# Neural Human Video Rendering by Learning Dynamic Textures and Rendering-to-Video Translation

Lingjie Liu, Weipeng Xu, Marc Habermann, Michael Zollhöfer, Florian Bernard, Hyeongwoo Kim, Wenping Wang, Christian Theobalt



**Abstract**—Synthesizing realistic videos of humans using neural networks has been a popular alternative to the conventional graphics-based rendering pipeline due to its high efficiency. Existing works typically formulate this as an image-to-image translation problem in 2D screen space, which leads to artifacts such as over-smoothing, missing body parts, and temporal instability of fine-scale detail, such as pose-dependent wrinkles in the clothing. In this paper, we propose a novel human video synthesis method that approaches these limiting factors by explicitly disentangling the learning of time-coherent fine-scale details from the embedding of the human in 2D screen space. More specifically, our method relies on the combination of two convolutional neural networks (CNNs). Given the pose information, the first CNN predicts a dynamic texture map that contains time-coherent high-frequency details, and the second CNN conditions the generation of the final video on the temporally coherent output of the first CNN. We demonstrate several applications of our approach, such as human reenactment and novel view synthesis from monocular video, where we show significant improvement over the state of the art both qualitatively and quantitatively.

**Index Terms**—Video-based Characters, Deep Learning, Neural Rendering, Learning Dynamic Texture, Rendering-to-Video Translation

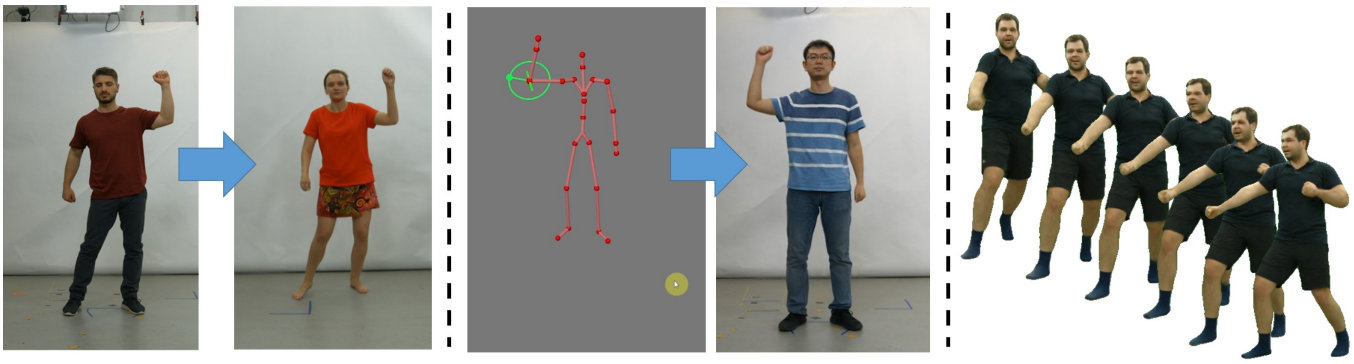
## 1 Introduction

Synthesizing realistic videos of humans is an important research topic in computer graphics and computer vision, which has a broad range of applications in visual effects (VFX) and games, virtual reality (VR) and telepresence, AI assistants, and many more. In this work, we propose a novel machine learning approach for synthesizing a realistic video of an actor that is driven from a given motion sequence. Only a monocular video and a personalized template mesh of the actor are needed as input. The motion of the actor in the target video can be controlled in different ways. For example by transferring the motion of a different actor in a source video, or by controlling the video footage directly based on an interactive handle-based editor.

Nowadays, the de-facto standard for creating video-realistic animations of humans follows the conventional graphics-based human video synthesis pipeline based on highly detailed animated 3D models. The creation of these involves multiple non-trivial, decoupled, manual and time-consuming steps: These include 3D shape and appearance scanning or design, hand-design or motion capture of target motions and deformations, and time-consuming photorealistic rendering. Aiming to streamline and expedite this process, in recent years graphics and vision researchers developed data-driven methods to generate realistic images [1], [2], [3] and videos [4], [5], [6], [7], [8] of humans. Many of these use variants of adversarially trained convolutional neural networks to translate coarse conditioning inputs, which encode human appearance and/or pose, into photo-realistic imagery.

A prominent problem with existing methods is that fine-scale details are often over-smoothed and temporally incoherent, e.g. wrinkles often do not move coherently with the garments but look like lying on a separated spatially fixed layer floating in the screen space (see the supplementary video). While some approaches try to address these challenges by enforcing temporal coherence in the adversarial training objective [4], [5], [6], we argue that most problems are due to a combination of two limiting factors: 1) Conditioning input is often a very coarse and sparse 2D or 3D skeleton pose rather than a more complete 3D human animation model. 2) Image translation is learned only in 2D screen space. This fails to properly disentangle appearance effects from residual image-space effects that are best handled by 2D image convolutions. Since appearance effects are best described on the actual 3D body surface, they should be handled by suitable convolutions that take the manifold structure into account. As a consequence of these effects, networks struggle to jointly generate results that show both, complete human body imagery without missing body parts or silhouette errors, as well as plausible temporally coherent high-frequency surface detail.


- *This work was done when L. Liu was an intern at Max Planck Institute for Informatics.*
- *L. Liu and W. Wang are with the Department of Computer Science, The University of Hong Kong, Hong Kong, P.R.China. E-mail: liulingjie0206@gmail.com, wenping@cs.hku.hk*
- *W. Xu, M. Habermann, F. Bernard, H. Kim, C. Theobalt are with the Graphics, Vision and Video Group at Max Planck Institute for Informatics, 66123 Saarbrcken, Germany. E-mail: wxu@mpi-inf.mpg.de, mhaberma@mpi-inf.mpg.de, fbernard@mpi-inf.mpg.de, hyeongwoo@mpi-inf.mpg.de, theobalt@mpi-inf.mpg.de*
- *M. Zollhöfer is with the Department of Computer Science, Computer Graphics Laboratory at Stanford University, CA 94305, United States. E-mail: zollhoefer@cs.stanford.edu*



(a) Motion transfer (b) Interactive editing (c) Monocular bullet time effect

Fig. 1. We present an approach for synthesizing realistic videos of humans. Our method allows for: a) motion transfer between a pair of monocular videos, b) interactively controlling the pose of a person in the video, and c) monocular bullet time effects, where we freeze time and virtually rotate the camera.

We propose a new human video synthesis method that tackles these limiting factors and explicitly disentangles learning of time-coherent pose-dependent fine-scale detail from the time-coherent pose-dependent embedding of the human in 2D screen space. Our approach relies on a monocular training video of the actor performing various motions, and a skinned person-specific template mesh of the actor. The latter is used to capture the shape and pose of the actor in each frame of the training video using an off-the-shelf monocular performance capture approach. Our video synthesis algorithm uses a three-stage approach based on two CNNs and the computer graphics texturing pipeline: 1) Given the target pose in each video frame encoded as a surface normal map of the posed body template, the first CNN is trained to predict a dynamic texture map that contains the pose-dependent and time-coherent high-frequency detail. In this normalized texture space, local details such as wrinkles always appear at the same uv-location, since the rigid and articulated body motion is already factored out by the monocular performance capture algorithm, which significantly simplifies the learning task. This frees the network from the task of having to synthesize the body at the right screen space location, leading to temporally more coherent and detailed results. 2) We apply the dynamic texture on top of the animated human body model to render a video of the animation that exhibits temporally stable high-frequency surface details, but that lacks effects that cannot be explained by the rendered mesh alone. 3) Finally, our second CNN conditions the generation of the final video on the temporally coherent output of the first CNN. This refinement network synthesizes foreground-background interactions, such as shadows, naturally blends the foreground and background, and corrects geometrical errors due to tracking/skinning errors, which might be especially visible at the silhouettes.

To the best of our knowledge, our approach is the first dynamic-texture neural rendering approach for human bodies that disentangles human video synthesis into explicit texture-space and image-space neural rendering steps: pose-dependent neural texture generation and rendering to realistic video translation. This new problem formulation yields more accurate human video synthesis results, which better preserve the spatial, temporal, and geometric coherence of the actor's appearance compared to existing state-of-the-art methods.

As shown in Figure 1, our approach can be utilized in various applications, such as human motion transfer, interactive reenactment and novel view synthesis from monocular video. In our experiments, we demonstrate these applications and show that our approach is superior to the state of the art both qualitatively and quantitatively.

Our main contributions are summarized as follows:

- A novel three-stage approach that disentangles learning pose-dependent fine-scale details from the pose-dependent embedding of the human in 2D screen space.
- High-resolution video synthesis of humans with controllable target motions and temporally coherent fine-scale detail.

## 2 Related Work

In the following, we discuss human performance capture, classical video-based rendering, and learning-based human performance cloning, as well as the underlying image-to-image translation approaches based on conditional generative adversarial networks.

**Classical Video-based Characters.** Classically, the domain gap between coarse human proxy models and realistic imagery can be bridged using image-based rendering techniques. These strategies can be used for the generation of video-based characters [9], [10], [11], [12] and enable free-viewpoint video [13], [14], [15], [16], [17]. Even relightable performances can be obtained [18] by disentangling illumination and scene reflectance. The synthesis of new body motions and viewpoints around an actor is possible [9] with such techniques.

**Modeling Humans from Data.** Humans can be modeled from data using mesh-based 3D representations. For example, parametric models for different body parts are widely employed [19], [20], [21], [22], [23], [24] in the literature. Deep Appearance Models [25] learn dynamic view-dependent texture maps for the human head. The paGAN [26] approach builds a dynamic avatar from a single monocular image. Recently, models of the entire human body have become popular [27], [28]. There are also some recent

works on cloth modeling [29], [30], [31]. One drawback of these models is that they do not model the appearance of dressed humans, i.e., the color of different garments. To tackle this problem, generative models based on neural networks have been applied to directly synthesize 2D images of humans without having to model the 3D content. First, these approaches have been applied to individual parts of the human body [32], [33], [34]. Also models that capture the appearance of clothing have been proposed [35]. Nowadays, similar techniques are applied for the complete human body, i.e., for the synthesis of different poses [1], [2], [3], [7]. In contrast to previous approaches, we employ dense conditioning and learn dynamic high-frequency details in texture space to enable the temporally coherent generation of video.

**Deep Video-based Performance Cloning.** Very recently, multiple approaches for video-based human performance cloning have been proposed [4], [5], [6], [8], [36], [37], [38] that output realistic video sequences. These approaches learn complex image-to-image mappings, i.e., from renderings of a skeleton [4], [36], [37], [38], dense mesh [6], [8], or joint position heatmaps [5], to real images. Liu et al. [8] proposed to translate simple synthetic computer graphics renderings of a human character into realistic imagery. *Everybody Dance Now* [4] predicts two consecutive video frames and employs a space-time discriminator to obtain temporally more coherent synthesis results. Deep performance cloning [5] combines paired and unpaired training based on a two-branch network for better generalization. The vid2vid [6] approach learns high-resolution video-to-video translation based on a sequential RNN generator and uses optical flow for explicitly forward warping the last frame estimate. All theses approaches learn an image-to-image mapping in 2D screen space based on a set of 2D convolution and deconvolution kernels. We argue that many artifacts of these approaches, e.g., the synthesized images are over-smoothed and temporally incoherent in fine-scale detail, are due to two limiting factors: 1) Only sparse 2D or 3D skeleton conditioning and 2) learning image translation in 2D screen space. In contrast to existing methods, we tackle these limiting factors and explicitly disentangle learning of time-coherent pose-dependent detail in texture space from the pose-dependent embedding of the human in 2D screen space.

**Surface-based Modeling with Deep Learning.** Several previous works have integrated neural synthesis into surface-based modeling [39], [40], [41], [42], [43], [44]. Deferred Neural Rendering [42] proposed an end-to-end training strategy to learn neural textures and deferred neural rendering jointly. They produced photo-realistic renderings for static scenes and faces with imperfect 3D reconstructed geometry. Some works also focus on neural synthesis for human bodies. For example, Densepose [41] predicts UV coordinates of image pixels from the RGB inputs, and the works [40], [43], [44] synthesize a new image of a person in a given pose based on a single image of that person. This is done by estimating dense 3D appearance flow to guide the transfer of pixels between poses. Textured Neural Avatars [40] learns full body neural avatars with static textures based on pretrained Densepose [41] results. In contrast, our work aims at generating dynamic textures for photo-realistic renderings of human bodies, which is a more challenging task.

**3D Performance Capture of Humans.** Monocular data based on recent performance capture techniques can provide the paired training corpora required for learning video-based performance cloning. Historically, 3D human performance capture has been based on complex capture setups, such as multi-view reconstruction studios with a large number of cameras [45], [46], [47], [48], [49]. The highest quality approaches combine active and passive depth sensing [17], [50], [51], [52]. Recent dense tracking approaches build on top of joint detections, either in 2D [53], [54], in 3D [55], [56], [57], or a combination thereof [58], [59], [60]. The set of sparse detections provides initialization for optimization-based tracking approaches to start near the optimum to facilitate convergence. Many approaches simplify performance capture by tracking only the degrees of freedom of a low-dimensional skeleton [61], [62], [63], thus resolving some of the ambiguities of truly dense capture. There is also a trend of using a reduced number of cameras, aiming to bring human performance capture to a commodity setting. For example, some approaches enable capturing human performances from two [64] or a sparse set of cameras [65]. Recently, even lighter approaches [66], [67], [68], [69], [70], [71], [72] have been developed to deal with the rising demand for human performance capture in commodity settings, e.g., to enable virtual and augmented reality applications. Monocular dense 3D human performance capture [73] is still a popular research problem, with recently real-time performance being demonstrated for the first time [74].

**Conditional Generative Adversarial Networks.** Generative adversarial networks (GANs) [75], [76], [77], [78] have been very successful in learning to generate arbitrary imagery using a generator network based on convolutional neural networks with an encoder-decoder structure [79]. They either start from scratch using a random vector [75], [76], or they learn conditional image-to-image synthesis based on an input image from a different domain [77], [78]. U-Nets [80] with skip connections are often employed as generator networks. The discriminator network is trained based on a binary classification problem [75] or is patch-based [78]. The generator and the discriminator are jointly trained based on a minimax optimization problem. Very recently, high-resolution images have been generated using GANs [81], [82] with a progressive training strategy and using cascaded refinement networks [83]. While most of these techniques are trained in a fully supervised manner based on paired training data, some approaches tackle the harder problem of learning the translation between two domains based on unpaired data [84], [85], [86], [87]. Some recent works studied the problem of video-to-video synthesis. Vid2vid [6] learns high-resolution video-to-video translation based on a sequential RNN generator and uses optical flow for explicitly forward warping the last frame estimate. The recently proposed Recycle-GAN [88] approach enables unpaired learning of a coherent video-to-video mapping.

In our work, we employ two vid2vid networks, where the first network has the task of generating a time-coherent texture with high-frequency details (e.g. in clothing), and the second network has the task of producing the final output image by refining a rendering of a mesh that is textured with

the output of the first network.

## 3 METHOD

In this section we describe our neural human video synthesis approach. As illustrated in Fig. 2, given a monocular video of a performing actor and a textured mesh template of the actor, our method learns a person-specific embedding of the actor's appearance. To generate the training data, we first employ an off-the-shelf monocular human performance capture method [74] to track the motion of the actor in the video (Sec. 3.1). Based on the tracking results, we generate the (partial) dynamic texture by back-projecting the video frames to the animated template mesh. Having the motion data, partial dynamic textures, and the original video frames as the training corpus, our approach proceeds in three stages: In the first stage, we train our *texture synthesis network (TexNet)* to regress a partial texture image, which depicts the *pose-dependent* appearance details, such as wrinkles, given a certain pose as input. Here, the pose information is encoded in a (partial) normal map in the uv-space in order to obtain an *image-based pose encoding in texture space*. In the second stage, we complete the predicted *partial* texture image to a *complete* texture image (Sec. 3.2), and render the mesh with this *complete* texture image. In the third stage, we translate the renderings into a realistic video with our *refinement network (RefNet)* (Sec. 3.3). During testing, our method takes a motion clip from arbitrary sources (e.g., motion capture, artist-designed, etc.), and generates a video of the actor performing the input motion.

### 3.1 Training Data Generation

In this section we describe the human character model, how its texture mapping is obtained, and how the human motion is captured.

**Image Sequence.** Let $\mathcal{I}_1, \dots, \mathcal{I}_f$ be a given image sequence comprising $f$ frames of a human actor that performs motions. The $j$-th frame $\mathcal{I}_j \in [0,1]^{w \times h \times 3}$ is an RGB image of dimension $w \times h$.

**3D Character Model.** For each subject we create a 3D character model based on the multi-view image-based 3D reconstruction software Agisoft Photoscan[1]. To this end, we capture approximately a hundred images from different view points of the actor in a static neutral pose (upright standing and the arms forming a "T-pose", see Fig. 2 "Character model"). This data is then directly used as input to Photoscan, which produces a textured 3D model of the person, as shown in Fig. 2 ("Character model" and "Static texture"). Then, we rig the character model with a parameterized skeleton model, similarly as done in other approaches (e.g. [74]). Based on this procedure we obtain a parameterized surface mesh model with vertex positions $\mathcal{M}(\theta) \in \mathbb{R}^{n \times 3}$, where $n$ is the number of mesh vertices and $\theta \in \mathbb{R}^{33}$ is the pose parameter vector, where among the 33 scalar values 6 are global rigid pose parameters, and 27 are pose articulation parameters in terms of joint angles.

**Texture Mapping.** For texture mapping, we unwrap the human body surface mesh and map it onto the unit square $[0,1]^2$ using the quasi-harmonic surface parameterization method of [89], which reduces the parametric distortion by attempting to undo the area distortion in the initial conformal mapping. To this end, the mesh is first cut along the spine, followed by two cuts along the legs, as well as three cuts along the arms and the head. Then, this boundary is mapped to the boundary of the square. A so-created RGB texture $\mathcal{T} \in [0,1]^{w \times h \times 3}$ is shown in Fig. 2 ("Static texture").

**Human Performance Capture.** We employ the recent real-time dense motion capture method of [74]. Their two-stage energy-based method first estimates the actor's pose by using a sparse set of body and face landmarks, as well as the foreground silhouette. The output of the motion capture stage is the pose vector $\theta$, which can be used to pose the surface model, resulting in a deformed mesh with vertex positions $\mathcal{M}(\theta)$. Next, the reconstruction is refined on the surface level to account for local non-rigid deformations that cannot be captured by a pure skeleton-based deformation. To this end, per-vertex displacements are estimated using a dense silhouette and photometric constraints.

**Target Dynamic Texture Extraction.** After the performance capture, we generate the pose-specific partial dynamic texture $\mathcal{T}_j$ by back-projecting the input image frame $\mathcal{I}_j$ onto the performance capture result, i.e., the deformed mesh $\mathcal{M}(\theta_j)$. Note that the generated dynamic textures are incomplete, since we only have front view observation, due to the monocular setup.

Although the reconstructed 3D body model yields a faithful representation of the true body geometry, small tracking errors between the digital model and the real human are inevitable. A major issue is that such small misalignments would directly result in an erroneous texture map $\mathcal{T}_j$ (e.g. a common case is that a hand in front of the torso leads to the incorrect assignment of the hand color to a torso vertex, see Fig. 3). Using such noisy texture maps would be disadvantageous for learning, as the network would need to spend capacity on understanding and (implicitly) fixing these mismatches. Instead, based on a simple image-based analysis we filter out the erroneous parts and thereby avoid training data corruption. The filtering method consists of four simple steps:

(i) First, we generate an average texture map $\overline{\mathcal{T}} \in [0,1]^{w \times h \times 3}$ by averaging all colors of $\mathcal{T}_1, \dots, \mathcal{T}_f$ along the temporal axis. Note that texels that correspond to occluded mesh vertices of $\mathcal{M}(\theta_j)$, i.e. zero values in the texture map $\mathcal{T}_j$, are not taken into account for averaging.

(ii) Next we use a $k$-means clustering procedure to cluster all the colors present in the average texture map $\overline{\mathcal{T}}$ so that we obtain a small number of $k$ *prototype colors* that are "typical" to the specific sequence at hand.

(iii) Then, for all frames $j$ we assign to each texel of $\mathcal{T}_j$ its nearest prototype color, which is then used to compute a (per-texel) histogram of the prototype colors over all the frames (again, only considering visible parts).

(iv) Finally, for each texel we check whether there is a prototype color that only occurs very rarely. If yes, we suppose that it is caused by the transient color of a tracking error (e.g. a wrongly tracked hand), and therefore discard the color assignment for this texel in all frames where the insignificant color is present.

1. http://www.agisoft.com/

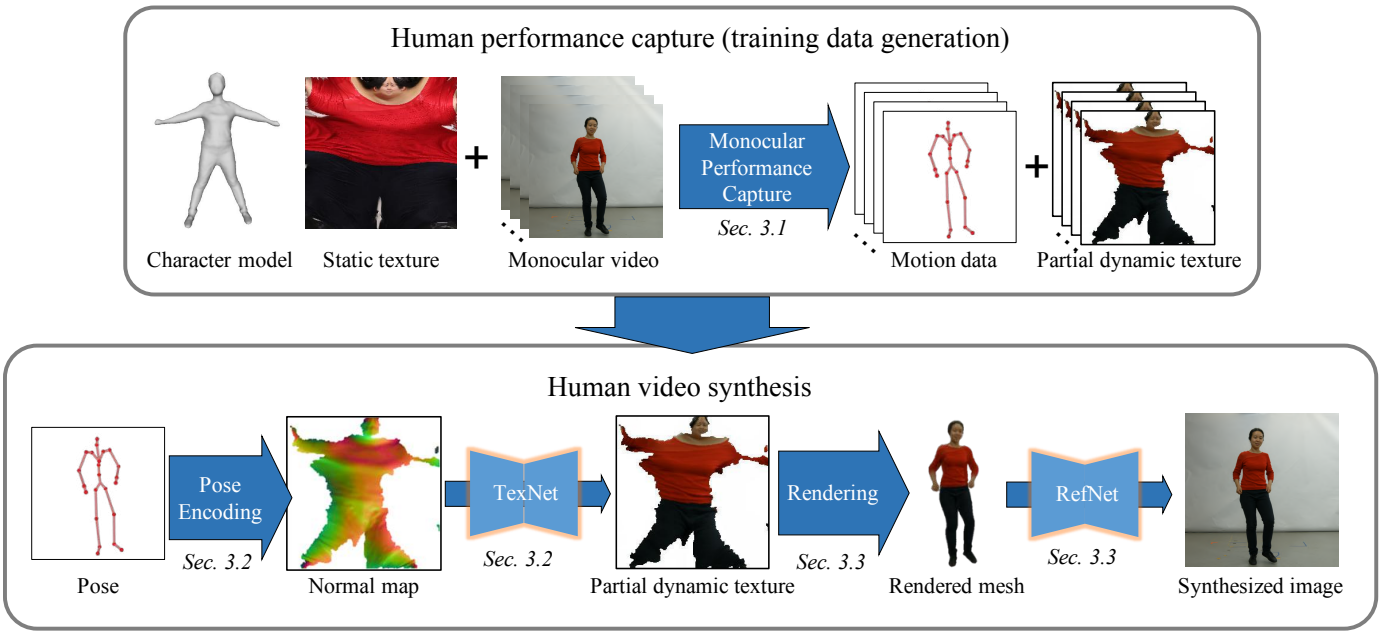



Fig. 2. Overview of our approach. The top shows the human performance capture stage that is used for training data generation. Here, a parametric human character model is used in combination with a static texture for tracking the human motion in a monocular video and encoding the motion to the partial normal map. The output are motion data and dynamic (per-frame) partial textures, which capture pose-dependent high-frequency details (e.g. cloth wrinkles). The bottom part shows the human video synthesis stage. First, a pose-dependent partial normal map is generated by animating the 3D static template according to the motion data and unwarping the visible region of the human body mesh to uv space (e.g. obtained by motion capture as on the top, user-defined, or from any other source). This partial normal map serves as a pose encoding in texture space, which is then used as input to a *texture synthesis network* (TexNet) for computing a pose-dependent partial texture map. The mesh rendered with this texture is then used as input to the *refinement network* (RefNet) that produces the final output by blending the foreground and background, modelling shadows, and correcting geometric errors.

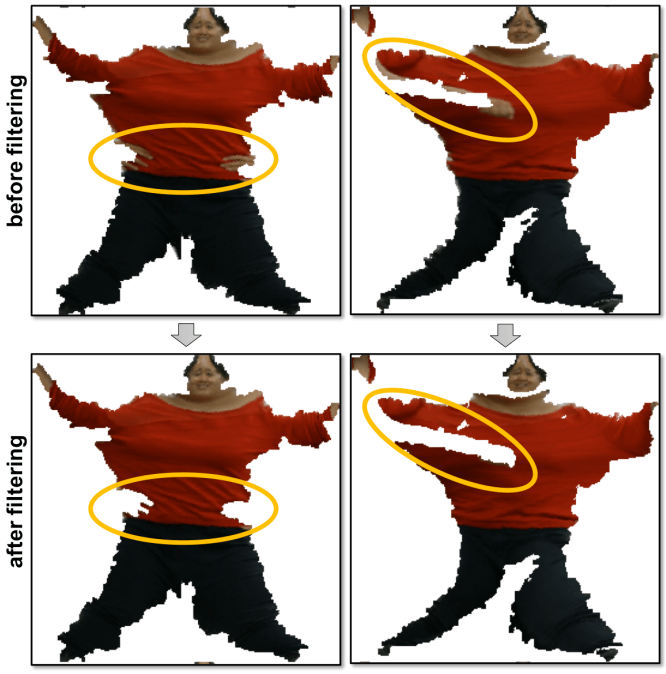



Fig. 3. Effect of our filtering procedure. The top row shows the texture map before filtering, and the bottom row shows it after filtering.

By doing so, erroneous color assignments are excluded from the partial textures to enhance network training quality. In Fig. 3 we illustrate the effect of our filtering procedure. In addition, to avoid the background pixels from being projected onto the mesh, we apply a foreground mask, generated with the video segmentation method of [90], on the input images when doing the back-projection.

Subsequently, we fill in the discarded texels based on the average texture $\overline{\mathcal{T}}$. The so-created partial dynamic textures $\mathcal{T}_j$, together with the tracking results $\theta_j$, are then used as training data to our networks.

### 3.2 Dynamic Texture Synthesis

Now we describe our first network, the *texture synthesis network* (TexNet), which generates a *pose-dependent texture* given the corresponding pose $\theta$ as conditional input. With that, we are able to generate pose-dependent high-frequency details directly in texture space, such as for example cloth wrinkles, which otherwise would require complex and computationally expensive offline rendering approaches (e.g. cloth simulation).

**Pose Encoding.** Since the texture that we aim to generate is represented in texture space (or *uv-space*), it is advantageous to also use an input that lives in the same domain. Hence, we have chosen to represent the pose using a *partial normal map in texture space* (cf. Fig. 2, "Partial normal map"), which we denote by $\mathcal{N} \in (\mathcal{S}^2)^{w \times h \times 3}$, for $\mathcal{S}^2$ being a unit 2-sphere embedded in 3D space (i.e. the set of all unit length 3D vector). We note that here we use the camera coordinate system for normal calculation, since the

appearance/illumination would change if the person faces a different direction. In order to allow for texels that do not have an assigned normal, we include the zero vector in $\mathcal{S}^2$. Compared to other pose representations, such as for example a depth map of a 3D skeleton, using such an *image-based* pose encoding in texture space facilitates simplified learning because the network does not need to additionally learn the translation between different domains (see the ablation study). The partial normal map $\mathcal{N}_j$ is created based on the 3D body reconstruction $\mathcal{M}(\theta_j)$ at frame $j$.

To this end, for each vertex of the fitted 3D model that is visible in the current frame, we compute its (world-space) surface normal, and then create the partial normal map using the mesh's uv-mapping (Sec. 3.1). Note that those areas in the partial normal map that correspond to invisible vertices are set to zero, cf. Fig. 2 ("Partial normal map").

**Texture Synthesis Network.** The TexNet has the purpose of creating a pose-dependent texture from a given input partial normal map, as illustrated in Fig. 2. As such, we aim to learn the network parameters $\Theta$ that parameterize the TexNet $f_\Theta^{\text{tex}}$ translating a given partial normal map $\mathcal{N} \in (\mathcal{S}^2)^{w \times h}$ to a pose-dependent texture $\mathcal{T} \in [0,1]^{w \times h \times 3}$. For training the network, we require pairs of partial normal maps and target partial texture maps $\{(\mathcal{N}_j, \mathcal{T}_j) : 1 \leq j \leq f\}$, which are directly computed from the input sequence $\mathcal{I}_1, \ldots, \mathcal{I}_f$ based on motion capture as described in Sec. 3.1. During test time, for each frame $\mathcal{I}_j$ the partial normal map $\mathcal{N}_j$ is extracted using the 3D reconstruction $\mathcal{M}(\theta_j)$, and the texture map $\mathcal{T}_j = f_\Theta^{\text{tex}}(\mathcal{N}_j)$ is synthesized by the network.

**Network Architecture.** Since the recent *vid2vid* network [6] was shown to synthesize photo-realistic and temporally consistent videos, we build our network upon its state-of-the-art architecture. It considers the temporal consistency in a local window (we set the window size to 3 in our experiments). This is achieved by leveraging optical flow based warping together with conditional generative adversarial networks (cGANs). The cGANs jointly learn the generator function $f_\Theta^{\text{tex}}$ to produce the output texture map $\mathcal{T} = f_\Theta^{\text{tex}}(\mathcal{N})$ from a given conditioning input partial normal map $\mathcal{N}$, along with a discriminator function $\mathcal{D}$. The latter has the purpose to classify whether a given texture map $\mathcal{T}$ is a synthesized texture (produced by the generator $f_\Theta^{\text{tex}}$) or a real texture. The general cGAN loss function reads:

$$\mathcal{L}^{\text{cGAN}}(f_\Theta^{\text{tex}}, \mathcal{D}) = \mathbb{E}_{\mathcal{T},\mathcal{N}}(\log \mathcal{D}(\mathcal{T}, \mathcal{N})) + \mathbb{E}_{\mathcal{N}}(\log(1 - \mathcal{D}(f_\Theta^{\text{tex}}(\mathcal{N}), \mathcal{N}))) \,. \tag{1}$$

To obtain realistic individual frames, as well as a temporally consistent sequence of frames, a per-frame cGAN loss term $\mathcal{L}^{\text{frm}}$ is used in combination with a video cGAN loss term $\mathcal{L}^{\text{vid}}$ that additionally incorporates the previous two frames. Furthermore, the term $\mathcal{L}^{\text{flow}}$ is used to learn the optical flow fields. The total learning problem now reads:

$$\min_{f_\Theta^{\text{tex}}} \max_{\mathcal{D}^{\text{frm}}, \mathcal{D}^{\text{vid}}} \mathcal{L}^{\text{frm}}(f_\Theta^{\text{tex}}, \mathcal{D}^{\text{frm}}) + \mathcal{L}^{\text{vid}}(f_\Theta^{\text{tex}}, \mathcal{D}^{\text{vid}}) + \lambda \mathcal{L}^{\text{flow}} \,. \tag{2}$$

**Training.** We use approximately 12,000 training pairs, each of which consists of the ground truth texture map $\mathcal{T}$ as well as the partial normal map $\mathcal{N}$. For training, we set a hyper-parameter of $\lambda = 10$ for the loss function, and use the Adam optimizer ($lr = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.99$), which we run for a total number of 10 epochs with a batch size of 8. For each sequence of $256 \times 256$ images, we use 8 Nvidia Tesla V100 GPUs to train for about 2 days.

**Runtime During Testing.** A forward pass of TexNet takes 8ms/frame to generate a $256 \times 256$ image on a Nvidia Tesla V100 GPU.

### 3.3 High-fidelity Video Synthesis

By synthesizing the texture using TexNet, we bake pose-specific high-frequency details into the texture. This texture is now used for generating the final output by means of a *refinement network* (RefNet). The RefNet has the task of synthesizing the background, as well as dealing with background-foreground interactions, such as shadows. Moreover, it implicitly learns to correct geometric errors due to tracking misalignments and due to skinning errors.

**Training Data.** In order to train the RefNet, we first run TexNet in order to obtain the (partial) dynamic texture map of all frames. Subsequently, we fill in the invisible texels based on the average texture (across the temporal axis) to obtain a full texture map. Then, we use the full texture map to render the mesh of the 3D reconstruction obtained by motion capture. The RefNet is now trained on this data for the task of synthesizing the original input image, given the rendered mesh, cf. Fig. 2.

**Network Architecture.** The architecture is the same as the TexNet, with the main difference being that instead of learning a function that maps a partial normal map to a color texture, we now learn a function $f_\Phi^{\text{ref}}$ that maps a rendered image to a realistic output, see Fig. 2. The loss function is analogous to Eq. 2 with $f_\Phi^{\text{ref}}$ in place of $f_\Theta^{\text{tex}}$.

**Training.** We use approximately 12,000 training pairs, each of which consists of the rendered image and the original RGB image. For training, we set a hyper-parameter of $\lambda = 10$ for the loss function, and use the Adam optimizer ($lr = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.99$) which we run for a total of 10 epochs with a batch size of 8. For each sequence of $256 \times 256$ images, we use 8 Nvidia Tesla V100 GPUs to train for about 2 days. For higher resolution results of $512 \times 512$, we need about 6 days on the same GPUs.

**Runtime During Testing.** A forward pass of RefNet requires 8ms/frame to generate $256 \times 256$ images on a Nvidia Tesla V100 GPU, 15ms/frame for $512 \times 512$, and 33ms/frame for $1024 \times 1024$.

## 4 Experiments

To evaluate our approach and provide comparisons to existing methods, we conduct experiments on the 7 video sequences from [8]. Each sequence comprises approximately 12,000 frames, where the subjects are instructed to perform a wide range of different motions, so that the space of motions is sufficiently covered by the training data. We split each sequence into a training sequence and a test sequence, where the last quarter of each sequence is used for testing. In addition, we captured a new sequence to demonstrate the use of our approach in a novel-view synthesis setting, and we also evaluate our method based on a community video as driving sequence. In the following, we show our qualitative results on the motion transfer and novel-view synthesis


Fig. 4. Example frames for our motion transfer results. The 1st row shows frames from the source videos, the 2nd row shows the meshes rendered with the synthesized textures (input to our RefNet), and the 3rd row shows our final results. See our supplementary video for complete results.

tasks and provide comparisons to previous state-of-the-art methods. Then, we perform an ablation study to evaluate the importance of each component of our approach.

### 4.1 Motion Transfer

For the motion transfer application, we make use of pairs of monocular video sequences and our goal is to synthesize a video of the target actor performing the motion of the source actor, i.e., to transfer the motion from the source video to the target video. To this end, we estimate the optimal pose of the target person for each frame by solving a inverse kinematics (IK) problem as in [8], which encourages the corresponding keypoints on both skeletons, including the joints and facial landmarks, to match each other in 3D as much as possible. Note that directly applying the source's skeletal pose parameters to the target skeleton may fail to produce acceptable results in general for two reasons: First, this would require that both skeletons have exactly the same structure, which may be overly restrictive in practice. Second, even more importantly, differences in the rigging of the skeleton would lead to incorrect poses if the pose parameters of the source skeleton are applied directly to


Fig. 5. Qualitative comparison against previous state-of-the-arts on the motion transfer application. The first row shows the input sequence that is used to drive the motion, the second row shows the results obtained from our method, and the remaining rows show results obtained by the methods from Liu et al. [8], Wang et al. [6], Chan et al. [4], Ma et al. [91], Esser et al. [3].

Fig. 6. Bullet time video frame examples. Our method can be used to synthesize new views of the actor using just a monocular video.

Fig. 7. Reenactment result of using internet video footage as driving motion.

the target skeleton. Several example frames of the motion transfer results are shown in Fig. 4. We can see from the mesh rendered with the synthesized dynamic texture (see Fig. 4, 2nd row) that our TexNet is able to capture the pose-dependant details, such as wrinkles, while the RefNet yields realistic images, where artifacts due to tracking/skinning errors are corrected and the natural blending and interaction (shadows) between foreground and background are synthesized. We point out that even the results of non-frontal motions look plausible. In our supplementary video we show additional animated results. Our approach can also take a user-designed motion as source motion input, which allows the user to interactively reenact the actor using a handle-based editor (see the demonstration in our supplementary video). Furthermore, we stress test our approach by using internet video footage as driving motion. Although the driving motion is very different from the motions in our training corpus, our approach generates plausible results (see Fig. 7 and the supplementary video).

We compare our approach with the following five methods on two sequences: Esser et al. [3], Ma et al. [91], Liu et al. [8], Chan et al. [4], and Wang et al. [6]. For fair comparison, we apply the input in the same formats to the networks in the comparison methods as they require. Specifically, the input to Esser et al. [3], Ma et al. [91] and Chan et al. [4] is the motion of a 2D skeleton. A part-based RGBD representation is used as input for Liu et al. [8]. The tracking results obtained with OpenPose [92] and DensePose [41] are used as input to Wang et al. [6].

The qualitative comparisons are provided in Fig. 5. Again, we refer the reader to the supplementary video for better visual comparisons. As can be seen from the video, our approach yields temporally more coherent results and exhibits less artifacts than the competing methods. Especially, the artifact of missing limbs is significantly alleviated in our results. Also note that, in contrast to our method, the methods of Esser et al. [3], Ma et al. [91] and Wang et al. [6] do not preserve the identity (body shape) of the actors, since their motion transfer is done in the 2D image space (e.g. with 2D landmarks positions), while ours is done in the skeleton pose space. Furthermore, our approach yields geometrically more consistent results. For example, wrinkles in our results move coherently with the garments, rather than being attached to a separated spatially fixed layer in screen space, as can be observed for the other methods. These benefits come from a well-designed three-stage pipeline that first generates a dynamic texture with time-coherent high-frequency details and then renders the mesh with the dynamic texture, which is eventually refined in screen space. To help understanding how each component of the pipeline contributes to the final result, we provide thorough ablations in Section 4.4, including the use of rendered mesh with dynamic texture rather than a sparse skeleton or rendered meshes with average/static texture as input to the second network, and the importance of a partial normal map as input to the first network, etc.

We also compare the output of TexNet with the texture map retrieved by a simple nearest-neighbor-based approach. The similarity of two motions is defined as the $\ell_2$-norm of the difference of the motions represented by 30 joint angles $\theta$ ($\theta \in (-\pi, \pi]$). We fetch the texels from the texture map of the closest pose and fill-in the invisible region using the average texture. The results are clearly worse and show many spatial and temporal artifacts (see the supplementary video).

### 4.2 Novel-View Synthesis

Novel-view synthesis is an important task for many real-world applications, such as VR-based telepresence and the iconic "bullet time" visual effect for the film industry. Our proposed approach can deliver such results based on just a monocular video. To demonstrate this, we captured a monocular video sequence and showcase the bullet time visual effect based on our approach. In each video, the actor is asked to perform a similar set of motions (Karate exercise) for multiple repetitions in eight different global rotation angles (rotated in 45 degrees steps) with respect to

the camera. This lets the camera capture similar poses from different viewing directions. The captured video is tracked and used for training our networks. For testing, we select a fixed pose out of the motion sequences, and then use a virtual camera orbiting around the actor to generate the conditional input images to our approach. This allows us to synthesize realistic video of the actor frozen in a certain pose, viewed from different angles. Some example frames are shown in Fig. 6 and the complete video can be found in the supplementary material. Note that we do not synthesize background, i.e., the rotating floor and walls, but render them with Blender[2] with the same orbiting camera. Then, we segment out the foreground of our synthesized video, using the method of [90], and composite the foreground and the rendered background.

### 4.3 User Study

Following many other image synthesis methods, we evaluate our approach in terms of user perception via a user study and also provide comparisons to existing methods in this manner. Therefore, we show pairs of video synthesis results from 6 different methods to 18 users. These six methods include ours and the methods of Esser et al. [3], Ma et al. [91], Liu et al. [8], Chan et al. [4], and Wang et al. [6]. Our result is always included in each pair, thus performing the direct comparison between our method and each of the existing methods. In total, 30 pairs of videos from two sequences are shown to the users. The user study video and the labels of all pairs are provided in the supplementary material. After watching the videos, the users are asked to select the one from each pair that appears more natural and realistic. In Table. 1 we provide the percentages of votes for our method, when compared to the respective existing method. We can see that our results are considered more realistic than all existing methods. Although Wang et al. [6] is slightly more preferable on sequence 2, we show in the supplementary video that their method only transfers the appearance but incorrectly scales the person to match the driving actors shape. Note that this user study does not allow relative comparison among the previous methods, since they are not directly shown to the user side by side.

TABLE 1
Comparison of our method with existing methods through a user study. The percentages of votes for our method are provided. Numbers larger than 50 mean that our results are considered more realistic.

| Methods | Seq 1 | Seq 2 | All |
|---|---|---|---|
| **Esser et al. [3]** | 90.74 | 94.44 | 92.59 |
| **Ma et al. [91]** | 100.00 | 96.30 | 98.15 |
| **Liu et al. [8]** | 88.68 | 72.55 | 80.61 |
| **Chan et al. [4]** | 67.92 | 68.52 | 68.22 |
| **Wang et al. [6]** | 79.63 | 46.30 | 62.96 |

### 4.4 Ablation Study

We evaluate the importance of individual components of our approach via a quantitative ablation study. To this end, we split one video into a training (12130 frames) and a test set (4189 frames). We evaluate the error on the test set with respect to the ground truth. As we are mainly interested in synthesizing the appearance of the human body, we compute the error only on the foreground region.

**Relevance of TexNet.** First, we investigate the importance of using the dynamic texture generation based on TexNet. For this analysis, we consider the two cases where we train the RefNet based on two alternative inputs: 1) the static texture from the 3D reconstruction (cf. Fig. 2 "Static texture"), and 2) the average texture computed from the visible texels of the texture extracted from the training video (cf. Sec. 3.2). The reconstruction error of these two and our approach are shown in Tab. 2 ("Average texture (RefNet)", "Static texture (RefNet)", and "Ours (TexNet + RefNet)"). We can see that our full pipeline significantly outperforms these two baseline methods in terms of average per-pixel mean error and SSIM (see the supplementary video for the visual results).

**Importance of filtering stage.** We have also analyzed the importance of the filtering stage as used for the target texture extraction (Sec. 3.1). To this end, we trained one network on unfiltered data, see Tab. 2 ("Without filtering (TexNet) + RefNet"). It can be seen that our full approach outperforms this network. Although quantitatively the improvements may appear small due to the relatively small area that is affected, we have found that the filtering qualitatively improves the results significantly, see Fig. 8.

**Importance of partial normal map input.** We have also analyzed the importance of the partial normal map as input to our TexNet. For this analysis, we consider two cases: 1) we train TexNet using a rendered 3D skeleton and its depth as input ("Rendered 3D skeleton (TexNet) + RefNet"), and 2) a direct mapping (only RefNet) from the rendered 3D skeleton to the final image ("Rendered 3D skeleton (RefNet)"). As shown in Tab. 2, our full pipeline outperforms these two baselines. For the first case, compared to a depth map of a 3D skeleton, using a partial normal map to encode the pose as the input to TexNet is more effective and more robust since it does not need more effort to learn the translation between different domains. Also, in the second case, we can see that the dense mesh representation is more informative than the sparse skeleton and therefore can achieve better results (see the supplementary video for the visual results).

TABLE 2
Quantitative evaluation. We report the mean (for the whole sequence) of the L2 error and SSIM for the region of the person in the foreground. Our full approach obtains the best scores.

| | L2 error | SSIM |
|---|---|---|
| **Rendered 3D skeleton (TexNet) + RefNet** | 9.558 | 0.763 |
| **Rendered 3D skeleton (RefNet)** | 9.726 | 0.755 |
| **Average texture (RefNet)** | 9.133 | 0.771 |
| **Static texture (RefNet)** | 8.958 | 0.775 |
| **Without filtering (TexNet) + RefNet** | 8.744 | 0.781 |
| **Ours (TexNet + RefNet)** | **8.675** | **0.784** |

**Size of training dataset.** We also evaluate the dependence of the performance on the size of the training dataset. In this experiment, we train TexNet and RefNet with 6000, 9000, 12130 frames of the target sequence. See Table 3 for the

2. https://www.blender.org/


Fig. 8. Ablative results for the proposed filtering procedure used for the target texture extraction. We show three instances, where the left images show the result of the rendered mesh with a dynamically generated texture without filtering, and the right images show the analogous images with filtering. When not using filtering, one can clearly see additional artifacts in the hand areas.

quantitative results, and the supplementary video for the visual results. As expected, larger training sets have more pose variety and hence can produce better results. For better generalizability, the poses in our training data should be as diverse as possible. If the testing pose is very different from any of the training poses, the synthesis quality will degrade but still look reasonable due to the generalizability of the networks (see, for example. the results with Youtube videos as driving sequences in the supplementary video).

TABLE 3
Quantitative evaluation on the dependency of the performance on the training dataset size. We report the mean (for the whole sequence) of the L2 error and SSIM for the region of the person in the foreground. Our full training set obtains the best scores.

| | L2 error | SSIM |
|---|---|---|
| **6000 frames** | 10.003 | 0.749 |
| **9000 frames** | 9.287 | 0.767 |
| **12130 frames** | **8.675** | **0.784** |

## 5 Discussion and Limitations

In addition to the presented use-cases of motion transfer, interactive reenactment, and novel-view synthesis, another potential application of our approach is the generation of annotated large-scale human image or video datasets. Particularly, with the recent popularity of deep learning, such datasets could be used for many different computer vision tasks, such as human detection, body pose estimation, and person re-identification.

Our experimental results demonstrate that our method outperforms previous approaches for the synthesis of human videos. However, there are still some issues that could be addressed in future work. One important issue is that the currently used neural network architectures (TexNet and RefNet) are computationally expensive to train. In order to move on to very high image resolutions, one needs to reduce the network training time. For example, training each network for an image resolution of $256 \times 256$ takes already two days, and training it for an image resolution of $512 \times 512$ takes about 6 days on 8 high-end GPUs, and training for an image resolution of $1024 \times 1024$ takes about 10 days on 8 high-end GPUs. Another point that is a common issue in machine learning approaches is generalization. On the one hand, our trained networks can only produce reasonable results when the training data has a similar distribution to the test data. For example, it would not be possible to train a network using frontal body views only, and then synthesize reasonable backsides of a person. On the other hand, in our current approach we train person-specific networks, whereas it would be desirable to train networks for more general settings. While we cannot claim that the results produced by our approach are entirely free of artifacts, we have demonstrated that in overall the amount and severity of artifacts is significantly reduced compared to other methods. Another limitation is that we are not able to faithfully generate the fingers, since the human performance capture method cannot track finger motion. This can be alleviated in future works by incorporating a more complicated hand model and finger tracking components. Furthermore, the artifacts regarding the hands and feet are due to the 3D tracking used for generating the training data. The error in the 3D tracking would lead to a misalignment between the ground truth image and the rendered mesh in the second stage, which makes it hard for the network to directly learn this mapping.

## 6 Conclusion

We have presented a novel method for video synthesis of human actors. Our method is a data-driven approach that learns, from a monocular video, to generate realistic video footage of an actor, conditioned on skeleton pose input. In contrast to most existing methods that directly translate the sparse pose information into images, our proposed approach explicitly disentangles the learning of time-coherent fine-scale pose-dependent details from the embedding of the human in 2D screen space. As a result, our approach leads to significant better human video synthesis results, as we have demonstrated both qualitatively and quantitatively.

## References

[1] L. Ma, Q. Sun, X. Jia, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *NIPS*, 2017.
[2] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, "Synthesizing images of humans in unseen poses," in *Computer Vision and Pattern Recognition (CVPR), 2018*, 2018.
[3] P. Esser, E. Sutter, and B. Ommer, "A variational u-net for conditional appearance and shape generation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[4] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody Dance Now," *arXiv e-prints*, p. arXiv:1808.07371, Aug. 2018.
[5] K. Aberman, M. Shi, J. Liao, D. Lischinski, B. Chen, and D. Cohen-Or, "Deep Video-Based Performance Cloning," *arXiv e-prints*, p. arXiv:1808.06847, Aug. 2018.
[6] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
[7] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable GANs for pose-based human image generation," in *CVPR 2018*, 2018.
[8] L. Liu, W. Xu, M. Zollhöfer, H. Kim, F. Bernard, M. Habermann, W. Wang, and C. Theobalt, "Neural rendering and reenactment of human actor videos," *ACM Trans. Graph.*, vol. 38, no. 5, Oct. 2019. [Online]. Available: https://doi.org/10.1145/3333002
[9] F. Xu, Y. Liu, C. Stoll, J. Tompkin, G. Bharaj, Q. Dai, H.-P. Seidel, J. Kautz, and C. Theobalt, "Video-based characters: Creating new human performances from a multi-view video database," in *ACM SIGGRAPH 2011 Papers*, ser. SIGGRAPH '11. New York, NY, USA: ACM, 2011, pp. 32:1–32:10. [Online]. Available: http://doi.acm.org/10.1145/1964921.1964927

[10] K. Li, J. Yang, L. Liu, R. Boulic, Y.-K. Lai, Y. Liu, Y. Li, and E. Molla, "Spa: Sparse photorealistic animation using a single rgb-d camera," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 27, no. 4, pp. 771–783, Apr. 2017. [Online]. Available: https://doi.org/10.1109/TCSVT.2016.2556419

[11] D. Casas, M. Volino, J. Collomosse, and A. Hilton, "4d video textures for interactive character appearance," *Comput. Graph. Forum*, vol. 33, no. 2, pp. 371–380, May 2014. [Online]. Available: http://dx.doi.org/10.1111/cgf.12296

[12] M. Volino, D. Casas, J. Collomosse, and A. Hilton, "Optimal representation of multiple view video," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[13] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel, "Free-viewpoint video of human actors," *ACM Trans. Graph.*, vol. 22, no. 3, Jul. 2003.

[14] G. Borshukov, D. Piponi, O. Larsen, J. P. Lewis, and C. Tempelaar-Lietz, "Universal capture-image-based facial animation for the matrix reloaded," in *ACM Siggraph 2005 Courses*. ACM, 2005, p. 16.

[15] G. Li, Y. Liu, and Q. Dai, "Free-viewpoint video relighting from multi-view sequence under general illumination," *Mach. Vision Appl.*, vol. 25, no. 7, pp. 1737–1746, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1007/s00138-013-0559-0

[16] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 600–608.

[17] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan, "High-quality streamable free-viewpoint video," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 69, 2015.

[18] G. Li, C. Wu, C. Stoll, Y. Liu, K. Varanasi, Q. Dai, and C. Theobalt, "Capturing relightable human performances under general uncontrolled illumination." *Comput. Graph. Forum*, vol. 32, no. 2, pp. 275–284, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/cgf/cgf32.html#LiWSLVDT13

[19] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[20] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 194:1–194:17, Nov. 2017, two first authors contributed equally.

[21] P. Bérard, D. Bradley, M. Nitti, T. Beeler, and M. Gross, "High-quality capture of eyes," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 223:1–223:12, 2014.

[22] E. Wood, T. Baltrusaitis, L. P. Morency, P. Robinson, and A. Bulling, "A 3d morphable eye region model for gaze estimation," in *ECCV*, 2016.

[23] C. Wu, D. Bradley, P. Garrido, M. Zollhöfer, C. Theobalt, M. Gross, and T. Beeler, "Model-based teeth reconstruction," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 220:1–220:13, 2016.

[24] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, pp. 245:1–245:17, Nov. 2017. [Online]. Available: http://doi.acm.org/10.1145/3130800.3130883

[25] S. Lombardi, J. Saragih, T. Simon, and Y. Sheikh, "Deep appearance models for face rendering," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 68:1–68:13, Jul. 2018.

[26] K. Nagano, J. Seo, J. Xing, L. Wei, Z. Li, S. Saito, A. Agarwal, J. Fursund, and H. Li, "pagan: Real-time avatars using dynamic textures," in *SIGGRAPH Asia 2018 Technical Papers*, ser. SIGGRAPH Asia '18. New York, NY, USA: ACM, 2018, pp. 258:1–258:12.

[27] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: Shape completion and animation of people," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 408–416, Jul. 2005. [Online]. Available: http://doi.acm.org/10.1145/1073204.1073207

[28] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[29] G. Pons-Moll, S. Pujades, S. Hu, and M. Black, "Clothcap: Seamless 4d clothing capture and retargeting," *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, vol. 36, no. 4, 2017, two first authors contributed equally.

[30] Z. Lahner, D. Cremers, and T. Tung, "Deepwrinkles: Accurate and realistic clothing modeling," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[31] J. Yang, J.-S. Franco, F. Hetroy-Wheeler, and S. Wuhrer, "Analyzing clothing layer deformation statistics of 3d human motions," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[32] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2242–2251.

[33] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," *CoRR*, vol. abs/1712.01057, 2017.

[34] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, N. Nießner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt, "Deep Video Portraits," *ACM Transactions on Graphics 2018 (TOG)*, 2018.

[35] C. Lassner, G. Pons-Moll, and P. V. Gehler, "A generative model of people in clothing," in *Proceedings IEEE International Conference on Computer Vision (ICCV)*. Piscataway, NJ, USA: IEEE, Oct. 2017. [Online]. Available: http://files.is.tuebingen.mpg.de/classner/gp/

[36] C. Si, W. Wang, L. Wang, and T. Tan, "Multistage adversarial losses for pose-based human image synthesis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[37] A. Pumarola, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Unsupervised person image synthesis in arbitrary poses," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[38] P. Esser, J. Haux, T. Milbich, and B. orn Ommer, "Towards learning a realistic rendering of human behavior," in *The European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[39] N. Neverova, R. Alp Guler, and I. Kokkinos, "Dense pose transfer," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[40] A. Shysheya, E. Zakharov, K.-A. Aliev, R. Bashirov, E. Burkov, K. Iskakov, A. Ivakhnenko, Y. Malkov, I. Pasechnik, D. Ulyanov, A. Vakhitov, and V. Lempitsky, "Textured neural avatars," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[41] R. A. Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7297–7306.

[42] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: image synthesis using neural textures," *ACM Trans. Graph.*, vol. 38, pp. 66:1–66:12, 2019.

[43] Y. Li, C. Huang, and C. C. Loy, "Dense intrinsic appearance flow for human pose transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[44] W. Liu, W. L. L. M. Zhixin Piao, Min Jie, and S. Gao, "Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis," in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[45] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, "Image-based visual hulls," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 369–374.

[46] J. Starck and A. Hilton, "Surface capture for performance-based animation," *IEEE Computer Graphics and Applications*, vol. 27, no. 3, pp. 21–31, 2007.

[47] M. Waschbüsch, S. Würmlin, D. Cotting, F. Sadlo, and M. Gross, "Scalable 3d video of dynamic scenes," *The Visual Computer*, vol. 21, no. 8-10, pp. 629–638, 2005.

[48] C. Cagniart, E. Boyer, and S. Ilic, "Free-form mesh tracking: a patch-based approach," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1339–1346.

[49] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik, "Dynamic shape capture using multi-view photometric stereo," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, p. 174, 2009.

[50] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi, "Fusion4d: Real-time performance capture of challenging scenes," *ACM Trans. Graph.*,

vol. 35, no. 4, pp. 114:1–114:13, Jul. 2016. [Online]. Available: http://doi.acm.org/10.1145/2897824.2925969

[51] M. Dou, P. Davidson, S. R. Fanello, S. Khamis, A. Kowdle, C. Rhemann, V. Tankovich, and S. Izadi, "Motion2fusion: Real-time volumetric performance capture," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 246:1–246:16, Nov. 2017.

[52] R. Wang, L. Wei, E. Vouga, Q. Huang, D. Ceylan, G. Medioni, and H. Li, "Capturing dynamic textured surfaces of moving targets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[53] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[54] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional Pose Machines," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[55] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, "Deep kinematic pose regression," in *ECCV Workshop on Geometry Meets Deep Learning*, 2016.

[56] D. Mehta, H. Rhodin, D. Casas, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation using transfer learning and improved cnn supervision," in *3DV*, 2017.

[57] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[58] A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt, "Efficient ConvNet-based marker-less motion capture in general scenes with a low number of cameras," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3810–3818.

[59] R. Rosales and S. Sclaroff, "Combining generative and discriminative models in a framework for articulated pose estimation," *International Journal of Computer Vision*, vol. 67, no. 3, pp. 251–276, 2006.

[60] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," vol. 36, no. 4, July 2017.

[61] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1746–1753.

[62] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 97.

[63] Y. Liu, C. Stoll, J. Gall, H.-P. Seidel, and C. Theobalt, "Markerless motion capture of interacting characters using multi-view image segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1249–1256.

[64] C. Wu, C. Stoll, L. Valgaerts, and C. Theobalt, "On-set Performance Capture of Multiple Actors With A Stereo Camera," in *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, vol. 32, no. 6, November 2013, pp. 161:1–161:11. [Online]. Available: http://doi.acm.org/10.1145/2508363.2508418

[65] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 98.

[66] Q. Zhang, B. Fu, M. Ye, and R. Yang, "Quality Dynamic Human Body Modeling Using a Single Low-cost Depth Camera," in *CVPR*. IEEE, 2014, pp. 676–683.

[67] F. Bogo, M. J. Black, M. Loper, and J. Romero, "Detailed full-body reconstructions of moving people from monocular RGB-D sequences," in *International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 2300–2308.

[68] T. Helten, M. Muller, H.-P. Seidel, and C. Theobalt, "Real-time body tracking with one depth camera and inertial sensors," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[69] T. Yu, K. Guo, F. Xu, Y. Dong, Z. Su, J. Zhao, J. Li, Q. Dai, and Y. Liu, "Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 910–919.

[70] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *European Conference on Computer Vision (ECCV)*, 2016.

[71] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural body fitting: Unifying deep learning and model based human pose and shape estimation," in *2018 International Conference on 3D Vision (3DV)*, Sep. 2018, pp. 484–494.

[72] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3d human pose and shape from a single color image," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[73] W. Xu, A. Chatterjee, M. Zollhoefer, H. Rhodin, D. Mehta, H.-P. Seidel, and C. Theobalt, "Monoperfcap: Human performance capture from monocular video," *ACM Transactions on Graphics*, 2018. [Online]. Available: http://gvv.mpi-inf.mpg.de/projects/wxu/MonoPerfCap

[74] M. Habermann, W. Xu, M. Zollhöfer, G. Pons-Moll, and C. Theobalt, "Livecap: Real-time human performance capture from monocular video," *ACM Trans. Graph.*, vol. 38, no. 2, Mar. 2019. [Online]. Available: https://doi.org/10.1145/3311970

[75] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[76] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016.

[77] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, arXiv:1411.1784. [Online]. Available: https://arxiv.org/abs/1411.1784

[78] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017, pp. 5967–5976.

[79] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[80] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[81] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *ICLR*, 2018.

[82] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *CVPR*, 2018.

[83] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *ICCV*, 2017, pp. 1520–1529.

[84] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017, pp. 2242–2251. [Online]. Available: https://junyanz.github.io/CycleGAN/

[85] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *ICCV*, Oct. 2017, pp. 2868–2876.

[86] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *NIPS*, 2017.

[87] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Computer Vision and Pattern Recognition (CVPR), 2018*, 2018.

[88] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-gan: Unsupervised video retargeting," in *ECCV*, 2018.

[89] R. Zayer, C. Rossl, and H.-P. Seidel, "Discrete tensorial quasi-harmonic maps," in *Shape Modeling and Applications*. IEEE, 2005, pp. 276–285.

[90] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[91] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *Conference on Computer Vision and Pattern Recognition (CVRP) 2018*, 2018.

[92] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," in *arXiv preprint arXiv:1812.08008*, 2018.