clips.

5.2. Qualitative Comparisons

Figure 4 visually compares our method with others. Our method outperforms state-of-the-art methods in terms of the image quality and pose accuracy. Posewarp fails to construct some parts of the body. FewShotV2V cannot generalize to new person with small number of training videos (their paper used 1500 videos). It only outlines the appearance of source person with plenty of artifacts. While LWG can synthesize a person with regular shapes, it is not able to generate person with complicated shape such as dress and

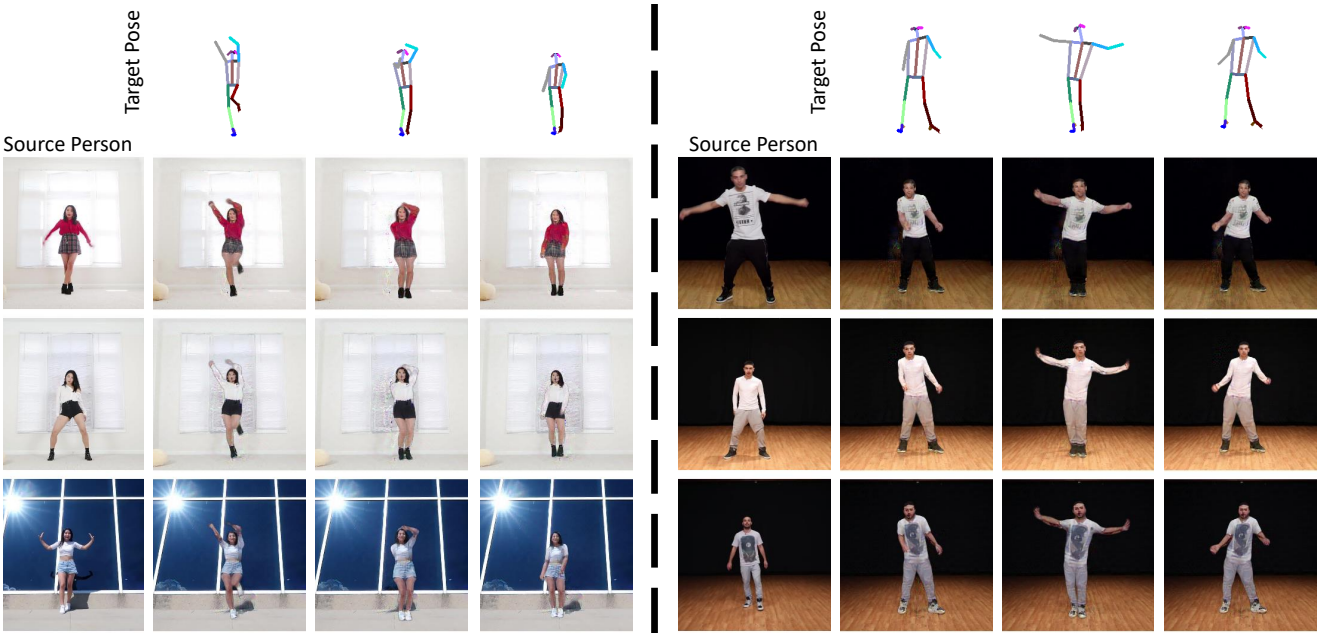


Figure 6. More examples of our method. Our method can generate personalized geometry and realistic texture for a wide variety of source persons. *More video results can be found at* https://youtu.be/ZJ15X-sdKSU.

Table 3. Performance with varied numbers of source images.

Source Images	1	5	20	50	100
SSIM↑	0.845	0.852	0.861	0.867	0.870
LPIPS↓	0.167	0.174	0.157	0.158	0.153
FReID↓	4.29	3.97	3.78	3.60	3.41
Pose Error↓	6.86	6.68	6.58	6.53	6.58

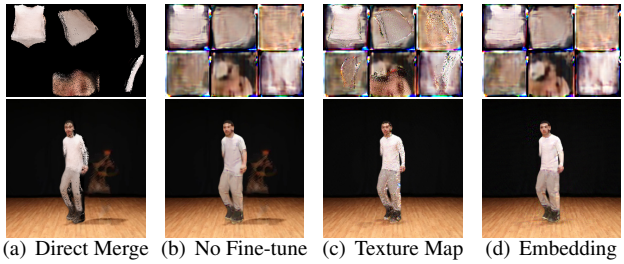
long hair, as detailed shape information cannot be modeled by HMR [18] used in LWG. Furthermore, HMR cannot accurately estimate the target pose, making generated results temporally discontinuous. Our method is capable of modeling the complicated personalized geometry of each person and preserve detailed appearance.

Figure 6 shows more examples of motion transfer generated by our method, which produces geometry with accurate pose and personalized shape details for a large variety of people. Besides, the texture is well preserved for the synthesized human, resulting in high-fidelity motion transfer.

5.3. Ablation Study

Number of Source Images. We vary the number of source images at test time from 1 to 100 and present the results in Figure 5 and Table 3. Our method achieves high quality motion transfer with only one source image, and the generation quality improves as more source images are utilized.

Texture Map. Figure 7 shows the learned texture maps and human images of four strategies: directly averaging the source textures, without fine-tuning, fine-tuning texture map, and fine-tuning the embedding. Quantitative metrics are shown in Table 1 ($\text{Ours}_{\text{w/o Fine-tuning}}$ means fig. 7(b), $\text{Ours}_{\text{w/ Fine-tuning}}$ means fig. 7(d)). Although fine-tuning texture map has better quantitative metrics, the noises of the texture make the synthesized video quite unnatural for human eyes. Texture generator fills in the invisible parts of the source texture and fine-tuning the embedding further improves the quality without suffering from visual artifacts.



(a) Direct Merge (b) No Fine-tune (c) Texture Map (d) Embedding

Figure 7. Texture maps of different fine-tuning schemes.

6. Conclusion

We proposed a novel method for few-shot human motion transfer, which decouples the task into generation of personalized geometry and texture. We designed a geometry generator that can extract shape information from source person images and inject it into generating personalized geometry of the source person in the target pose. In addition, a texture generator merges source textures and fills in invisible texture map. Extensive experiments demonstrate that the proposed method outperforms previous approaches for synthesizing realistic human motion. We see our method may be limited in coping with non-rigid moving parts like whipping hair or shaking dress. One future direction is to have the geometry generator take multiple continuous frames as input, and learn the temporally consistent motion.

References

[1] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[2] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7297–7306, 2018. 2, 3, 5, 6

[3] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8340–8348, 2018. 2, 3, 6, 7

[4] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017. 6

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017. 6

[7] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5933–5942, 2019. 1, 2, 3, 7

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018. 3, 5, 6

[9] Haoye Dong, Xiaodan Liang, Ke Gong, Hanjiang Lai, Jia Zhu, and Jian Yin. Soft-gated warping-gan for pose-guided person image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 474–484, 2018. 1

[10] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8857–8866, 2018. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014. 1

[12] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R Scott. Clothflow: A flow-based model for clothed person generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10471–10480, 2019. 1

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, 2017. 7

[14] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017. 1, 3

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016. 5

[17] Donggyu Joo, Doyeon Kim, and Junmo Kim. Generating a fusion image: One's identity and another's shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1635–1643, 2018. 1

[18] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7122–7131, 2018. 8

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 6

[20] Verica Lazova, Eldar Insafutdinov, and Gerard Pons-Moll. 360-degree textures of people in clothing from a single image. In *2019 International Conference on 3D Vision (3DV)*, pages 643–653. IEEE, 2019. 2

[21] Jessica Lee, Deva Ramanan, and Rohit Girdhar. Metapix: Few-shot video retargeting. In *International Conference on Learning Representations*, 2020. 2

[22] Yining Li, Chen Huang, and Chen Change Loy. Dense intrinsic appearance flow for human pose transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3693–3702, 2019. 1, 2

[23] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural rendering and reenactment of human actor videos. *ACM Transactions on Graphics (TOG)*, 38(5):1–14, 2019. 2

[24] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5904–5913, 2019. 2, 6, 7

[25] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2

[26] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 406–416, 2017. 2

[27] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 99–108, 2018. 1

[28] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 123–138, 2018. 2, 3

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 4

[30] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2304–2314, 2019. 2

[31] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, et al. Textured neural avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2387–2397, 2019. 2, 3, 5, 6

[32] Chenyang Si, Wei Wang, Liang Wang, and Tieniu Tan. Multistage adversarial losses for pose-based human image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 118–126, 2018. 2

[33] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2377–2386, 2019. 2, 6, 7

[34] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuiliere, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3408–3416, 2018. 1, 2

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[36] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 480–496, 2018. 7

[37] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 3

[38] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. Few-shot video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 5013–5024, 2019. 2, 6, 7

[39] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1, 2

[40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8798–8807, 2018. 1, 2

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 7

[42] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4471–4480, 2019. 6

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 7

[44] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara Berg. Dance dance generation: Motion transfer for internet videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1, 2

[45] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2347–2356, 2019. 1, 2