Textured Neural Avatars

Aliaksandra Shysheya [1,2] Egor Zakharov [1,2] Kara-Ali Aliev [1] Renat Bashirov [1]
Egor Burkov [1,2] Karim Iskakov [1] Aleksei Ivakhnenko [1] Yury Malkov [1]
Igor Pasechnik [1] Dmitry Ulyanov [1,2] Alexander Vakhitov [1,2] Victor Lempitsky [1,2]

[1]Samsung AI Center, Moscow [2]Skolkovo Institute of Science and Technology, Moscow

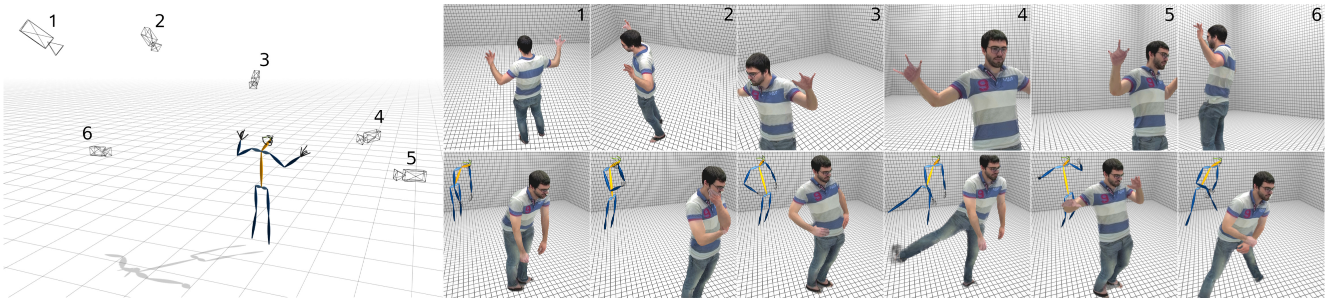

Figure 1: We propose a new model for neural rendering of humans. The model is trained for a single person and can produce renderings of this person from novel viewpoints (top) or in the new body pose (bottom) unseen during training. To improve generatlization, our model retains explicit texture representation, which is learned alongside the rendering neural network.

Abstract

We present a system for learning full body neural avatars*, i.e. deep networks that produce full body renderings of a person for varying body pose and varying camera pose. Our system takes the middle path between the classical graphics pipeline and the recent deep learning approaches that generate images of humans using image-to-image translation. In particular, our system estimates an explicit two-dimensional texture map of the model surface. At the same time, it abstains from explicit shape modeling in 3D. Instead, at test time, the system uses a fully-convolutional network to directly map the configuration of body feature points w.r.t. the camera to the 2D texture coordinates of individual pixels in the image frame. We show that such system is capable of learning to generate realistic renderings while being trained on videos annotated with 3D poses and foreground masks. We also demonstrate that maintaining an explicit texture representation helps our system to achieve better generalization compared to systems that use direct image-to-image translation.*

1. Introduction

Capturing and rendering human body in all of its complexity under varying pose and imaging conditions is one of the core problems of both computer vision and computer graphics. Recently, there is a surge of interest that involves deep convolutional networks (ConvNets) as an alternative to traditional computer graphics means. Realistic *neural rendering* of body fragments e.g. faces [37, 43, 62], eyes [24], hands [47] is now possible. Very recent works have shown abilities of such networks to generate views of a person with a varying body pose but with fixed camera position, and using an excessive amount of training data [1, 12, 42, 67]. In this work, we focus on the learning of *neural avatars*, i.e. generative deep networks that are capable of rendering views of individual people under varying body pose defined by a set of 3D positions of the body joints and under varying camera positions (Figure 1). We prefer to use body joint positions to represent the human pose, as joint positions are often easier to capture using marker-based or marker-less motion capture systems.

Generally, neural avatars can serve as an alternative to classical ("neural-free") avatars based on a standard computer graphics pipeline that estimates a user-personalized body mesh in a neutral position, performs skinning (deformation of the neutral pose), and projects the resulting 3D surface onto the image coordinates, while superimposing person-specific 2D texture. Neural avatars attempt to shortcut the multiple stages of the classical pipeline and to replace them with a single network that learns the mapping from the input (the location of body joints) to the output (the

2D image). As a part of our contribution, we demonstrate that, however appealing for its conceptual simplicity, existing pose-to-image translation networks generalize poorly to new camera views, and therefore new architectures for neural avatars are required.

Towards this end, we present a neural avatar system that does full body rendering and combines the ideas from the classical computer graphics, namely the decoupling of geometry and texture, with the use of deep convolutional neural networks. In particular, similarly to the classic pipeline, our system explicitly estimates the 2D textures of body parts. The 2D texture within the classical pipeline effectively transfers the appearance of the body fragments across camera transformations and body articulations. Keeping this component within the neural pipeline boosts generalization across such transforms. The role of the convolutional network in our approach is then confined to predicting the texture coordinates of individual pixels in the output 2D image given the body pose and the camera parameters (Figure 2). Additionally, the network predicts the body foreground/background mask.

In our experiments, we compare the performance of our *textured neural avatar* with a direct video-to-video translation approach [67], and show that explicit estimation of textures brings additional generalization capability and improves the realism of the generated images for new views and/or when the amount of training data is limited.

2. Related work

Our approach is closely related to a vast number of previous work, and below we discuss a small subset of these connections.

Building **full-body avatars** from image data has long been one of the main topics of the computer vision research. Traditionally, an avatar is defined by a 3D geometric mesh of a certain neutral pose, a texture, and a skinning mechanism that transform the mesh vertices according to pose changes. A large group of works has been devoted to body modeling from 3D scanners [51], registered multiview sequences [53] as well as from depth and RGB-D sequences [7, 69, 74]. On the other extreme are methods that fit skinned parametric body models to single images [6, 8, 30, 35, 49, 50, 59]. Finally, research on building full-body avatars from monocular videos has started [4, 3]. Similarly to the last group of works, our work builds an avatar from a video or a set of unregistered monocular videos. The classical (computer graphics) approach to modeling human avatars requires explicit physically-plausible modeling of human skin, hair, schlera, clothing surface, as well as explicit physically-plausible modeling of motion under pose changes. Despite considerable progress in reflectivity modeling [2, 18, 38, 70, 72] and better skinning/dynamic surface modeling [23, 44, 60], the computer graphics approach still requires considerable "manual" effort of designers to achieve high realism [2] and to pass the so-called uncanny valley [46], especially if real-time rendering of avatars is required.

Image synthesis using deep convolutional neural networks is a thriving area of research [27, 20] and a lot of recent effort has been directed onto synthesis of realistic human faces [15, 36, 61]. Compared to traditional computer graphics representations, deep ConvNets model data by fitting an excessive number of learnable weights to training data. Such ConvNets avoid explicit modeling of the surface geometry, surface reflectivity, or surface motion under pose changes, and therefore do not suffer from the lack of realism of the corresponding components. On the flipside, the lack of ingrained geometric or photometric models in this approach means that generalizing to new poses and in particular to new camera views may be problematic. Still a lot of progress has been made over the last several years for the neural modeling of personalized talking head models [37, 43, 62], hair [68], hands [47]. Notably, the recent system [43] has achieved very impressive results for neural face rendering, while decomposing view-dependent texture and 3D shape modeling.

Over the last several months, several groups have presented results of neural modeling of full bodies [1, 12, 42, 67]. While the presented results are very impressive, the approaches still require a large amount of training data. They also assume that the test images are rendered with the same camera views as the training data, which in our experience makes the task considerably simpler than modeling body appearance from arbitrary viewpoint. In this work, we aim to expand the neural body modeling approach to tackle the latter, harder task. The work [45] uses a combination of classical and neural rendering to render human body from new viewpoints, but does so based on depth scans and therefore with a rather different algorithmic approach.

A number of recent works **warp a photo of a person** to a new photorealistic image with modified gaze direction [24], modified facial expression/pose [9, 55, 64, 71], or modified body pose [5, 48, 56, 64], whereas the warping field is estimated using a deep convolutional network (while the original photo effectively serves as a texture). These approaches are however limited in their realism and/or the amount of change they can model, due to their reliance on a single photo of a given person for its input. Our approach also disentangles texture from surface geometry/motion modeling but trains from videos, therefore being able to handle harder problem (full body multiview setting) and to achieve higher realism.

Our system relies on the **DensePose** body surface parameterization (UV parameterization) similar to the one used in the classical graphics-based representation. Part of our system performs a mapping from the body pose to the surface parameters (UV coordinates) of image pixels. This makes our approach related to the DensePose approach [28] and the earlier works [29, 63] that predict UV coordinates of image pixels from the input photograph. Furthermore, our approach uses DensePose results [28] for pretraining.

Our system is related to approaches that extract **textures from multi-view image collections** [26, 39] or multi-view video collections [66] or a single video [52]. Our approach is also related to free-viewpoint video compression and rendering systems, e.g. [11, 16, 21, 66]. Unlike those works, ours is restricted to scenes containing a single human. At the same time, our approach aims to generalize not only to new camera views but also to new user poses unseen in the training videos. The work of [73] is the most related to ours in this group, as they warp the individual frames of the multi-view video dataset according to the target pose to generate new sequences. The poses that they can handle, however, are limited by the need to have a close match in the training set, which is a strong limitation given the combinatorial nature of the human pose configuration space.

3. Methods

Notation. We use the lower index i to denote objects that are specific to the i-th training or test image. We use uppercase notation, e.g. B_i to denote a stack of maps (a third-order tensor/three-dimensional array) corresponding to the i-th training or test image. We use the upper index to denote a specific map (channel) in the stack, e.g. B_i^j. Furthermore, we use square brackets to denote elements corresponding to a specific image location, e.g. $B_i^j[x, y]$ denotes the scalar element in the j-th map of the stack B_i located at location (x, y), and $B_i[x, y]$ denotes the vector of elements corresponding to all maps sampled at location (x, y).

Input and output. In general, we are interested in synthesizing images of a certain person given her/his pose. We assume that the pose for the i-th image comes in the form of 3D joint positions defined in the camera coordinate frame. As an input to the network, we then consider a map stack B_i, where each map B_i^j contains the rasterized j-th segment (bone) of the "stickman" (skeleton) projected on the camera plane. To retain the information about the third coordinate of the joints, we linearly interpolate the depth-value between the joints defining the segments, and use the interpolated values to define the values in the map B_i^j corresponding to the bone pixels (the pixels not covered by the j-th bone are set to zero). Overall, the stack B_i incorporates the information about the person and the camera pose.

As an output of the whole system, we expect an RGB image (a three-channel stack) I_i and a single channel mask M_i, defining the pixels that are covered by the avatar. Below, we consider two approaches: the *direct translation* baseline, which directly maps B_i into $\{I_i, M_i\}$ and the *textured neural avatar* approach that performs such mapping indirectly using texture mapping.

In both cases, at training time, we assume that for each input frame i, the input joint locations and the "ground truth" foreground mask are estimated, and we use 3D body pose estimation and human semantic segmentation to extract them from raw video frames. At test time, given a real or synthetic background image $\tilde{I}_i$, we generate the final view by first predicting M_i and I_i from the body pose and then linearly blending the resulting avatar into an image: $\hat{I}_i = I_i \odot M_i + \tilde{I}_i \odot (1 - M_i)$ (where $\odot$ defines a "location-wise" product, i.e. the RGB values at each location are multiplied by the mask value at this location).

Direct translation baseline. The direct approach that we consider as a baseline to ours is to learn an image translation network that maps the map stack B_i^k to the map stacks I_i and M_i (usually the two output stacks are produced within two branches that share the initial stage of the processing [20]). Generally, mappings between stacks of maps can be implemented using fully-convolutional architectures. Exact architectures and losses for such networks is an active area of research [19, 14, 31, 33, 65]. Very recent works [1, 12, 42, 67] have used direct translation (with various modifications) to synthesize the view of a person for a fixed camera. We use the video-to-video variant of this approach [67] as a baseline for our method.

Textured neural avatar. The direct translation approach relies on the generalization ability of ConvNets and incorporates very little domain-specific knowledge into the system. As an alternative, we suggest the textured avatar approach, that explicitly estimates the textures of body parts, thus ensuring the similarity of the body surface appearance under varying pose and cameras.

Following the DensePose approach [28], we subdivide the body into n=24 parts, where each part has a 2D parameterization. Each body part also has the texture map T^k, which is a color image of a fixed pre-defined size (256×256 in our implementation). The training process for the textured neural avatar estimates personalized part parameterizations and textures.

Again, following the DensePose approach, we assume that each pixel in an image of a person is (soft)-assigned to one of n parts or to the background and with a specific location on the texture of that part (body part coordinates). Unlike DensePose, where part assignments and body part coordinates are induced from the image, our approach at test time aims to predict them based solely on the pose B_i.

The introduction of the body surface parameterization outlined above changes the translation problem. For a given pose defined by B_i, the translation network now has

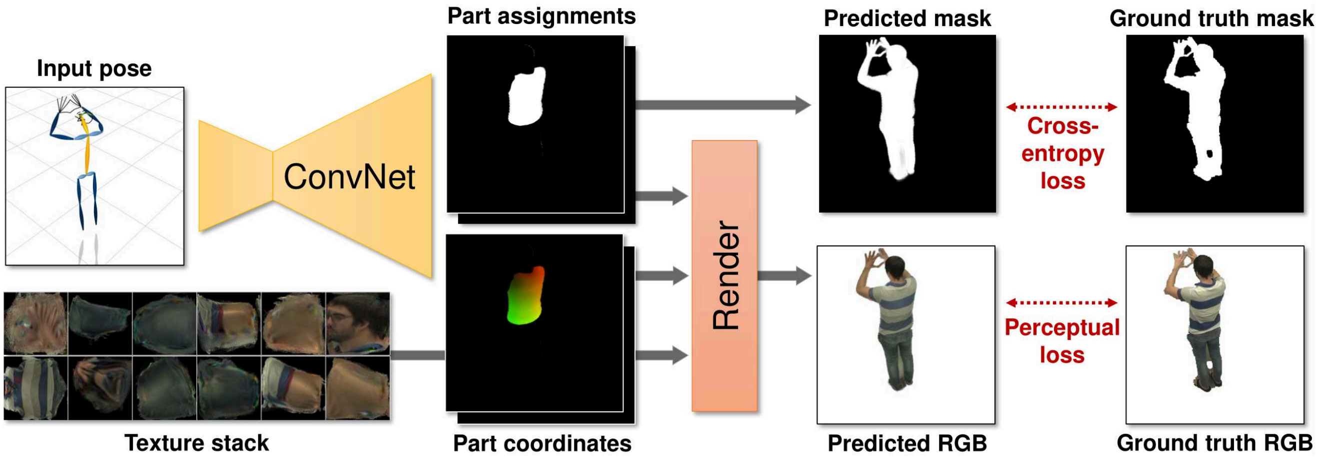

Figure 2: The overview of the textured neural avatar system. The input pose is defined as a stack of "bone" rasterizations (one bone per channel; here we show it highlighted in red). The input is processed by the fully-convolutional network (orange) to produce body part assignment map stack and the body part coordinate map stack. These stacks are then used to sample the body texture maps at the locations prescribed by the part coordinate stack with the weights prescribed by the part assignment stack to produce the RGB image. In addition, the last body assignment stack map corresponds to the background probability. During learning, the mask and the RGB image are compared with ground-truth and the resulting losses are backpropagated through the sampling operation into the fully-convolutional network and onto the texture, resulting in their updates.

to predict the stack P_i of body part assignments and the stack C_i of body part coordinates, where P_i contains $n{+}1$ maps of non-negative numbers that sum to identity (i.e. $\sum_{k=1}^{n} P_i^k[x,y] = 1$ for any position (x,y)), and C_i contains $2n$ maps of real numbers between 0 and w, where w is the spatial size (width and height) of the texture maps T^k.

The map channel P_i^k for $k = 0,\ldots,n{-}1$ is then interpreted as the probability of the pixel to belong to the k-th body part, and the map channel P_i^n corresponds to the probability of the background. The coordinate maps C_i^{2k} and C_i^{2k+1} correspond to the pixel coordinates on the k-th body part. Specifically, once the part assignments P_i and body part coordinates C_i are predicted, the image I_i at each pixel (x,y) is reconstructed as a weighted combination of texture elements, where the weights and texture coordinates are prescribed by the part assignment maps and the coordinate maps correspondingly:

$$s(P_i, C_i, T)[x,y] = \sum_{k=0}^{n-1} P_i^k[x,y] \cdot \\ T^k\left[C_i^{2k}[x,y], C_i^{2k+1}[x,y]\right] \ , \qquad (1)$$

where $s(\cdot,\cdot,\cdot)$ is the sampling function (layer) that outputs the RGB map stack given the three input arguments. In (1), the texture maps T^k are sampled at non-integer locations $(C_i^{2k}[x,y], C_i^{2k+1}[x,y])$ in a piecewise-differentiable manner using bilinear interpolation [32].

When training the neural textured avatar, we learn a convolutional network g_ϕ with learnable parameters ϕ to translate the input map stacks B_i into the body part assignments and the body part coordinates. As g_ϕ has two branches ("heads"), we denote with g_ϕ^P the branch that produces the body part assignments stack, and with g_ϕ^C the branch that produces the body part coordinates. To learn the parameters of the textured neural avatar, we optimize the loss between the generated image and the ground truth image $\bar{I}_i$:

$$\mathcal{L}_{\text{image}}(\phi, T) = d_{\text{Image}}\left(\bar{I}_i,\ s\left(g_\phi^P(B_i), g_\phi^C(B_i), T\right)\right) \qquad (2)$$

where $d(\cdot,\cdot)$ is a loss used to compare two images. In our current implementation we use a simple perceptual loss [25, 33, 65], which computes the maps of activations within pretrained fixed VGG network [58] for both images and evaluates the L1-norm between the resulting maps (12 first layers are used). More advanced adversarial losses [27] popular in image translation [19, 31] can also be used here.

During the stochastic optimization, the gradient of the loss (2) is backpropagated through (1) both into the translation network g_ϕ and onto the texture maps T^k, so that minimizing this loss updates not only the network parameters but also the textures themselves. As an addition, the learning also optimizes the mask loss that measures the discrepancy between the ground truth background mask $1-\bar{M}_i$ and the background mask prediction:

$$\mathcal{L}_{\text{mask}}(\phi, T) = d_{\text{BCE}}\left(\bar{1} - M_i,\ g_\phi^P(B_i)^n\right) \qquad (3)$$

where d_{BCE} is the binary cross-entropy loss, and $g_\phi^P(B_i)^n$ corresponds to the n-th (i.e. background) channel of the predicted part assignment map stack. After backpropagation

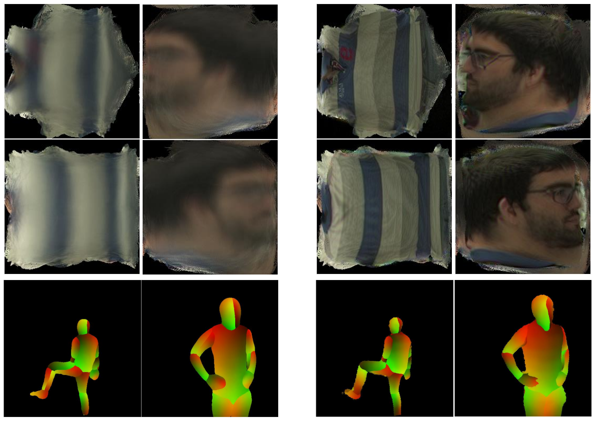

Figure 3: The impact of the learning on the texture (top, shown for the same subset of maps T^k) and on the convolutional network g_ϕ^C predictions (bottom, shown for the same pair of input poses). Left part shows the starting state (after initialization), while the right part shows the final state, which is considerably different from the start.

of the weighted combination of (2) and (3), the network parameters ϕ and the textures maps T^k are updated. As the training progresses, the texture maps change (Figure 2), and so does the body part coordinate predictions, so that the learning is free to choose the appropriate parameterization of body part surfaces.

Initialization of textured neural avatar. The success of our network depends on the initialization strategy. When training from multiple video sequences, we use the DensePose system [28] to initialize the textured neural avatar. Specifically, we run DensePose on the training data and pretrain g_ϕ as a translation network between the pose stacks B_i and the DensePose outputs.

An alternative way that is particularly attractive when training data is scarce is to initialize the avatar is through transfer learning. In this case, we simply take g_ϕ from another avatar trained on abundant data. The explicit decoupling of geometry from appearance in our method facilitates transfer learning, as the geometrical mapping provided by the network g_ϕ usually does not need to change much between two people, especially if the body types are not too dissimilar.

Once the mapping g_ϕ has been initialized, the texture maps T^k are initialized as follows. Each pixel in the training image is assigned to a single body part (according to the prediction of the pretrained g_ϕ^P) and to a particular texture pixel on the texture of the corresponding part (according to the prediction of the pretrained g_ϕ^C). Then, the value of each texture pixel is initialized to the mean of all image pixels assigned to it (the texture pixels assigned zero pixels are initialized to black). The initialized texture T and g_ϕ usually produce images that are only coarsely reminding the person, and they change significantly during the end-to-end learning (Figure 3).

4. Experiments

Below, we discuss the details of the experimental validation, provide comparison with baseline approaches, and show qualitative results. The project webpage[1] also contains the video of the learned avatars.

Architecture. We input 3D pose via bone rasterizations, where each bone, hand and face are drawn in separate channels. We then use standard image translation architecture [33] to perform a mapping from these bones' rasterizations to texture assignments and coordinates. This architecture consists of downsampling layers, stack of residual blocks, operating at low dimensional feature representations, and upsampling layers. We then split the network into two roughly equal parts: encoder and decoder, with texture assignments and coordinates having separate decoders. We use 4 downsampling and upsampling layers with initial 32 channels in the convolutions and 256 channels in the residual blocks. The ConvNet g_ϕ has 17 million parameters.

Datasets. We train neural avatars on two types of datasets. First, we consider collections of multiview videos registered in time and space, where 3D pose estimates can be obtained via triangulation of 2D poses. We use two subsets (corresponding to two persons from the 171026_pose2 scene) from the CMU Panoptic dataset collection [34], referring to them as `CMU1` and `CMU2` (both subsets have approximately four minutes / 7,200 frames in each camera view). We consider two regimes: training on 16 cameras (`CMU1-16` and `CMU2-16`) or six cameras (`CMU1-6` and `CMU2-6`). The evaluation is done on the hold-out cameras and hold-out parts of the sequence (no overlap between train and test in terms of the cameras or body motion).

We have also captured our own multiview sequences of three subjects using a rig of seven cameras, spanning approximately 30°. In one scenario, the training sets included six out of seven cameras, where the duration of each video was approximately six minutes (11,000 frames). We show qualitative results for the hold-out camera as well as from new viewpoints. In the other scenario described below, training was done based on a video from a single camera.

Finally, we evaluate on two short monocular sequences from [4] and a Youtube video in Figure 7.

Pre-processing. Our system expects 3D human pose as input. For non-CMU datasets, we used the OpenPose-

[1] https://saic-violet.github.io/texturedavatar/

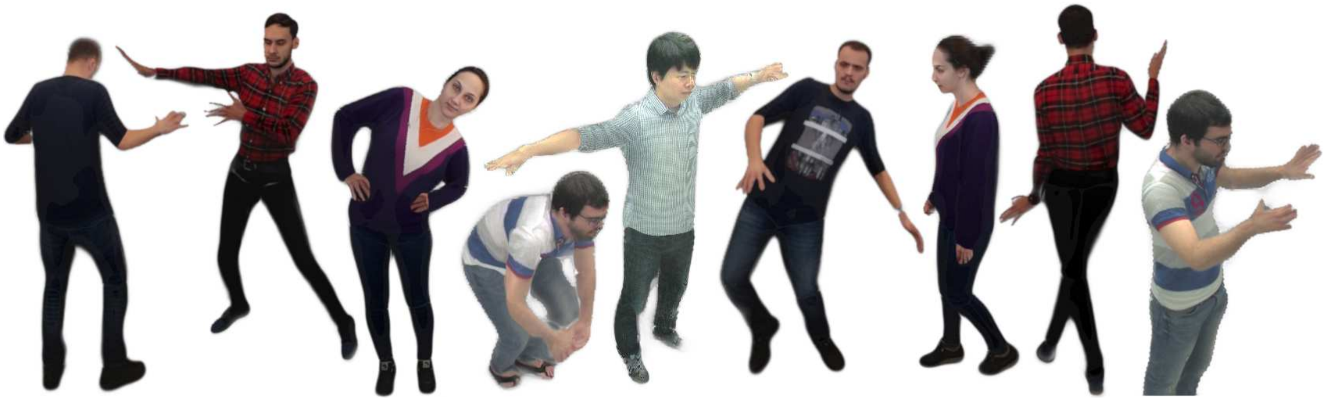

Figure 4: Renderings produced by multiple textured neural avatars (for all people in our study). All renderings are produced from the new viewpoints unseen during training.

	(a) User study		(b) SSIM score			(c) Frechet distance		
	Ours-v-V2V	Ours-v-Direct	V2V	Direct	Ours	V2V	Direct	Ours
CMU1-16	0.56	0.75	0.908	0.899	0.919	6.7	7.3	8.8
CMU2-16	0.54	0.74	0.916	0.907	0.922	7.0	8.8	10.7
CMU1-6	0.50	0.92	0.905	0.896	0.914	7.7	10.7	8.9
CMU2-6	0.53	0.71	0.918	0.907	0.920	7.0	9.7	10.4

Table 1: Quantitative comparison of the three models operating on different datasets (see text for discussion).

compatible [10, 57] 3D pose formats, represented by 25 body joints, 21 joints for each hand and 70 facial landmarks. For the CMU Panoptic datasets, we use the available 3D pose annotation as input (which has 19 rather than 25 body joints). To get a 3D pose for non-CMU sequences we first apply the OpenPose 2D pose estimation engine to five consecutive frames of the monocuular RGB image sequence. Then we concatenate and lift the estimated 2D poses to infer the 3D pose of the last frame by using a multi-layer perceptron model. The perceptron is trained on the CMU 3D pose annotations (augmented with position of the feet joints by triangulating the output of OpenPose) in orthogonal projection.

For foreground segmentation we use DeepLabv3+ with Xception-65 backbone [13] initially trained on PASCAL VOC 2012 [22] and fine-tuned on HumanParsing dataset [40, 41] to predict initial human body segmentation masks. We additionally employ GrabCut [54] with background/foreground model initialized by the masks to refine object boundaries on the high resolution images. Pixels covered by the skeleton rasterization were always added to the foreground mask.

Baselines. We consider two other systems, against which ours is compared. First, we use the video-to-video (*V2V*) system [67], using the authors code with minimal modifications that lead to improved performance. We provide it with the same input as ours, and we use images with blacked-out background (according to our segmentation) as desired output. On the CMU1-6 task, we have also evaluated a model with DensePose results computed on the target frame given as input (alongside keypoins). Despite much stronger (oracle-type) conditioning, the performance of this model in terms of considered metrics has not improved in comparison with V2V that uses body joints as input only.

The video-to-video system employs several adversarial losses and an architecture different from ours. Therefore we consider a more direct ablation (*Direct*), which has the same network architecture that predicts RGB color and mask directly, rather than via body part assignments/coordinates. The Direct system is trained using the same losses and in the same protocol as ours.

Multi-video comparison. We compare the three system (*ours*, *V2V*, *Direct*) in `CMU1-16`, `CMU2-16`, `CMU1-6`, `CMU2-6`. Using the hold-out sequences/motions, we then evaluated two popular metrics, namely structured self-similarity (SSIM) and Frechet Inception Distance (FID) between the results of each system and the hold-out frames (with background removed using our segmentation algorithm). Our method outperforms the other two in terms of SSIM and underperforms V2V in terms of FID. Representative examples are shown in Figure 5.

We have also performed user study using a crowdsourc-

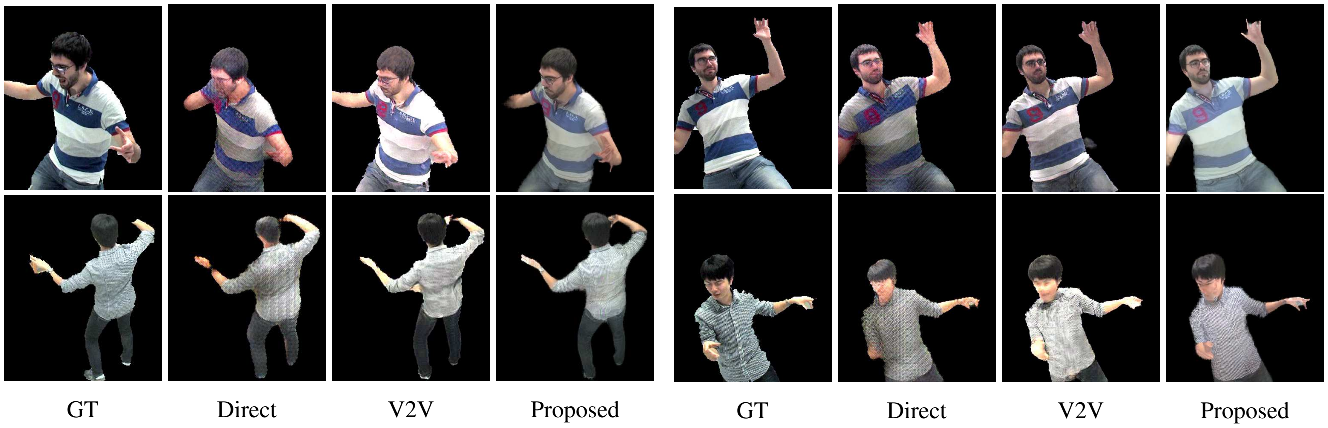

Figure 5: Comparison of the rendering quality for the Vid2vid, Direct and proposed methods on the CMU1-6 and CMU2-6 sequences. Images from six arbitrarily chosen cameras were used for training. We generate the views onto the hold-out cameras which were not used during training. The pose and camera in the lower right corner are in particular difficult for all the systems.

ing website, where the users were shown the results of ours and one of the other two systems on either side of the ground truth image, and were asked to pick a better match to the middle image. In the side-by-side comparison, the results of our method were always preferred by the majority of the crowd-sourcing users. We note that our method suffers from a disadvantage both in the quantitative metrics and in the user comparison, since it averages out lighting from different viewpoints. The more detailed quantitative comparison is presented in Table 1.

We show more qualitative examples of our method for a variety of models in Figure 4 and some qualitative comparisons with baselines in Figure 6.

Single video comparisons. We also evaluate our system in the single video case. We consider the scenario, where we train the model and transfer it to a new person by fitting it to a single video. We use single camera videos from one of the cameras in our rig. We then evaluate the model (and the baselines) on a hold-out set of poses projected onto the camera from the other side of the rig (around 30° away). We thus demonstrate that new models can be obtained using single monocular videos. For our models, we consider transferring from CMU1-16.

We thus pretrain V2V and our system on CMU1-16 and use the obtained weights of g_ϕ as initialization for fine-tuning to the single video in our dataset. The texture maps are initialized from scratch as described above. Evaluating on hold-out camera and motion highlighted strong advantage of our method. In the user study on two subjects, the result of our method has been preferred to V2V in 55% and 65% of the cases. We further compare our method and the system of [4] on the sequences from [4]. The qualitative comparison is shown in Figure 7. In addition, we generate an avatar from a youtube video. In this set of experiments, the avatars were obtained by fine-tuning from the same avatar (shown in Figure 6–left). Except for the considerable artefacts on hand parts, our system has generated avatars that can generalize to new pose despite very short video input (300 frames in the case of [4]).

5. Summary and Discussion

We have presented textured neural avatar approach to model the appearance of humans for new camera views and new body poses. Our system takes the middle path between the recent generation of methods that use ConvNets to map the pose to the image directly, and the traditional approach that uses geometric modeling of the surface and superimpose the personalized texture maps. This is achieved by learning a ConvNet that predicts texture coordinates of pixels in the new view jointly with the texture within the end-to-end learning process. We demonstrate that retaining an explicit shape and texture separation helps to achieve better generalization than direct mapping approaches.

Our method suffers from certain limitations. The generalization ability is still limited, as it does not generalize well when a person is rendered at a scale that is considerably different from the training set (which can be partially addressed by rescaling prior to rendering followed by cropping/padding postprocessing). Furthermore, textured avatars exhibit strong artefacts in the presence of pose estimation errors on hands and faces. Finally, our method assumes constancy of the surface color and ignores lighting effects. This can be potentially addressed by making our textures view- and lighting-dependent [17, 43].

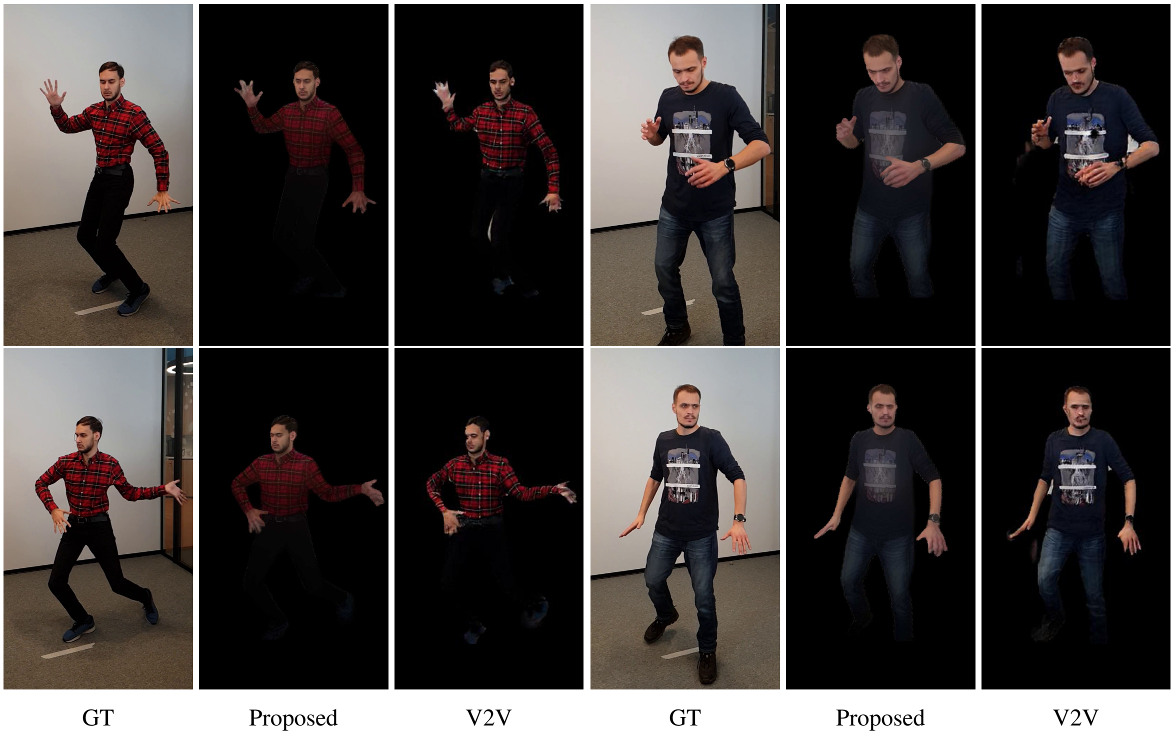

Figure 6: Results comparison for our multi-view sequences using a hold-out camera. Textured Neural Avatars and the images produced by the video-to-video (V2V) system correspond to the same viewpoint. Both systems use a video from a single viewpoint for training. *Electronic zoom-in recommended.*

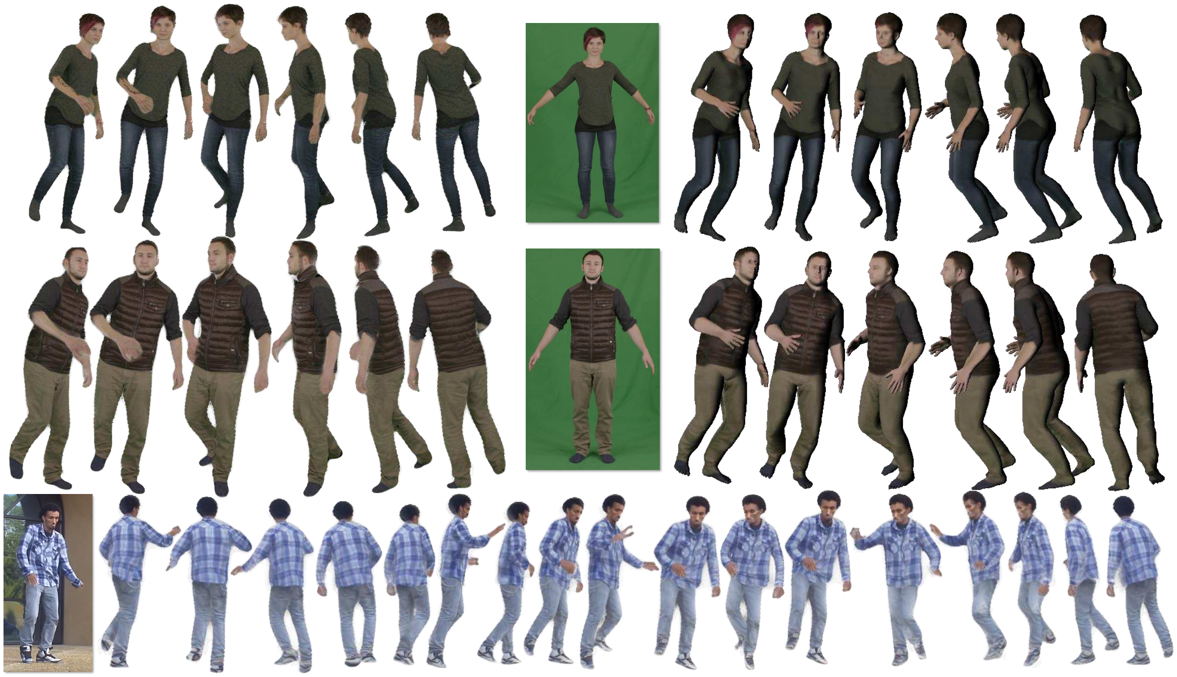

Figure 7: Results on external monocular sequences. Rows 1-2: avatars for sequences from [4] in an unseen pose (left – ours, right – [4]). Row 3 – the textured avatar computed from a popular Youtube video ('PUMPED UP KICKS DUBSTEP'). In general, our system is capable of learning avatars from monocular videos.

References

[1] Kfir Aberman, Mingyi Shi, Jing Liao, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Deep video-based performance cloning. *arXiv preprint arXiv:1808.06847*, 2018. 1, 2, 3

[2] Oleg Alexander, Mike Rogers, William Lambeth, Jen-Yuan Chiang, Wan-Chun Ma, Chuan-Chang Wang, and Paul Debevec. The Digital Emily project: Achieving a photorealistic digital actor. *IEEE Computer Graphics and Applications*, 30(4):20–31, 2010. 2

[3] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *2018 International Conference on 3D Vision (3DV)*, pages 98–109. IEEE, 2018. 2

[4] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *Proc. CVPR*, June 2018. 2, 5, 7, 8

[5] Guha Balakrishnan, Amy Zhao, Adrian V. Dalca, Frédo Durand, and John V. Guttag. Synthesizing images of humans in unseen poses. In *Proc. CVPR*, pages 8340–8348, 2018. 2

[6] Alexandru O Bălan and Michael J Black. The naked truth: Estimating body shape under clothing. In *Proc. ECCV*, pages 15–29. Springer, 2008. 2

[7] Federica Bogo, Michael J Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *Proc. ICCV*, pages 2300–2308, 2015. 2

[8] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Proc. ECCV*, pages 561–578. Springer, 2016. 2

[9] Jie Cao, Yibo Hu, Hongwen Zhang, Ran He, and Zhenan Sun. Learning a high fidelity pose invariant model for high-resolution face frontalization. *arXiv preprint arXiv:1806.08472*, 2018. 2

[10] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. CVPR*, 2017. 6

[11] Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 4d video textures for interactive character appearance. In *Computer Graphics Forum*, volume 33, pages 371–380. Wiley Online Library, 2014. 3

[12] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. *arXiv preprint arXiv:1808.07371*, 2018. 1, 2, 3

[13] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. ECCV*, 2018. 6

[14] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proc. ICCV*, pages 1520–1529, 2017. 3

[15] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. CVPR*, June 2018. 2

[16] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4):69, 2015. 3

[17] Paul E. Debevec, Yizhou Yu, and George Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Rendering Techniques '98, Proceedings of the Eurographics Workshop in Vienna, Austria, June 29 - July 1, 1998*, pages 105–116, 1998. 7

[18] Craig Donner, Tim Weyrich, Eugene d'Eon, Ravi Ramamoorthi, and Szymon Rusinkiewicz. A layered, heterogeneous reflectance model for acquiring and rendering human skin. In *ACM Transactions on Graphics (TOG)*, volume 27, page 140. ACM, 2008. 2

[19] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proc. NIPS*, pages 658–666, 2016. 3, 4

[20] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proc. CVPR*, pages 1538–1546, 2015. 2, 3

[21] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: real-time volumetric performance capture. *ACM Transactions on Graphics (TOG)*, 36(6):246, 2017. 3

[22] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 6

[23] Andrew Feng, Dan Casas, and Ari Shapiro. Avatar reshaping and automatic rigging using a deformable model. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 57–64. ACM, 2015. 2

[24] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *Proc. ECCV*, pages 311–326. Springer, 2016. 1, 2

[25] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pages 2414–2423, 2016. 4

[26] Bastian Goldlücke and Daniel Cremers. Superresolution texture maps for multiview reconstruction. In *Proc. ICCV*, pages 1677–1684, 2009. 3

[27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NIPS*, pages 2672–2680, 2014. 2, 4

[28] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense human pose estimation in the wild. In *Proc. CVPR*, June 2018. 3, 5

[29] Riza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. DenseReg: Fully convolutional dense shape regression in-the-wild. In *Proc. CVPR*, volume 2, page 5, 2017. 3

[30] Nils Hasler, Hanno Ackermann, Bodo Rosenhahn, Thorsten Thormählen, and Hans-Peter Seidel. Multilinear pose and

body shape estimation of dressed subjects from image sets. In *Proc. CVPR*, pages 1823–1830. IEEE, 2010. 2

[31] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, pages 5967–5976, 2017. 3, 4

[32] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proc. NIPS*, pages 2017–2025, 2015. 4

[33] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, pages 694–711, 2016. 3, 4, 5

[34] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5

[35] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proc. CVPR*, 2018. 2

[36] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 2

[37] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *arXiv preprint arXiv:1805.11714*, 2018. 1, 2

[38] Oliver Klehm, Fabrice Rousselle, Marios Papas, Derek Bradley, Christophe Hery, Bernd Bickel, Wojciech Jarosz, and Thabo Beeler. Recent advances in facial appearance capture. In *Computer Graphics Forum*, volume 34, pages 709–733. Wiley Online Library, 2015. 2

[39] Victor S. Lempitsky and Denis V. Ivanov. Seamless mosaicing of image-based texture maps. In *Proc. CVPR*, 2007. 3

[40] Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. Deep human parsing with active template regression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(12):2402–2414, Dec 2015. 6

[41] Xiaodan Liang, Chunyan Xu, Xiaohui Shen, Jianchao Yang, Si Liu, Jinhui Tang, Liang Lin, and Shuicheng Yan. Iccv. 2015. 6

[42] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural animation and reenactment of human actor videos. *arXiv preprint arXiv:1809.03658*, 2018. 1, 2, 3

[43] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (TOG)*, 37(4):68, 2018. 1, 2, 7

[44] 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):248, 2015. 2

[45] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien P. C. Valentin, Sameh Khamis, Philip L. Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B. Goldman, Cem Keskin, Steven M. Seitz, Shahram Izadi, and Sean Ryan Fanello. *LookinGood*: enhancing performance capture with real-time neural re-rendering. *ACM Trans. Graph.*, 37(6):255:1–255:14, 2018. 2

[46] Masahiro Mori. The uncanny valley. *Energy*, 7(4):33–35, 1970. 2

[47] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. GANerated hands for real-time 3d hand tracking from monocular RGB. In *Proc. CVPR*, June 2018. 1, 2

[48] Natalia Neverova, Riza Alp Güler, and Iasonas Kokkinos. Dense pose transfer. In *Proc. ECCV*, September 2018. 2

[49] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. Verona, Italy, 2018. 2

[50] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proc. CVPR*, June 2018. 2

[51] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)*, 34(4):120, 2015. 2

[52] Alex Rav-Acha, Pushmeet Kohli, Carsten Rother, and Andrew W. Fitzgibbon. Unwrap mosaics: a new representation for video editing. *ACM Trans. Graph.*, 27(3):17:1–17:11, 2008. 3

[53] Nadia Robertini, Dan Casas, Edilson De Aguiar, and Christian Theobalt. Multi-view performance capture of surface details. *International Journal of Computer Vision*, 124(1):96–113, 2017. 2

[54] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. ”grabcut”: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004. 6

[55] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proc. ECCV*, September 2018. 2

[56] Aliaksandr Siarohin, Enver Sangineto, Stphane Lathuilire, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proc. CVPR*, June 2018. 2

[57] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 6

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4

[59] J Starck and A Hilton. Model-based multiple view reconstruction of people. In *Proc. ICCV*, pages 915–922, 2003. 2

[60] Ian Stavness, C Antonio Sánchez, John Lloyd, Andrew Ho, Johnty Wang, Sidney Fels, and Danny Huang. Unified skin-

ning of rigid and deformable models for anatomical simulations. In *SIGGRAPH Asia 2014 Technical Briefs*, page 9. ACM, 2014. 2

[61] Diana Sungatullina, Egor Zakharov, Dmitry Ulyanov, and Victor Lempitsky. Image manipulation with perceptual discriminators. In *Proc. ECCV*, September 2018. 2

[62] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing Obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017. 1, 2

[63] Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proc. CVPR*, pages 103–110. IEEE, 2012. 3

[64] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proc. CVPR*, June 2018. 2

[65] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proc. ICML*, pages 1349–1357, 2016. 3, 4

[66] Marco Volino, Dan Casas, John P Collomosse, and Adrian Hilton. Optimal representation of multi-view video. In *Proc. BMVC*, 2014. 3

[67] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018. 1, 2, 3, 6

[68] Lingyu Wei, Liwen Hu, Vladimir Kim, Ersin Yumer, and Hao Li. Real-time hair rendering using sequential adversarial networks. In *Proc. ECCV*, September 2018. 2

[69] Alexander Weiss, David Hirshberg, and Michael J Black. Home 3d body scans from noisy image and range data. In *Proc. ICCV*, pages 1951–1958. IEEE, 2011. 2

[70] Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, et al. Analysis of human faces using a measurement-based skin reflectance model. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1013–1024. ACM, 2006. 2

[71] Olivia Wiles, A. Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proc. ECCV*, September 2018. 2

[72] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proc. ICCV*, pages 3756–3764, 2015. 2

[73] 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 Transactions on Graphics (TOG)*, 30(4):32, 2011. 3

[74] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *Proc. CVPR*, pages 7287–7296. IEEE Computer Society, 2018. 2