Novel-View Human Action Synthesis

Mohamed Ilyes Lakhal[1], Davide Boscaini[2], Fabio Poiesi[2], Oswald Lanz[2], and Andrea Cavallaro[1]

[1] Centre for Intelligent Sensing, Queen Mary University of London, UK
[2] Technologies of Vision, Fondazione Bruno Kessler, Italy
{m.i.lakhal,a.cavallaro}@qmul.ac.uk, {dboscaini,poiesi,lanz}@fbk.eu

Abstract. Novel-View Human Action Synthesis aims to synthesize the movement of a body from a virtual viewpoint, given a video from a real viewpoint. We present a novel 3D reasoning to synthesize the target viewpoint. We first estimate the 3D mesh of the target body and transfer the rough textures from the 2D images to the mesh. As this transfer may generate sparse textures on the mesh due to frame resolution or occlusions. We produce a semi-dense textured mesh by propagating the transferred textures both locally, within local geodesic neighborhoods, and globally, across symmetric semantic parts. Next, we introduce a context-based generator to learn how to correct and complete the residual appearance information. This allows the network to independently focus on learning the foreground and background synthesis tasks. We validate the proposed solution on the public NTU RGB+D dataset. The code and resources are available at https://bit.ly/36u3h4K.

1 Introduction

Novel-view human action synthesis is the problem of reproducing a person performing an action from a virtual viewpoint [1]. The ability to synthesize of one or more novel viewpoints of an action is attractive for extended reality [2], action recognition [3] and free-viewpoint video [4].

Recent works [5,6,7,8] have shown the ability to synthesize high-quality images, but with limiting assumptions on the input data. SiCloPe [5] takes a frontal input image and uses canonical views to reconstruct the 3D mesh through supervision. However, using a ground-truth mesh from real images is not a realistic assumption. Similarly, PIFu [6] predicts a dense 3D occupancy field using multiple input views. This method expects high-resolution ground-truth mesh and a neutral background which hinder generalization to real-world scenarios when backgrounds are cluttered. The method proposed in [9] creates an animated version of an image containing a person in the center. An initial mesh is first estimated and then corrected. However, the mesh construction part is computationally costly and incompatible with the extension of the model to videos. Furthermore, the texture filling is based on heuristics or requires human intervention. If multiple real views of the same scene are available, the rendering of an arbitrary virtual view can be successfully addressed [10,11]. For example,

the method proposed in Bansal *et al.* [11] combines the information available from multiple camera views to reconstruct the geometry of a static scene. Then, a neural network based model is used to compose the dynamics on top of the static scene. However, with only a single video (view) as input the problem becomes much more challenging and largely unexplored. To the best of our knowledge, VDNet [1] is the only previous work addressing it.

Thanks to the rapid development on human mesh recovery [12,13,14,15,16], we can obtain 3D representations from images or videos. Our approach consists of a two-stage pipeline. In the first stage we exploit a novel 3D reasoning to produce a sparse initialization for the virtual view. In the second stage we introduce Geometric texture Transfer Network (GTNet), a context-based generator that aims to correct and complete such initial guess by learning the residual appearance information. For each frame captured from the real view we estimate the 3D mesh of the human actor using [14] where the parameters of the Skinned Multi-Person Linear (SMPL) model [17] are learned to morph a canonical 3D model of the human body to fit the 2D projection of the human actor pose and shape. Given such 3D model, we transfer the appearance information from the 2D video to the 3D mesh. This results in a sparse texture on the 3D mesh because of occlusions. We propose to compute the missing information by exploiting the knowledge of the 3D model both at a local and global scale. Locally, missing values within a geodesic neighborhood are computed by interpolating the input sparse texture. More globally, if a part of the 3D model (e.g. an arm or a leg) lacks texture information but its symmetric counterpart contains it, we propagate it. The texture on the mesh (in 3D) obtained in this way is then projected (rendered) on the novel view (in 2D). The estimated 3D model thus acts as a proxy to transfer appearance information from the input (real) view to the target (virtual) view. The design of our approach is inspired by pixel warping methods [18,19] that create realistic human images from existing frames (or views). Differently from VDNet [1], we exploit the geometric properties of the input prior to facilitate the transfer to the target view. Unlike motion-transfer methods (*e.g.* [20]), we learn the geometry and the appearance of a novel (virtual) view.

2 Related work

Methods for novel-view synthesis that focus on humans can be based on computer graphics, learning, or combining 3D mesh representations and learning. These methods are discussed in this section jointly with a discussion on the importance of the modality used to synthesize the novel-view.

Novel-View Image Synthesis. Graphics based methods [21,6,22] rely on the abundance of ground-truth data to achieve high quality synthesis. For example [23,22,24] use image or sequence of frames to learn the displacement of clothing on top of the SMPL [17] model. Differently, the methods in [6,5] use high quality human mesh representations from the Renderpeople dataset[3]. These

[3] https://renderpeople.com/, accessed September 2020

representations enable the model to achieve high quality results, but fail to generalize in uncontrolled setups and need a few viewpoints, which may be hard to obtain, to perform the synthesis.

Learning (or data-driven) approaches [1,3,25,26] use spatial cues about the human subject to synthesize the target view. A drawback of such approaches is the poor generalization to unseen views and the difficulty to handle occlusions.

A new direction of work considers the use of a 3D model estimated directly from raw images [27,28,29]. Liu *et al.* [28] enforce feature warping of the input view in the network structure to synthesize the novel view.

Video Synthesis. We categorise the methods solving this problem into two classes: unconstrained or constrained synthesis. The first category tries to learn the distribution of the data during training. The video is therefore a sample from the learned distribution [30,31,32]. Since the datasets available are most often a sparse representation of the distribution of the true data, the generated videos generally are limited to few applications. The constrained video synthesis [20,33,34] relies on context (*e.g.* image sequence [20]) or spatial cues (*e.g.* keypoints [33,34]). Applications include action imitation [20] and video prediction [35].

Novel-View Video Synthesis. Recently, Lakhal *et al.* [1] introduced the task of *novel-view video synthesis* which shares the challenges of both the novel view synthesis (*i.e* dealing with occlusion) and video synthesis (*i.e* maintaining temporal consistency across frames). The assumptions are the availability of only one input view and the modalities about the target view which can be either given or computed. Furthermore, the problem is different from the pose-guided human image synthesis [25] where the background synthesis is not taken into account, the pose is not constrained to the view (*i.e.* cannot model the 3D structure of the scene), and these methods fail to maintain temporal consistency. In this paper, we show that by estimating the texture we can approximate the target feature with a simple mapping. Using the proposed context-based architecture, the network can focus on the background synthesis. We exploit the 3D mesh information and use the input-target view association as guidance in the novel-view synthesis process. Also, we explicitly handle the temporal consistency of the synthesized frames. Furthermore, we handle self occlusions using visible information and transfer it to neighboring occluded parts.

Prior modalities. Deep learning based methods made progress on estimating accurate modalities (*e.g.* depth and 2D/3D keypoints) from object priors (*e.g.* human). This includes human pose estimation [36], human part segmentation [37], or human mesh recovery [14,12]. The performance of a neural network-based generator for novel-view synthesis relies heavily on the modalities derived from the prior used about the target view (we only consider human priors). Early works rely on 2D keypoints of the human body joints [38,25,39,40,41,42,43,44].

The skeleton indicates the spatial location of the person on the target view. The network then has to learn how to extract and transfer appearance information from the input view to generate the target image. A segmentation map could be considered as another modality [45,46]. DensePose [47] maps the pixels of

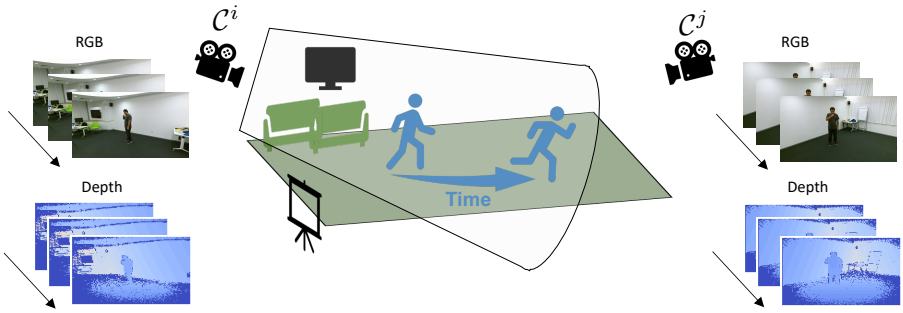

Fig. 1: Given a video of a person performing an action recorded from an input view $\mathcal{C}^i$, we synthesize how it would appear from a target (or virtual) view $\mathcal{C}^j$. Each view can be captured with a set of modalities (e.g. RGB, depth, skeleton).

an RGB image of human to a 3D surface and is used in [46,48]. Li *et al.* [29] proposed to represent the pose as a rendered 3D body mesh.

3 Method

3.1 Problem definition

Let $\mathcal{C} = \{\mathcal{C}^i\}_{i=1}^{V}$ be V static cameras (views) placed at different positions in a scene. Each view $i \in \{1, \ldots, V\}$ is represented as a sequence of RGB images $x_t^i \in \mathbb{R}^{w \times h \times 3}$ of width w and height h pixels and indexed by the timestep $t = 1, \ldots, T$. The sequence $x^i = \{x_t^i\}_{t=1}^{T}$ is an instance of the scene captured from the input view camera $\mathcal{C}^i$. Each view contains M different modalities $p^i = \{p_1^i, \ldots, p_M^i\}$ *e.g.* depth and skeleton (see Fig. 1). Each modality has to at least spatially localise the person in the scene (*i.e.,* foreground). The modalities of the virtual view are computed by transforming p^i with the information in both $\mathcal{C}^i$ and $\mathcal{C}^j$.

The aim of novel-view video synthesis is to reconstruct the RGB sequence x^j of a scene from an input view to a target view. Given a parametric function $G_{i \to j}$ called *generator* we have: $x^j = G_{i \to j}(x^i, p^j)$.

3.2 3D human body prior

We model the foreground using a 3D mesh assuming that the video stream contains only humans. We use the estimated 3D mesh as a proxy to transfer the appearance information from the input view to the novel view. In this section, we show that we can compute other modalities of the target view by fully exploiting the one-to-one correspondence between vertices of the 3D mesh (Fig. 2).

Foreground prior. We model the foreground (*i.e.* the target subject) using a geometric approach that exploits one-to-one correspondences between the vertices

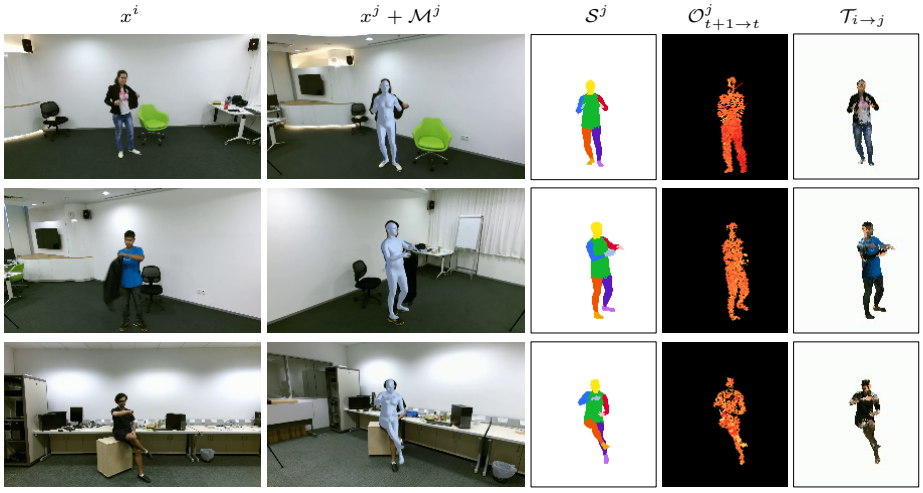

Fig. 2: Target view modalities obtained with the 3D human body prior extracted from NTU-RGB+D dataset [49]: (a) input view; (b) target view with the rendered mesh; (c) segmentation map; (d) foreground motion; (e) texture transfer.

of the 3D meshes from the real and virtual views. As in [14], we model the human body using the SMPL model [17]. A set of three connected vertices defines a face on the 3D mesh. The SMPL model is composed of $N_f = 13776$ faces that are uniquely identified by a face map $\mathcal{F}$. Given the camera $\mathcal{C}^i \in \mathcal{C}$ and a projection function (*e.g.* a renderer [50]) the mesh in rendered on the camera $\mathcal{C}^i$ producing a data structure $F^i \in \mathbb{R}^{w\times h}$ (*i.e.* projecting the faces onto the image plane of the camera $\mathcal{C}^i$) and the rendering of the 3D body mesh $\mathcal{M}^i$ (Fig 2(b)).

Human part segmentation. Let us decompose the human body representation into B parts (*e.g.* head and arms). Since for the SMPL the map $\mathcal{F}$ has a fix set of faces, we cluster it into parts[4] such that: $\mathcal{F} = \{\mathcal{F}_b\}_{b=1}^{B}$. Therefore, each face $f \in F^i$ can belong to any of the B classes or to the background.

Foreground motion. We exploit the data-structure $\{F^i_t\}_{t=1}^{T}$ to extract the foreground motion information. Specifically, because mesh vertices are uniquely identified over time, we can compute their displacement 3D vector. This 3D vector can be projected on the image to obtain the foreground motion flow. As in [35] we use a backward motion flow to warp the frame for each time step to help the foreground synthesis. Given a face $f \in F^i_{t+1}$ (resp. F^i_t) at pixel location (u_x, u_y) (resp. $(u_x^{'}, u_y^{'})$), let $\mathcal{O}^i_{t+1\to t}$ be the motion vector at (u_x, u_y), which is computed as $(u_x - u_x^{'}, u_y - u_y^{'})$.

Texture transfer. The structure $F^i_t \in \mathbb{R}^{w\times h}$ is the projection of the 3D human mesh of the person in x^i_t at time step t onto the image plane of camera $\mathcal{C}^i$. A key observation to make is that we can exploit the association between F^i_t and x^i_t in order to estimate a rough foreground on the image plane of the camera

[4] In practice, we manually annotate each of the N_f face into a unique body-part label.

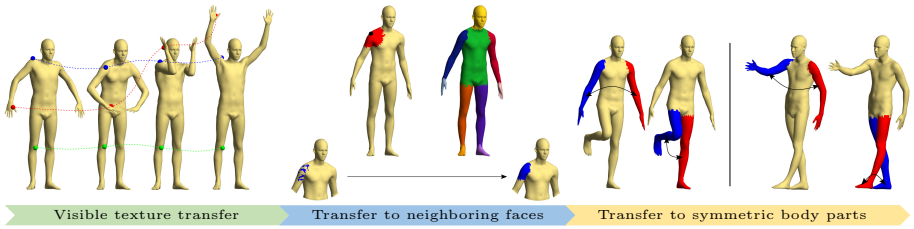

Fig. 3: Illustration of the proposed texture transfer. **Step I:** for every visible face index from the input view mesh, we accumulate its RGB pixel value over time. Then, we copy the pixel values to the target view mesh. **Step II:** we transfer the closest visible face with respect to a distance measure. **Step III:** we transfer texture across intrinsic symmetries, *i.e.* from the blue regions to the red regions. Intrinsic symmetries are independent on the pose of the subject.

$\mathcal{C}^j$. The proposed Symmetric Texture Transfer extends this idea to improve the target foreground appearance through three steps (Fig. 3). The first step consists of tracking each visible face in F_t^i over time. If a face $f \in F_t^i$ is at position (u_x, u_y) we copy the pixel value of x_t^i. The face-pixel association is then stored in a hashmap where the keys are the face number and the values are the pixels. If at time $t+k$ the face f is detected we add it and at time step T we keep the median of the detected pixels. The second step transfers pixels from the hashmap to an image indexed by F^j. Specifically, given a face $f \in \mathcal{F}$, we rank the neighboring faces as a function of the distance $\mathbf{dist} : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$ defined on the surface manifold of a template mesh. Because the Euclidean distance is not a suitable metric to measure distances of vertices on a deformable surface, we use the geodesic distance [51] that is invariant to intrinsic deformation of the mesh. The computation of the geodesic distances produces the matrix $\mathbf{F} \in \mathbb{R}^{M \times M}$, where the element in the u^{th} row and and v^{th} column is $\mathbf{F}_{uv} = \{\mathbf{dist}(f, f') | f, f' \in \mathcal{F}\}$. Using $\mathbf{F}$ we transfer the texture of the n-nearest neighbor face to the image. The final step uses symmetry between body part in order to transfer occluded pixels (see Supplementary Material).

We use a template gender neutral 3D mesh with a canonical pose to compute the pairwise distance map $\mathbf{F} \in \mathbb{R}^{N_f \times N_f}$ (see Fig. 3(a)). The reason is that for Euclidean distance it is computationally not possible to compute it for each frame of the dataset. Furthermore, computing a geodesic distance is much more computationally expensive than an Euclidean distance. The texture transfer $\mathcal{T}^s_{i \rightarrow j}$ approximates the foreground of the novel-view and is not used as a final prediction.

3.3 Geometric texture Transfer Network (GTNet)

The burden over the network to synthesize the novel view can be mitigated if we exploit the 3D mesh. The texture-transfer $\mathcal{T}^s_{i \rightarrow j}$ provides a good estimate of the foreground of target view x^j. Therefore we consider that $\mathcal{T}^s_{i \rightarrow j}$ is an

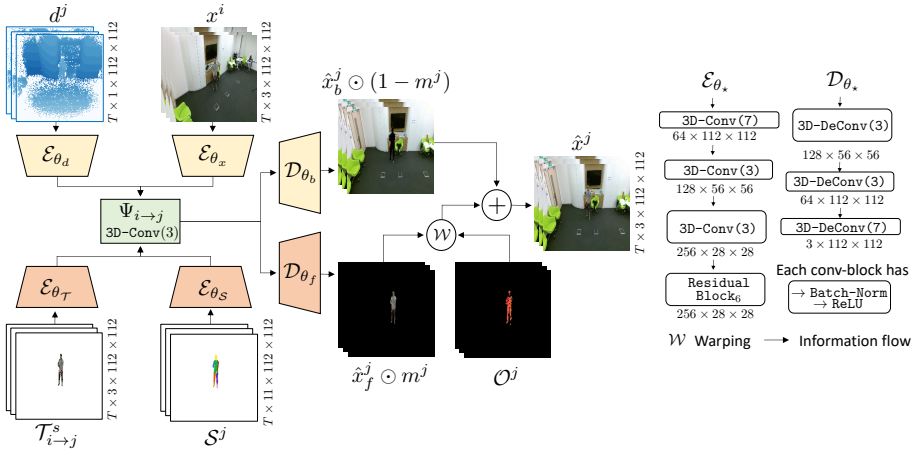

Fig. 4: Architecture of the proposed GTNet model. We encode each modality using a separate encoder to approximate the feature point of the target view with $\Psi_{i \to j}$. We decode the background and the foreground separately. Note that we also enforce explicit temporal modeling using the estimated foreground motion.

informative input to the network and we only need to learn the residual to correct elements like mis-transferred textures or lighting. Since the texture $\mathcal{T}^s_{i \to j}$ is a good approximation of the foreground, we chose a context-based network structure.

Architecture. GTNet jointly learns to synthesize the foreground and background (see Fig 4). The network takes the input view video x^i (resp. depth modality d^j) and encodes it with feature mapping $\mathcal{E}_{\theta_x}$ (resp. $\mathcal{E}_{\theta_d}$). These features constitute the background information. Similarly, we encode the texture transfer $\mathcal{T}^s_{i \to j}$ (resp. segmentation map $\mathcal{S}^s$) to represent the foreground information in the latent space. Now to approximate the target feature ϵ^j using the operator $\Psi_{i \to j}$ we rely on a 3D Convolutional Neural Network layer, this will enforce the temporal consistency on the bottleneck layer. We therefore have the following:

$$\hat{\epsilon}^j \approx \Psi_{i \to j}(\oplus_{k \in \mathcal{I}} \mathcal{E}_{\theta_k}(k)); \mathcal{I} = \{x^i, d^j, \mathcal{S}^j, \mathcal{T}^s_{i \to j}\}, \tag{1}$$

where $\oplus$ is the concatenation operation. As motivated earlier, we separate the synthesis of the foreground and the background using dedicated decoders $\mathcal{E}_{\theta_f}$ and $\mathcal{E}_{\theta_b}$, respectively. The synthesized foreground (resp. background) is obtained as: $\hat{x}^j_f = \mathcal{D}_{\theta_f}(\hat{\epsilon}^j)$ (resp. $\hat{x}^j_b = \mathcal{D}_{\theta_b}(\hat{\epsilon}^j)$). The synthesized video $\hat{x}^j$ is therefore:

$$\hat{x}^j = \hat{x}^j_f \odot m^j + \hat{x}^j_b \odot (1 - m^j), \tag{2}$$

where $\odot$ is the Hadamard product and m^j is the foreground mask obtained by the binarization of F^j.

In order to enforce temporal constraints, we propose to use the foreground motion from the mesh displacement vectors in the synthesized frame to add

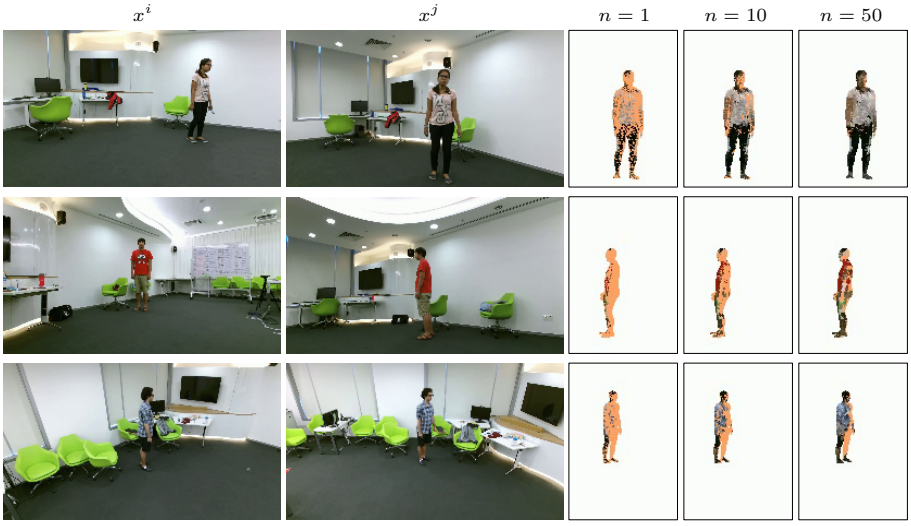

Fig. 5: Comparison of the traditional texture transfer methods (e.g. [28]) and the proposed transfer method with $n \in \{10, 50\}$ nearest neighbor transfer.

motion information. The frame synthesis of view j at time step t is defined as:

$$\hat{x}^j_{f,t} = \begin{cases} \hat{x}^j_{f,t} & \text{if } t = 1 \\ \tilde{x}^j_{f,t} + \zeta.\mathcal{W}(\hat{x}^j_{f,t-1}, O^j_{t+1\to t}) & \text{if } t \in [2..T], \end{cases} \tag{3}$$

where $\mathcal{W}$ is a residual warping function, $\hat{x}^j_{f,t}$ (resp. $O^j_{t+1\to t}$) is the foreground prediction (resp. foreground motion) of the view j. $\tilde{x}^j_{f,t}$ is the initial synthesized frame of the generator and ζ is a controlling factor defined empirically (Tab. 3). We force the model to focus on the residue with respect to the previous time step $t-1$. Note that when training a generator, $\mathcal{W}(\hat{x}^j_{f,t-1}, O^j_{t+1\to t})$ is computed by a forward pass (and freezing the weights). Thus when applying the reconstruction pixel-wise loss, the network would only learn the residual over $\tilde{x}^j_{f,t}$.

Training losses. Instead of the traditional L_1 used in the literature, we employ a Huber loss [52] to penalize the video synthesis produced by the generator. Differently from the L_2, the Huber loss is more robust to outliers and, differently from the L_1 loss, the Huber loss considers the directions of the error magnitude. The reconstruction loss L_r between the generated videos (both foreground and background branch) $\hat{x}^j$ and the ground-truth x^j at t is

$$L_r = \begin{cases} 0.5(\hat{x}^j_t, -x^j_t)^2, & \text{if } |\hat{x}^j_t, -x^j_t| < 1 \\ |\hat{x}^j_t, -x^j_t| - 0.5, & \text{otherwise} \end{cases} \tag{4}$$

To enforce the perceptual quality over the generated videos we use the temporal perceptual loss [1]. This loss extends the so-called perceptual loss [53] by penalising the generated videos on a spatio-temporal feature space using a 3D CNN network

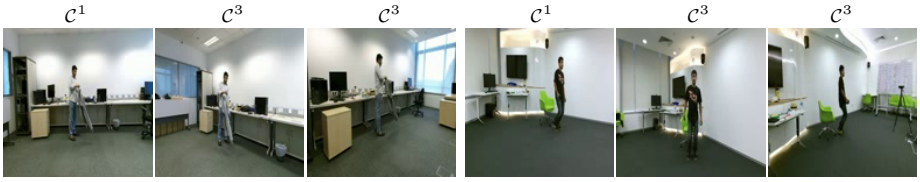

Fig. 6: Sample frames from the NTU RGB+D dataset. The 3 views are captured with cameras placed with horizontal angle of: $-45°, 0°, +45°$.

ϕ (called perceptual network). The temporal perceptual loss is defined as:

$$L_p = \sum_{k=1}^{L} \frac{1}{T_k w_k h_k c_k} \left\| \phi_k(\hat{x}^j) - \phi_k(x^j) \right\|_2, \tag{5}$$

where T_k, w_k, h_k, c_k are the temporal dimension (*i.e. timesteps*), width, height and the number of channel of at the k-th layer of the perceptual network ϕ, respectively. Furthermore, we use adversarial loss [54] in order to add high frequency details in the synthesized frames. Given our generator $G_{i \to j}$ and a discriminator D, the conditional adversarial loss is given as:

$$L_a = \mathbb{E}_{x^i, x^j}\Big[\log(D(x^i, x^j))\Big] + \mathbb{E}_{x^i}\Big[\log(1 - D(x^i, \hat{x}^j))\Big]. \tag{6}$$

The total training loss is given by $L = L_r + \lambda_p L_p + \lambda_a L_a$, with $\lambda_p = \lambda_a = 0.01$.

4 Experiments

This section evaluates the proposed GTNet. Sec. 4.1 describes the training protocol. Sec. 4.2 provides the ablation of each component of the proposed pipeline. Sec. 4.2 compares our method with the state-of-the art VDNet [1].

4.1 Experimental Setup

Dataset. We use NTU RGB+D [49], the only large-scale synchronized multi-view action recognition dataset (see Fig. 6), which consists of of videos captured using three synchronized cameras with two front views and one side view. The dataset contains 80 views with 40 distinct subjects and 60 actions. Following [1], we use the cross-subject split.

Evaluation metrics. We assess the performance using two criteria: (i) the generated video visual quality; (ii) the accuracy of the pose of the individual. For the visual quality, we use Structural Similarity (SSIM), Peak Signal-to-Noise-Ratio (PSNR) [55] (we also report their masked version [25]) and Fréchet Video Distance (FVD) [56]. We use Percentage of Correct Keypoints (PCK) [57] for the pose evalutation.

Implementation Details. To obtain a temporally consistent 3D mesh we combine OpenPose [36] with [14]. We used an NVIDIA Tesla V100 16GB RAM GPU to train our model. We use Adam optimizer [58] with $(\alpha_1, \alpha_2) = (0.5, 0.999)$ and a learning rate of $2{\cdot}10^{-5}$.

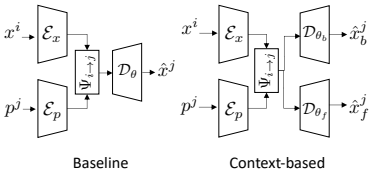

Fig. 7: Model ablation.

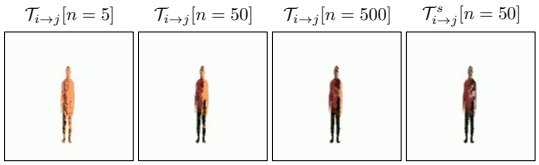

Fig. 8: Texture transfer vs. Symmetric texture transfer (occluded region: orange pixel).

Table 1: Quality of the foreground estimation. Key. M: mask; S.: SSIM; P. PSNR; Euc: Euclidean; []: nearest neighbour value n.

Step.	Notation.	M-S.	M-P.
II (Euc.)	$\mathcal{T}_{i\to j}[500]$	.952	26.85
II	$\mathcal{T}_{i\to j}[500]$	.952	26.86
II, III	$\mathcal{T}^s_{i\to j}[50]$	.953	26.92
I, II, III	$\mathcal{T}^s_{i\to j}[50]$	.954	27.16

Table 2: Baseline ablation. Key. M: mask; S.: SSIM; P. PSNR; Mod. modality; BL: baseline; Hb: Huber.

Model	Mod.	S.	M-S.	P.	M-P.	FVD
BL ($\Psi^{\text{lin}}_{i\to j}$)	$\mathcal{M}^j_{2D}$	.534	.957	17.62	26.13	10.81
BL ($\Psi^{\text{conv}}_{i\to j}$)	$\mathcal{M}^j_{2D}$	.628	.964	18.39	27.73	7.51
	$\mathcal{T}_{i\to j}$	.680	.969	19.83	29.13	6.79
	$\mathcal{T}^s_{i\to j}$	.688	.970	19.85	29.12	6.57
GTNet (L_1)	$\mathcal{T}_{i\to j}$	.693	.977	20.26	31.81	6.81
GTNet (Hb)	$\mathcal{T}_{i\to j}$	.709	.976	20.63	31.70	6.44

Table 3: Sensitivity analysis of the warping factor with $T = 24$.

ζ	0	.1	.01	.001
SSIM	.624	**.635**	.623	.612
PSNR	18.35	**18.41**	18.17	18.19

Table 4: Synthesis performance using different model weight using $T = 8$.

	VDNet [1]			GTNet	
#layers	6 (3D)	6	18	6	6 (3D)
#params	112.74M	34.70M	77.20M	12.35M	99.20M
SSIM	.821	.698	.711	.709	**.823**
M-SSIM	.972	N/A	N/A	.976	**.981**

4.2 Ablation Studies

We provide a detailed evaluation of each component of the proposed pipeline. Unless otherwise stated, we use a 2D-ResNet$_6$ [59] for the ablation.

Texture Transfer. We show the result of the contribution of each step in the texture transfer:

– **Step II (Euclidean): F** computed with Euclidean pairwise distance.
– **Step II: F** computed with geodesic pairwise distance.
– **Step II + III:** Symmetric texture transfer.
– **Step I + II + III:** Symmetric texture transfer with temporal context.

Results from Tab. 1 show that the symmetric texture transfer helps to better estimate the foreground with only a kernel size of $n = 50$ instead of 500 without the symmetry. Adding the temporal context for the hashmap construction improves further. Fig. 7 shows a challenging case of texture transfer between views with and without the body symmetry transfer.

Baseline Models. GTNet has a separate decoder for the foreground and the background. Therefore, we chose a generator with a single decoder as the baseline (see Fig. 6). GTNet is key to refine the texture transfer. To verify this we consider three variants of the input to the network: $\mathcal{M}^j_{2D}, \mathcal{T}_{i\to j}$, and $\mathcal{T}^s_{i\to j}$.

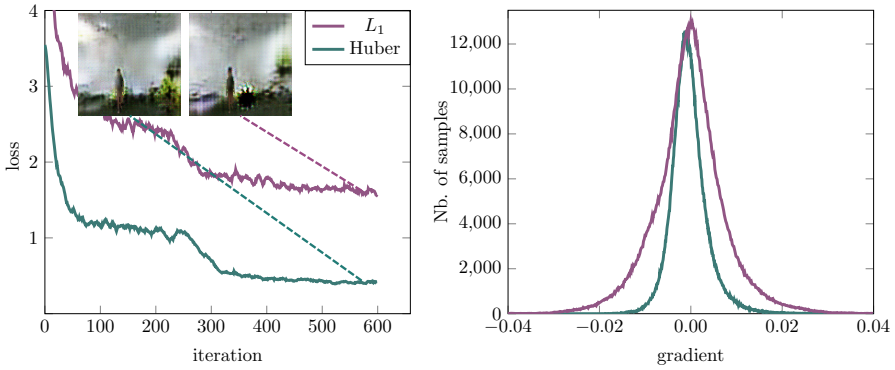

Fig. 9: Training curves analysis of GTNet.

We propose two variants of $\Psi_{i\to j}$ to assess the feature approximation:

– **Linear:** $\Psi^{\texttt{lin}}_{i\to j}(\epsilon^i, \pi^j) = \mathbf{W}_{ij}.\epsilon^i + \mathbf{W}_{jj}.\pi^j + b_j$ s.t $\mathbf{W}_{ij}, \mathbf{W}_{jj} \in \mathbb{R}^{m\times m}, b_j \in \mathbb{R}^m$.
– **Convolution:** $\Psi^{\texttt{conv}}_{i\to j}(\epsilon^i, \pi^j) = \texttt{conv}_{3\times 3}(\epsilon^i \oplus \pi^j)$.

The operator $\Psi_{i\to j}$ estimates the feature vector of the target view. The linear version assumes a linearity between the input-view feature and the target-view modalities, whereas, the convolution applies a concatenation operation followed by a convolution operation which refers to a complex mapping (*i.e.* non-linear) between the inputs. Results from Tab. 2 suggest that better feature approximation leads to better view synthesis. A linear mapping cannot approximate well the target feature ϵ^j. The convolution is the default feature approximation for GTNet.

$\mathcal{M}^j_{2D}$ is the straightforward modality to use for the synthesis using 3D mesh. Tab. 2 shows that Baseline($\mathcal{M}^j_{2D}$) underperforms compared to Baseline($\mathcal{T}_{i\to j}$). This suggests that the texture transfer helps the network to refine the foreground. Having a better estimate (*i.e.* $\mathcal{T}^s_{i\to j}$) improves further. With the Baseline the network has to focus on both synthesizing the foreground and background. Using the context based approach in GTNet helps the model to focus on the background synthesis and to refine only the foreground. The other conclusion is the texture $\mathcal{T}^s_{i\to j}$ approximates better the foreground.

Hyperparameters. We analyse the model performance while validating the effect of the loss, warping factor and model weights.

Using a Huber reconstruction loss in GTNet(Huber) improves the quality of the synthesized videos (see Tab. 2). We investigate this further by plotting the

Table 5: Models performance using a pose estimator [36].

Model	$\mathbf{L}_2$	**PCK** [57]		
		0.20	0.05	0.01
VDNet [1]	4.37	99.3	92.4	51.2
Baseline	4.06	99.4	**93.6**	55.3
GTNet	**3.95**	**99.5**	93.0	**57.6**

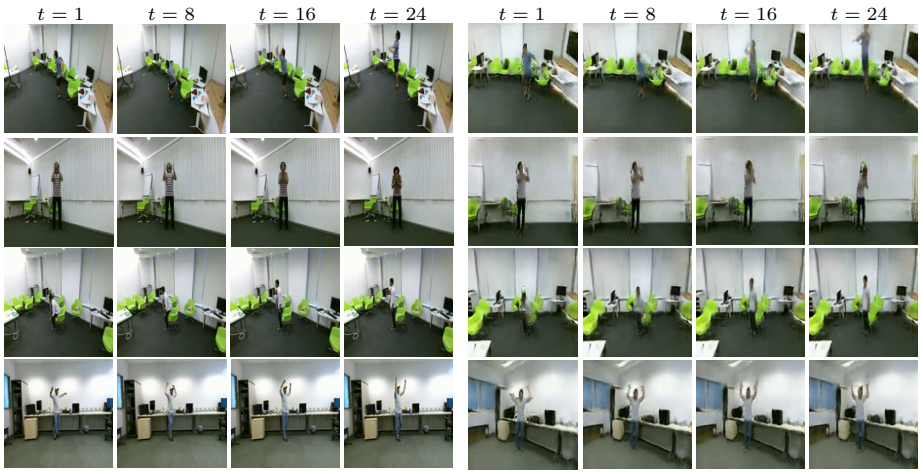

Fig. 10: Sample frames on novel-view synthesis. (left): input view video sequence x^i; (right): synthesized target view $\hat{x}^j$ using the proposed GTNet.

training loss and the histogram of the gradient at the last convolutional kernel of the decoder of GTNet (see Fig. 9). For the gradient we noticed a significantly smaller variance, which we deem to be due to the non-smoothness of the L_1 loss around the origin. By analysing the output of the generator trained with these two losses, we observe that the generator trained with L_1 loss outputs black artifacts during the first epochs which may cause mode collapse [60].

From Tab. 3 we note the improvement using the warping introduced in Eq. 3. This in fact helps the generator to only learn the residual from previous frame $\hat{x}^j_{t-1}$. We therefore keep $\zeta = .1$ as the default value for the warping function.

We report a good foreground synthesis even with a 2D-ResNet compared to VDNet. However, the network could not synthesize well the background. This is because without the temporal context the network will synthesize the background

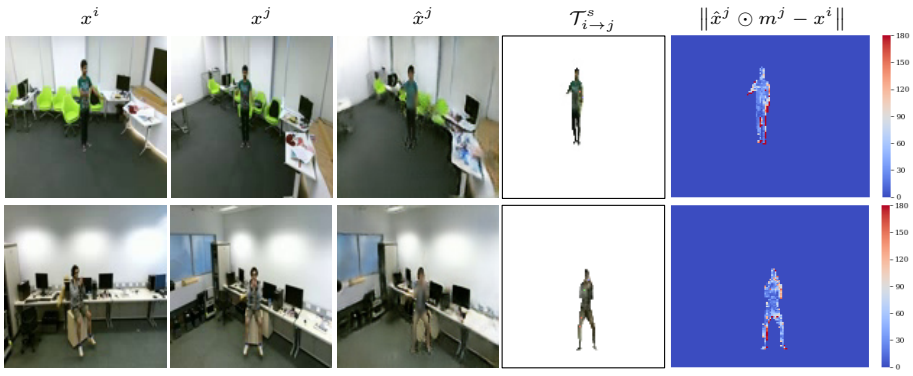

Fig. 11: Visualisation of the learned residual of $\mathcal{T}^s_{i\to j}$ using GTNet.

Table 6: Comparison between the proposed GTNet and VDNet [1] on NTU RGB+D. Key. ζ controlling factor (see Sec. 3.3).

Model	Modality	↑ SSIM	↑ M-SSIM	↑ PSNR	↑ M-PSNR	↓ FVD
VDNet [1]	s^j	.749	.964	20.78	28.27	7.35
	d^j	.794	.970	22.47	29.46	6.60
	$d^j + s^j$	.821	.972	23.18	29.70	5.78
GTNet($\zeta = 0$)	$\mathcal{M}^j_{2D}$	.703	.976	20.16	30.95	6.34
GTNet($\zeta = 0$)	$\mathcal{T}_{i\to j}$	.767	.979	22.03	31.98	5.62
GTNet($\zeta = 0$)	$\mathcal{T}_{i\to j} + \mathcal{S}^j$	.714	.978	20.44	31.90	6.42
GTNet($\zeta = 0$)	$\mathcal{T}_{i\to j} + d^j$	.778	.980	22.96	32.04	**4.32**
GTNet($\zeta = .1$)	$\mathcal{T}_{i\to j} + \mathcal{S}^j + d^j$	.787	.980	22.98	32.25	5.06
GTNet($\zeta = .1$)	$\mathcal{T}^s_{i\to j} + \mathcal{S}^j + d^j$	**.823**	**.981**	**23.81**	**32.50**	4.96

independently for each time step. With the 3D-ResNet we obtain better video synthesis compared to VDNet with many fewer trainable weights.

Comparison. Results from Tab. 6 are reported with 3D-ResNet$_6$. Overall, GTNet significantly improves over all the metrics compared to VDNet [1]. GTNet benefits from the depth d^j along with $\mathcal{T}_{i\to j}$. The model GTNet($\mathcal{T}^s_{i\to j} + \mathcal{S}^j + d^j; \zeta = .1$) produces superior quality results compared to VDNet.

$\mathcal{T}_{i\to j}$ is derived from the skeleton s^j. It is worth noting that GTNet($\mathcal{T}_{i\to j}$) is superior to VDNet(s^j) (see Tab. 6). The proposed GTNet produces temporally consistent videos (FVD scores). Tab. 5 reports the PCK scores of GTNet with the baseline and VDNet. The pose estimator estimates keypoints that are close to the ground-truth with GTNet.

Fig. 11 shows the ability of GTNet in refining the textures. Fig. 10 shows four examples of typical synthesis results using GTNet. The synthesized novel-view videos are sharper and we can clearly distinguish the movement of the subject. Fig. 12 compares three examples of GTNet and VDNet. We can note that indeed the motion is clearly distinct with our model. Note also that thanks to $\mathcal{T}^j_{i\to j}$ the body texture is preserved across the views. In the example of the third row, we can see that GTNet is able to keep the movement of the hand with the object interaction (hat). Fig. 13 shows a qualitative example of the pose estimation. The pose estimator is able to extract keypoints similar to the ones extracted from the ground-truth. This is because GTNet has better foreground synthesis (.981 M-SSIM, 32.50 M-PSNR) compared to VDNet (.972 M-SSIM, 29.70 M-PSNR).

5 Conclusions

We presented a novel approach to synthesize actions as if they were recorded from a novel view by exploiting geometric and appearance information extracted from the real view. Our geometric approach transfers the textures of the visible parts of the human (foreground) from images to a 3D mesh and re-projects them onto the novel, 2D view. Then, we designed a new encoder-decoder network architecture that learns how to synthesize the occluded parts of the foreground

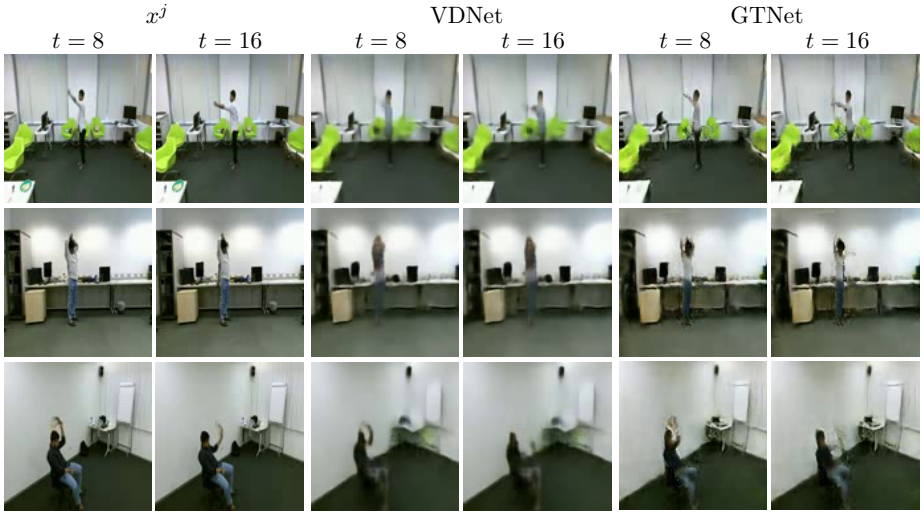

Fig. 12: Sample frames comparing VDNet and GTNet.

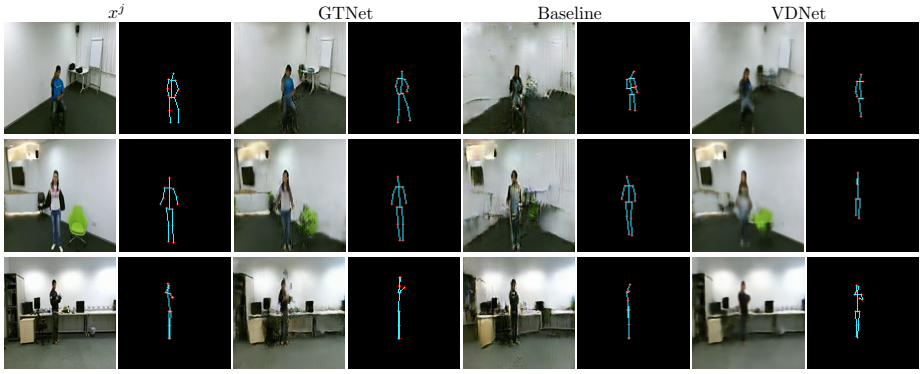

Fig. 13: Comparison of the estimated skeleton.

and that tackles the foreground and background tasks separately to achieve high synthesis fidelity. We obtain state-of-the-art synthesis results on the NTU RGB+D dataset.

Acknowledgements This project acknowledges the use of the ESPRC funded Tier 2 facility, JADE.

References

1. Lakhal, M.I., Lanz, O., Cavallaro, A.: View-LSTM: Novel-view video synthesis through view decomposition. In: Proceedings of the International Conference on Computer Vision (ICCV). (2019) 7576–7586

2. Bertel, T., Campbell, N.D.F., Richardt, C.: MegaParallax: Casual 360°Panoramas with Motion Parallax. IEEE Transactions on Visualization and Computer Graphics **25** (2019) 1828–1835
3. Li, J., Wong, Y., Zhao, Q., Kankanhalli, M.: Unsupervised Learning of View-invariant Action Representations. In: Neural Information Processing Systems (NeurIPS). (2018)
4. Rematas, K., Kemelmacher-Shlizerman, I., Curless, B., Seitz, S.: Soccer On Your Tabletop. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 4738–4747
5. Natsume, R., Saito, S., Huang, Z., Chen, W., Ma, C., Li, H., Morishima, S.: SiCloPe: Silhouette-Based Clothed People. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 4475–4485
6. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In: Proceedings of the International Conference on Computer Vision (ICCV). (2019) 2304–2314
7. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. ACM Transactions on Graphics (TOG) **38** (2019)
8. Thies, J., Zollhöfer, M., Theobalt, C., Stamminger, M., Nießner, M.: Image-guided neural object rendering. In: Proceedings of the International Conference on Learning Representations (ICLR). (2020)
9. Weng, C.Y., Curless, B., Kemelmacher-Shlizerman, I.: Photo Wake-Up: 3D Character Animation From a Single Photo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 5901–5910
10. Mustafa, A., Hilton, A.: Semantically Coherent Co-Segmentation and Reconstruction of Dynamic Scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 5583–5592
11. Bansal, A., Vo, M., Sheikh, Y., Ramanan, D., Narasimhan, S.: 4d visualization of dynamic events from unconstrained multi-view videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2020) 5365–5374
12. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In: Proceedings of the European Conference on Computer Vision (ECCV). (2016) 561–578
13. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-End Recovery of Human Shape and Pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 7122–7131
14. Kanazawa, A., Zhang, J.Y., Felsen, P., Malik, J.: Learning 3D Human Dynamics From Video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 5607–5616
15. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the Loop. In: Proceedings of the International Conference on Computer Vision (ICCV). (2019) 2252–2261
16. Pavlakos, G., Kolotouros, N., Daniilidis, K.: TexturePose: Supervising Human Mesh Estimation With Texture Consistency. In: Proceedings of the International Conference on Computer Vision (ICCV). (2019) 803–812
17. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A Skinned Multi-Person Linear Model. ACM Transactions on Graphics (TOG) **34** (2015) 248:1–248:16

18. Pishchulin, L., Jain, A., Wojek, C., Andriluka, M., Thormälen, T., Schiele, B.: Learning people detection models from few training samples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2011) 1473–1480
19. Xu, F., Liu, Y., Stoll, C., Tompkin, J., Bharaj, G., Dai, Q., Seidel, H.P., Kautz, J., Theobalt, C.: Video-based Characters: Creating New Human Performances from a Multi-view Video Database. ACM Transactions on Graphics (TOG) **30** (2011) 32:1–32:10
20. Siarohin, A., Lathuillre, S., Tulyakov, S., Ricci, E., Sebe, N.: First Order Motion Model for Image Animation. In: Neural Information Processing Systems (NeurIPS). (2019)
21. Chen, X., Song, J., Hilliges, O.: Monocular Neural Image Based Rendering With Continuous View Control. In: Proceedings of the International Conference on Computer Vision (ICCV). (2019) 4089–4099
22. Bhatnagar, B.L., Tiwari, G., Theobalt, C., Pons-Moll, G.: Multi-Garment Net: Learning to Dress 3D People From Images. In: Proceedings of the International Conference on Computer Vision (ICCV). (2019) 5419–5429
23. Alldieck, T., Magnor, M., Bhatnagar, B.L., Theobalt, C., Pons-Moll, G.: Learning to Reconstruct People in Clothing From a Single RGB Camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 1175–1186
24. Alldieck, T., Magnor, M., Xu, W., Theobalt, C., Pons-Moll, G.: Video Based Reconstruction of 3D People Models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 8387–8397
25. Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., Van Gool, L.: Pose Guided Person Image Generation. In: Neural Information Processing Systems (NeurIPS). (2017)
26. Zhao, B., Wu, X., Cheng, Z.Q., Liu, H., Jie, Z., Feng, J.: Multi-View Image Generation from a Single-View. In: Proceedings of the ACM International Conference on Multimedia (ACM-MM). (2018) 383391
27. Zanfir, M., Oneata, E., Popa, A.I., Zanfir, A., Sminchisescu, C.: Human Synthesis and Scene Compositing. In: Proceedings of the National Conference on Artificial Intelligence (AAAI). (2020) 12749–12756
28. Liu, W., Piao, Z., Jie, M., Luo, W., Ma, L., Gao, S.: Liquid Warping GAN: A Unified Framework for Human Motion Imitation, Appearance Transfer and Novel View Synthesis. In: Proceedings of the International Conference on Computer Vision (ICCV). (2019) 5903–5912
29. Li, Y., Huang, C., Loy, C.C.: Dense Intrinsic Appearance Flow for Human Pose Transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 3688–3697
30. Saito, M., Matsumoto, E., Saito, S.: Temporal generative adversarial nets with singular value clipping. In: Proceedings of the International Conference on Computer Vision (ICCV). (2017)
31. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: MoCoGAN: Decomposing motion and content for video generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 1526–1535
32. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Neural Information Processing Systems (NeurIPS). (2016)
33. Yang, C., Wang, Z., Zhu, X., Huang, C., Shi, J., Lin, D.: Pose guided human video generation. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 204–219

34. Wang, T.C., Liu, M.Y., Tao, A., Liu, G., Kautz, J., Catanzaro, B.: Few-shot video-to-video synthesis. In: Neural Information Processing Systems (NeurIPS). (2019)
35. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Flow-Grounded Spatial-Temporal Video Prediction from Still Images. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 609–625
36. Raaj, Y., Idrees, H., Hidalgo, G., Sheikh, Y.: Efficient Online Multi-Person 2D Pose Tracking With Recurrent Spatio-Temporal Affinity Fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 4615–4623
37. Yang, L., Song, Q., Wang, Z., Jiang, M.: Parsing r-cnn for instance-level human analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 364–373
38. Siarohin, A., Sangineto, E., Lathuilière, S., Sebe, N.: Deformable GANs for Pose-Based Human Image Generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 3408–3416
39. Pumarola, A., Agudo, A., Sanfeliu, A., Moreno-Noguer, F.: Unsupervised Person Image Synthesis in Arbitrary Poses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 8620–8628
40. Liqian, M., Qianru, S., Stamatios, G., Luc, V.G., Bernt, S., Mario, F.: Disentangled Person Image Generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 99–108
41. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody Dance Now. In: Proceedings of the International Conference on Computer Vision (ICCV). (2019) 5932–5941
42. Esser, P., Sutter, E., Ommer, B.: A Variational U-Net for Conditional Appearance and Shape Generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 8857–8866
43. Balakrishnan, G., Zhao, A., Dalca, A.V., Durand, F., Guttag, J.: Synthesizing Images of Humans in Unseen Poses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 8340–8348
44. Qian, X., Fu, Y., Xiang, T., Wang, W., Qiu, J., Wu, Y., Jiang, Y.G., Xue, X.: Pose-Normalized Image Generation for Person Re-identification. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 661–678
45. Raj, A., Sangkloy, P., Chang, H., Hays, J., Ceylan, D., Lu, J.: SwapNet: Image Based Garment Transfer. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 679–695
46. Dong, H., Liang, X., Gong, K., Lai, H., Zhu, J., Yin, J.: Soft-Gated Warping-GAN for Pose-Guided Person Image Synthesis. In: Neural Information Processing Systems (NeurIPS). (2018)
47. Alp Gler, R., Neverova, N., Kokkinos, I.: DensePose: Dense Human Pose Estimation in the Wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 7297–7306
48. Neverova, N., Alp Guler, R., Kokkinos, I.: Dense Pose Transfer. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 128–143
49. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 1010–1019
50. Kato, H., Ushiku, Y., Harada, T.: Neural 3D Mesh Renderer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 3907–3916

51. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S.J., Hoppe, H.: Fast exact and approximate geodesics on meshes. ACM Transactions on Graphics (TOG) **24** (2005) 553560
52. Huber, P.J.: Robust Estimation of a Location Parameter. Annals of Mathematical Statistics **35** (1964) 73–101
53. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In: Proceedings of the European Conference on Computer Vision (ECCV). (2016) 694–711
54. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets. In: Neural Information Processing Systems (NeurIPS). (2014) 2672–2680
55. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing **13** (2004) 600–612
56. Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: FVD: A new Metric for Video Generation. In: Proceedings of the International Conference on Learning Representations (ICLR) Workshops. (2019)
57. Yang, Y., Ramanan, D.: Articulated Human Detection with Flexible Mixtures of Parts. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **35** (2013) 2878–2890
58. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: Proceedings of the International Conference on Learning Representations (ICLR). (2015)
59. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In: Proceedings of the International Conference on Computer Vision (ICCV). (2017) 2242–2251
60. Che, T., Li, Y., Jacob, A.P., Bengio, Y., Li, W.: Mode Regularized Generative Adversarial Networks. In: Proceedings of the International Conference on Learning Representations (ICLR). (2017)