Creating and Reenacting Controllable 3D Humans with Differentiable Rendering

Thiago L. Gomes[1] Thiago M. Coutinho[1] Rafael Azevedo[1]
Renato Martins[2] Erickson R. Nascimento[1]
[1]Universidade Federal de Minas Gerais (UFMG), Brazil [2] Universit Bourgogne Franche-Comt, France
{thiagoluange,thiagocoutinho,rafaelvieira}@dcc.ufmg.br,
renato.martins@u-bourgogne.fr, erickson@dcc.ufmg.br

Abstract

This paper proposes a new end-to-end neural rendering architecture to transfer appearance and reenact human actors. Our method leverages a carefully designed graph convolutional network (GCN) to model the human body manifold structure, jointly with differentiable rendering, to synthesize new videos of people in different contexts from where they were initially recorded. Unlike recent appearance transferring methods, our approach can reconstruct a fully controllable 3D texture-mapped model of a person, while taking into account the manifold structure from body shape and texture appearance in the view synthesis. Specifically, our approach models mesh deformations with a three-stage GCN trained in a self-supervised manner on rendered silhouettes of the human body. It also infers texture appearance with a convolutional network in the texture domain, which is trained in an adversarial regime to reconstruct human texture from rendered images of actors in different poses. Experiments on different videos show that our method successfully infers specific body deformations and avoid creating texture artifacts while achieving the best values for appearance in terms of Structural Similarity (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), Mean Squared Error (MSE), and Frchet Video Distance (FVD). By taking advantages of both differentiable rendering and the 3D parametric model, our method is fully controllable, which allows controlling the human synthesis from both pose and rendering parameters. The source code is available at https://www.verlab.dcc.ufmg.br/retargeting-motion/wacv2022.

1. Introduction

The research in computer vision and graphics communities has made great strides forward in the last two decades, with milestones being passed from semantic segmentation [30], tridimensional reconstruction [31, 38, 39] to synthesis of human motion [5, 20], realistic images [54] and videos [44]. As more vision and graphics methods are integrated into new approaches such as differentiable rendering, more systems will be able to achieve high accuracy and quality in different tasks, in particular, generative methods for synthesizing videos with plausible human reenactment.

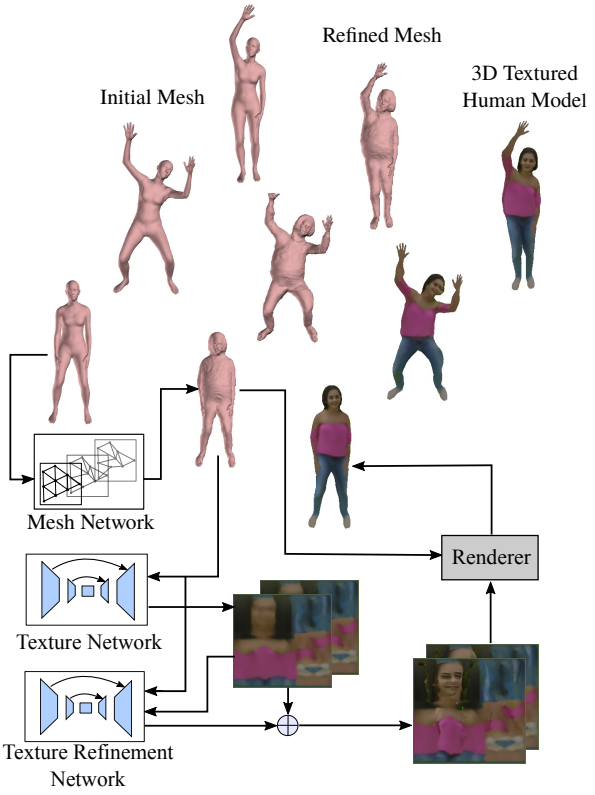

Figure 1: **3D texture-mapped human synthesis.** Our method receives a frame of a person, extracts her/his mesh (left side) and outputs a refined 3D shape and appearance representations of people (right side).

Over the past few years, a remarkable performance of the Generative Adversarial Networks (GANs) [10] has shed a new light for the problem of synthesizing faithful real-world images of humans. Although GAN-based approaches have been achieving high quality results in generating videos [2, 47, 22] and images [14, 12, 54, 44, 4]

of people, in general, they suffer with the high variability in the poses, unseen images from viewpoints not present in the training data, and are often limited to reason directly on the projected 2D image domain. Image-based rendering techniques [8, 9], for their turn, are effective solutions to create 3D texture-mapped models of people, being capable to synthesize images from any arbitrary viewpoint without using large number of images. On the other hand, image-based rendering methods require a painstaking design of the algorithm and are not capable of improving the visual quality of the synthesized images by using more data when available.

In this paper, we take a step towards combining learning and image-based rendering approaches in a new end-to-end architecture that synthesizes human avatars capturing both body geometry and texture details. The proposed architecture comprises a graph convolution network (GCN) operating over the mesh with differentiable rendering to achieve high-quality results in image generation of humans. Our approach receives as input a frame of a person, estimates a generic mesh in the desired pose and outputs a representation of their shape and appearance, as shown in Figure 1. The method provides a final 3D representation that is compatible with traditional graphic pipelines. Specifically, our architecture estimates a refined mesh and a detailed texture map to properly represent the person's shape and appearance for a given input frame.

While 2D image-to-image translation methods are capable of generating plausible images, many applications such as Virtual Reality (VR) and Augmented Reality (AR) [6, 3, 30] require a fully 3D representation of the person. Additionally, the view-dependence hinders the creation of new configurations of scenes where the avatar can be included. Although view interpolation can be applied to estimate a transition between a pair of camera poses, it may create artifacts and unrealistic results when adding the virtual avatar into the new scene using unseen camera poses. Video-based rendering systems are greatly benefited through the use of realistic 3D texture-mapped models, which make possible the inclusion of virtual avatars using unrestricted camera poses and the automatic modification and re-arrangement of video footage. In addition to being able to render human avatars from different viewpoints, the 3D shape also allows synthesizing new images under different illumination conditions. We experimentally showed the capabilities of our new architecture to estimate realistic 3D texture-mapped models of humans. Experiments on a variety of videos show that our method excels several state-of-the-art methods achieving the best values for appearance in terms of Structural Similarity (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), Mean Squared Error (MSE), and Frchet Video Distance (FVD).

Contributions. The main contributions of this paper are threefold: i) a novel formulation for transferring appearance and reenacting human actors that produces a fully 3D representation of the person; ii) a graph convolutional architecture for mesh generation that leverages the human body structure information and keeps vertex consistency, which results in a refined human mesh model; iii) a new architecture that takes advantages of both differentiable rendering and the 3D parametric model and provides a fully controllable human model, *i.e*., the user can control the human pose and rendering parameters.

2. Related Work

Human animation by neural rendering. Recently, we witnessed the explosion of neural rendering approaches to animate and synthesize images of people on unseen poses [44, 4, 22]. According to Tewari *et al*. [41], neural rendering is a deep image or video generation approach that enables explicit or implicit control of scene properties. The main difference among these approaches is how the control signal is provided to the network. Lassner *et al*. [18] proposed a GAN called ClothNet. ClothNet produces random people with similar pose and shape in different clothing styles using as a control signal a silhouette image. In a similar context, Esser *et al*. [4] used a conditional U-Net to synthesize new 2D human views based on estimated edges and body joint locations. Chan *et al*. [2] applied an adversarial training to map a 2D source pose to the appearance of a target subject. Wang *et al*. [44] presented a general video-to-video synthesis framework based on conditional GANs, where a 2D dense UV mapping to body surface (DensePose [37]) is used as a control signal. Similarly, Neverova *et al*. [32] and Sarkar *et al*. [40] rely on DensePose as input to synthesize new views.

These image-to-image translation approaches only allow for implicit control by way of representative samples, *i.e*., they can copy the scene parameters from a reference image/video, but not manipulate these parameters explicitly. To enable the control of the position, rotation, and body pose of a person in a target image/video, Liu *et al*. [22] proposed to use a medium-quality controllable 3D template model of people. In the same line, Liu *et al*. [24] proposed a 3D body mesh recovery module based on a parametric statistical human body model SMPL [25], which disentangles human body into joint rotations and shape. Wu *et al*. [46] produced photorealistic free-view-video from multi-view dynamic human captures. Although these methods are capable of generating plausible 2D images, they cannot generate a controllable 3D models of people, unlike our approach, which is a desired feature in many tasks such as in rendering engines and games or virtual and augmented reality contexts [6, 3, 30].

3D human pose and mesh reconstruction. Substantial advances have been made in recent years for human pose

and 3D model estimation from still images. Bogo *et al*. [1] proposed the SMPLify method, a fully automated approach for estimating 3D body shape and pose from 2D joints in images. SMPLify uses a CNN to estimate 2D joint locations and then optimize the re-projection errors of an SMPL body model [25] to these 2D joints. Similarly, Kanazawa *et al*. [13] used unpaired 2D keypoint annotations and 3D scans to train an end-to-end network to regress the 3D mesh parameters and the camera pose. Kolotouros *et al*. [17] proposed SPIN, an hybrid approach combining the ideas of optimization-based method from [1] and regression deep network [13] to design an efficient method less sensitive to the optimization initialization, while keeping accuracy from the optimization-based method.

Despite remarkable results in pose estimation, these methods are limited to estimate coarse quality generic meshes and textureless characters. Gomes *et al*. [8, 9] proposed an image-based rendering technique to create 3D textured models of people synthesized from arbitrary viewpoints. However, their method is not end-to-end and it is not capable of improving the visual quality of the synthesized images by using information of all available data in the training step. Lazova *et al*. [19] automatically predict a full 3D textured avatar, including geometry and 3D segmentation layout for further generation control; however, their method cannot predict fine details and complex texture patterns. Methods like PiFu [38, 39], for their turn, are limited to static 3D models per frame. Their mesh represents a rigid body and the users cannot drive the human body to new poses, *i.e*., it is not possible to make animations where the model raise his/her hands since the arms and hands are not distinguished from the whole mesh.

Graph networks (GCN) and adversarial learning. GCNs recently emerged as a powerful representation for learning from data lying on manifolds beyond n-dimensional Euclidean vector spaces. They have been widely adopted to represent 3D point clouds such as PointNet [35] or Mesh R-CNN [7], and notably, to model the human body structure with state-of-the-art results in tasks such as human action recognition [50], pose estimation [53, 43] and human motion synthesis [49, 5, 48]. Often these GCNs have been combined and trained in adversarial learning schemes, as in human motion [5], and pose estimation [13]. Our work leverages these features from GCNs and adversarial training to estimate 3D texture-mapped human models.

Differentiable rendering. Differentiable renderers (DR) are operators allowing the gradients of 3D objects to be calculated and propagated through images while training neural networks. As stated in [15], DR connects 2D and 3D processing methods and allows neural networks to optimize 3D entities while operating on 2D projections. Loper and Black [26] introduced an approximate differentiable render which generates derivatives from projected pixels to the 3D parameters. Kato *et al*. [11] approximated the backward gradient of rasterization with a hand-crafted function. Liu *et al*. [23] proposed a formulation of the rendering process as an aggregation function fusing the probabilistic contributions of all mesh triangles with respect to the rendered pixels. Niemeyer *et al*. [33] represented surfaces as 3D occupancy fields and used a numerical method to find the surface intersection for each ray, then they calculate the gradients using implicit differentiation. Mildenhall *et al*. [28] encoded a 3D point and associated view direction on a ray using periodic activation functions, then they applied classic volume rendering techniques to project the output colors and densities into an image, which is naturally differentiable. More recently, techniques [21, 34] based on neural radiance field (NeRF) learning [29] are being proposed to synthesize novel views of human geometry and appearance. While these methods achieved high-quality results, they generally require multi-view data collected with calibrated cameras and have high computational cost, notably during the inference/test time.

In this paper, we propose a carefully designed architecture for human neural rendering, leveraging the new possibilities offered by differentiable rendering techniques [26, 23, 36, 52]. We present an end-to-end learning method that i) does not require data capture with calibrated systems, ii) is computationally efficient in test time, and iii) leverages DR and adversarial training to improve the estimation capabilities of fully controllable realistic 3D texture-mapped models of humans.

3. Methodology

In order to generate deformable 3D texture-mapped human models, our end-to-end architecture has three main components to be trained during the rendering. The first component models local deformations on the human 3D body shape extracted from the images using a three-stage GCN. In the second component, a CNN is trained to estimate the human appearance map. Similar to the GCN, the CNN is trained in a self-supervised regime using the gradient signals from the differentiable renderer. Finally, the third component comprises an adversarial regularization in the human appearance texture domain to ensure the reconstruction of photo-realistic images of people.

All frames of a person are used to train the mesh and the texture networks. The frames provide different viewpoints of the person to build his/her texture map and to estimate the mesh deformation according to the motion. We apply a stage-wise training strategy. First, we train the mesh model using the silhouettes, which produce a texture-less human model. In the sequence, we freeze the mesh network and train a global texture model. Then, we freeze both the mesh model and the global texture model and learn the texture refinement generator and the discriminators parameters. In

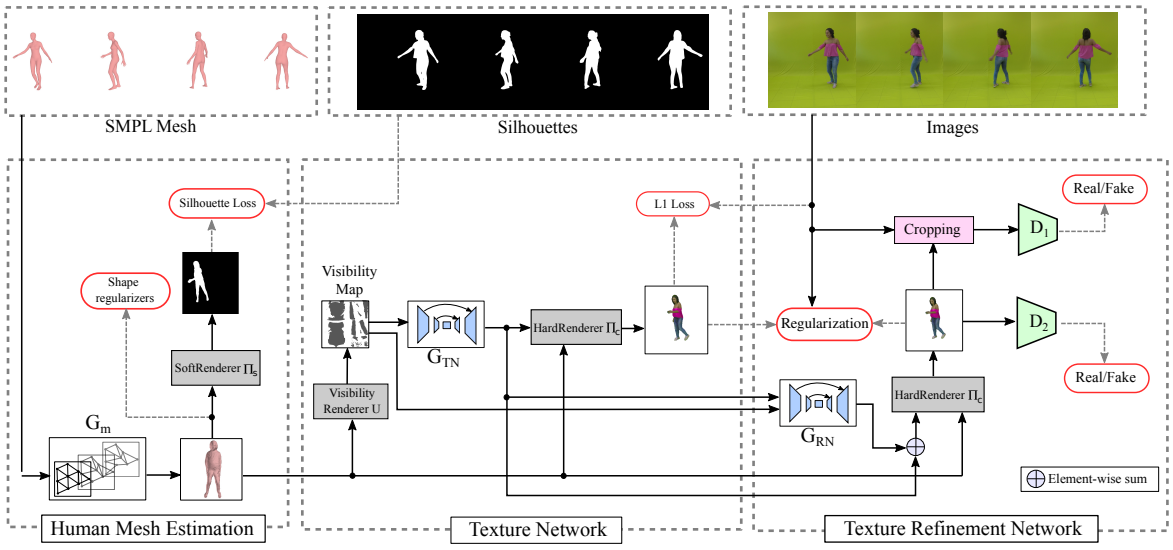

Figure 2: **Human synthesis architecture.** Our architecture has three main networks: **a)** the *Human Mesh Estimation* that comprises a three-stage GCN and learns to fit and deform the mesh based on rendered silhouettes and shape regularizers; **b)** the *Texture Network*, a CNN that is trained conditioned on the visibility map generated from the deformed mesh to create a coarse texture by rendering and optimizing on the l_1 norm; **c)** the *Texture Refinement Network*, a second CNN that is conditioned on the visibility map and the coarse texture to generate the detailed texture map of the person.

the inference time, we feed our architecture with generic meshes parameterized by the SMPL model, and then to create a refined mesh and a detailed texture map to properly represent the person's shape and appearance. Figure 2 outlines these components and their relations during the training phase and Figure 1 shows these components during the inference phase.

3.1. Human Mesh Estimation

Shape and pose representation. We adopt a shape representation that explores the global and local information of the human body. We encode the global information using the SMPL model parametrization [25], which is composed of a learned human shape distribution $\mathcal{M}$, 24 3D joint angles $\boldsymbol{\theta} \in \mathbb{R}^{72}$ (defining 3D rotations of the skeleton joint tree), and shape coefficients $\boldsymbol{\beta} \in \mathbb{R}^{10}$ that model the proportions and dimensions of the human body. We used SPIN [17] to regress the SMPL parameters due to its trade-off between accuracy and efficiency. While the global information provided by SMPL parametrization enables global control of human poses, they do not encode the fine details of the human body geometry (local information). We then characterize the local information by adding a set of offsets to the mesh produced by the SMPL parametrization from the GCN mesh network.

Mesh Refinement Network. After computing the global human shape and pose information $P = \text{SMPL}(\boldsymbol{\theta}, \boldsymbol{\beta})$, we model local deformations on the mesh with the GCN Mesh Network G_m. Since the SMPL parametrization only provides a coarse 3D shape, and it cannot accurately model fine structures like clothes, we feed the mesh network with the initial SMPL mesh to refine its vertex positions with a sequence of refinement stages. The network is designed to learn the set of offsets to the SMPL mesh to increase the realism of the generated views. Drawing inspiration from the network architecture of [7], our refinement network is composed of three blocks with six graph convolutional layers with intermediate features of size 128. Each block in our mesh network performs four operations:

- *Vertex normal computation.* This operation computes the normal surface vector for each vertex as being the weighted sum of the normals of faces containing the vertex, where the face areas are used as the weights. The resulting normal is assigned as the node feature f_i for the vertex v_i in the GCN.
- *Graph convolution.* This convolutional layer propagates information along mesh edges using a aggregation strategy. Similar to [7], given the input vertex feature f_i, the layer updates the feature as $f_i' = \text{ReLU}(W_0 f_i + \sum_{j \in \mathcal{N}} W_1 f_j)$, where W_0 and W_1 are learned weighting matrices, and $\mathcal{N}(i)$ gives the i-th vertexs neighbors in the mesh.
- *Vertex refinement.* To improve the quality of the vertex position estimation, this operation computes vertex offsets as $u_i = \tanh(W[f_i'; f_i])$. W is a learned weighting matrix.

- *Vertex refinement clamping.* To avoid strong deformations (large $||u_i||_2$), we constraint and bound the position update of each vertex v_i as $v_i' = v_i + \min(\max(u_i, -K(v_i)), K(v_i))$, where $K(v)$ is the 3D update bounds allowed to the vertex v, depending on the body part it belongs to. Each body part, *e.g.*, face, footprints, hands, head, torso, arms, feet, *etc*., have predefined bound thresholds. This operation ensures that the offsets do not exceed the threshold defined to that body part, and that the refinement of the mesh geometry do not affect the body's topology.

Loss function. For learning the mesh refinement, our model exploits two differentiable renderers that emulate the image formation process. Techniques such as [23, 36] enable us to invert such renderers and take the advantage of the learning paradigm to infer shape and texture information from the 2D images.

During the training, the designed loss minimizes the differences between the image silhouette extracted from the input real image I_s and the image silhouette $\hat{I}_s$ of the human body obtained by rendering the deformed 3D human model M into the image by *SoftRenderer*, a differentiable renderer $\Pi_s(M)$. *SoftRenderer* is a differentiable that synthesises the silhouette of the actor. We can define the loss of the Mesh Network G_m as:

$$\mathcal{L}_m = \lambda_{gl}\mathcal{L}_{gl} + \lambda_{gn}\mathcal{L}_{gn} + \mathcal{L}_s, \tag{1}$$

where $\mathcal{L}_{gl}$ and $\mathcal{L}_{gn}$ regularize the Laplacian and the normals consistency of the mesh respectively [36], λ_{gl} and λ_{gn} are the weights for the geometrical regularizers, and

$$\mathcal{L}_s = 1 - \frac{\left\|\hat{I}_s \otimes I_s\right\|_1}{\left\|(\hat{I}_s + I_s) - \hat{I}_s \otimes I_s\right\|_1}, \tag{2}$$

is the silhouette loss proposed by Liu *et al*. [23], where $\hat{I}_s = \Pi_s(M)$, $M = G_m(P)$ is the refined body mesh model, and $\otimes$ is the element-wise product.

3.2. Human Texture Generation

We represent the appearance of the human model as a texture map in a fixed UV space that is applied to the refined mesh, in our case, the SMPL mesh with offsets. Our full pipeline for texture generation is depicted in Figure 2-b-c and consists of two stages: a coarse texture generation and then texture refinement. In an initial stage, given the refined 3D meshes of the actor M, the Texture Network G_{TN} learns to render the appearance of the actor in the image. This texture is also further used to condition and regularize the Texture Refinement Network G_{RN} in the second stage.

Texture Network. We start estimating a coarse texture map with a U-Net architecture [12]. The input of the network is a visibility map x_v and it outputs the texture map x_p. The visibility map indicates which parts in the texture map must be generated to produce a realistic appearance for the refined mesh. The visibility map is extracted by a parametric function $x_v = U(M)$ that maps points of refined mesh M with positive dot product between the normal vector and the camera direction vector to a point x_v in the texture space. We implement U as a render of the 2D UV-map considering only faces with positive dot product between the normal vector and the camera direction vector. Thus, the network can learn to synthesize texture maps on demand focusing on the important parts for each viewpoint.

Figure 2-b shows a schematic representation of the Texture Network training. The *HardRenderer* $\Pi_c(M, x_p)$, represents the colored differentiable renderer that computes the output coarse textured image $\hat{I}$, of model M and texture map x_p. In our case, $\hat{I} = \Pi_c(M, G_{TN}(U(M))$. Conversely to the *SoftRenderer*, the differentiable *HardRenderer* is used to propagate the detailed human texture appearance information (color). Specifically, we train the Texture Network to learn to generate a coarse texture by imposing the l_1 norm in the person's body region of the color rendered image as:

$$\mathcal{L}_{pt} = \left\|(\hat{I} - I) \otimes B\right\|_1 / \|B\|_1, \tag{3}$$

where I is the real image in the training set and B is the union of the visibility masks and real body regions.

Texture refinement. We further improve the coarse texture to represent finer details. For that, we design the Texture Refinement Network G_{RN} to condition the generation of a new texture map from the coarse texture, on a coherent output of the Texture Network G_{TN} and the visibility map.

In our adversarial training, the Texture Refinement Network acts as the generator network G_{RN} and engages in a minimax game against two task-specific discriminators: the Face Discriminator D_1 and Image Discriminator D_2. The generator is trained to synthesize texture maps in order to fool the discriminators which must discern between "real" images and "fake" images, where "fake" images are produced by the neural renderer using the 3D texture-mapped model estimated by the Mesh and Texture Networks. While the discriminator D_1 sees only the face region, the Image Discriminator D_2 sees the whole image.

These three networks are trained simultaneously and drive each other to improve their inference capabilities, as illustrated in Figure 2-c. The Texture Refinement Network learns to synthesize a more detailed texture map to deceive the discriminators which in turn learn differences between generated outputs and ground truth data. The total loss function for the generator and discriminators for the rendering is then composed of three terms:

$$\begin{aligned} &\min_{G_{RN}}(\max_{D_1}\mathcal{L}_{GAN}(G_{RN}, D_1) + \\ &\max_{D_2}\mathcal{L}_{GAN}(G_{RN}, D_2) + \mathcal{L}_r(G_{RN})), \end{aligned} \tag{4}$$

where both $\mathcal{L}_{GAN}$ terms address the discriminators and $\mathcal{L}_r$ is a regularization loss to reduce the effects from outlier poses. Each adversarial loss is designed as follows:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_y[\log D(y)] + \mathbb{E}_{x_v, x_p}[\log(1 - D(\Pi_c(M, G(x_v, x_p))))], \quad (5)$$

where x_v is the visibility map, x_p is the output of the Texture Network, M is the refined mesh, and y is the corresponding segmented real image $I \otimes B$.

Finally, to reduce the effects of wrong poses, which causes mismatches between the rendered actor silhouette and silhouette of the real actor, we also add a regularization loss to prevent the GAN to apply the color of the background into the human texture. The first term of the regularization loss acts as a reconstruction of the pixels by imposing the l_1 norm in the person's body region and the second term enforces eventual misaligned regions to stay close to the coarse texture:

$$\mathcal{L}_r = \alpha_1 \left\| (I - \hat{I}^{RN}) \otimes B \right\|_1 / \|B\|_1 + \alpha_2 \left\| (\hat{I}^{TN} - \hat{I}^{RN}) \otimes C \right\|_1 / \|C\|_1, \quad (6)$$

where α_1 and α_2 are the weights, $\hat{I}^{TN}$ is the rendered image using the coarse texture, $\hat{I}^{RN}$ is the rendered image using the refined texture, and C is the misaligned regions without the face region, *i.e.*, the image region where the predicted silhouette and estimated silhouette are different.

4. Experiments and Results

Datasets and baselines. For the training of both texture models and the mesh human deformation network we considered four-minute videos provided by [9], where the actors perform random movements, allowing the model to get different views from the person in the scene. We use the SMPL model parameters calculated by SPIN [17] and the silhouette image segmented by MODNet [16] for each frame of the video. In the evaluation/test time of our 3D human rendering approach, and to provide comparisons to related methods, we conducted experiments with publicly available videos used by Chan *et al*. [2], Liu *et al*. [24], and Gomes *et al*. [9] as the evaluation sequences.

We compare our method against five recent methods including V-Unet [4], Vid2Vid [44], EBDN [2], Retarget [9] and the Impersonator [24]. V-Unet is a famous method of image-to-image translation that uses conditional variational autoencoders to generate images based on a 2D skeleton and an image from the target actor. The approach Retarget is an image-based rendering method based on image rendering designed to perform human retargeting. Impersonator, Vid2Vid, and EBDN are generative adversarial models trained to perform human neural rendering.

Metrics and evaluation protocol. We adopted complementary metrics to evaluate the quality of the approaches to asset different aspects of the generated images such as structure coherence, luminance, contrast, perceptual similarity [51], temporal, and spatial coherence. The metrics used to perform quantitative analysis are SSIM [45], LPIPS [51], Mean Square Error (MSE), and Frchet Video Distance (FVD) [42]. Following the protocol proposed by [9], we executed all the methods in the motion sequences and transfered them to the same background. This protocol allows us to generate comparisons with the ground truth and compute the metrics for all the generated images with their respective real peers. Then, we group the values generated by the metrics in two ways: Motion Types and Actors. In the first one, for each metric, we calculate the average values of all actors making a motion sequence (*e.g.*, "spinning"), while in the second one, we calculate the average values of all movements performed by a given actor. This grouping allows us to analyze the capability of the methods to render the same actor performing various movements with different levels of difficulty and also to compare their capacity to render actors with different morphology performing the same movement.

Implementation details. We trained our body mesh refinement network for 20 epochs with batch size of 4. We used AdamW [27] with parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$, weight decay = 1×10^{-2}, and learning rate of 1×10^{-4} with a linear decay routine to zero starting from the middle of the training. We empirically set λ_1 and λ_2 to 1.0 and 0.5, respectively. In the Vertex refinement clamping component, we defined the set of thresholds as follows: $K \in \{$face = 0.0; footprints = 0.0; hands = 0.0; head = 0.04; torso = 0.06; arms = 0.02; forearms = 0.04; thighs = 0.04; calves = 0.03; feet = 0.02$\}$ meters. All the training and inference were performed in a single Titan XP (12 GB), where the GCN mesh model and the human texture networks took around 6 and 20 hours per actor, respectively. The inference time takes 92 ms per frame (90 ms in the GCN model deformation and 2 ms in the texture networks).

Due to remarkable performance of pix2pix [12] in synthesizing photo-realistic images, we build our Texture Network upon its architecture. The optimizers for the texture models were configured as the same as the Mesh Network, except for the learning rates. The learning rate for the whole body and face discriminators, the global texture and refinement texture generators were set as 2×10^{-5}, 2×10^{-5}, 2×10^{-3}, and 2×10^{-4}, respectively. The parameters of the texture reconstruction was set to $\alpha_1 = 100$ and the regularization as $\alpha_2 = 100$. We observed that smaller values led to inconsistency in the final texture. For the training regime, we used 40 epochs with batch size 8. The global texture model was trained separately from the other models for 2,000 steps, then we freeze the model, the texture

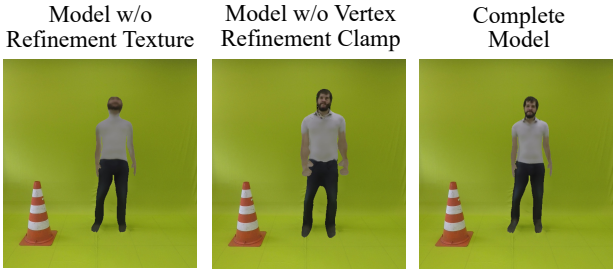

Figure 3: **Ablation study.** a) Results of the texture training without the refinement stage; b) Model without the Vertex Refinement Clamp layer. We observe an excessive growth of the mesh without update thresholds. The texture produced lacks details and even could not preserve the actor's face; c) shows the results for our complete model.

Table 1: **Ablation study**. SSIM, LPIPS, MSE, and FVD comparison by motion types. Best in bold.

Method	Metrics			
	SSIM[1]	LPIPS[2]	MSE[2]	FVD[2]
Texture Refinement Removal	**0.869**	0.136	262.39	795.15
Vertex Refinement Clamping	0.866	0.142	288.04	829.60
Complete Model	0.868	**0.134**	**259.79**	**769.54**
	[1]*Higher is better*		[2]*Lower is better*	

refinement generator and the discriminators were trained.

Ablation Study. We evaluate the contributions of different parts of the method to the overall view synthesis performance. We investigated the benefits from the Vertex refinement clamping component in the Mesh Refinement Network (MRN) and the use of adversarial training in the texture generation. For the first experiment, we removed the vertex refinement thresholds, letting the mesh grow loosely. All other steps of texture training were maintained. Table 1 shows that the performance dropped drastically when compared to our original model. A qualitative analysis of the results in Figure 3-b demonstrates that removing the Vertex refinement clamping component led to strong wrong deformations in the hands and feet, *i.e.*, the regions with higher pose estimation errors.

In the adversarial training analysis, we maintained the original Mesh Refinement Network and removed the Texture Refinement Network and its discriminators, training only the Global Texture Network using Equation 3. Figure 3-a shows the texture quality of the models trained with and without the adversarial regime. After removing the GAN the model could not generate textures with fine details, producing blurry results. This result is also reported in the metrics of Table 1, where we show the average values calculated from all motion sequences in the test data in which the model without GAN is outperformed in all results besides SSIM. This result is coherent, since SSIM is based on low-level image features, and blurred textures can lead to higher SSIM scores.

4.1. Results

Quantitative comparison with state of the art. We performed the neural rendering for actors with different body shapes, gender, clothing styles, and sizes for all considered video sequences. The video sequences used in the actors' animations contained motions with different levels of difficulty, which aims to test the generalization capabilities of the methods in unseen data. Table 2 shows the performance for each method considering all motion and actors types in the dataset. We can see that our method achieves superior peformance as compared to the methods in most of the motion sequences and actor types considering the SSIM, LPIPS, MSE, and FVD metrics. These results indicate that our method is capable of deforming the mesh according to the shape of the given actors and then, rendering a texture optimized to fit the morphology of the person in the scene. Furthermore, our training methodology, that considers multiple views from the actor and the shape parameters, allows the generation of consistent rendering with less visual artifacts when the character is performing challenging movements, such as bending or rotating.

Qualitative visual analysis. The visual inspection of synthesized actors also concur with the quantitative analysis. Figure 4 shows the best frames for each movement using four actors in the dataset. Our model and Retarget [9] are the only approaches capable of keeping the body scale of the authors along all scenes, while the other methods failed, in particular in the movements *shake hands* and *walk*. Besides generating coherent poses, our technique also generated more realistic textures in comparison to the other methods. Comparing the results of the movements *jump* and *spinning*, one can visualize some details as the shadow of the shirt sleeve of the actor and the shirt collar, respectively. Figure 5 illustrates a task of transferring the appearance in videos with different scenes and camera-to-actor translations and camera intrinsics. These results also demonstrate our capability of generating detailed face and body texture, producing a good fit to synthesize views of actors observed from different camera setups.

5. Conclusions

In this paper we proposed a method that produces a fully 3D controllable representation of people from images. Our key idea is leveraging differentiable rendering, GCNs, and adversarial training to improve the capability of estimating realistic 3D texture-mapped models of humans. Taking advantages of both differentiable rendering and the 3D parametric model, our method is fully controllable, which allows controlling the human pose and rendering parameters.

Table 2: **Comparison with state-of-the-art human neural rendering**. SSIM, LPIPS, MSE, and FVD comparison by motion and actors types. Best in bold.

Metric	Method	Motion type								Actor type			
		jump	walk	spinning	shake hands	cone	fusion dance	pull down	box	S1	S2	S3	S4
SSIM[1]	EBDN [2]	0.878	0.880	0.855	0.859	0.878	0.820	0.857	0.858	0.867	0.898	0.844	0.834
	Imper [24]	0.877	0.880	0.852	0.859	0.877	0.816	0.855	0.856	0.867	0.896	0.842	0.831
	Retarget [9]	0.881	0.885	0.855	0.860	0.879	0.820	0.861	0.869	0.872	0.902	0.846	0.834
	Vid2Vid [44]	0.880	0.884	0.856	0.858	0.878	0.821	0.859	0.866	0.868	0.901	0.848	0.835
	V-Unet [4]	0.870	0.871	0.843	0.847	0.862	0.797	0.847	0.857	0.855	0.886	0.830	0.826
	Ours	**0.884**	**0.890**	**0.860**	**0.865**	**0.885**	**0.824**	**0.866**	**0.873**	**0.876**	**0.908**	**0.852**	**0.838**
LPIPS[2]	EBDN [2]	0.141	0.122	0.139	0.138	0.143	0.215	0.151	0.170	0.159	0.145	0.147	0.159
	Imper [24]	0.151	0.134	0.151	0.151	0.155	0.239	0.168	0.184	0.161	0.165	0.170	0.171
	Retarget [9]	**0.125**	0.099	**0.130**	0.131	0.128	**0.206**	**0.131**	**0.127**	**0.133**	0.129	**0.133**	**0.143**
	Vid2Vid [44]	0.131	0.105	0.126	0.136	0.133	0.203	0.142	0.133	0.148	0.131	0.129	0.147
	V-Unet [4]	0.147	0.132	0.157	0.161	0.174	0.243	0.166	0.158	0.184	0.160	0.166	0.158
	Ours	0.127	**0.097**	**0.130**	**0.130**	**0.124**	**0.206**	0.132	**0.127**	0.136	**0.124**	0.134	**0.143**
MSE[2]	EBDN [2]	306.92	269.58	312.69	312.12	266.17	463.79	384.23	361.57	324.40	314.10	331.83	368.19
	Imper [24]	313.43	275.22	344.94	314.92	267.39	504.00	404.79	377.16	277.98	328.07	358.30	436.57
	Retarget [9]	237.16	178.57	286.86	270.25	237.86	434.64	301.66	245.86	**243.95**	260.14	294.88	297.46
	Vid2Vid [44]	257.33	206.42	286.18	332.09	253.04	452.77	349.28	288.54	312.40	274.80	269.81	356.26
	V-Unet [4]	295.04	269.59	354.33	377.02	328.68	559.75	417.00	346.13	381.77	344.71	362.63	384.66
	Ours	**231.20**	**153.23**	**278.75**	**254.38**	**218.76**	**418.58**	**286.02**	**237.42**	247.10	**236.49**	**276.17**	**279.42**
FVD[2]	EBDN [2]	887.56	273.00	918.94	**423.08**	725.49	952.22	1,113.46	853.26	791.98	751.45	**560.27**	826.71
	Imper [24]	1,770.31	656.07	1,531.64	1,266.14	1,051.42	1,322.72	1,440.94	1,719.55	1,270.41	1,092.82	1,395.81	1,214.64
	Retarget [9]	1,119.50	330.91	**674.99**	478.93	767.68	791.01	**988.35**	**760.33**	**715.00**	**653.30**	720.49	**515.61**
	Vid2Vid [44]	**879.94**	266.6	1,085.49	396.31	790.79	997.42	997.96	1,069.85	778.53	719.80	762.46	574.08
	V-Unet [4]	1,491.63	845.44	1,721.81	1,257.20	1,415.24	1,712.93	2,437.98	1,816.94	2,239.14	1,352.10	1,856.78	1,108.34
	Ours	1,114.43	**233.81**	1019.83	542.24	**614.88**	**746.22**	1010.24	874.69	881.49	697.68	718.21	551.06

[1] *Higher is better* [2] *Lower is better*

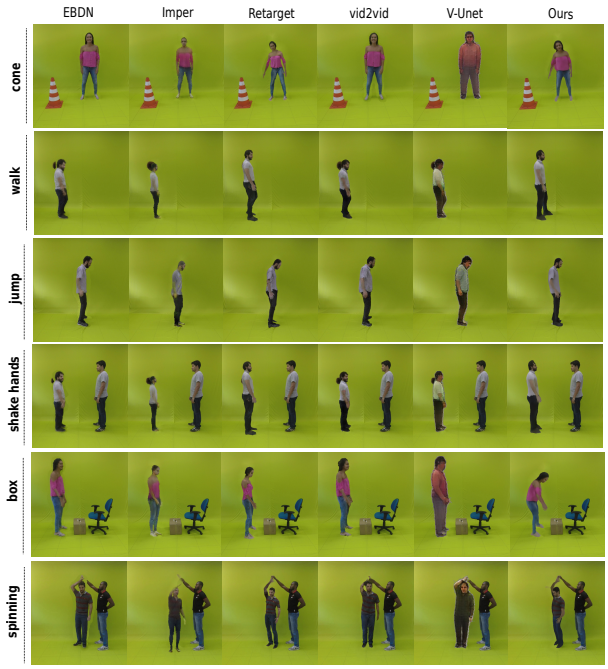

Figure 4: **Qualitative comparison.** Movements with four different actors are represented in the rows. The results for each competitor are represented in the columns.

Furthermore, we have introduced a graph convolutional architecture for mesh generation that refines the human body structure information, which results in a more accurate human mesh model. The experiments show that the proposed method has superior quality compared to recent neural rendering techniques in different scenarios, besides having several advantages in terms of control and ability to generalize to furthest views.

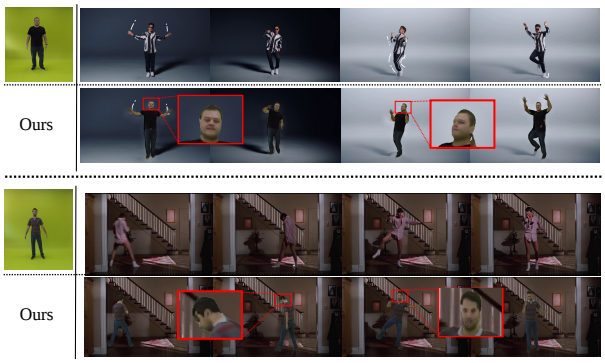

Figure 5: **Human appearance transfer and animation with different scenes and camera setups**. The first line of each scene illustrates the original frames of the source video. On the second line is the transferred appearance of the animated virtual actor using our proposed method. The red squares highlight the face generation quality.

Acknowledgments. The authors would like to thank CAPES, CNPq, FAPEMIG, and Petrobras for funding different parts of this work. R. Martins was also supported by the French National Research Agency through grants ANR MOBIDEEP (ANR-17-CE33-0011), ANR CLARA (ANR-18-CE33-0004) and by the French Conseil Rgional de Bourgogne-Franche-Comt. We also thank NVIDIA Corporation for the donation of a Titan XP GPU used in this research.

References

[1] 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 *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[2] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *IEEE/CVF International Conference on Computer Vision*, pages 5933–5942, 2019.

[3] Long Chen, Thomas W. Day, Wen Tang, and Nigel W. John. Recent developments and future challenges in medical mixed reality. In Wolfgang Broll, Holger Regenbrecht, and J. Edward Swan II, editors, *2017 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2017, Nantes, France, October 9-13, 2017*, pages 123–135. IEEE Computer Society, 2017.

[4] Patrick Esser, Ekaterina Sutter, and Bjrn Ommer. A variational u-net for conditional appearance and shape generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[5] João Pedro Moreira Ferreira, Thiago M. Coutinho, Thiago L. Gomes, José F. Neto, Rafael Azevedo, Renato Martins, and Erickson R. Nascimento. Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. *Computer & Graphics*, 94:11–21, 2021.

[6] Abir Gallala, Bassem Hichri, and Peter Plapper. Survey: The evolution of the usage of augmented reality in industry 4.0. *IOP Conference Series: Materials Science and Engineering*, 521:012017, may 2019.

[7] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *IEEE Conference on Computer Vision and Pattern Recognition*, October 2019.

[8] Thiago L. Gomes, Renato Martins, João P. K. Ferreira, and Erickson R. Nascimento. Do as I do: Transferring human motion and appearance between monocular videos with spatial and temporal constraints. In *IEEE Winter Conference on Applications of Computer Vision*, pages 3355–3364. IEEE, 2020.

[9] Thiago L. Gomes, Renato Martins, João Pedro Moreira Ferreira, Rafael Azevedo, Guilherme Torres, and Erickson R. Nascimento. A shape-aware retargeting approach to transfer human motion and appearance in monocular videos. *International Journal of Computer Vision*, 129(7):2057–2075, 2021.

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

[11] Yoshitaka Ushiku Hiroharu Kato and Tatsuya Harada. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[14] 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.

[15] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv preprint*, 2020.

[16] Zhanghan Ke, Kaican Li, Yurou Zhou, Qiuhua Wu, Xiangyu Mao, Qiong Yan, and Rynson W.H. Lau. Is a green screen really necessary for real-time portrait matting? *ArXiv*, abs/2011.11961, 2020.

[17] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *IEEE International Conference on Computer Vision*, 2019.

[18] Christoph Lassner, Gerard Pons-Moll, and Peter V. Gehler. A generative model for people in clothing. In *IEEE International Conference on Computer Vision*, 2017.

[19] Verica Lazova, Eldar Insafutdinov, and Gerard Pons-Moll. 360-degree textures of people in clothing from a single image. In *2019 International Conference on 3D Vision, 3DV 2019, Québec City, QC, Canada, September 16-19, 2019*, pages 643–653. IEEE, 2019.

[20] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[21] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *CoRR*, abs/2106.02019, 2021.

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

[23] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *IEEE International Conference on Computer Vision*, Oct 2019.

[24] Wen Liu, Zhixin Piao, Min Jie, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping GAN: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *IEEE International Conference on Computer Vision*, 2019.

[25] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics*, 2015.

[26] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, 2014.

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

[29] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421, 2020.

[30] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. 2020.

[31] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[32] Natalia Neverova, Riza Alp Güler, and Iasonas Kokkinos. Dense pose transfer. In *European Conference on Computer Vision*, 2018.

[33] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

[34] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[36] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

[37] Iasonas Kokkinos Riza Alp Güler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. *arXiv*, 2018.

[38] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[39] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[40] Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. Neural re-rendering of humans from a single image. In *European Conference on Computer Vision*, 2020.

[41] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Niebner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhofer. State of the art on neural rendering. *Computer Graphics Forum*, 39(2):701–727, 2020.

[42] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019.

[43] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. In *European Conference on Computer Vision*, 2020.

[44] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NIPS*, 2018.

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

[46] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[47] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. Reenactgan: Learning to reenact faces via boundary transfer. In *European Conference on Computer Vision*, 2018.

[48] Zijian Huang Qifeng Chen Xuanchi Ren, Haoran Li. Self-supervised dance video synthesis conditioned on music. In *ACM MM*, 2020.

[49] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[50] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2018.

[51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[52] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image GANs meet differentiable rendering for inverse graphics and interpretable 3D neural rendering. *arXiv preprint arXiv:2010.09125*, 2020.

[53] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, 2017.