TemporalUV: Capturing Loose Clothing with Temporally Coherent UV Coordinates

You Xie[†], Huiqi Mao[‡], Angela Yao[‡], Nils Thuerey[†]
[†]Department of Informatics, Technical University of Munich
[‡]Department of Computer Science, National University of Singapore
{you.xie, nils.thuerey}@tum.de, huiqi.mao@u.nus.edu, ayao@comp.nus.edu.sg

Abstract

We propose a novel approach to generate temporally coherent UV coordinates for loose clothing. Our method is not constrained by human body outlines and can capture loose garments and hair. We implemented a differentiable pipeline to learn UV mapping between a sequence of RGB inputs and textures via UV coordinates. Instead of treating the UV coordinates of each frame separately, our data generation approach connects all UV coordinates via feature matching for temporal stability. Subsequently, a generative model is trained to balance the spatial quality and temporal stability. It is driven by supervised and unsupervised losses in both UV and image spaces. Our experiments show that the trained models output high-quality UV coordinates and generalize to new poses. Once a sequence of UV coordinates has been inferred by our model, it can be used to flexibly synthesize new looks and modified visual styles. Compared to existing methods, our approach reduces the computational workload to animate new outfits by several orders of magnitude.

1. Introduction

In image or video generation tasks [37, 44] that involve people, it is crucial to obtain accurate representations of the 3D human shape and appearance to efficiently generate modified content. In this context, *UV coordinates* are a popular 2D representation that establish dense correspondences between 2D images and 3D surface-based representations of the human body. UV coordinates go beyond skeleton landmarks to encode human pose and shape, and are widely used in image/video editing, augmented reality, and human-computer interaction [12, 14, 15]. In this paper, we tackle video generation of people, with a focus on efficiency and capturing loose clothing. Unlike previous works [33, 40, 42] which use large networks to capture motion and appearance, we train a model to generate temporally coherent UV coordinates. We use a single, fixed texture to store appearance information so that our model can solely focus on learning UV dynamics.

Human body UV coordinates can be derived indirectly from estimates of 3D shape models [6, 19, 26, 29] like SMPL [23]. Alternatively, direct estimation methods like DensePose [2] and UltraPose [43] bypass intermediate 3D models to directly output UV coordinates from a single RGB image. The convenience of direct methods has led to DensePose being widely used in animation and editing applications [25, 27, 28, 49]. Nevertheless, the UV coordinates obtained from SMPL and DensePose approximate only human body silhouettes in tight clothing. They do not capture loose clothing, such as long skirts or wide pants (see comparisons in Figure 6 and 7). In addition, the methods for UV estimation work only on individual images. For video inputs, they are applied frame-by-frame [32, 47] without considering the temporal relationship between frames. As such, the UV coordinates are inconsistent over time, so any re-targeted sequences will shift and jitter.

In this paper, we focus on improving the spatial coverage and temporal coherence of UV coordinates generated from a sequence of 2D images. We target the ability to retain the full body plus clothing silhouette for arbitrary styles of clothing. Our approach is agnostic to the UV source, which we demonstrate via inputs from both DensePose [2] and SMPL model estimates [19]. For temporal coherence, we aim at achieving the point-to-point correspondences among different frames via UV coordinate maps, so that video sequences can be generated with one fixed texture.

A core challenge of learning a model for extended and temporally coherent UV coordinates lies in the lack of data for direct supervision. Hence, we propose a novel learning scheme that combines both supervised and unsupervised components. We first pre-process a sequence of UV coordinates obtained from DensePose or SMPL via spatial extension and temporal stabilization to obtain training data for an initial training stage. We then shift the learning gradually from supervised, with the pre-processed data, to unsu-

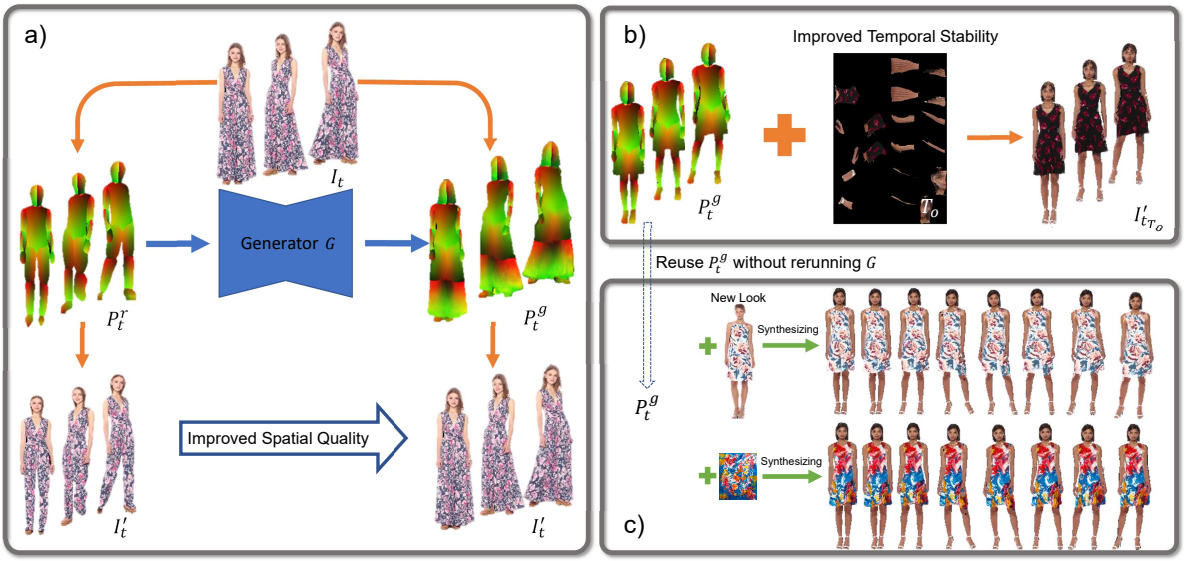


Figure 1. a) Our method generates temporally coherent UV coordinates that capture loose clothing from off-the-shelf human pose UV estimates such as SMPL and DensePose [2, 23]. b) Generated UV coordinates allow us to recover entire sequences from a constant texture map. c) Virtual try-on and modifications of the look can be easily achieved with minimal computation via a simple lookup.

pervised, driven by a differentiable UV mapping pipeline between the texture and image space.

Our results demonstrate that using loss terms formulated in both UV and image space are crucial for generating high-quality UV coordinates with temporal coherence. As our generator does not take RGB images as input, the UV coordinates generated from our trained model can be directly paired with different texture maps to generate virtual try-on videos with a very simple lookup step. This is device-independent and orders of magnitude more efficient than other methods, which generate video outputs by evaluating neural networks. To summarize, our main contributions are

- a model-agnostic method to extend UV coordinates to capture the complete appearance of the human body,
- an approach to train neural networks that generate completed and temporally coherent UV coordinates without the need for ground truth, and
- a highly efficient way to generate virtual try-on videos with arbitrary clothing styles and textures.

2. Related work

Pose-guided generation. Pose-guided methods [3, 16, 22, 24, 27, 31, 36] generate images of a person with designated target poses. To achieve realistic and high-quality generations, most methods [16, 22, 27] tend to work with dense targets with 3D shape or surface models. Specifically, these methods rely on UV coordinates generated either from estimated SMPL model parameters [23] or directly via DensePose [2]. Neither the SMPL model nor DensePose is good at dealing with loose clothing, such as dresses. In this paper, we also work with UV coordinates, though our pipeline focuses on *improving* the quality of the raw UV coordinates from SMPL and DensePose to take on loose clothing. Pose-guided video generation, also known as human motion transfer, generate videos based on a sequence of target poses [13, 34, 35, 47]. The appearance information is sourced from either images (image-to-video [34, 35, 46, 47]) or videos (video-to-video [1, 7, 9, 20]).

Image-to-video. An early example is MonkeyNet [34]. While Monkeynet decouples appearance and motion information, it uses keypoints, which is insufficient for high-quality capture of human body or clothing with complex textures. We use DensePose UV coordinates as pose representation to improve this problem.

Closely related to our work is DwNet [47], which also uses DensePose UV coordinates as inputs. DwNet applies an encoder-decoder architecture that warps the human body from source to target poses. However, DwNet can be difficult to train due to its use of highly non-linear warping grids. The generator also needs to be re-run for computing the warping grid each time the source image changes. Instead of predicting warping grids, our method works directly on the UV coordinates, making it independent of the source images. Once the target sequence of UV coordinates is generated, it can be applied for different textures without re-running the model regardless of complexity.

Video-to-video. These methods [1, 7, 9, 20] have access to a source video and can therefore create richer models of the source subject than single image sources. In particular, [9] generates videos with spatial transformation of target poses, allowing it to capture loose clothing. However, all these works rely on 2D keypoints, making it hard to con-

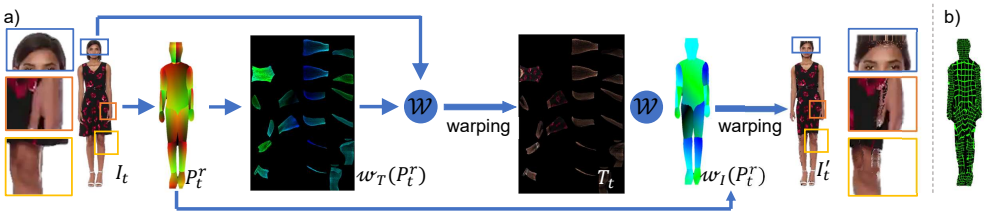


Figure 2. a) Example mapping from I_t to T_t via P_t^r, and back to I'_t. P_t^r cannot fully recover the image, and misses skirt, hair, and shoulder parts. Besides, colours inside the human body are also partially incorrect. b) Mapping results of P_t^r with T_{grid} as input. Most quadrants are preserved, indicating that the corresponding features are not destroyed after the UV mapping.

sider complicated visual styles. In contrast, we make use of a texture representation and aim to improve the quality of the UV mapping. Our model is trained without the need to access the textures in advance, which allows us to work with different texture inputs, regardless of complexity.

Generalized video generation. Early methods modelled the entire video clip as a single latent representation [33,40]. Follow-up work MoCoGAN [39] used a disentangled representation, separating appearance and motion. However, the model is not conditional, so it cannot generate videos conditioned on target appearances or motions, for example. Our method separates appearance and motion by design and allows for easy control and modification of either factor. We specify appearance via a (fixed) texture map, while motions are represented by UV coordinates over time.

End-to-end video re-targeting works RecycleGAN [4] and Vid2Vid [41] generate videos with content and motion from separate source videos. These methods train target-specific models, in that a new network is trained for each target video. In contrast, re-targeting in our case involves only a simple and efficient look-up.

3. Preliminaries

3.1. Notation & definitions

An image $I_t \in \mathbb{R}^{s_x \times s_y \times 3}$ for frame t in a sequence stores RGB information at a location $\mathbf{x} \in \mathbb{R}^{s_x \times s_y}$. The appearance of a person in I_t can also be represented in a texture $T_t \in \mathbb{R}^{t_x \times t_y \times 3}$ with locations $\mathbf{u}$. The image I_t and texture T_t are related via the the UV coordinates $P_t \in \mathbb{R}^{s_x \times s_y}$, where

$$P_t(\mathbf{x}) = \mathbf{u}, \quad \text{s.t.} \quad I_t(\mathbf{x}) = T_t(\mathbf{u}). \tag{1}$$

To ensure differentiability, we treat I_t, P_t and T_t as continuous functions in space via a suitable interpolation operator; we use bi-linear interpolation in our work. In practice, the three fields are represented as time sequences over t.

The corresponding texture T_t for an image I_t can be generated by warping I_t with function $\mathcal{W}$ via the warping grid $\omega_T(P_t)$; conversely, the image content can also be recovered as I'_t from the texture T_t and UV coordinates P_t with warping grid ω_I from T_t to I_t (see Figure 2):

$$T_t = \mathcal{W}(I_t,\ \omega_T(P_t)) \quad \text{and} \quad I'_t = \mathcal{W}(T_t,\ \omega_I(P_t)). \tag{2}$$

Note the warping function $\mathcal{W}(I, \omega)$, for every location $\mathbf{x}$ in I, returns a bi-linear interpolation of I at location $\omega(\mathbf{x})$.

In our work, we refer to the UV outputs from DensePose [2] or unwrapped from the 3D mesh of models like SMPL [19] as *raw* UV coordinates, denoted by P_t^r for frame t. Raw UV coordinates are typically restricted by the human body silhouette. As such, loose clothing parts are cut off (see the missing skirt parts in Figure 1a and Figure 2a. Additionally, the raw UV P_t^r is not one-to-one. Multiple pixels $\mathbf{x}$ of I_t may be mapped to the same $\mathbf{u}$ in T_t, leading to a loss of information in T_t. These two shortcomings may result in extreme and undesirable differences between the original I_t and the reconstructed I'_t (see example in Figure 2a. For a sequence of images over time, the differences are further compounded. As P_t^r can only be estimated frame-wise, resulting textures T_t tend to lack correspondence over time.

3.2. Problem formulation

Given the non-idealities of P_t^r, we aim to develop a system that can output a sequence of refined UV coordinates P_t^g leading to faithful reconstructions $I'_t = I_t$. Additionally, we aim for an independent and lightweight appearance representation in the form of a single texture T_o, which is constant over time.

We start by defining a model G parameterized by θ to estimate refined UV coordinates P_t^g from raw UV P_t^r:

$$P_t^g = G(P_t^r; \theta). \tag{3}$$

For I'_t to be of high quality and for P_t^g to be temporally stable, we consider appearance and temporal loss functions

$$\mathcal{L}_{\text{app}} = \sum_{t=0}^{N}(||I'_t - I_t||^2) = \sum_{t=0}^{N}(||\mathcal{W}(T_t, \omega_I(P_t^g)) - I_t||^2),$$
$$\mathcal{L}_{\text{temp}} = \sum_{t=0}^{N}(||T_t - T_o||^2) = \sum_{t=0}^{N}(||\mathcal{W}(I_t, \omega_T(P_t^g)) - T_o||^2), \tag{4}$$

where N represents the sequence length and T_o a constant texture. Minimizing $||I'_t - I_t||^2$ leads to improvements of I'_t. Minimizing $||T_t - T_o||^2$ encourages a constant texture, which in turn largely alleviates inconsistent correspondences over time.

4. Method

One could learn θ of model G if raw UV (P_t^r) were paired ground truth UV coordinates fulfilling the constraints in Equation 4. Such ground truth data does not exist in practice, so we are forced to consider indirect approaches. Naively applying an unsupervised or self-supervised training is ill-conditioned and error-prone, due to the strong non-linearities in mappings between I_t, T_t, and P_t. As such, we propose an approach to combine both supervised and unsupervised learning.

We start with a data pre-processing step (Sections 4.1 to 4.3) that gradually refines P_t^r to establish "ground-truth". It is worth noting that we handle the two parts of Equation 4 separately due to the strong non-linearity and large distance between P_t^r and P_t^g. After an initial training of G with the pre-processed data, we then incorporate unsupervised losses from the image space (Section 4.4) to train a final model G that jointly improves spatial and temporal quality. The trained model G generates full-silhouette UV coordinates for different poses.

Since appearance or RGB information is encoded only in the texture T_o, which is used for the loss and preprocessing of the data but not a part of the network inputs, the resulting UV coordinate sequence P_t^g can be directly used for video generation with any given texture. Subsequently, generating a new output sequence with changed colours or patterns is highly efficient.

4.1. UV extension

Raw UV inputs omit important details (see example in Figure 2a. First, we aim to achieve full silhouette coverage for P_t^r. To better understand the relationship between I_t, P_t^r and T_t, we visualize UV mapping results for a synthetic grid texture $\mathcal{W}(T_{grid}, \omega_I(P_t^r))$ in Figure 2b. Here, T_{grid} contains an evenly distributed grid quadrants, which remain well-preserved, suggesting that the UV mapping with P_t^r retains a piece-wise regular surface manifold, albeit with different scaling factors.

The grid structure suggests that neighbouring points in I_t remain neighbours in T_t, and additional entries can be added to the raw UV coordinates P_t^r via extrapolation from neighbouring points. It is worth pointing out that for traditional UV generation, cutting the object surface and minimizing surface distortion are two challenging steps [30]. The raw UV coordinates provide an initial unwrapping of the body, hence we focus on solving the latter challenge of minimizing distortions when extrapolating content in the UV coordinates.

In this paper, we extend the UV coordinates through energy minimization, employing a *virtual mass-spring* system. Mass-spring systems are commonly used in the simulation of clothing [18, 45]. Additionally, [21] and [38] have shown that the potential energy of a mass-spring system is minimized at the equilibrium state. Due to space limitations, we defer the full exposition to the Supplementary. In our formulation, springs naturally encode the area preservation constraints among new extrapolated points and their neighbouring points in a small region of the texture map. The spring forces drive the new extrapolated points to new positions until the system finds an equilibrium state with reduced distortion.

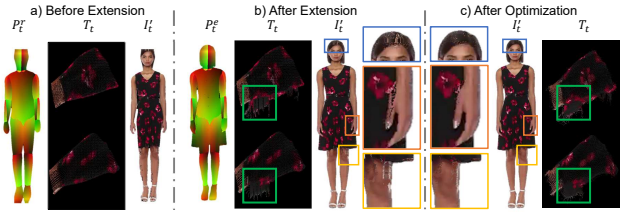


Figure 3. a) Raw UV coordinates, b) with application of UV extension and c) optimization. The UV extension allows missing parts such as the dress to be mapped into the correct parts of T_t, while UV optimization makes I_t' closer to I_t.

We denote the UV coordinates after the extension with P_t^e. An example result is shown in Figure 3b. We can see that the missing parts from the raw UV coordinates computed via DensePose are recovered successfully, such as the side of the dress.

4.2. UV optimization

After UV extension, artifacts in I_t' may remain (See Figure 3b. One cause of these artifacts is duplicate UV coordinates in P_t^r, as it is not constrained to be a one-to-one mapping, especially for direct methods such as DensePose. To further improve P_t, we directly minimize $\mathcal{L}_{\text{app}}(P_t)$ via gradient descent, initializing P_t with the extended UV map P_t^e. The gradient $\frac{\partial \mathcal{L}_{\text{app}}(P_t)}{\partial P_t}$ can be estimated via the intermediate warping grids $\omega_T(P_t^r)$, and $\omega_I(P_t^r)$. Details are provided in the Supplementary.

Following common practice in non-linear settings, we add a gradient and Laplacian regularizer to encourage smooth solutions [5] and minimize $\mathcal{L}_{\text{app}} + L_r$, where

$$L_r = \alpha_1(||\nabla P_t||_F^2) + \alpha_2 \sum_{i,j=0,1} ||H_{ij}(P_t)||_F^2, \quad (5)$$

H is the Hessian and $||\cdot||_F$ denotes the Frobenius norm. It is visible in Figure 3c that most of the artifacts in I_t' have been removed by the optimization procedure, and the image content is significantly closer to the reference. We denote the optimized UVs with P_t^o.

4.3. UV temporal relocation

Minimizing $\mathcal{L}_{\text{temp}}$ in Equation 4 will improve the temporal stability of the texture maps. To do so, we find point correspondences $Q_t(\mathbf{u})$ between T_t and T_o so that $T_t(\mathbf{u}) = T_o(Q_t(\mathbf{u}))$. The correspondences allow new UV coordinates P_t^f to map I_t back to the constant T_o instead of

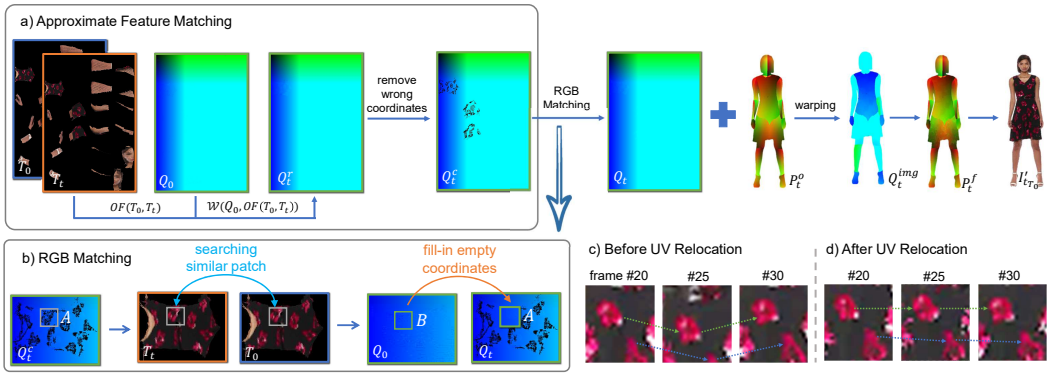


Figure 4. Overview and results of temporal UV generation. a) Approximate feature matching is achieved via the optical flow (OF) from T_o to T_t. b) RGB matching is applied to correct the coordinates resulting from errors in OF. Images in c) are generated with P_t^o and T_o, i.e., $I'_{t_{T_o}} = \mathcal{W}(T_o, \omega_I(P_t^o))$. Images in d) are similarly generated with P_t^f. Green and blue arrows are shown here to track the two patterns in the images. After the temporal relocation step, results are more temporally coherent.

T_t. For simplicity, we assign as the constant T_o the texture from frame 0 of a sequence, i.e. $T_o = T_0$.

We initialize the point correspondences with optical flow from T_o to T_t, i.e. $OF(T_o, T_t)$, as shown in Figure 4a. An approximate correspondence between T_t and T_o can be written as $Q_t^r(\mathbf{u}) = \mathcal{W}(Q_0(\mathbf{u}), OF(T_o, T_t))$. In theory, the reconstruction T'_t can then be reconstructed from T_o and $Q_t^r(\mathbf{u})$ via a *lookup* step, i.e. $T'_t(\mathbf{u}) = T_o(Q_t^r(\mathbf{u}))$.

Note that errors in $OF(T_o, T_t)$ makes $Q_t^r(\mathbf{u})$ only an approximate correspondence, and there are still differences between the reconstructed T'_t and the true T_t. To correct these errors, we remove the coordinates in $Q_t^r(\mathbf{u})$ where the texture content does not match, i.e. $T'_t(\mathbf{u}) \neq T_t(\mathbf{u})$. We then fill them in with regions from T_o to obtain the final $Q_t(\mathbf{u})$. The filling is based on a simple similarity measure of the RGB values. We defer the details to the Supplementary.

Comparisons of results before and after the temporal relocation step are shown in Figure 4c-d. The images $I'_{t_{T_o}}$ recovered from T_o are more temporally coherent after the relocation step.

4.4. Temporal UV model training

So far, we have improved the spatial and temporal quality of the raw UV P_t^r separately. We now consider the two objectives jointly in a spatio-temporal manner and apply an adversarial training for G from Equation 3. The learned G can then generate complete UV coordinates P_t^g at test time given raw UV coordinates P_t^r. Below, we define several unsupervised loss terms in both the UV and RGB image space to guide the training and produce high-quality outputs.

Spatial loss (UV space). Recall that P_t^f is now the UV coordinates that relate image I_t to the constant texture T_o based on the UV relocation step in section 4.3. We make use of a supervised L_2 loss $L_2 = \left\| G(P_t^r) - P_t^f \right\|_F^2$ and adversarial loss via a discriminator D_s:

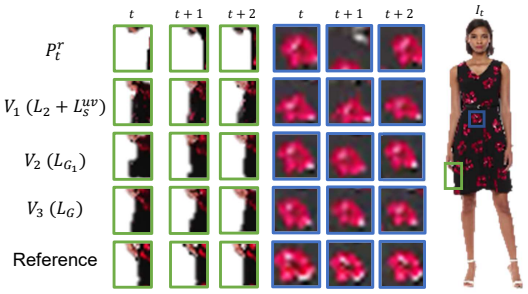


Figure 5. Comparisons of three successive frames, $(I'_{t-1_{T_o}}, I'_{t_{T_o}}, I'_{t+1_{T_o}})$, among results of P_t^r, V_1, V_2, and V_3. We show examples of the same region in the image to illustrate the temporal coherence of the generated videos. We can see that V_1 is more coherent than P_t^r because of the temporal relocation when preparing the training data. V_2 and V_3 show further improvements due to the temporal stability losses in UV and image spaces.

$$\begin{aligned} L_s^{uv} &= -log(D_s(G(P_t^r))), \\ L_{D_s} &= -logD_s(P_t^f) - log(1 - D_s(G(P_t^r))). \end{aligned} \tag{6}$$

Temporal stability loss (UV space). We consider a smoothing loss between neighbouring frames $t-1$ and $t+1$:

$$\begin{aligned} L_{smo} = &\left\| G(P_{t-1}^r) - G(P_t^r) \right\|_F^2 + \left\| G(P_t^r) - G(P_{t+1}^r) \right\|_F^2 \\ &+ \left\| G(P_{t-1}^r) - 2 \times G(P_t^r) + G(P_{t+1}^r) \right\|_F^2, \end{aligned} \tag{7}$$

and add an unsupervised adversarial loss via a second discriminator network D_t:

$$\begin{aligned} L_t^{uv} &= -log(D_t(G(P_{t-1}^r), G(P_t^r), G(P_{t+1}^r))), \\ L_{D_t} &= -log(D_t(P_{t-1}^f), f(P_{t-1}^f), f(f(P_{t-1}^f))) \\ &\quad - log(1 - D_t(G(P_{t-1}^r), G(P_t^r), G(P_{t+1}^r))), \end{aligned} \tag{8}$$

where f are randomized geometric transformations (e.g., translation, rotation or scaling). Note that ground truth over time is not available in our setting. We synthesize ground

truth by randomly choosing a transformation f and applying it to P^f_{t-1}. This yields a reference for time t; applying the transformation again at $t+1$ yields an additional reference to form a synthetic triplet. The triplets serve as ground truth for the adversarial training of Equation 8 and guide the generation of smooth UV coordinates over time.

Spatial loss (image space). With the mapping pipeline from UV coordinates to images, an image-based L2 loss is applied at training time:

$$L^{\text{img}}_s = \|I_{g_t} - I_t\|^2_F, \text{ where } I_{g_t} = \mathcal{W}(T_o, \omega_I(G(P^r_t))). \quad (9)$$

Temporal stability loss (image space). Similar to the UV space, we define a temporal adversarial loss via an additional discriminator D_{img} in image space:

$$\begin{aligned} L^{img}_t &= -\,logD_{img}(I_{g_{t-1}}, I_{g_t}, I_{g_{t+1}}), \\ L_{D_{img}} &= -\,logD_{img}(I_{t-1}, I_t, I_{t+1}) \\ &\quad -\,log(1 - D_{img}(I_{g_{t-1}}, I_{g_t}, I_{g_{t+1}})). \end{aligned} \quad (10)$$

To summarize, the full loss of G is given by

$$\begin{aligned} L_G =& \lambda_2 L_2 + \lambda_{uv,s} L^{uv}_s + \lambda_{smo} L_{smo} \\ &+ \lambda_{uv,t} L^{uv}_t + \lambda_{img,s} L^{img}_s + \lambda_{img,t} L^{img}_t. \end{aligned} \quad (11)$$

In practice, we found it difficult to keep the losses in image space stable at the beginning of the training. Hence, we train G first with the partial loss L_{G_1}, where

$$L_{G_1} = \lambda_2 L_2 + \lambda_{uv,s} L^{uv}_s + \lambda_{smo} L_{smo} + \lambda_{uv,t} L^{uv}_t, \quad (12)$$

for 5×10^4 steps. We freeze G, and only train D_{img} for 5×10^4 steps to ensure that D_{img} is commensurate with G. We then train all networks jointly with the full loss L_G for another 10×10^4 steps. Generator G is built with ResNet architecture, using 30 (for DensePose P^r_t) or 20 (for SMPL P^r_t) residual blocks. All of our discriminators D_s, D_t, and D_{img} follow the same encoder structure using 5 convolutional layers followed by a dense layer. We use 120 continuous frames without background from the Fashion dataset [47] as the training data. For every step, we randomly crop small regions of size 32×32 from P^r_t to be used as input. The Adam optimizer is applied for training. Other learning details are given in the Supplementary.

Model inference. After the training, UV coordinates P^g_t with full clothing silhouettes can be generated via G. We can achieve pose-guided generation when a sequence of raw target poses is provided. Since we focus on the UV coordinates and inputs to G, which do not include texture information, virtual try-on can also be easily achieved in our pipeline by changing texture maps to which the UV coordinates are applied. Once P^g_t is generated, the image sequence I'_t requires a minimal number of calculations to be produced (essentially, only one texture lookup per output pixel). As we will demonstrate below, this is vastly more efficient than, e.g., evaluating a full CNN.

	PSNR↑	LPIPS↓ $\times 10^{-2}$	tOF↓ $\times 10^4$	tLP↓ $\times 10^{-2}$	T-diff↓ $\times 10^5$
P^r_t	22.1	8.1	1.69	1.0	5.42
V_1	**23.1**	7.9	1.84	1.4	**3.93**
V_2	**23.1**	**7.6**	1.70	**0.9**	4.33
V_3	22.9	7.7	**1.65**	1.0	4.19

Table 1. Quantitative comparisons between P^r_t and our three different versions, V_1, V_2, and V_3. For a fair comparison, the body shapes of V_1, V_2, and V_3 are cropped to be in line with P^r_t. Our method shows significant improvements on both spatial (PSNR and LPIPS) and temporal (tOF, T-diff) evaluation metrics.

5. Ablation study

This section shows how different parts of Equation 11 influence the generated results. We start with a basic model trained with the losses L_2 and L^{uv}_s and denote this V_1. We then add temporal losses in the UV space, L_{smo} and L^{uv}_t, for training and denote this as V_2. The full model trained with L_G is denoted as V_3.

Figure 5 shows two qualitative comparisons. All three versions successfully fill in the missing parts of the DensePose UV map and are close to the reference (green patch, skirt edge). To evaluate the temporal coherence, we zoom in on the motion of the flower patterns (blue patch). Results from the raw UV coordinates P^r are unsteady since its temporally unstable UV content leads to a misalignment of the texture over time. V_1 has better coherence due to the UV relocation (section 4.3) applied to the training data. V_2 and V_3 show progressive improvements thanks to the temporal stability losses and the image space losses.

As quantitative evaluation of the spatial performance, we compute peak signal-to-noise ratio (PSNR) and perceptual LPIPS [48]. For temporal stability, we follow [8] and estimate the differences of warped frames, i.e., T-diff $= \|I_{g_t}, \mathcal{W}(I_{g_t}, v_t)\|_1$, where v_t typically denotes the intra-frame motion computed by optical flow. In our setting we use the UV coordinates for v_t instead (details in the Supplementary Material). Additionally, we evaluate with two temporal coherence metrics [11]: $tOF : \left\|OF(I_t, I_{t+1}) - OF(I_{g_t}, I_{g_{t+1}})\right\|_1$ and $tLP : \left\|LPIPS(I_t, I_{t+1}) - LPIPS(I_{g_t}, I_{g_{t+1}})\right\|_1$. Except for PSNR, lower values are better for all metrics.

From Table 1, we see that that V_1 has the worst results in terms of tOF and tLP. Its LPIPS is also worse than V_2 and V_3 because V_1 is trained purely with spatial losses in the UV space. Hence, supervision via preprocessed data P^f_t is insufficient. Note, however, that V_1 shows the best T-diff score, as T-diff mainly relies on the calculation of v_t and is easily "fooled" by overly smooth content. V_2 and V_3 add temporal constraints and show better temporal behaviour in terms of tOF and tLP. Compared with V_2, V_3 exhibits a similar spatial performance though it yields better temporal stability. This is especially the case if we evaluate without cropping to fit P^r_t (see Supplementary). This

		example in Figure 4	a)	b)	c)
PSNR ↑	$P^r(I_t)$	26.44	26.10	14.39	15.33
	$P^o(I_t)$	**36.40**	**40.92**	**36.91**	**30.11**
LPIPS↓ ($\times 10^{-1}$)	$P^r(I_t)$	0.65	1.09	1.30	1.38
	$P^o(I_t)$	**0.12**	**0.09**	**0.09**	**0.24**

Figure 6. Comparisons between DensePose UVs P_t^r and optimized UVs P_t^o. Here, we only show examples of the skirt part in T_t to clarify the differences. We can see that P_t^o can preserve most of the skirt information in T_t, and I_t' of P_t^o are closer to I_t than that of P_t^r. The quantitative evaluation also shows that our results after UV optimization (described in section 4.2) are closer to the reference.

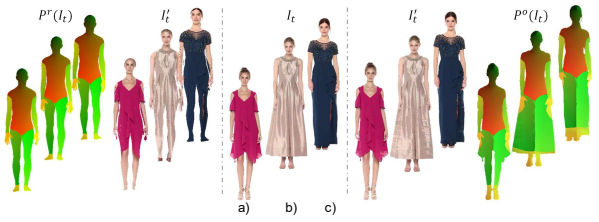
also verifies that loss functions from the image space can be successfully applied to guide the training.

Optimized UVs (P_t^o). In addition to Figure 3c in section 4.2, more samples of the optimized UVs P_t^o are shown in Figure 6 and the Supplementary. The comparison of PSNR and LPIPS scores in Figure 6 verifies that our optimization pipeline significantly improves the spatial content. Similar conclusions can be drawn for the UV coordinates derived from SMPL (see Figure 7).

6. Results and evaluation

Direct comparison of P_t^g. We provide a direct comparison between raw UVs P_t^r and those generated by our approach P_t^g in Figure 8. Apart from the body itself, it is visible that our outputs I_t', generated with UVs from both SMPL and DensePose models, also recover the hair, sleeves, and the skirt. Hence, we have fulfilled the goal of capturing the full appearance of a person, rather than the body silhouette.

Comparison with state of the art. In Figure 9, we compare with the closest method DwNet, which also focuses on video generation from a single image with UV coordinates.



		a)	b)	c)
PSNR ↑	$P^r(I_t)$	16.61	13.33	19.68
	$P^o(I_t)$	**31.64**	**33.27**	**33.88**
LPIPS↓ ($\times 10^{-1}$)	$P^r(I_t)$	1.25	2.18	1.41
	$P^o(I_t)$	**0.10**	**0.43**	**0.21**

Figure 7. Comparisons between P_t^r from SMPL and P_t^o. We can see that after optimization, P_t^o preserves more of the loose clothing and I_t' closely matches I_t. Quantitative evaluations also show that our results are much closer to the reference.

DwNet smoothes the texture of the clothing as the quality of its output is limited by the accuracy of the warping module. However, our method focuses on UV coordinates, and obtains the appearance information directly from the texture map, so our results are significantly sharper. Our results are also closer to the reference images, leading to better spatial evaluations like PSNR and LPIPS. For temporal quality, the tOF and tLP values indicate that our results have better temporal stability than DwNet. Consistent conclusions can also be drawn from the user studies illustrated in the Supplementary.

We note that the DwNet model needs to be rerun once the texture is changed. In contrast, once the UV coordinates of a sequence have been generated, our method can re-texture a sequence without evaluating any trained models. Instead, we simply map the updated texture via our UV coordinates; this is a simple lookup that is several orders of magnitude fewer in operations than DwNet. Such a low computational load would, e.g., allow for running a virtual try-on pipeline in real-time on otherwise low-performance end-devices.

Generated video with different textures. Our generation network completely separates the UV representation from the RGB appearance information, which is only encoded in the constant texture T_o. As such, the UV coordinates generated from our model are compatible with any other texture that aligns with the arrangement of the original T_o. This makes it easy to create virtual try-on applications by modifying the texture. In particular, the source clothing can be obtained from any image source, e.g. another photo or a texture image. We show re-textured examples in Figure 1, Figure 10 and the Supplementary. Note that as our focus is on capturing clothing, hence we reuse the texture of the human parts (face, hands and legs) from the source videos for these virtual try-on results.

Limitations. Our method generates an entire video via P_t^g and T_o. Currently, we simply choose T_0 for T_o. However, T_0 may not have sufficient coverage for some situa-

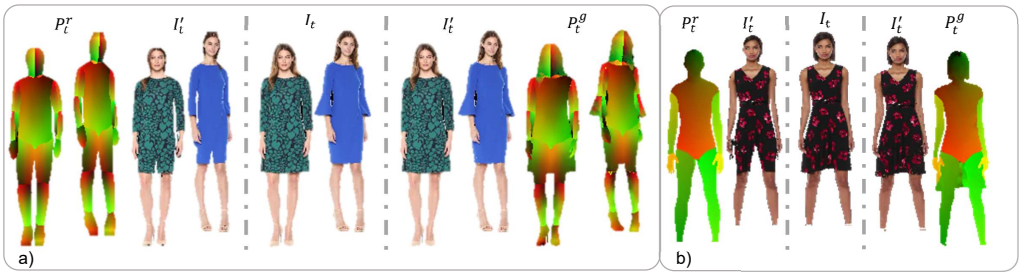
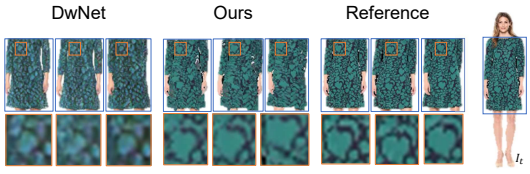


Figure 8. Comparison between P_t^g and P_t^r. P_t^g in (a) and (b) are generated from DensePose and SMPL model, respectively. Our I'_t are closer to the reference I_t, which indicates that P_t^g has better capacity to preserve more information of I_t.



	PSNR ↑	LPIPS ↓	tOF ↓ ($\times 10^4$)	tLP ↓ ($\times 10^{-2}$)	FPO (*times/pixel*)
DwNet	22.02	0.069	1.48	0.67	57.5M
Ours	**23.23**	**0.048**	**1.21**	**0.35**	**5**

Figure 9. Comparison with state-of-the-art method DwNet. Our results are closer to the reference, which is also supported by the evaluation metrics below. Additionally, we also compare the number of floating point operations (FPO) for every pixel during video generation. Without rerunning trained models, our method shows a significant reduction of computation.

tions, e.g. the backside of the clothing. This could be improved by incorporating additional steps for texture completion [10, 17]. Another limitation arises from boundary occlusions. While we aim at coherent point correspondence among different frames, occlusions occurred in the boundary areas make it impossible to find the obscured point at frame $t+1$ for the corresponding point at frame t, which brings a noticeable degree of high-frequency noise near the boundaries during fast motions of the body or clothing. But quantitative metrics and our user study show that our results yield better temporal coherence than state-of-the-art methods. Besides, this problem could benefit from additional image space smoothing over time.

7. Conclusion

We have presented a novel algorithm to generate stable UV coordinates for image sequences that capture the full appearance of a human body, including loose clothing and hair. Central in arriving at this goal are a custom pre-computation pipeline and a spatio-temporal adversarial learning approach. Our method allows for high-quality video generation and also enables very quick turnaround times for style modifications. Based on the one-time process to generate a UV coordinate sequence, our method allows for the repeated synthesis of output videos via a single underlying texture with vastly reduced computations compared to existing approaches. Currently, we primarily focus on clothing, because the complicated textures and various poses of the body make it a challenging application. It provides an appropriate test bed that encapsulates the capabilities of our pipeline. However, our pipeline can be potentially generalized to UVs of other objects, e.g., animals, cars and furniture. This enables us to achieve video generation and texture editing of arbitrary objects easily in our future work.

Figure 10. P_t^g is compatible with different textures to generate a desired target sequence. (a)-(c) are generated using DensePose UV coordinates, while (d) uses SMPL UVs.

8. Acknowledgement

This research / project is supported by the Ministry of Education, Singapore, under its MOE Academic Research Fund Tier 2 (STEM RIE2025 MOE-T2EP20220-0015) and the ERC Consolidator Grant *SpaTe* (ERC-2019-COG-863850).

References

[1] Kfir Aberman, Mingyi Shi, Jing Liao, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Deep video-based performance cloning. In *Computer Graphics Forum*, volume 38, pages 219–233. Wiley Online Library, 2019. 2

[2] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 1, 2, 3

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

[4] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European conference on computer vision (ECCV)*, pages 119–135, 2018. 3

[5] Misha Belkin, Partha Niyogi, and Vikas Sindhwani. On manifold regularization. In *International Workshop on Artificial Intelligence and Statistics*, pages 17–24. PMLR, 2005. 4

[6] 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. 1

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

[8] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1105–1114, 2017. 6

[9] Kun Cheng, Hao-Zhi Huang, Chun Yuan, Lingyiqing Zhou, and Wei Liu. Multi-frame content integration with a spatio-temporal attention mechanism for person video motion transfer. *arXiv preprint arXiv:1908.04013*, 2019. 2

[10] Julian Chibane and Gerard Pons-Moll. Implicit feature networks for texture completion from partial 3d data. In *European Conference on Computer Vision*, pages 717–725. Springer, 2020. 8

[11] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics (TOG)*, 39(4):75–1, 2020. 6

[12] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7093–7102, 2018. 1

[13] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bowen Wu, Bing-Cheng Chen, and Jian Yin. Fw-gan: Flow-navigated warping gan for video virtual try-on. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1161–1170, 2019. 2

[14] Albert Rial Farras, Sergio Escalera Guerrero, and Meysam Madadi. Rgb to 3d garment reconstruction using uv map representations. 2021. 1

[15] Baris Gecer, Jiankang Deng, and Stefanos Zafeiriou. Ostec: One-shot texture completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7628–7638, 2021. 1

[16] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided image generation. *arXiv preprint arXiv:1811.11459*, 2018. 2

[17] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided human image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12135–12144, 2019. 8

[18] Chenfanfu Jiang, Theodore Gast, and Joseph Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 4

[19] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020. 1, 3

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

[21] Tiantian Liu, Adam W Bargteil, James F O'Brien, and Ladislav Kavan. Fast simulation of mass-spring systems. *ACM Transactions on Graphics (TOG)*, 32(6):1–7, 2013. 4

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

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

[24] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in neural information processing systems*, pages 406–416, 2017. 2

[25] Liqian Ma, Zhe Lin, Connelly Barnes, Alexei A Efros, and Jingwan Lu. Unselfie: Translating selfies to neutral-pose portraits in the wild. In *European Conference on Computer Vision*, pages 156–173. Springer, 2020. 1

[26] Meysam Madadi, Hugo Bertiche, and Sergio Escalera. Smplr: Deep smpl reverse for 3d human pose and shape recovery. *arXiv preprint arXiv:1812.10766*, 2018. 1

[27] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *Proceedings of the European conference on computer vision (ECCV)*, pages 123–138, 2018. 1, 2

[28] Natalia Neverova, James Thewlis, Riza Alp Guler, Iasonas Kokkinos, and Andrea Vedaldi. Slim densepose: Thrifty learning from sparse annotations and motion cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10915–10923, 2019. 1

[29] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2018. 1

[30] Roi Poranne, Marco Tarini, Sandro Huber, Daniele Panozzo, and Olga Sorkine-Hornung. Autocuts: simultaneous distortion and cut optimization for uv mapping. *ACM Transactions on Graphics (TOG)*, 36(6):1–11, 2017. 4

[31] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8620–8628, 2018. 2

[32] Albert Pumarola, Vedanuj Goswami, Francisco Vicente, Fernando De la Torre, and Francesc Moreno-Noguer. Unsupervised image-to-video clothing transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1

[33] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839, 2017. 1, 3

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

[35] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Advances in Neural Information Processing Systems*, pages 7137–7147, 2019. 2

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

[37] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. In *European Conference on Computer Vision*, pages 717–734. Springer, 2020. 1

[38] Florian Theil. Surface energies in a two-dimensional mass-spring model for crystals. *ESAIM: Mathematical Modelling and Numerical Analysis*, 45(5):873–899, 2011. 4

[39] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 3

[40] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621, 2016. 1, 3

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

[42] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1160–1169, 2020. 1

[43] Haonan Yan, Jiaqi Chen, Xujie Zhang, Shengkai Zhang, Nianhong Jiao, Xiaodan Liang, and Tianxiang Zheng. Ultrapose: Synthesizing dense pose with 1 billion points by human-body decoupling 3d model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10891–10900, 2021. 1

[44] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018. 1

[45] Jian Dong Yang and Shu Yuan Shang. Cloth modeling simulation based on mass spring model. In *Applied Mechanics and Materials*, volume 310, pages 676–683. Trans Tech Publ, 2013. 4

[46] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. Pose-guided human animation from a single image in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15039–15048, 2021. 2

[47] Polina Zablotskaia, Aliaksandr Siarohin, Bo Zhao, and Leonid Sigal. Dwnet: Dense warp-based network for pose-guided human video generation. *arXiv preprint arXiv:1910.09139*, 2019. 1, 2, 6

[48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[49] Tyler Zhu, Per Karlsson, and Christoph Bregler. Simpose: Effectively learning densepose and surface normals of people from simulated data. In *European Conference on Computer Vision*, pages 225–242. Springer, 2020. 1