Generating 3D Human Animations from Single Monocular Images

Tanya Marwah

Technical Report Number: CMU-RI-TR-19-12

April 18, 2019

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee:
Professor Kris M. Kitani, Chair, CMU
Professor Katerina Fragkiadaki, CMU
Ye Yuan, CMU

Thesis proposal submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics

for Jingle...

Abstract

This work presents a methodology to infer the complete texture of an articulated 3D model of a person from a single image. We use a rich and compact mesh representation as the basis of our articulated human model and detail it with the predicted texture – a UV texture map. We propose a two-stream approach that decouples geometry and texture inference, and combines the outputs of the two streams using a differentiable renderer, which enables end-to-end self-supervised learning. In order to predict a consistent texture that captures the nuances of the input image, we propose two key innovations – 1) a generative model that outputs complete UV texture map from partial observations; and 2) multi-view curriculum training that enables our model to be applied on a wide range of images. We show through our experiments that our model is able to predict a complete texture of the human model that captures the regularities in appearance such as symmetry in clothing, facial features, etc. To highlight some important differences compared to prior work on geometric and texture inference from a single image, we also show how our parameterization of texture and human pose can easily be extended to generate animations based on learned humanoid control policies.

Acknowledgements

I would like to thank my advisor Dr. Kris M. Kitani for his guidance and support in the past two years. I have benefited greatly from his insights and his readiness to brainstorm over different ideas. His patience and constant encouragement were crucial motivating factors throughout this work.

This work would not have been possible without the help from Ye Yuan. His insights on the problem statements and his willingness to discuss different ideas especially when I was stuck is greatly appreciated. I would like to thank Mohit Sharma and Arjun Sharma for introducing me to GAIL and explaining the codebase line-by-line. I am also indebted to all the members of K-Lab, a cohort that I am proud to have been a part of.

I would also like to thank Rohan Sawhney, for his advice throughout my research. Gaurav Mittal and Shubham Agrawal for the brainstorming sessions, useful insights and suggestions. Finally, a big shout out to Kevin Zhang for all the editing that he was always ready to help me out with.

Two people whom I can never thank enough but also never miss an opportunity to do so are my parents. Thank you Mom and Dad.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

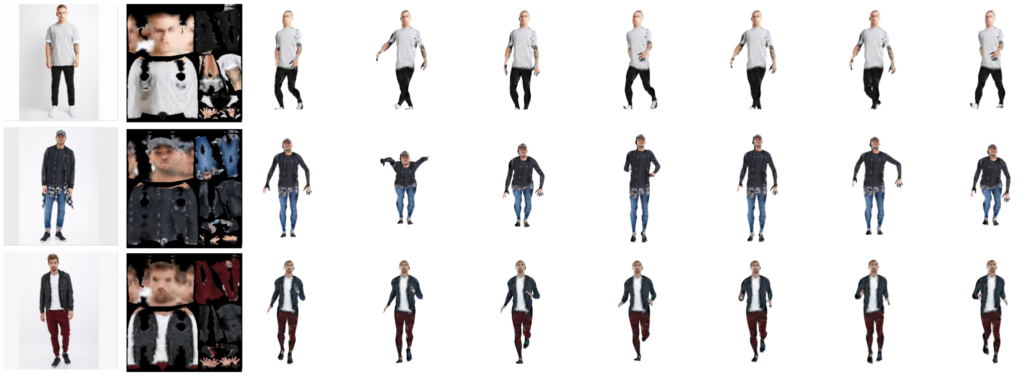


Figure 1.1: We present a methodology to create textured human mesh models from a single monocular image. As an application of our approach, we animate our models using control policies learned from motion clips, making the person in the input image come alive.

Endowing AI systems with the ability to formulate a three-dimensional understanding of human appearance from a single RGB image is an important component technology for applications such as person re-identification, biometrics, virtual reality and augmented reality. In the context of virtual reality or gaming, a method that can generate the complete texture of a subject from a single RGB image can provide a simple way for content creators to create virtual 3D human avatars, without the need for expensive imaging equipment. However, jointly inferring the texture map and 3D human model from a single image is challenging because: (1) scale ambiguity, clothing and occlusion make it hard to obtain the 3D model of a person from a single monocular image; and (2) a complete texture map needs to be inferred for each part of the body, such as skin tone, clothing and facial details, even if parts are occluded or are not visible in the image. With these challenges in mind, we introduce a methodology to generate the complete texture of a 3D model of a human from a single image. In particular, we use a model-based approach for estimating the body-shape and pose of the 3D mesh model [26] and concurrently utilize it towards a data-driven approach for estimating the UV texture map of the body (*e.g.*, shirt, pants, skin tone, face).

In this thesis we introduce an approach to estimates 3D shape, pose and texture that utilizes a two-stream computational architecture that decouples geometric (body shape and pose) reasoning and texture (clothing, skin tone, face layout) reasoning, and merges the two streams using a differentiable graphics renderer to ensure that the 2D projection of the estimated 3D model is consistent with the input image.

For the geometric stream, we use a deep neural network, referred to as the *geometry decoder*, to estimate the shape and pose parameters of a 3D mesh model. This portion of the network estimates the parameters of the Skinned Multi-Person Linear [26] (SMPL) model – a parameterized 3D mesh model for the human body. SMPL uses a pose prior – 3D joint angles and a shape prior – low dimensional shape space to model realistic humans.

For the texture stream, we use a cascaded refinement networks type [5] architecture, referred to as the *texture decoder*, to estimate the texture in the UV space [14]. We propose two crucial techniques to help the texture decoder complete the UV texture map. (1) We generate an *incomplete texture map* by first projecting the visible vertices of the predicted 3D mesh model onto the input image and then writing the color at those points at their corresponding UV coordinates. We then condition our texture decoder on the acquired incomplete texture map as well as the input image. This provides the initial context to our network as each pixel on the UV texture map as a fixed semantic meaning (*e.g.*, head, arms, torso). (2) We propose a novel *multiview curriculum learning* (MCL) approach to make the network gradually render images from viewing angles moving away from the original image. We constrain these synthetically rendered images with an adversarial loss which forces the network to hallucinate appropriate textures for non-visible body parts. We show that combining both of these techniques allows for single-shot generation of texture of the entire mesh model with fine visual details. Finally, the use of a differentiable renderer [20] allows us to penalize variation between the final rendered image and the input image, which not only allows for self-supervision and end-to-end training, but also alleviates the needs for labelled data.

To the best of our knowledge ours is the first end-to-end methodology that infers the complete texture of a 3D human mesh model from a single image in a completely unsupervised manner (*i.e.*, without relying on additional inputs) by utilizing a parametric 3D human mesh model. The main technical innovations of our work lies within the texture decoder which is able to (1) infer a complete UV through a sequence of refinement steps; and (2) perform multi-view curriculum learning to gradually learn a view invariant texture inference model. In contrast to prior works such as [29, 39] we infer the complete texture of the 3D mesh model by utilizing the UV parameterization and hence can synthetically render the subject's image from any view and in any desired pose. We also show how our parameterization of texture and human pose are widely applicable and can easily be extended to generate animations based on learned humanoid control policies in Figure 1.1. Furthermore, they are compatible with existing graphics engines such as Blender, Unity and Unreal Engine and hence can be utilized by animators and content creators for a wide range of applications.

Chapter 2

Related Work

2.1 2D Generation of Human Body

A large body of recent work has focused on using deep generative models [9,22] for various tasks like image-to-image translation [16], image super-resolution [24] and generating photo-realistic images of objects and human faces [19,34]. Most of these generative approaches however do not deal with articulated-object generation, hence the representations learned by these models fail to capture an interpretable latent structure of the objects. This issue is even more pronounced when we try to generate humans, as encapsulating the human anatomy along with all its varying visual details can be very difficult. Work like [35] uses generative adversarial networks (GANs) to change human pose in one image to another, however they do so by conditioning on either a target image or pose. [30] makes use of dense 2D to 3D correspondences provided by [11] to change the pose of a person in a 2D image, conditioned on a pose donor. Though all these approaches achieve photo-realistic results, they limit themselves to the 2D pixel space and need to be conditioned on a strong signal (target pose or pose donor). In contrast, we are not constrained to the pixel space but rather infer the complete texture of an articulated 3D model. This representation can be manipulated (*e.g.*, camera motion, pose change) and rendered in a meaningful way without a conditioning image, hence allows for physics-based character animation.

Recent work by [23] proposes a generative model for people in clothing for the full body, wherein, the authors use a conditional variational autoenconder (cVAE) to generate people in accurate appearances. However, their approach requires annotations for different clothing items. Our method eliminates the need for such annotations. Further, the aforementioned method is limited to generating a 2D RGB image, while we infer the complete 3D representation of a human body from a single image.

2.2 3D Modeling of Human Body

The estimation of human models from a single monocular image has been a long-standing problem in computer vision. Most early works focus on learning 2D body pose from a single image. However, deep neural networks have been successfully used to estimate the 3D pose from a single image as well [28, 36]. Alternatively, there has also been prior work on creating realistic animated human body from an RGB image. Most such works have used parameteric mesh models to model the complex deformations possible in humans [6, 10]. In [26], the authors proposed Skinned Multi-Person Linear model (SMPL) a generative human body model that is parameterized using 3D joint angles and a 10 dimensional linear shape subspace.

[3] is one of the first works to look at the problem of inferring 3D human pose and shape for a SMPL model from a single image. They follow an optimization based approach that penalizes the error between the projection of 3D model joints and detected 2D joints. However, such optimization based approaches are often susceptible to local minima. Very recently, there have been many works [17,31,32,36] that have used deep neural networks for estimating the parameters of the SMPL model. These works use novel methods to compensate for the lack of ground truth 3D annotations *e.g.*, the use of factorized adversarial prior in [17], self-supervision by using differentiable rendering in [36] and using a region based 2D representation as a prior to estimate the 3D pose and shape in [31]. However, unlike all the above methods, we not only infer the geometry (pose and shape) of the human body but also address the highly ambiguous task of inferring the concomitant texture of the estimated 3D model.

2.3 Texture Generation

Very few works have looked at the problem of inferring texture for human models. Recently, [1, 2] have looked at the problem of estimating accurate 3D model and texture of arbitrary people from a monocular video. This video, however, requires a moving person in a single T-pose, such that a consensus shape for the human body is calculated from each frame of the video. This consensus shape is then compared with the shape at each frame to get the texture map for the person. Although the above method does obtain excellent results, it requires a monocular video of a person in a specific T-pose from all different viewpoints, while in contrast we only require a single monocular image. Moreover, the above optimization techniques are insufficient to infer the texture from a single image as they cannot reason about the invisible body parts.

Prior works such as [3, 18, 32, 36] has looked at the problem of inferring 3D human pose and shape for an SMPL model from a single image. Recent work [38] also estimates the appropriate vertex deformations of the SMPL model to match the clothing geometry of the subject in the image by matching the silhouettes of the deformed mesh with image inputs. However, they do not address the problem of texture inference and instead manually fit visible texture to the back which is only applicable when the image contains the frontal view of the subject. [29] introduces a deep visual hull algorithm that captures the geometric details of the subject in the given image and also predicts the visual appearance of the person in new poses. However, in contrast to our approach that predicts a UV texture map, their algorithm uses a target pose conditioned image-to-image translation network to generate the person in the target view.

In [18] the authors look at the problem of inferring the shape and texture of simple object categories, such as birds and cars, from a single image. They assume a relatively low-dimensional spherical mesh as the mean shape ($\sim$600 vertices) for the underlying object and match its shape to the object by vertex deformation. Further they assume that the objects are symmetric and predict the texture for their mesh using a method referred to as 'texture flow'. We show in section 5.6.1 that such a flow-based approach is insufficient to find semantic correspondence between the input image and the texture for the human model, and fails to assign plausible texture to each body part. To address these challenges, we instead generate an incomplete UV texture map for a person from the input image. Further, we don't assume any symmetry but introduce a novel multiview curriculum learning approach which allows our model to hallucinate the parts of human body that are non-visible in the input image.

Chapter 3

Geometry Inference

Given an image $\tilde{I}$ of a person, our goal is to predict a texture map Y of an articulated human mesh model that captures the visual attributes (, skin tone, clothing, facial details) of the person from a single image. Figure 4.1 shows a brief outline of our method. We follow an encoder-decoder framework and first encode the input image into a latent representation ϕ, which is then used by two decoders: 1) a *geometry decoder* that predicts the shape and pose of our parametrized human mesh model, and 2) a *texture decoder* that infers the UV texture map of the mesh model. We then use a differentiable renderer [20] to render the image of the predicted textured human model, allowing us to train our network in an end-to-end fashion. We therefore present a data-driven approach to estimate the complete texture of a 3D human mesh model in an unsupervised way from just a single image. In the following sections we describe the details of all the steps involved.

3.1 Human Mesh Model

We use the Skinned Multi-Person Linear (SMPL), a parameterized 3D mesh model which factors the human body into *shape* (height, weight and body proportions) and *pose* (sit, stand, jump,). The shape of the model is defined by a vector $\beta \in \mathbb{R}^{10}$ containing the first 10 coefficients of a PCA shape space. The pose parameter $\theta \in \mathbb{R}^{69}$ consists of the relative 3D rotation of 23 joints in axis-angle representation. The final triangular mesh of the SMPL model has $F = 13776$ faces and $N = 6890$ vertices, with vertex positions obtained by shaping a template mesh conditioned on θ and β, transforming bones of the skeleton according to θ, and finally deforming the surface via linear blend skinning. In order to enable self-supervision, we also infer the camera parameters from the input image. Similar to [17], we assume a weak-perspective camera model with orthographic projection. The camera parameters consists of the global rotation R in axis-angle representation, 2D translation $t \in \mathbb{R}^2$ and scale $s \in \mathbb{R}$.

We introduce a network which we refer to as the *geometry decoder G* that takes the latent representation ϕ and predicts the parameters of the SMPL model as well as the camera. Therefore, G decodes the representation ϕ to predict an 85 dimensional vector $\Theta = (\theta, \beta, R, s, t)$. Similar to [4,7,17] we indeed realize that regressing over the parameters in an iterative error feedback loop (IEF) improves the performance of the decoder. The training of our *geometry decoder* is similar to [17] with one major difference. Our methodology, for now, trains on datasets that have the ground truth 3D joint locations and SMPL parameters available, and hence we do not use the factorized adversarial prior in [17] for our training.

3.1.1 Geometric Loss

In order to train the geometry decoder and learn the parameters of the 3D model, we use three different types of losses: (1) 3D mesh parameter loss $\mathcal{L}_{\text{smpl}}$; (2) 3D joint location loss $\mathcal{L}_{\text{3D}}$; (3) 2D re-projection loss $\mathcal{L}_{\text{2D}}$, which are estimated as follows,

$$\mathcal{L}_{\text{smpl}} = \sum_i \|[\beta_i, \theta_i] - [\tilde{\beta}_i, \tilde{\theta}_i]\|_2^2 \,, \tag{3.1}$$

where $(\tilde{\beta}_i, \tilde{\theta}_i)$ are the ground truth shape and pose parameters of the SMPL model and $(\beta_i$, $\theta_i)$ are the predicted parameters.

$$\mathcal{L}_{\text{3D}} = \sum_i \|X_i - \tilde{X}_i\|_2^2 \,, \tag{3.2}$$

where $\tilde{X}_i, X_i$ are the ground truth and predicted 3D joint locations respectively, which can be calculated from β and θ by first computing their rest positions with the regression matrix provided by [26] and transforming them with forward kinematics.

$$\mathcal{L}_{\text{2D}} = \sum_i \|x_i - \tilde{x}_i\|_2^2 \,, \tag{3.3}$$

where $\tilde{x}_i$ are the ground truth 2D keypoint locations and x_i are the locations after orthographic projection of X_i using the predicted camera parameters (R, s, t). Therefore,

$$\mathcal{L}_{\text{geom}} = \lambda_1 \mathcal{L}_{\text{smpl}} + \lambda_2 \mathcal{L}_{3D} + \lambda_3 \mathcal{L}_{2D} \, . \tag{3.4}$$

Chapter 4

Texture Inference

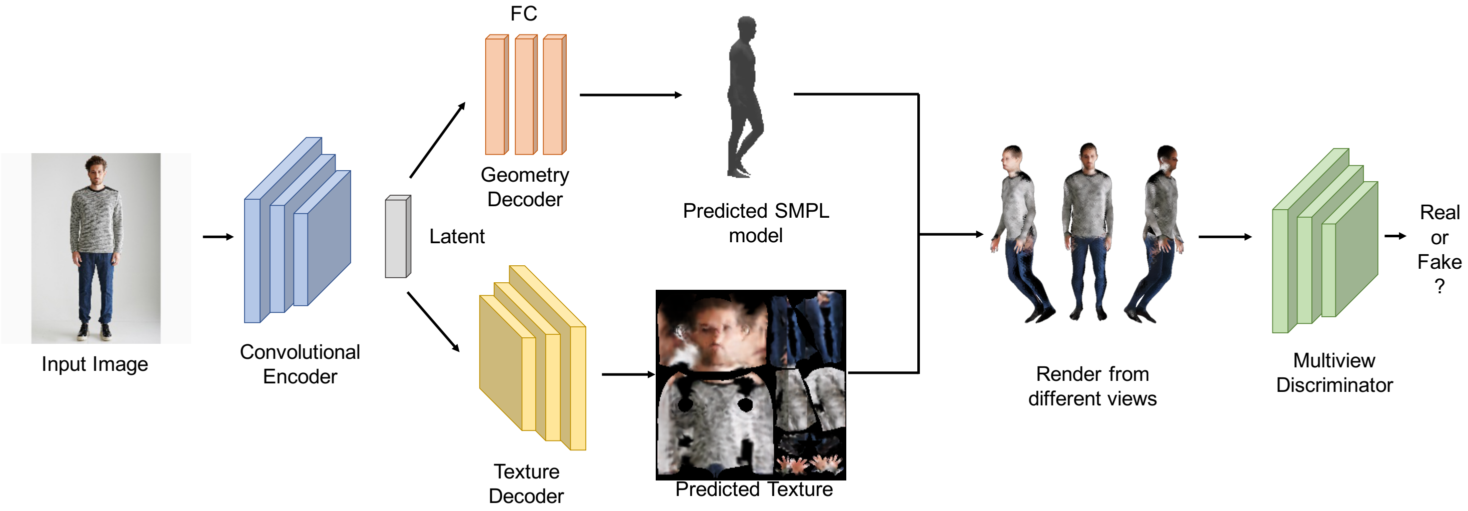
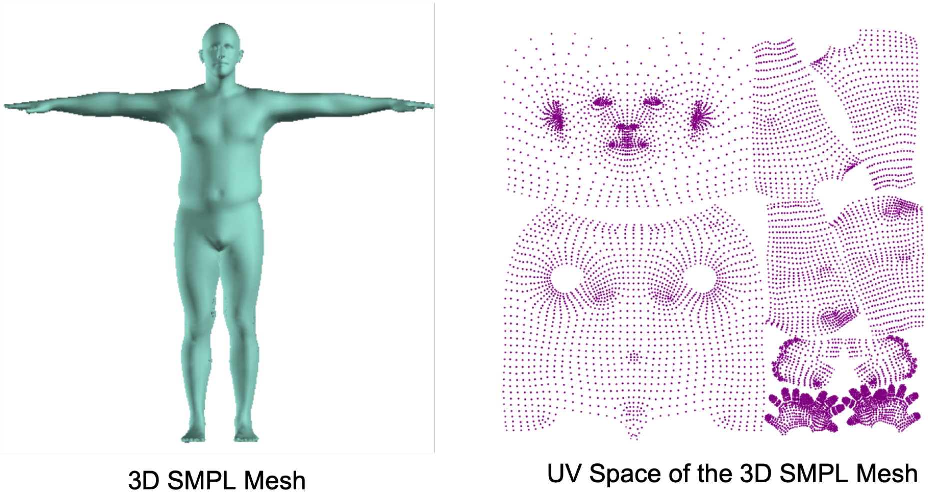


Figure 4.1: Training architecture. Our model first encodes the input image to a latent representation. This representation is then shared by *geometry decoder* and *texture decoder* to infer the geometric and visual attributes of the 3D model.

The prime focus of this thesis is to infer the texture of a fully articulated human mesh model. While previous literature attempts to infer the 3D pose or the 3D SMPL parameters from a single two-dimensional RGB image, in order to fully enable the use of these models for tasks such as virtual and augmented reality, data-augmentation *etc*, it is imperative that we also infer the visual attributes of a human model. Since what a person wears, their skin tone and other visual attributes are some of their most identifiable characteristics.

The use of a surface based representation for modeling the human body allows us to use their UV parameterization and apply different textures to the underlying 3D model. Given the UV parameterization, each vertex of the mesh has a unique UV coordinate in the texture map, and the color sampled at that coordinate is assigned to the vertex when rendering the mesh. Given this procedure, each vertex of the SMPL model has a fixed semantic meaning associated with it, *e.g.*, face, torso, legs, *etc*. The texture map also has static semantics associated with each UV coordinate. We therefore pose the task of predicting the texture of the human body model as predicting the UV texture map of the SMPL model.



3D SMPL Mesh

UV Space of the 3D SMPL Mesh

Figure 4.2: Figure showing the 3D SMPL model and the corresponding UV space. Each vertex v_i in the 3D mesh model gets mapped to a unique coordinate u_i in the UV space.

4.1 Incomplete Texture Map

As people have very rich and complex visual details associated with them, generating a texture map from scratch is a difficult task. This is especially true in our setting, where we only have a single image of the person and many parts of the body are non-visible or occluded. Even with texture flow [18], *i.e.*, learning a vector field that maps colors from the input image to the texture map, the learned flow often fails to assign pixels to the correct body part, which we elaborate upon this in Section 5.6.1. Thus, we propose to first generate an *incomplete texture map* Y_0 by de-rendering colors from the address parts of the person in the image to give the network some initial context. For each pixel p in Y_0, we find the UV face (u_1, u_2, u_3) enclosing p and compute the corresponding barycentric coordinates $(\alpha_1, \alpha_2, \alpha_3)$. u_i is the UV coordinate of vertex v_i in our mesh model. The color of p can be computed by $\sum_{i=1}^{3} \alpha_i c_i$, where c_i is the color of each vertex obtained by projecting v_i onto the input image using our predicted camera parameters and sampling color at the 2D coordinate. We note that in the above process only visible UV faces are considered and pixels belonging to non-visible faces will not be colored. To determine the visibility of a face, we use z-buffering to register each pixel in the final rendered image with the face that has the smallest z value.

4.2 Texture Decoder

With the incomplete texture map Y_0, we pose the prediction of the completed UV texture map Y as an image completion problem. We introduce a *texture decoder* T that takes the incomplete texture map and the latent representation ϕ of the input image $\tilde{I}$, and predicts the complete texture map Y of our humanoid mesh model inferred by our geometry decoder G. Hence, conditioned on the context provided by Y_0 and ϕ, the texture decoder T needs to predict the texture of body parts that are non-visible or occluded. Once we have the completed texture map Y, we use a differentiable renderer $\mathcal{R}$ [20] with the learned camera parameters to render the final image I. We use the proposed losses during the texture generation to penalize different forms of variation and ensure that our learned texture is visually plausible.

4.2.1 Reconstruction Loss.

We find the L_1 distance between the final rendered image I and the input image $\tilde{I}$:

$$\mathcal{L}_{L_1} = \|I - \tilde{I}\|_1 \,. \tag{4.1}$$

4.2.2 Perceptual Similarity Loss

To ensure that the generated image is visually similar to the original ground truth image, we use the perceptual similarity loss $\mathcal{L}_{\text{percep}}$. Perceptual loss computes the L_1 distance in the image feature space which is provided by deep neural networks [40].

This helps bring the statistics of the intermediate features of the real image and the predicted image close to each other and thus allows for visually similar texture generation,

$$\mathcal{L}_{\text{percep}} = \sum_{v=1}^{N} \|C^k(I) - C^k(\tilde{I})\|_1 \,, \tag{4.2}$$

where C^k is the k^{th} feature extractor of a deep network.

4.2.3 Multiview Curriculum Learning

While above losses encourage the network to learn the texture of the 3D model, they only provide stimulus for body parts that are visible in the input image. In order to estimate the complete UV texture map, our network also needs to hallucinate parts of the body that are either non-visible or occluded. Hence, we require a loss function that provides appropriate feedback to the network based on the texture inferred for these non-visible parts. However, such a loss is not straightforward due to the absence of any ground truth UV-texture and images of the same person in different poses.

Fortunately, the use of a differentiable renderer allows us to introduce a novel and widely applicable multiview curriculum learning (MCL) approach. MCL takes into account the fact that we can synthesize multiple views of the same person by utilizing multiple renderings, each parameterized by a different camera angle. MCL gradually increases the difficulty of the task assigned to the texture decoder in the following way. We first render the image with the predicted frontal camera view. Every 3000 iterations, we also render the image from two additional views by rotating the camera by δ degrees in two directions, until we cover a full 360 degrees perspective. For each iteration, rendered images from different views are sent to a discriminator network. The discriminator is tasked to differentiate generated images from the real input image. By doing so we are gradually making the network predict texture for the entire body without the need for actual multi-view input.

We use the patch-GAN discriminator introduced in [16] with LSGAN [27] objective during training. The loss function to train the discriminator thus becomes

$$
\begin{aligned}
\mathcal{L} = & \frac{1}{V} \sum_{i=1}^{V} \frac{1}{2} \mathbb{E}_{\tilde{I} \sim p_{\text{data}}} \left[(D(\tilde{I}))^2 \right] \\
& + \frac{1}{2} \mathbb{E}_{I_i \sim p(I_i)} \left[(D(I_i) - 1)^2 \right] \ ,
\end{aligned}
\tag{4.3}
$$

where V are the number of views that we consider for that iteration, and I_i is the image rendered from i^{th} view. The loss that is used to train the texture decoder T is

$$
\mathcal{L}_{\text{MCL}} = \frac{1}{V} \sum_{i=1}^{V} \frac{1}{2} \mathbb{E}_{I_i \sim p(I_i)} [(D(I_i)) - 1)^2] \, . \tag{4.4}
$$

Therefore, the total loss for the *texture decoder* is

$$
\mathcal{L}_{\text{texture}} = \mu_1 \mathcal{L}_{L_1} + \mu_2 \mathcal{L}_{\text{perceptual}} + \mu_3 \mathcal{L}_{\text{MCL}}, \tag{4.5}
$$

where μ_i denotes the relative weights of the different losses.

Chapter 5

Experimental Evaluation

The aim of our experiments is to show that our proposed approach is able to estimate a well textured articulated 3D mesh model of a person from a single monocular image. We also aim to show that our model is able to estimate textures for images which exhibit large variance in both appearance (visual attributes) and pose.

5.1 Datasets.

We evaluate our approach on three very diverse datasets: SURREAL [37]; Human3.6M [15]; and DeepFashion [25]. SURREAL is a large-scale dataset with realistic looking synthetically generated images of people. Since the SURREAL dataset contains ground truth UV texture maps, we use it as our initial dataset to compare our generated UV texture to the ground truth UV texture. To verify that our model is able to generate plausible textures under large pose-variance, we experiment on Human3.6M (H36M) dataset. H36M is a large-scale video dataset with annotated 3D human skeletons. H36M contains videos of various subjects performing daily activities which often results in very complex poses. Finally, to verify that our model is able to generate textures with large appearance diversity we use the DeepFashion (In-shop Clothes Retrieval Benchmark) dataset. DeepFashion consists of images of fashion models with very large visual diversity *e.g.*, different kind of clothing types, large variance in clothing colors *etc*. We believe all of the above datasets together reflect the large visual diversity observed in human images.

5.2 Training Details

5.2.1 Network Architecture

The input to our network is an RGB image of size 224×224. We use a network similar to ResNet-18 [12] to encode the input image to the latent variable $\phi \in \mathbb{R}^{300}$. For the *texture decoder*, using a fully-connected layer we first convert the latent representation ϕ to a 196 dimensional vector, which we reshape into a 14×14 2D map. This map is then channel-wise concatenated with a down-sampled incomplete UV texture map and follows a *cascaded refinement network* type architecture [5] which gradually increases the size of the generated texture from 14×14 to 224×224, hence generating a complete UV texture map. The *geometry decoder* consists of 2 fully-connected layers of size 512 each, and takes as input the latent representation ϕ and output an 85 dimensional vector containing the parameters for the SMPL as well as the camera. We pre-train the *geometry decoder* and fine-tune it while learning the *texture decoder* to get improved results. Learning both the *texture decoder* and *geometry decoder* from scratch is a difficult task as initially the geometry network does not predict plausible parameters for the SMPL model and camera, which results in an incorrect incomplete UV map, an essential input to our *texture decoder*.

5.2.2 Hyperparameters

The hyperparameters used for training the *geometry decoder* are $\lambda_1 = 2$, $\lambda_2 = 1$, $\lambda_3 = 1$, and for *texture decoder* the hyperparameters used are $\mu_1 = 1$, $\mu_2 = 10$, $\mu_3 = 0.1$ and $\mu_4 = 2$ (used only while training the Surreal dataset [37]). We train our model using Adam optimizer [21] with learning rate 3×10^{-4}, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We update the parameters of the multiview discriminator every 5 iterations of updating *texture* and *geometry decoders*. Our code base is in python and will be released after the review process is over.

5.3 Metrics

We perform separate quantitative analysis for the on the two decoders. Moreover we provide the complete UV texture and the final rendered textured 3D human mesh model for the qualitative assessment of the reader. We use the recently proposed perceptual similarity loss [8] between the final rendered image and the input image as a metric to quantitatively evaluate our approach. We emphasize here that although our primary focus is the estimation of texture, estimating the correct 3D pose and shape is a crucial part of our pipeline. For quantitative analysis of our geometry decoder we show that our model is able to accurately predict the SMPL model parameters and compare with [31] and [17] in Section 5.5. We provide the implementation details and the hyper-parameters used in the supplementary material.

5.4 Qualitative Analysis

Figure 5.1 and Figure 5.2 show the qualitative results for our experiments on the three datasets. Figure 5.2 shows multiple predictions for each dataset. The image on the left column is the given ground truth image, while each row represents a textured human model as predicted by our network, and viewed from different camera locations. As can be seen, for each dataset our model is able to infer meaningful texture for body parts that are not visible. Additionally, the geometry decoder is also able to predict the shape of the human mesh model accurately. Further, the predicted textures show that our model is able to infer various details in a person's visual appearance such as clothing patterns, skin color, *etc*.

For SURREAL dataset (Figure 5.2 first two rows), the results show that our learned model is able to predict meaningful texture for body parts that are not visible in the given image. For example, the network is able to hallucinate the complete face even though most of the face is not visible. Since the SURREAL dataset has ground truth UV maps available, we introduce a supervised loss on the generated UV texture maps which provides the network with a strong human-body based regularization. This regularization enforces a structural constraint on our model and allows it to hallucinate the unseen body parts correctly.

Figure 5.2 (middle two rows) shows the predicted results for DeepFastion dataset. Additionally, Figure 5.1 visualize the input image along with the incomplete texture created by de-rendering this image and the output texture as inferred by our model. We would like to note that the DeepFashion dataset has very high variance in clothing and does not contain any ground truth UV texture maps. Given these constraints, our network is still able to predict meaningful texture outputs, as can be seen in the above figure. The reason for this is twofold. First, given the incomplete UV image our network only has to inpaint the textures that are not visible in the input image.

Second, our multiview curriculum learning approach enforces appropriate body priors on the network. This ensures that different views of the same person do not visually diverge from each other.

Although our proposed approach is able to infer the textures for unseen body parts in a completely unsupervised manner, we do want to however point out a common failure case. Instead of perfectly hallucinating parts unseen body parts such as hairs, ears, , for the face, our network seems to capture similar details from different views. This is due to the fact that all the training images in our subset of DeepFashion dataset has the face of a person visible at all time. Thus, during multiview curriculum training, the discriminator learns this data dependent bias and forces the hallucination of a face for other side views, and hence has a propensity to generate multiple faces in the UV texture map. Finally, our visualization shows that unlike [30, 35] our method doesn't require a target image to change the view or the pose of the person, but can generate as many views of a textured model as we want due to our explicit 3D mesh representation of the human model.

Finally, we verify our proposed approach on the Human3.6M (H36M) dataset. H36M contains complex and highly-varied poses but little variation in the visual appearance of the subjects. Similar to DeepFashion, H36M lacks any ground truth UV texture annotations. Figure 5.2 (bottom two rows) visualizes our model prediction on H36M. As seen in the above figure, our model is able to correctly synthesize textures for people in two very different scenarios, , a back-view of the person and a complex crawling pose. Both of these poses are challenging for the model as a large part of the body is not visible. However, our network is still able to infer the overall texture of the person and certain nuances like type of clothing (T-shirt, shorts,) well. However, it is not able to reconstruct faces perfectly. We believe this is because faces are much more complex to reconstruct compared to other body parts, given the large amount of details they encapsulate. Despite this, without any supervision from a target pose or a ground truth texture map, our model is able to understand the texture semantics associated with different clothing types and colors accurately.

To further differentiate our method from 2D pose transfer [30,35] and approaches not designed for articulated objects [18], we animate our learned textured human models with control policies learned from motion clips using DeepMimic [33]. These animations are sketched in Figure 1.1 and are best seen in the video link provided[1].

[1] https://drive.google.com/open?id=1aPNxYS2BEY-sUrmGCYbKcj7JKbBYePx4

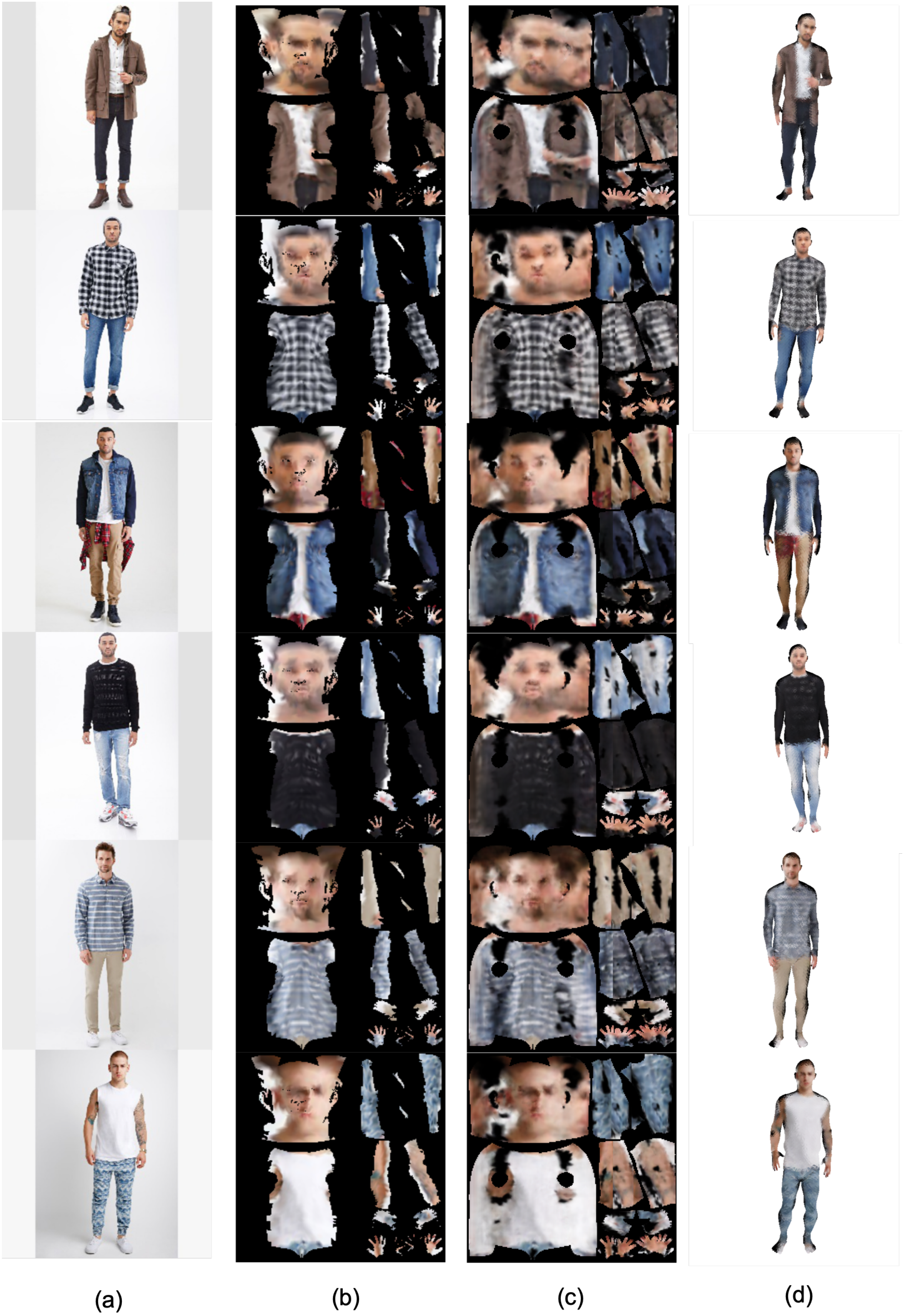


Figure 5.1: Results of our methodology with visualization of the intermediate results. (a) Input image. (b) Incomplete UV texture map. (c) Completed UV texture map. (d) Final output rendered using the predicted camera view.

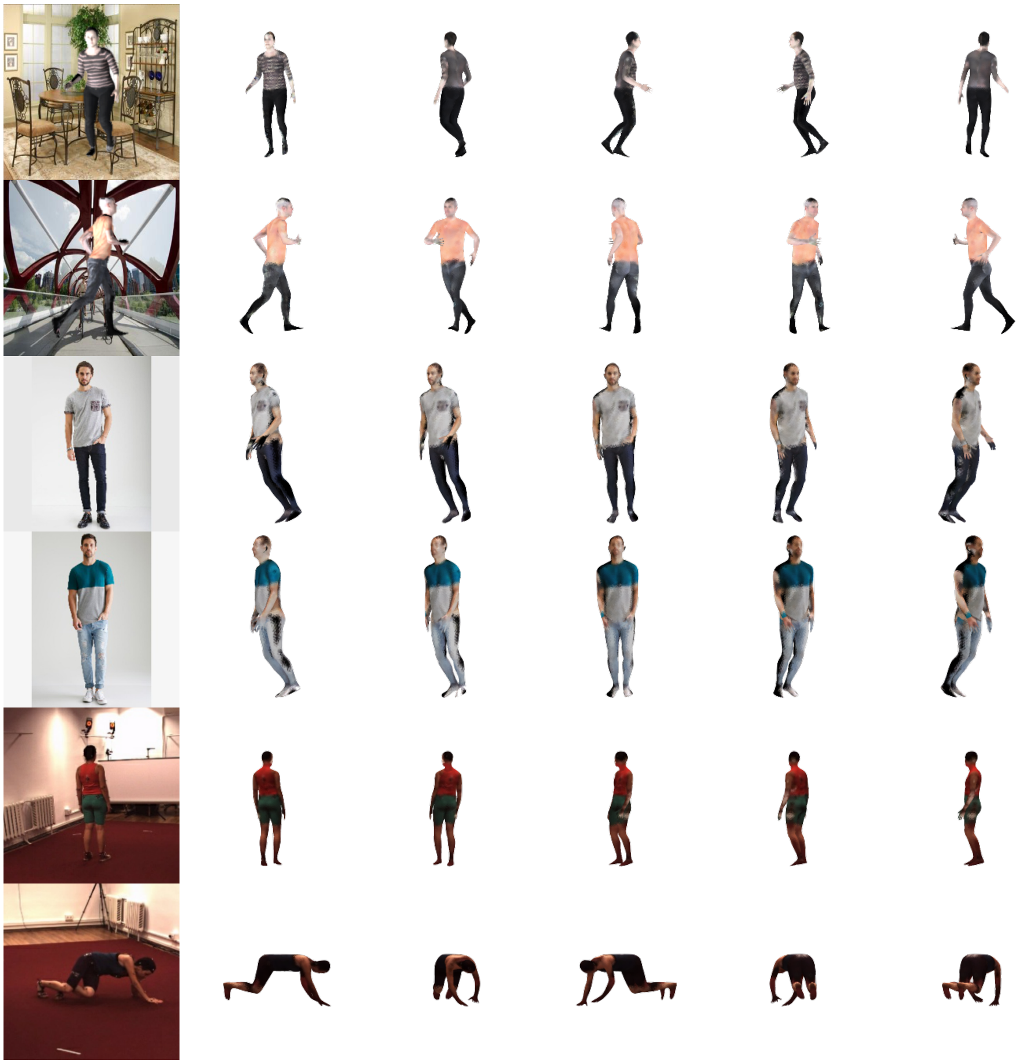


Figure 5.2: Results for multiple datasets. Input images (left column) followed by predicted textured models viewed from different angles.

5.5 Quantitative Results

Method	MPJPE
HMR (Trained on H3.6M)	77.6
Neural Body Fitting	**59.9**
Ours	80.3

Table 5.1: Evaluation of our geometry decoder with mean error per joint position (MPJPE).

Dataset	Ours	Texture Flow
Surreal	**0.114**	0.259
Fashion	**0.140**	0.316
Human3.6M	**0.216**	0.412

Table 5.2: Comparison of our method with texture flow [18]

Method	LPIPS
Ours	**0.140**
Ours w/o MCL	**0.143**
Ours w/o perceptual similarity	0.152
Ours w/o incomplete UV&MCL	0.159

Table 5.3: Ablation Study result

5.5.1 Texture Decoder

We use the Learned Perceptual Similarity metric [40] (version 0.1) for the quantitative assessment of our texture decoder. The LPIPS metric ranges between 0 to 1, with 0 being the most similar. To avoid any bias from the background, we calculate LPIPS on the masked input image which was obtained using Mask-RCNN [13]. Table 5.2 shows the LPIPS value of our network compared with the texture flow baseline [18]. As seen above, our method outperforms the baseline on all three datasets.

5.5.2 Geometry Decoder

First we compare our *geometric decoder* performance to two recent state-of-the-art works [17] and [31]. Since H36M contains complex poses and also has ground truth annotations available, we use it to evaluate our geometry decoder. Table 5.1 shows results for the mean per joint position error (MPJPE) in *mm* on the test set (S2, S3, S4, S10). As can be seen, our model performs reasonably compared with the other baselines. We would like to reiterate that our main aim is not to out-perform the above state-of-the-art methods. Hence, we only use direct supervision and avoid any additional loss such as the factorized adversarial loss [17] or use of semantic body part segmentation [31].

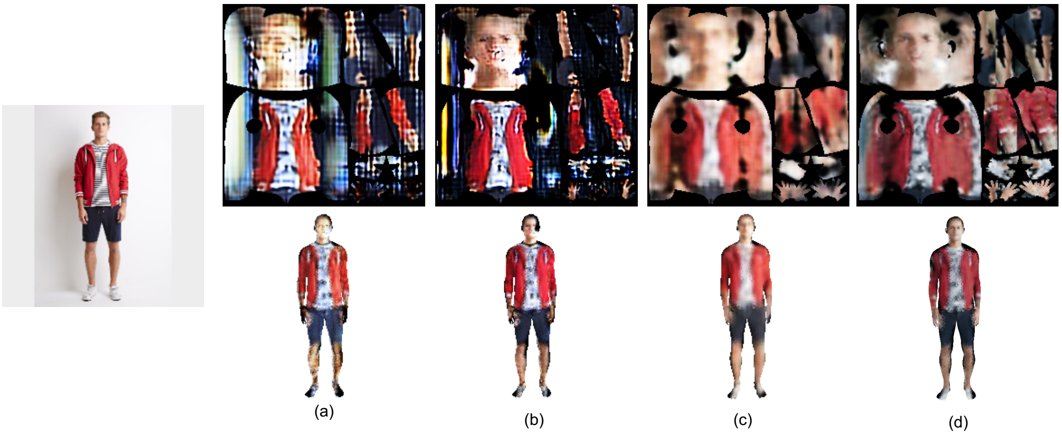


Figure 5.3: Results of the ablation study. (a) Input images. (b) Results without conditioning on incomplete UV and multiview curriculum learning. (c) Results without multiview curriculum learning. (d) Results without using perceptual similarity metric. (e) Results with our full method.

5.6 Texture Decoder Ablation Study

Table 5.3 and Figure 5.3 analyze and demonstrate the role of all the terms used for the generation of the textured model. We calculate the LPIPS metric between the original image and the rendered image using the predicted camera parameters. Figure 5.3 shows that $\mathcal{L}_{\text{perceptual}}$ is important to capture the overall structure of UV texture map. However, the two most important components responsible for generating realistic textures are: 1) conditioning the texture decoder on incomplete texture map and 2) using $\mathcal{L}_{\text{MCL}}$, , multiview curriculum learning to hallucinate the body parts that are not visible. As can be seen in Figure 5.3(b), when the images are not conditioned on the incomplete texture map, they fail to capture the important details of the person's visual appearance. Whereas, if there is no multiview curriculum learning, the network is completely unable to hallucinate the non-visible body parts of the person.

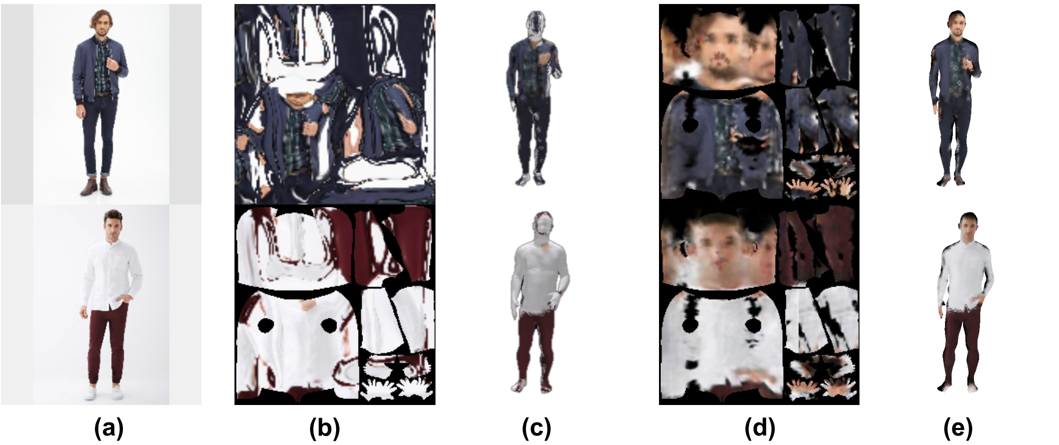


Figure 5.4: Figure comparing our our methodology with [18]. (a) Input images. (b) Predicted texture maps using texture flow. (c) Rendered images with Texture flow, (d) Predicted texture maps using our method. (e) Our rendered images.

5.6.1 Comparison with Texture Flow

The closest to our methodology is the work by [18]. As noted in Chapter 2, the scope of their texture generation is limited as they only generate textures for non-articulated objects such as birds and cars. [18] uses a flow-based vector field to map colors from the input image onto the predicted texture map. This is referred to as 'texture flow'. However, we found such a flow-based technique failed to produce reliable results. We believe this is because predicting the flow field for the entire human body is very challenging. Further, such a flow based technique cannot utilize the semantic correspondence between the input image and the texture map as our incomplete texture map does.

Another major difference between [18] and our proposed approach is the assumption of symmetric body parts in [18]. The authors in [18] assume symmetric body parts, which allows them to color the non-visible parts of the given object. However, such an assumption does not transfer to the human body, *e.g.* the front of the head is clearly not symmetric with the back. To alleviate this, we instead introduce a multiview curriculum learning approach, which forces the network to generate textures that renders consistent images from different camera positions.

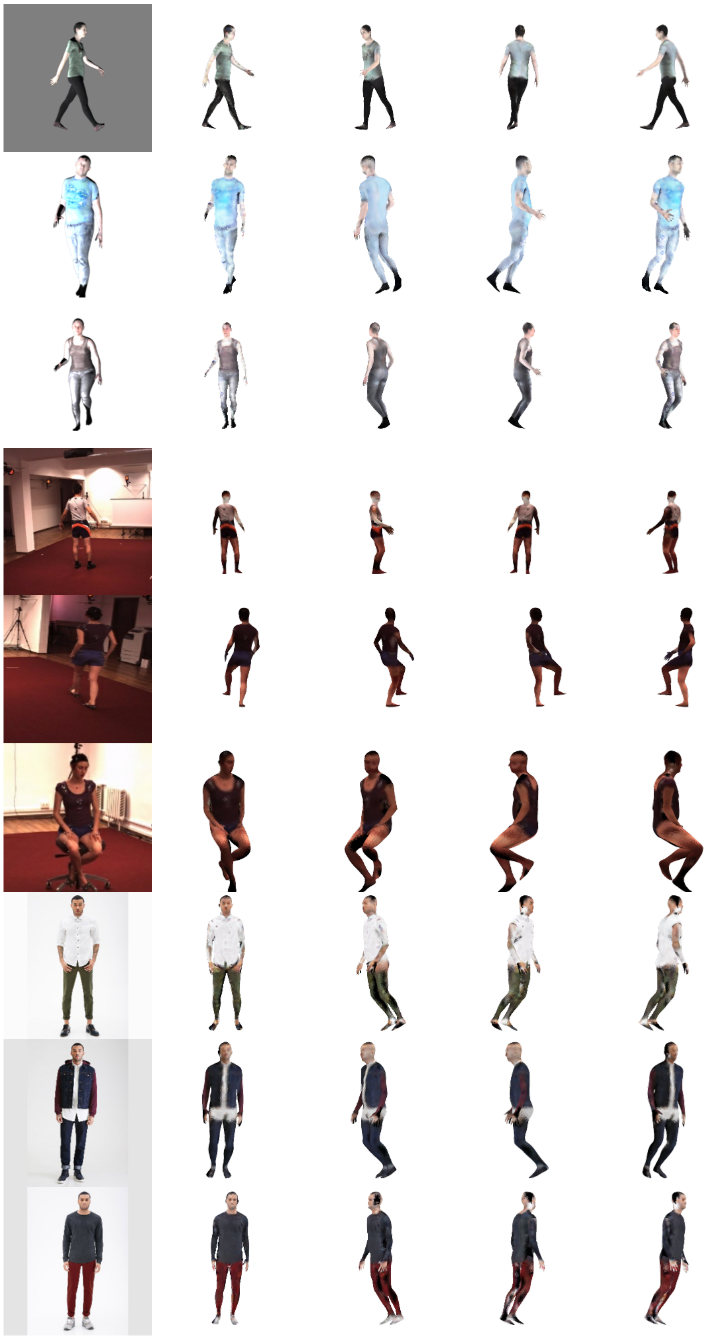


Figure 5.5: Additional results on Surreal, Human3.6M and Fashion datasets, each with three rows. The first column is the input image, the second column is the predicted model from the same viewing angle as the input image, and the other columns show the model from other viewing angles.

Chapter 6

Conclusion and Future Direction

We propose an approach towards single-shot inference of the complete texture of an articulated 3D human mesh model from a single monocular image. Our model consists of a geometry and a texture decoder, which are responsible for the 3D pose and shape, and texture inference respectively. To allow our model to hallucinate the unseen body parts from the input image, we introduce two novel techniques. First, we condition our texture decoder on an incomplete texture map, which is obtained via de-rendering the input image. Second, we introduce a curriculum based multiview learning approach which forces the network to output coherent textures for the entire human body. To evaluate our proposed approach, we experiment on multiple datasets. We report both qualitative and quantitative results, and show that our proposed approach is able to outperform the comparing baselines. Finally, we also show our textured articulated models are naturally compatible with physics-based animation, allowing us to make a person come alive from just a single static image.

While our methodology is able to learn appropriate textures that capture the visual details of a person, deforming the mesh vertices so that they match the clothing geometry was not the focus of this thesis. However, we feel that it is indeed an imperative step towards retrieving realistic 3D human models and an important direction for future work. Moreover, our methodology can prove to be extremely useful for tasks such as end-of-tail data augmentation. Given that we predict a 3D Human Model along with its texture, our methodology can be utilized to generate images/videos of people in various poses and performing multiple different actions from different views and perspectives. Such images and videos can therefore be utilized to augment existing datasets which can help better train models for applications like person re-identification, human action recognition, *etc*.

Bibliography

[1] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Detailed human avatars from monocular video. *arXiv preprint arXiv:1808.01338*, 2018.

[2] T. Alldieck, M. A. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll. Video based reconstruction of 3d people models. *arXiv preprint arXiv:1803.04758*, 2018.

[3] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[4] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016.

[5] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, page 3, 2017.

[6] Y. Chen, T.-K. Kim, and R. Cipolla. Inferring 3d shapes and deformations from single views. In *European Conference on Computer Vision*, pages 300–313. Springer, 2010.

[7] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1078–1085. IEEE, 2010.

[8] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] P. Guan, A. Weiss, A. O. Balan, and M. J. Black. Estimating human shape and pose from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1381–1388. IEEE, 2009.

[11] R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. *arXiv preprint arXiv:1802.00434*, 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. corr, vol. abs/1512.03385, 2015.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on Computer Vision*, pages 2980–2988. IEEE, 2017.

[14] J. F. Hughes, A. Van Dam, J. D. Foley, M. McGuire, S. K. Feiner, D. F. Sklar, and K. Akeley. *Computer graphics: principles and practice*. Pearson Education, 2014.

[15] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2014.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[17] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[18] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. *arXiv preprint arXiv:1803.07549*, 2018.

[19] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[20] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018.

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] C. Lassner, G. Pons-Moll, and P. V. Gehler. A generative model of people in clothing. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 6, 2017.

[24] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.

[25] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015.

[27] X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. arxiv preprint. *arXiv preprint ArXiv:1611.04076*, 2(5), 2016.

[28] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *International Conference on Computer Vision*, volume 1, page 5, 2017.

[29] R. Natsume, S. Saito, Z. Huang, W. Chen, C. Ma, H. Li, and S. Morishima. Siclope: Silhouette-based clothed people. *arXiv preprint arXiv:1901.00049*, 2018.

[30] N. Neverova, R. A. Güler, and I. Kokkinos. Dense pose transfer. *arXiv preprint arXiv:1809.01995*, 2018.

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

[32] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to estimate 3d human pose and shape from a single color image. *arXiv preprint arXiv:1805.04092*, 2018.

[33] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *arXiv preprint arXiv:1804.02717*, 2018.

[34] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[35] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe. Deformable gans for pose-based human image generation. In *CVPR 2018-Computer Vision and Pattern Recognition*, 2018.

[36] H.-Y. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki. Self-supervised learning of motion capture. In *Advances in Neural Information Processing Systems*, pages 5236–5246, 2017.

[37] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pages 4627–4635. IEEE, 2017.

[38] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[39] C. Weng, B. Curless, and I. Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. *CoRR*, abs/1812.02246, 2018.

[40] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.