Dynamic Appearance Modelling from Minimal Cameras

Lewis Bridgeman Jean-Yves Guillemaut Adrian Hilton
CVSSP, University of Surrey, UK
l.bridgeman, j.guillemaut, a.hilton@surrey.ac.uk

Figure 1: Our method generates a model of full-body texture appearance of a subject from a small multi-camera setup (3 in the above example), capable of expressing dynamic variation in texture with respect to body pose.

Abstract

We present a novel method for modelling dynamic texture appearance from a minimal set of cameras. Previous methods to capture the dynamic appearance of a human from multi-view video have relied on large, expensive camera setups, and typically store texture on a frame-by-frame basis. We fit a parameterised human body model to multi-view video from minimal cameras (as few as 3), and combine the partial texture observations from multiple viewpoints and frames in a learned framework to generate full-body textures with dynamic details given an input pose. Key to our method are our multi-band loss functions, which apply separate blending functions to the high and low spatial frequencies to reduce texture artefacts. We evaluate our method on a range of multi-view datasets, and show that our model is able to accurately produce full-body dynamic textures, even with only partial camera coverage. We demonstrate that our method outperforms other texture generation methods on minimal camera setups.

1. Introduction

Realistic human reconstruction and avatars generated from video have a wide range of application areas, from immersive content production for virtual and augmented reality experiences, to applications in the gaming and entertainment industries. However, reconstruction of humans from video is still an ongoing problem; the literature now focuses on generating increasingly detailed texture and geometry and from fewer cameras. Various works have addressed this problem, including methods that aim to produce human reconstructions or avatars from single images or monocular videos [3, 31, 23, 21, 35, 4, 5], and multi-view methods which rely on large multi-camera studios in a controlled environment [37, 14, 2].

Typically multi-view methods achieve a higher level of detail versus monocular methods, both in terms of geometry and texture quality. These methods can produce complete textures at every time instance, containing the variation in appearance caused by changing pose [37, 11]. However, these methods require large and expensive multi-camera studios, reducing the ease of acquiring this type of data. There has also been limited work addressing the problem of parameterizing the changing texture appearance.

Conversely, monocular methods are able to produce reasonable estimates of human shape given limited input [3, 21] and frequently make use of model-based reconstruction or recent advances in implicit-function human shape estimation [21, 35]. However, these methods only generate a single static texture map, which fails to express variation in appearance with respect to pose.

We present a method that sits between these two separate branches of work. Our method learns to generate whole-body dynamic texture appearance from a minimal set of camera viewpoints. We estimate a human shape reconstruction from multi-view video, and use texture observations on this mesh to train a dynamic texture appearance model. This model is capable of generating complete textures for any in-

put pose in the dataset, without providing complete texture supervision. The model is able to express the variations in human appearance with respect to pose, such as changes in illumination and wrinkling clothing. Furthermore, the model greatly compresses the textures while maintaining detail. Our contributions are as follows:

- Learning to generate full-body dynamic texture with partial visibility from a minimal set of camera views
- Reproduction of high-resolution dynamic texture from 3D pose input
- A novel training schema that exploits the separation of texture into high and low spatial frequency bands

Finally, we include a comparative evaluation with previous texture generation methods, demonstrating a quantitative and qualitative improvement in texture detail.

2. Related Work

There has been extensive work on estimating textured 3D human models from single images [3, 31, 23, 21, 35] and monocular video [4, 5]. Detailed textured reconstructions of humans in clothing are created from single images in [31, 35], however these methods are restricted to single poses. The methods in [3, 21, 23] use a learned approach to generate an animatable textured avatar, and make use of the SMPL model [29]. The meshes produced are high quality, as these methods estimate a more detailed surface beyond the level of detail prescribed by the SMPL model. However, the textures on the back of the model are only estimated by a neural network, and thus usually lack finer detail.

Monocular video of a human in motion is used to generate avatars in [4, 5]. These methods also use the SMPL model as the basis for their avatars, but combine multiple frames to refine the shape and texture estimates. These methods generate a static texture over the whole sequence, which lacks the dynamic appearance required of a realistic avatar. The methods in [40, 18] pre-compute a personalised template, which is fit to monocular video. This results in a detailed geometry, but these methods also fail to capture the dynamic variation in texture with respect to changing pose.

All of these methods are able to produce detailed geometry and texture, given limited camera views. However, there is a significant loss of detail in the texture in unseen regions. Additionally, none are able to produce a dynamically varying texture that captures the changes in shading and clothing appearance with varying body pose.

Multi-view methods can create more detailed reconstructions, but usually require constrained environments [36, 14, 17, 37, 20]. Extensive multi-camera setups are used for volumetric performance capture in [17, 14, 37], which can be used to generate highly detailed shape and texture reconstructions. However, these methods all assume that full texture coverage is available every frame, and do not model the dynamic texture with respect to pose.

The work in [20] targets volumetric performance capture from minimal cameras, by using deep learning to generate plausible geometry where traditional methods fail. However, they do not address the issue of generating full textures when the number of cameras has been reduced.

A neural texture appearance rendering method is presented in [36], which steps outside the traditional mesh-based pipeline. A network is trained on multi-view videos of a single subject, which learns to generate regions of texture, and learns a mapping of these regions into an image given an input 3D pose. This method is able to produce renderings of an avatar given a pose input, however, it fails to model dynamic texture appearance.

4D video textures [11] are a layered texture representation that compresses view-dependent variation in appearance. However it requires separate texture stacks each frame, meaning there is no temporal compression.

In [9], PCA is used to compress texture observations across different poses and camera views, using an optical flow solution to correct for misalignment between textures. This model is able to accurately reproduce textures from the input dataset, but there is no way to reproduce textures given an input pose. That is, appearance variation is parameterized with respect to the PCA basis functions.

The first use of variational autoencoders (VAEs) to model dynamic textures appears in [28]. This work focuses on face modelling, and uses a multi-camera rig to generate view-specific textures. Their conditional autoencoder learns to generate view-dependent texture maps for any input pair of viewing direction and latent vector. Their method uses a 40-camera rig, whereas our method aims to reduce the reliance on such extensive hardware.

Recent research uses GANs to predict posed face images from a single image of a face in a neutral pose [30]. The network is able to predict the texture for unseen regions like the inside of the mouth, but the results become increasingly deformed as the viewing angle differs from the input image.

The multi-view methods discussed can produce detailed texture appearance with complete coverage, but require a large number of cameras to do so. Few address the compression or modelling of the resultant textures, and none parameterise them with respect to body pose. Conversely, monocular methods are able to produce full-body texture, but these are static, and lack detail in unseen regions. Our work aims to address this gap in the literature with a model that expresses full-body pose-based variation in human appearance, generated from only a minimal set of viewpoints.

Multi-view texturing techniques combine observations from different cameras using a weighted average [15, 33, 7]. The weights are typically computed from various factors, including the size of the mesh region in each camera, or the angle between the surface normal and the camera di-

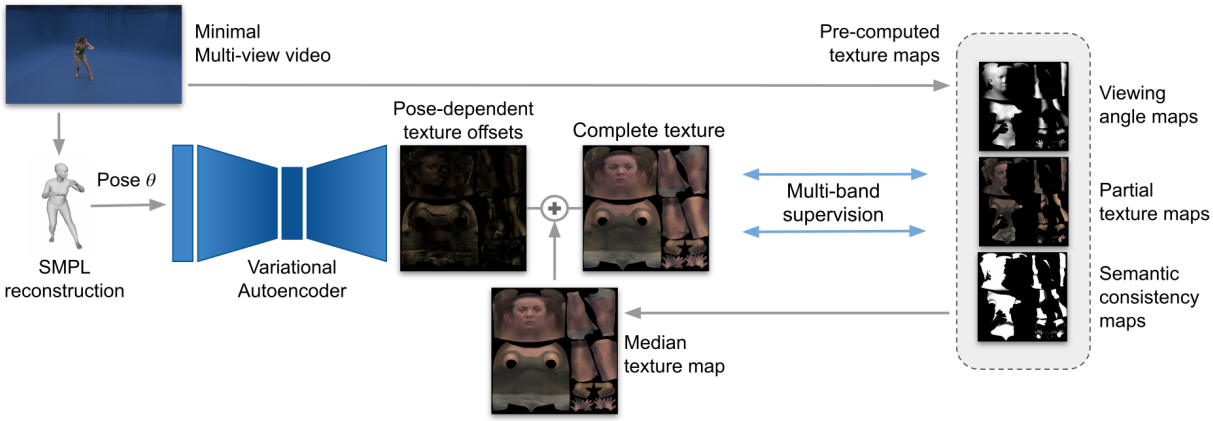

Figure 2: Overview of our method. We reconstruct the body shape from a minimal multi-camera setup. Our network is trained on partial texture observations to generate full dynamic texture maps for any pose in the input dataset. We employ a novel supervision method that exploits multi-frequency-band texture blending.

rection. When creating view-dependent textures, the angle between the camera direction and the viewing direction is also used [15, 38]. More sophisticated techniques are used in [6, 34, 32, 25, 41, 16] to alleviate artefacts caused by misaligned textures, different camera settings, or inaccurate geometry. The methods in [6, 13] employ a multi-band blending approach, whereby the images are separated into frequency bands, which are each weighted separately, which helps to reduce ghosting artefacts. The methods in [41, 16] correct misalignment by warping the images, using grid-based warping or optical flow respectively. However, these methods all require full camera coverage of the subject at each time-frame in order to generate complete textures. Additionally, these methods only produce textures on a per-frame basis.

In this work, we advance on traditional texturing techniques to produce a model of dynamic texture appearance in an approach that learns the full-body surface appearance from a minimal set of views, and synthesises dynamic texture with respect to pose for any pose in the input dataset.

3. Methodology

We introduce a method to generate a pose-driven model of dynamic human appearance from a minimal set of cameras. Our trained model can produce complete full-body dynamic textures for every pose in the dataset, even when complete texture supervision is not available. Using multi-view video of a single subject in a range of poses but with minimal camera coverage, we generate a temporally consistent reconstruction, from which we are able to produce partial texture maps from every camera. These are used to train a VAE that learns to output a complete texture for a given pose in the dataset, containing pose-dependent variation including shading and wrinkling of clothing. A full overview of our method can be seen in Figure 2. We describe the generation of training data in Section 3.1, and the method by which we train the network in Section 3.2.

3.1. Data Pre-processing

3.1.1 Model-based Human Body Reconstruction

Our method requires a temporally consistent texture map layout. To achieve this with a minimal camera setup, we use the SMPL model [29] to provide a coarse estimate of the body shape. The SMPL model is a statistical body model representing unclothed humans, parameterised by body shape β and pose θ. We employ a method based on [19] to align the SMPL model to pose detections in the input videos [10]. The energy function given in Equation 1 is minimised to align the SMPL model to the multi-view video sequence.

$$E_M(\beta,\theta,t) = E_J(\beta,\theta,t) + \lambda_\theta E_\theta(\theta) \\ +\lambda_\beta E_\beta(\beta) + \lambda_{DCT} E_{DCT}(\beta,\theta,t,C) \quad (1)$$

where θ and β are the SMPL pose and shape parameters respectively, and t is the root translation. We optimise a constant set of shape parameters over the whole sequence to ensure temporal consistency. E_J is the joint fitting term that minimises the distance from the projected SMPL model joints to their corresponding 2D joint estimates. E_{DCT} is a discrete cosine transform smoothing term, which minimises the distance between the reconstructed joint trajectories and a low-dimensional DCT approximation with 10 coefficients, C. Finally, E_θ is a Gaussian-mixture model pose prior, and E_β is a body-shape prior, as in [19]. The optimized meshes and pose parameters θ are used to train the pose-driven texture model.

3.1.2 Texture & Map Generation

We produce partial texture maps for every frame and camera view of the reconstructed sequences by projecting the input

(a) Partial texture (b) Viewing angle map (ω_{dot}) (c) Semantically segmented texture (d) Consensus semantic texture (e) Semantic consistency map (ω_{con})

Figure 3: The partial textures and maps used for supervision.

images on to the reconstructed body (Figure 3a).

In multi-view texturing algorithms [6, 34, 32, 25], the observations from different cameras are combined using a weighted blending function, which can depend on a range of parameters. This is because in practice there will be differences in the observed surface appearance between viewpoints, due to non-Lambertian lighting, and different camera settings, which can cause seams in the combined texture map. Additionally, inconsistencies between the estimated and ground truth surface can result in ghosting artefacts. We take inspiration from this approach in training our network, by generating maps that that represent the weights of the blending function, which we use to weight the losses between different texture observations (Section 3.2.2). We adopt a weighting function that favours texture observations on the faces of our mesh that are oriented towards the camera. We do this by means of generating a map that encodes the dot product between the normals of the mesh and the viewing direction of the camera (Figure 3b).

The texture maps contain misprojection artefacts, since the SMPL model is only capable of representing unclothed humans thus will not perfectly align with the input images. This means that the texture maps may contain regions of background texture, body parts may project onto each other, and textures from different cameras will not directly align (see Figure 4). To ensure that these erroneous regions are not used in supervision, we produce masks that eliminate semantic inconsistencies in the textures. We generate semantic body-part segmentation images [24], which we project onto the SMPL model to produce semantic texture maps (Figure 3c), which we one-hot encode to produce an S-channel map, where S is the number of semantic labels. We combine the semantic texture maps across all cameras and all frames into a single 'consensus' semantic map (Figure 3d) by computing a pixel-wise average. The 'consensus' map represents the 'probability' of each pixel belonging to a particular semantic label. We then produce semantic weight maps by computing a pixel-wise dot product between the 'consensus' map and the partial maps from each frame (Figure 3e). These maps allow us to down-weight the misprojected texture regions during training. A similar approach is taken in [4], in which clothing segmentation is utilised to

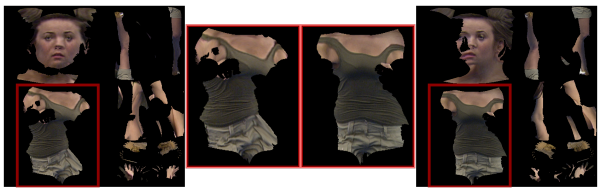

Figure 4: An example of misalignment between texture observations from two cameras.

reduce texture spilling. However, this approach generates a single clothing label for each pixel in their consensus map. By generating probabilities for multiple semantic labels, we are able to avoid significantly reducing the amount of supervision provided at the edges semantic regions, where the predicted semantic label tends to fluctuate.

Our final map is a visibility map, which denotes whether a point is observable from a particular camera. We combine our various maps into a single weighted map which we use during training:

$$\mathcal{W}_\theta^c = \omega_{dot,\theta}^c \odot \omega_{con,\theta}^c \odot \omega_{vis,\theta}^c \tag{2}$$

where ω_{dot} is the viewing-angle map, ω_{con} is the semantic consistency map, ω_{vis} is the visibility map, and $\odot$ denotes an element-wise multiplication.

We also use the combined weight maps $\mathcal{W}$ to generate an average texture map M, which is a weighted median across the partial texture maps from all cameras and frames.

3.2. Model & Training

Our pose-driven texture model is a variational encoder-decoder pair, which takes a 3D human body pose as input, and returns a map of pose-dependent texture offsets (Section 3.2.1). The offsets are added to a median texture computed over the whole dataset to produce a complete texture map with pose-dependent details. We supervise our network with the visible texture regions from each camera. To learn to reproduce high resolution texture detail from partially overlapping texture maps, we incorporate techniques from the image-based rendering and projective texturing literature into our loss functions, to teach the network to combine texture observations from different camera

images without losing high-frequency detail or introducing artefacts. We explain the network architecture in Section 3.2.1, discuss our dual-band filtering approach in Section 3.2.2, and we describe the loss functions used to train our network in Section 3.2.3.

3.2.1 Architecture

We employ a VAE for its smooth distribution within the latent space. This assists the network in filling unobserved texture regions, since it can benefit from the supervision of these regions in similar poses. Our network comprises an encoder which takes our input pose θ and produces a latent space $z(\theta)$, followed by a decoder, which returns a 512 $\times$ 512 offset texture map that is added to a pre-computed median texture. The encoder features 4 fully connected layers, which generate a 128-dimensional latent vector. The decoder uses 2 more linear layers, followed by 7 decoder blocks, each of which comprises an up-sampling function following by a residual block, with 3 $\times$ 3 convolutions throughout. Our pose input comprises joint rotation matrices concatenated with 3D joint-positions, with length 252. We elect to output a 512 $\times$ 512 texture map, as this is enough to capture the detail in our evaluation datasets.

3.2.2 Multi-banded Texture Blending

Multi-band blending is employed in [6], whereby the images are decomposed into high and low frequency bands, and different weights are used to blend each component. Multi-band blending helps to reduce seams in the low-frequencies while avoiding ghosting artefacts in the high-frequencies. In [6] the low-frequency band output is a weighted average of all images, and in the high-frequency band the output pixel values are taken from the image with the maximum blending weight. To apply multi-band blending in our loss function, we separate both the ground truth partial texture and the network output into two frequency bands. A partial convolution [26] is applied using a Gaussian kernel of size 21 $\times$ 21px and standard deviation σ of 4. A mask constrains the partial convolution to pixels within the observed partial texture map. The low-frequency texture is subtracted from its full-band counterpart to produce a high-frequency band texture. The blending function is then applied as a weighted loss. The low-frequency textures are weighted according to the viewing-angle map. However in a differentiable framework, we are unable to use a max operator on the high-frequency band as in [6]. Instead, we narrow the blending transition region in the high-frequencies by using an exponentiated version of the original viewing angle map. We find that an exponent of 3 produces the best results, providing a balance of reduced ghosting in high-frequencies without reducing high-frequency supervision in the blended regions altogether.

3.2.3 Loss Functions

In section 3.1.2 we described the generation of ground truth partial textures I^c_θ, where θ is the pose and c is the camera used to generate the texture. These partial textures are used as a ground truth for training our texture model, using our combined blending weight maps $\mathcal{W}^c_\theta$ (Equation 2) to determine their relative influence on the loss functions.

We use mean squared error (MSE) as our principal reconstruction loss, which we apply to the low and high frequency bands separately:

$$L_{MSE} = \sum_{\theta,c} \|\mathcal{W}^c_\theta \odot (\hat{O}_\theta + M - I^c_\theta)\|^2 \tag{3}$$

where $\hat{O}_\theta$ is the reconstructed texture offset for pose θ, which are added to the median texture M. I^c_θ is the partial texture observation for pose θ from camera c.

We also use a perceptual loss [22] whereby we minimise the difference between features extracted from the reconstructed and target texture. We extract features from a pre-trained VGG-16 network at layer relu3_3. This loss is given by:

$$L_{per} = \sum_{\theta,c} \|V(\mathcal{W}^c_\theta \odot (\hat{O}_\theta + M)) - V(\mathcal{W}^c_\theta \odot I^c_\theta)\|^2 \tag{4}$$

where V denotes the VGG-16 network [27]. Again, this loss is applied to the high and low frequency bands separately.

Our final reconstruction loss minimises the offset values produced by the network. This stops the resultant textures from deviating too far from the median texture. We implement this as another perceptual loss, given by:

$$L_{median} = \sum_{\theta,c} \|V(\hat{O}_\theta) - V(0)\|^2 \tag{5}$$

where 0 is a zero-valued image of the same shape as $\hat{O}_\theta$. We find that this extra supervision aids the network in producing plausible results in texture regions with comparatively less supervision.

Finally, we incorporate a KL-divergence loss, given by:

$$L_{KLD} = KL\Big(\mathcal{N}\big(\mu^z_\theta, \sigma^z_\theta\big) \| \mathcal{N}\big(0, \mathbf{I}\big)\Big) \tag{6}$$

where μ^z_θ and σ^z_θ are the latent mean and standard deviation for pose θ. Our total loss is then computed as:

$$\begin{aligned} L_{total} = \lambda_1 L_{MSE} + \lambda_2 L_{per} + \\ \lambda_3 L_{median} + \lambda_4 L_{KLD} \end{aligned} \tag{7}$$

In our experiments we use values of λ_1=300, λ_2=500, λ_3=15 and λ_4=1e-4.

Dataset	Sub-sequences	Frames
Roxanne [37]	10	461
Dan [11]	13	1286
Tomas [8]	4	214

Table 1: Properties of the three evaluation datasets.

Figure 5: Qualitative results on the Roxanne, Dan and Thomas datasets for the 3-camera arrangement. From left to right: ground truth image, Metashape [1], video inpainting [12], PCA model, static median texture, and our proposed method.

	Roxanne [37]			Dan [11]			Thomas [8]			Average		
Cameras	8	4	3	8	4	3	8	4	3	8	4	3
Static	0.544	0.537	0.530	0.417	0.415	0.416	0.633	0.625	0.615	0.531	0.525	0.520
PCA	0.555	0.465	0.506	0.450	0.424	0.410	0.670	0.631	0.617	0.558	0.507	0.511
Metashape [1]	0.678	0.559	0.514	0.559	0.452	0.396	0.725	0.599	0.559	0.654	0.537	0.490
Video inpainting [12]	**0.723**	0.565	0.523	**0.609**	**0.480**	0.422	**0.767**	0.608	0.564	**0.700**	**0.551**	0.503
Proposed	0.598	**0.566**	**0.549**	0.462	0.437	**0.426**	0.668	**0.633**	**0.618**	0.576	0.546	**0.531**

Table 2: The SSIM scores for all methods on the Roxanne, Dan and Thomas datasets, computed on a range of camera setups.

4. Experiments and Results

We evaluate our method on three public multi-view datasets, Roxanne [37], Dan [11], and Tomas [8]. Each dataset features multiple sub-sequences of a subject undergoing various actions. Details of each dataset are presented in Table 1. We select 8 equispaced cameras from each dataset to use as our baseline, from which we select smaller subsets for evaluation.

There are no methods in the literature, to our knowledge, that aim to address our specific problem of full dynamic texture generation from partial observations. Instead, we compare to four texture generation methods: Metashape [1], a commercial multi-view texturing software; a static median texture map; a PCA model; and a partial texture map that with video-based inpainting [12] used to fill in the unseen regions.

Metashape: Agisoft Metashape is a state-of-the-art commercial photogrammetry and texturing software. We use only its texturing capability, by providing it with camera calibration plus our generated meshes. Its texturing algorithm employs several processing steps to provide a high-quality reconstruction, including color calibration, texture mosaicing with multi-band blending, hole-filling and a ghosting filter. The algorithm performs well at dealing with artefacts caused by misaligned geometry, but it only generates textures on a per-frame basis, meaning it is unable to exploit other frames to fill unobserved regions.

Median Texture Map: We also compare against a single, static texture map, which is generated using observations over the whole dataset. This is computed as a weighted median, using the weighting maps described in Section 3.1.2. Although this method provides complete texture coverage, it does not account for the large changes in surface texture

with varying body pose.

PCA model: We build a PCA model with 8 components (matching the compression ratio of our trained model). The PCA model is built using the partial textures imputed with median pixel values. A PCA model is capable of compressing the textures across a dataset, but lacks the advantage of being driven by pose, and cannot learn to fill unseen regions with pose-dependent details.

Video Inpainting: This method combines the partial textures using a traditional multi-band blending technique. The unobserved regions in each frame are then filled using a video inpainting method [12]. The video inpainting method advances on standard inpainting networks by exploiting information from a window of frames. We apply the inpainting method to windows of 9 frames (the maximum our GPU allows).

4.1. Quantitative Evaluation

We evaluate on three datasets, with a varying number of cameras used as input. A separate network is trained for each subject and each camera arrangement. We evaluate on arrangements of 8, 4, and 3 cameras. In theory, our dynamic texture model could be trained using fewer, or even a single camera; in practice we find that our reconstruction method and state-of-the-art monocular methods, provide a reconstruction result that is too temporally unstable. Our method does not require complete texture observations in every pose, however we do require every surface point to be observed in at least one sub-sequence in order to generate a median texture. Therefore in datasets where the subject remains facing in one direction throughout, we change our camera selection between sub-sequences to provide texture coverage of every body part in at least one sequence.

We evaluate the full pipeline of reconstruction, texturing and model training for each subset of cameras. We use the structural similarity index (SSIM) [39] as an evaluation metric, to compare the reprojected textured models with the original input images. Thus the metric is a measure of both the quality of the reconstruction, as well as the quality of the dynamic texture appearance model. We compute the SSIM score only in the overlapping regions between the projected mesh and the ground truth segmentation. We evaluate against the cameras that were not used for training, meaning the metric represents the ability for the method to infer correct pose-dependent texture in unseen regions.

We use the mesh computed in our pipeline as the proxy geometry for all four texturing methods. We compute the SSIM scores on all three datasets, and for sets of 8, 4 and 3 cameras. The results are presented in Table 2.

These results show that for 3, and often 4 cameras, our method outperforms the other texture generation techniques. This is because our method is able to produce more plausible results in the unobserved texture regions. As the number of cameras increases, the Metashape and video inpainting methods achieve better results as they receive a fuller texture coverage but do not compress the textures as our method does. We consistently outperform the static texture method, showing that dynamic texture details are necessary to accurately depict a subject in varying pose. Despite having the same compression ratio, our model overwhelmingly outperforms the the PCA model.

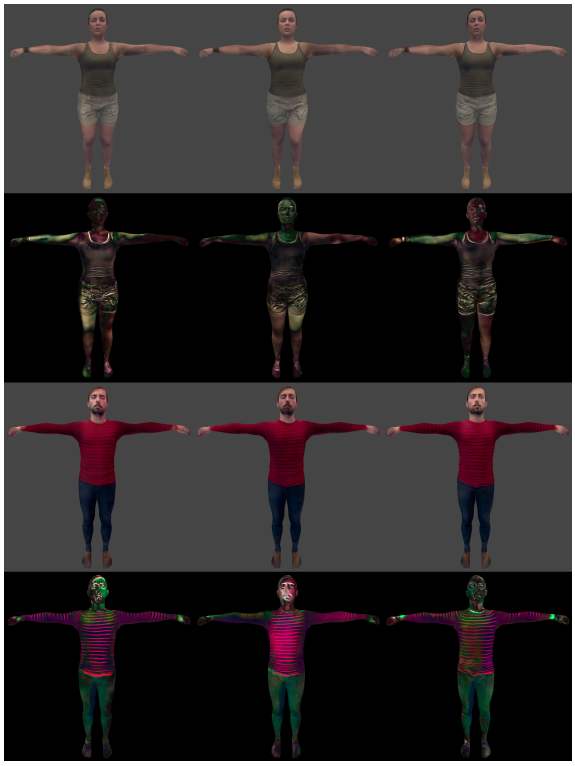

Figure 6: Dynamically varying texture maps in the Roxanne and Dan datasets, from 3 cameras. Our output textures, and the difference from the median texture.

4.2. Qualitative Results

We qualitatively compare our results against the four texture generation methods, as well as the ground truth images. These results are presented in Figure 5 for the 3-camera setup on all three datasets. The results demonstrate that our method is able to capture the dynamic clothing and shading details from only partial observations, and reproduce these given an input pose. Our method loses some high-frequency detail, which is a well known problem with VAEs, but manages to correct for misprojection and misalignment artefacts that are still present in other methods. The metashape textures contain artefacts where the algorithm has failed to detect misprojections or fill unseen regions; the video inpainting results have fewer artefacts, but the inpainted regions

are still not totally plausible. The static median method produces a complete texture without artefacts, but loses all pose-dependent variation and high-resolution detail. The PCA model has slightly more detail than the median texture, however the finer details are lost in comparison to our model. Overall, the results show that our method is the best at filling in unseen texture regions, while maintaining a high level of pose-dependent detail.

The dynamically varying texture appearance over a short sequence from the Thomas dataset [9] can be seen in Figure 1, which was generated from 3 cameras. The model is capable of expressing the variation in shading and wrinkling on the t-shirt with respect to the pose of the subject. Ours and the video inpainting method are the only approaches to exploit temporal information in generating textures for every frame. However, the video inpainting method relies on changing camera coverage in a short window of frames, whereas our model is able to exploit texture observations across the entire dataset. Further examples of dynamic variation in texture can be seen in Figure 6, although the dynamically varying texture effects are best viewed in our video, which is included in the supplementary material.

4.3. Compression

Our model is effective at expressing the dynamic texture appearance of a dataset, as shown in Section 4.1. However, it is also able to compress the dataset size significantly. We compute the size of each full dataset (one 512 $\times$ 512 texture map per frame), and compare it to our 12.5 MB texture model. The uncompressed dataset sizes and compression ratios are presented in Table 3. Additionally, inference of a complete texture map using our model only takes 4.5 ms on average on an NVIDIA 1080Ti GPU, making our method suitable for real-time playback of a sequence. We chose the number of components for our PCA model such that it matched the compression ratio of our model, however the results in Table 2 and Figure 5 show that our model produces much more detailed results.

Dataset	Frames	Size (MB)	Compression
Roxanne [37]	461	135	10.8
Dan [11]	1286	376	30
Tomas [8]	214	63	5

Table 3: Compression ratios of the three evaluation datasets using our model.

4.4. Ablation Study

We justify our multi-band loss function with an ablation study, the results of which are presented in Table 4 and Figure 7. As well as quantitatively improving results, Figure 7 clearly demonstrates how our multi-banded filtering helps to avoid seam artefacts in the network output.

Figure 7: Results on the Roxanne dataset with our multi-band filtering (left) and without (right).

Cameras	8	4	3
Proposed	**0.576**	**0.546**	**0.531**
Proposed w/o filtering	0.553	0.534	0.526

Table 4: Ablation study on multi-band filtering loss.

5. Discussion & Conclusions

We have proposed a novel method for the modelling of dynamic texture appearance of a subject captured from a minimal multi-camera setup. From as few as 3 cameras we are able to reconstruct their geometry, and generate full-body textures for every pose in the dataset that contain dynamically varying appearance, including wrinkling of clothing and changes in shading. Our method employs a novel multi-band blending loss function and blend-weight maps to reduce texture artefacts and preserve detail, and our quantitative and qualitative evaluation demonstrate that our method is capable of producing better results than previous texture generation methods.

Our model is capable of compressing the temporally varying textures of the datasets, and our total network takes up only 12.5 MB, allowing us to achieve high compression ratios. Furthermore, our network is capable of texture inference at faster than real-time, making it well-suited to real-time graphics applications.

The current network can generate textures for poses in the training dataset, and plausible textures for very similar poses. However it is likely that a larger dataset would be necessary in order to allow the network to generalise to a wider range of poses. Future work could address the limitations of VAEs, which are known for producing blurry outputs - this could pertain to investigating GANs or hybrid VAE-GANs. Currently we are using the unclothed SMPL model, which does not account for loose clothing or long hair, so future work could also incorporate a more detailed human shape estimation stage.

Acknowledgements

This work was funded by EPSRC Grant EP/N50977/1.

References

[1] *Agisoft Metashape Professional (Version 1.6.1) (Software)*, 2020. https://www.agisoft.com/downloads/installer/. 6

[2] Naveed Ahmed, Edilson de Aguiar, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. Automatic generation of personalized human avatars from multi-view video. In *Virtual Reality Software and Technology*, 2005. 1

[3] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2

[4] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *International Conference on 3D Vision*, 2018. 1, 2, 4

[5] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3D people models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[6] Adam Baumberg. Blending images for texturing 3D models. In *Proceedings of the British Machine Vision Conference*, 2002. 3, 4, 5

[7] Fausto Bernardini, Ioana M. Boier-Martin, and Holly E. Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Trans. Vis. Comput. Graph.*, 7:318–332, 2001. 2

[8] Adnane Boukhayma and Edmond Boyer. Video based animation synthesis with the essential graph. *International Conference on 3D Vision*, 2015. 5, 6, 8

[9] Adnane Boukhayma, Vagia Tsiminaki, Jean-Sébastien Franco, and Edmond Boyer. Eigen appearance maps of dynamic shapes. In *European Conference on Computer Vision*, 2016. 2, 8

[10] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[11] Dan Casas, Marco Volino, John P. Collomosse, and Adrian Hilton. 4D video textures for interactive character appearance. *Comput. Graph. Forum*, 33:371–380, 2014. 1, 2, 5, 6, 8

[12] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3D gated convolution and temporal patchgan. *In Proceedings of the International Conference on Computer Vision*, 2019. 6, 7

[13] Zhaolin Chen, Jun Zhou, Yisong Chen, and Guoping Wang. 3D texture mapping in multi-view reconstruction. In *ISVC*, 2012. 3

[14] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam G. Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Trans. Graph.*, 34:69:1–69:13, 2015. 1, 2

[15] Paul E. Debevec and Jagannath Malik. Modeling and rendering architecture from photographs. In *SIGGRAPH 1996*, 1996. 2, 3

[16] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson de Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. *Computer Graphics Forum (Proc. of Eurographics EG)*, 27(2):409–418, Apr 2008. Received the Best Student Paper Award at Eurographics 2008. 3

[17] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Fanello, Graham Fyffe, Christoph Rhemann, Jonathan Taylor, Paul Debevec, and Shahram Izadi. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Trans. Graph.*, 38(6), 2019. 2

[18] Marc Habermann, Weipeng Xu, M. Zollhöfer, Gerard Pons-Moll, and C. Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Trans. Graph.*, 38:14:1–14:17, 2019. 2

[19] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V. Gehler, Javier Romero, Ijaz Akhter, and Michael J. Black. Towards accurate marker-less human shape and pose estimation over time. In *International Conference on 3D Vision*, 2017. 3

[20] Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. Deep volumetric video from very sparse multi-view performance capture. In *European Conference on Computer Vision*, 2018. 2

[21] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[22] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 5

[23] Verica Lazova, Eldar Insafutdinov, and Gerard Pons-Moll. 360-degree textures of people in clothing from a single image. In *International Conference on 3D Vision*, 2019. 1, 2

[24] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *arXiv preprint arXiv:1910.09777*, 2019. 4

[25] Lifeng Wang, Sing Bing Kang, R. Szeliski, and Heung-Yeung Shum. Optimal texture map reconstruction from multiple views. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001. 3, 4

[26] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *European Conference on Computer Vision*, 2018. 5

[27] Shuying Liu and Weihong Deng. Very deep convolutional neural network based image classification using small training sample size. *3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015. 5

[28] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Trans. Graph.*, 37(4):68:1–68:13, 2018. 2

[29] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics*, 34(6):248:1–248:16, 2015. 2, 3

[30] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. Pagan: Real-time avatars using dynamic textures. 37(6), Dec. 2018. 2

[31] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2

[32] Wolfgang Niem and Jochen Wingbermühle. Automatic reconstruction of 3D objects using a mobile monoscopic camera. *Proceedings. International Conference on Recent Advances in 3D Digital Imaging and Modeling*, pages 173–180, 1997. 3, 4

[33] Eyal Ofek, Erez Shilat, Ari Rappoport, and Michael Werman. Multiresolution textures from image sequences. *IEEE Computer Graphics and Applications*, 17:18–29, 1997. 2

[34] Claudio Rocchini, Paolo Cignoni, Claudio Montani, and Roberto Scopigno. Multiple texture stitching and blending on 3D objects. In *Rendering Techniques*, 1999. 3, 4

[35] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2

[36] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, R S Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, I. M. Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, and Victor S. Lempitsky. Textured neural avatars. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2382–2392, 2019. 2

[37] J. Starck and A. Hilton. Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007. 1, 2, 5, 6, 8

[38] Jonathan Starck, Joe Kilner, and Adrian Hilton. A free-viewpoint video renderer. *Journal of Graphics, GPU, and Game Tools*, 14:57 – 72, 2009. 3

[39] Zhengjiang. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004. 7

[40] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. Monoperfcap: Human performance capture from monocular video. *ACM Trans. Graph.*, 37(2):27:1–27:15, May 2018. 2

[41] Qian-Yi Zhou and V. Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Transactions on Graphics (TOG)*, 33:1 – 10, 2014. 3