Procedural Humans for Computer Vision

Charlie Hewitt, Tadas Baltrušaitis, Erroll Wood, Lohit Petikam,
Louis Florentin and Hanz Cuevas Velasquez

Mesh Labs Cambridge, Microsoft

January 2023

Recent work has shown the benefits of synthetic data for use in computer vision, with applications ranging from autonomous driving [17, 18] to face landmark detection [20] and reconstruction [19]. There are a number of benefits of using synthetic data from privacy preservation and bias elimination [1, 12] to quality and feasibility of annotation [19]. Generating human-centered synthetic data is a particular challenge in terms of realism and domain-gap, though recent work has shown that effective machine learning models can be trained using synthetic face data alone [20]. We show that this can be extended to include the full body by building on the pipeline of Wood et al. [20] to generate synthetic images of humans in their entirety, with ground-truth annotations for computer vision applications.

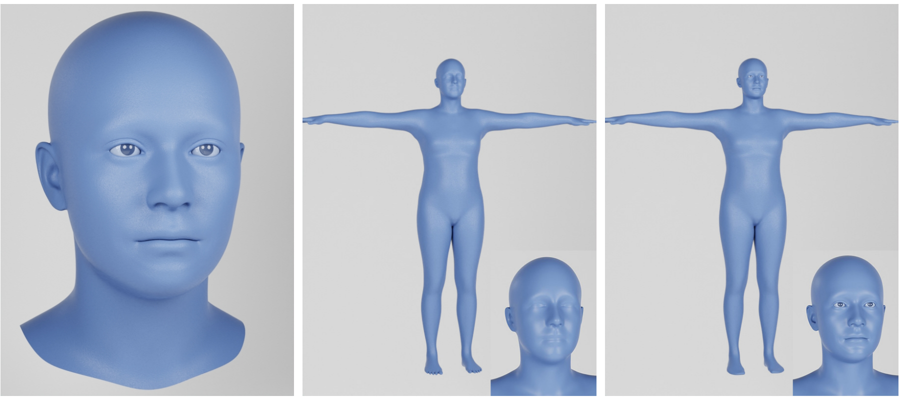
In this report we describe how we construct a parametric model of the face and body, including articulated hands; our rendering pipeline to generate realistic images of humans based on this body model; an approach for training DNNs to regress a dense set of landmarks covering the entire body; and a method for fitting our body model to dense landmarks predicted from multiple views.

1 Shape Model

1.1 Model Construction

Our body model combines the high fidelity face model of Wood et al. [20] with the popular body and hand model SMPL-H [14], which itself combines the articulated hands of MANO [14] with the SMPL body model [9]. So, we obtain a parametric model of the full human body with control of body shape and pose as in SMPL-H [14], and of facial and identity and expression as in Wood et al. [20], see Figure 1.

The body mesh is made up of $N = 12943$ vertices and 12726 polygons with a skeleton of $K = 54$ joints: 22 for the body (the SMPL skeleton [9]), 15 per hand (as in MANO/SMPL-H [14]) and 2 for the eyes.



(a) Head model of Wood et al. [20]

(b) SMPL-H body model [14]

(c) Our combined body and head model

Figure 1: Template meshes of the constituent models (left two) used in our combined shape model (right), with head region inset for full-body models. Our combined model has significantly higher fidelity on the face than the SMPL-H body model [14].

The body mesh vertex positions are defined by mesh generating function $\mathcal{M}(\vec{\gamma},\vec{\beta},\vec{\psi},\vec{\theta}):\mathbb{R}^{|\vec{\gamma}|+|\vec{\beta}|+|\vec{\psi}|+|\vec{\theta}|}\rightarrow\mathbb{R}^{N\times 3}$ which takes parameters $\vec{\gamma}\in\mathbb{R}^{|\vec{\gamma}|}$ for face identity, $\vec{\beta}\in\mathbb{R}^{|\vec{\beta}|}$ for body identity, $\vec{\psi}\in\mathbb{R}^{|\vec{\psi}|}$ for expression, and $\vec{\theta}\in\mathbb{R}^{K\times 3}$ for skeletal pose.

$$\mathcal{M}(\vec{\gamma},\vec{\beta},\vec{\psi},\vec{\theta}) = \mathcal{L}(\mathcal{T}(\vec{\gamma},\vec{\beta},\vec{\psi},\vec{\theta}),\vec{\theta},\mathcal{J}(\vec{\gamma},\vec{\beta});\mathbf{W})$$

where $\mathcal{L}(\mathbf{X},\vec{\theta},\mathbf{J};\mathbf{W})$ is a standard linear blend skinning (LBS) function that rotates vertex positions $\mathbf{X}\in\mathbb{R}^{N\times 3}$ about joint locations $\mathbf{J}\in\mathbb{R}^{K\times 3}$ by local joint rotations $\vec{\theta}$, with per-vertex hand-authored skinning weights $\mathbf{W}\in\mathbb{R}^{K\times N}$ determining how rotations are interpolated across the mesh.

$\mathcal{T}(\vec{\gamma},\vec{\beta},\vec{\psi},\vec{\theta}):\mathbb{R}^{|\vec{\beta}|+|\vec{\gamma}|+|\vec{\psi}|+|\vec{\theta}|}\rightarrow\mathbb{R}^{N\times 3}$ constructs an unposed body mesh by adding displacements to the template mesh $\overline{\mathbf{T}}\in\mathbb{R}^{N\times 3}$, which represents the average body in T-pose with neutral expression:

$$\mathcal{T}(\vec{\gamma},\vec{\beta},\vec{\psi},\vec{\theta})^{j}{}_{k} = \overline{T}^{j}{}_{k} + \gamma_i S^{ij}{}_{k} + \beta_i U^{ij}{}_{k} + \psi_i E^{ij}{}_{k} + P(\vec{\theta})^{j}{}_{k}$$

given linear face identity basis $\mathbf{S}\in\mathbb{R}^{|\vec{\gamma}|\times N\times 3}$, body identity basis $\mathbf{U}\in\mathbb{R}^{|\vec{\beta}|\times N\times 3}$, expression basis $\mathbf{E}\in\mathbb{R}^{|\vec{\psi}|\times N\times 3}$ and $P(\vec{\theta})$ which represents pose-dependent blendshape offsets for pose parameters $\vec{\theta}$ (see SMPL [9] for more details). Note the use of Einstein summation notation in this definition and below. Finally, $\mathcal{J}(\vec{\gamma},\vec{\beta}):\mathbb{R}^{|\vec{\gamma}|+|\vec{\beta}|}\rightarrow\mathbb{R}^{K\times 3}$ moves the joint locations to account for changes in identity:

$$\mathcal{J}(\vec{\gamma},\vec{\beta})^{j}{}_{k} = J(\overline{T}^{j}{}_{k} + \gamma_i S^{ij}{}_{k} + \beta_i U^{ij}{}_{k})$$

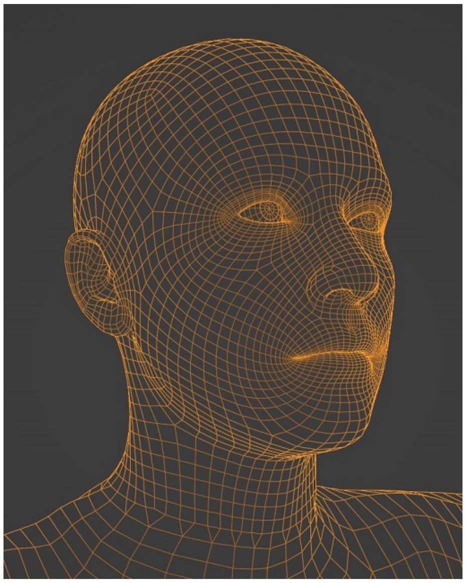


Figure 2: Head and neck topology of our body model.

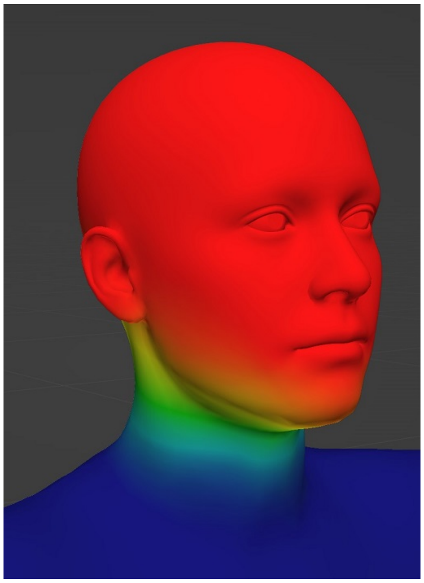


Figure 3: Head mask used for blending bases.

Where J is a modified version of the SMPL joint regressor, learnt as part of the SMPL model.

The face identity, $\mathbf{S}$, and expression $\mathbf{E}$ bases are those from Wood et al. [20], and the body identity basis $\mathbf{U}$ is that from the neutral SMPL-H model, which is a PCA basis learnt from scans of humans. The pose-dependent blendshapes are also taken from the neutral SMPL-H model.

The size of the face identity basis is $|\vec{\gamma}| = 260$, and the body identity basis is $|\vec{\beta}| = 300$. The expression basis has $|\vec{\psi}| = 224$ components.

1.1.1 Template Mesh

To construct the template mesh, $\overline{T}$, we manually align the template of Wood et al. [20] to the head of the SMPL template. Once aligned the head of SMPL and lower neck of the new head are removed and the two partial meshes merged. The topology around the join was hand-crafted to create a smooth transition given the different density of the two meshes, see Figure 2.

1.1.2 UV Space

To create the UV space we started from the SMPL UV space and manually aligned the UVs for the new head vertices with the original boundaries and features of the head in the SMPL UV space. This means that the UV space of SMPL and our model are functionally identical so textures can be reused directly. Elements that are not present in SMPL such as the eyes and mouth parts were added in previously unused areas of the UV space.

1.1.3 Basis Transfer

Given that we have changed the topology of the mesh significantly from both Wood et al. [20] and SMPL, we need to adapt all of the bases associated with these source models to work for ours. That is face and body identity, expression, pose-dependent blendshapes, skinning weights and the joint regressor.

We calculate a mapping function $\mathcal{F}_Q : \mathbb{R}^{|Q| \times m} \rightarrow \mathbb{R}^{N \times m}$ which transforms vertex data from a given topology Q to that of our model, where $|Q|$ is the number of vertices in model Q. Specifically, we calculate $\mathcal{F}_{head}$ and $\mathcal{F}_{smpl}$.

This mapping function $\mathcal{F}_Q$ is determined by finding, for every vertex in our template mesh $\overline{\mathbf{T}}$, the closest point on the surface of the template mesh of Q. This is then stored as barycentric coordinates on the triangle which that point lies within, we can then transform the input data for a given vertex in our model by taking the sum, weighted by barycentric coordinates, of the data from the vertices of that triangle in Q. This approach works because all data (identity basis, pose basis etc.) for all models is stored per vertex.

We can then apply this mapping function to bases directly to map them from the original head or body models to our model, e.g., the face identity basis: $\mathbf{S} = \mathcal{F}_{head}(\mathbf{S}_{head})$. Vertex groups can be mapped by creating a mask containing one where a vertex in the group and zero where it isn't, and applying $\mathcal{F}$ to this mask. Vertices are then determined to be in the vertex group for our model if the mapped value for the vertex is above some threshold value.

In order to prevent the head identity basis affecting the lower neck area we mask it to only apply to the head. Similarly, to prevent the SMPL identity basis and pose dependent blendshapes affecting the head, we mask these to only apply to the body. In order to preserve variation in head position due to body identity, we take the average of the SMPL identity basis over the masked area and apply it uniformly to all masked vertices. The mask is constructed by taking the SMPL skinning weights for the head joint, mapped to our model, and adding the eyes and mouth parts from Wood et al. [20], see Figure 3.

It is also not sufficient to simply map the SMPL joint regressor to our model using $\mathcal{F}$, as the per-joint regressor must sum to exactly one. Instead we calculate a one-to-one vertex mapping from SMPL to our model by taking the closest vertex pairing on the two template meshes. Given that our model is based on SMPL we get an exact match for all but the head and neck joints, where we get a very close approximation. As we have added additional joints for the eyes, we also need to construct eye joint regressors. We do this by simply taking the four extreme points of each eyeball in the x and y directions.

Similarly the skinning weights for the eyeballs need to be overridden, removing any influence from other joints and setting the influence of the applicable eye joint for all eyeball vertices to one. The mouth parts skinning weights are also overridden to completely follow the head joint, as the mapping function above can result in the neck joint having some influence.

1.2 Identity Sampling

To sample face identity we fit a multi-variate Gaussian to male, female and all identities in the training set of Wood et al. [20] (all gender labels are self-reported). This lets us randomly sample male, female and neutral (non-binary) facial identities.

For body identity, SMPL has three versions of the model: male, female and neutral but for our model we use just one, neutral, model. As such it is useful to be able to transfer shape parameters from gendered models to neutral, that is for gendered parameters $\vec{\beta}_g$, find neutral parameters $\vec{\beta}_n$ such that $\mathrm{SMPL}_g(\vec{\beta}_g) = \mathrm{SMPL}_n(\vec{\beta}_n)$ where SMPL_g is the mesh generating function for the SMPl model of gender g.

So for template vertices $\mathbf{T}$, shape basis $\mathbf{S}$ and shape parameters $\vec{\beta}$, we want to find template offset $\vec{o}$ and identity mapping $\mathbf{M}$ such that

$$\mathbf{T}_g + \vec{\beta}_g \cdot \mathbf{S}_g = \mathbf{T}_n + (\vec{o}_g + \vec{\beta}_g \cdot \mathbf{M}_g) \cdot \mathbf{S}_n$$

To do this we solve the two sub-problems finding the least squares solution for:

$$\begin{aligned} \mathbf{T}_g &= \mathbf{T}_n + \vec{o}_g \cdot \mathbf{S}_n \\ \mathbf{T}_g - \mathbf{T}_n &= \vec{o}_g \cdot \mathbf{S}_n \\ \vec{o} &= \mathbf{S}_n / (\mathbf{T}_g - \mathbf{T}_n) \end{aligned}$$

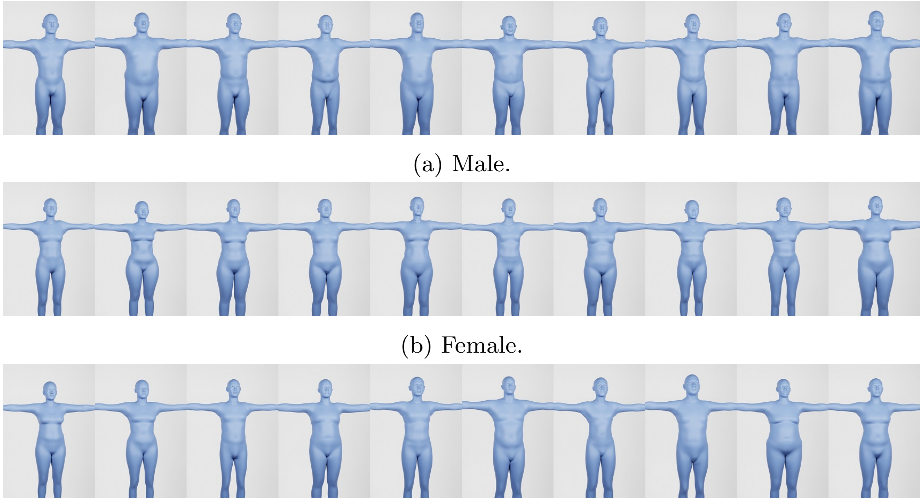
and

$$\begin{aligned} \vec{\beta}_g \cdot \mathbf{S}_g &= \vec{\beta}_g \cdot \mathbf{M}_g \cdot \mathbf{S}_n \\ \mathbf{S}_g &= \mathbf{M}_g \cdot \mathbf{S}_n \\ M_g^i &= \mathbf{S}_n / S_g^i \end{aligned}$$

for the latter we solve for each element of the shape basis, S_g^i individually.

So given this mapping information, $\vec{o}_g$ and $\mathbf{M}_g$, for both male and female SMPL models ($g \in [m, f]$) we can simply sample the unit normal to get identity $\vec{\beta}_g$ and for given gender, g, transfer to the neutral identity (usable by our model) $\vec{\beta}_n = \vec{o}_g + \vec{\beta}_g \cdot \mathbf{M}_g$.

We currently have no concept of dependence between the body and face identities, meaning there can be significant mismatch in the shape. In practice we find that sampling with coherent gender produces plausible results in most cases. Joint sampling of face and body identity could be an interesting direction for future work. Example sampled identities can be seen in Figure 4. In general we sample male, female and neutral identities in equal proportion to ensure we cover the gender spectrum sufficiently.



(a) Male.

(b) Female.

(c) Neutral.

Figure 4: Randomly sampled identities.

2 Rendering Pipeline

We build on the pipeline of Wood et al. [20] using the Cycles rendering engine [2]. Many elements are identical such as the hair and environment libraries. We additionally add a shadow-catching plane to integrate the subject better with the scene now that the legs/feet are included. Figure 5 shows some of the final renders from the pipeline.

Primary differences are the texture library used for the rest of the body, clothing for the body (which now needs to adapt to pose changes rather than being static as in Wood et al. [20]), and the pose library. Details of these additions are given in the following section. Figure 6 shows how we construct a synthetic image of a human from the component parts.

We are able to generate a wide variety of ground truth data from our pipeline, along with RGB images, in the same fashion as Wood et al. [20]. Figure 7 shows some example ground truth annotations for an image generated using our pipeline.

2.1 Textures

For the face we use the high quality skin texture library of Wood et al. [20], as shown in Figure 8a. For the body we use a set of 25 high quality textures from 3D body scans [16], as shown in Figure 8b. For each scan we extract albedo, displacement and an approximated bump map for high-frequency details in the SOMA UV space described above.

Sampling face and body textures independently can result in significant mis-

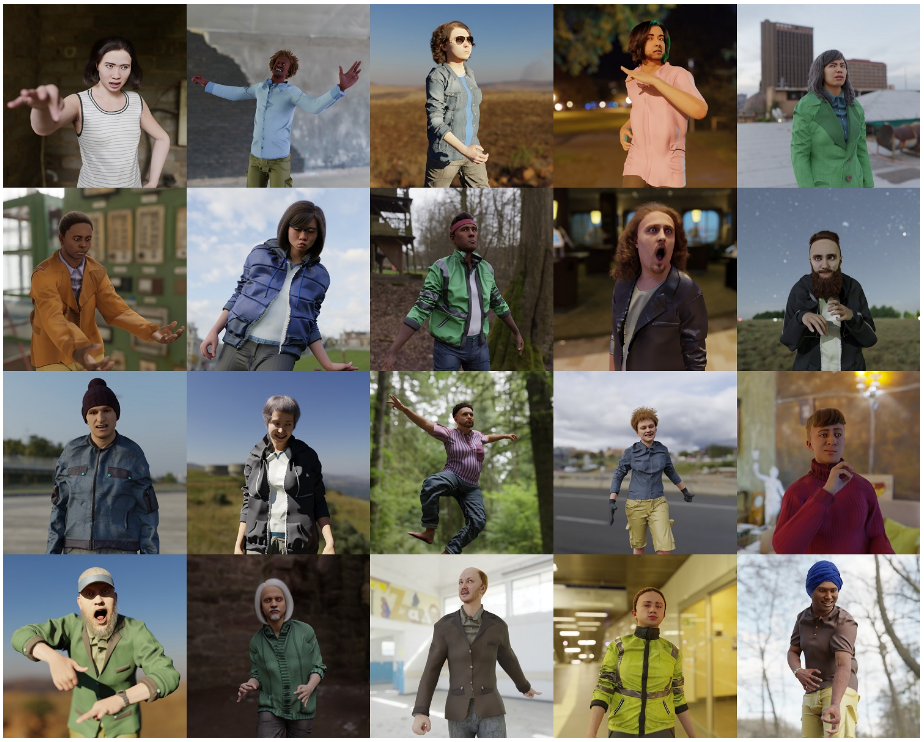
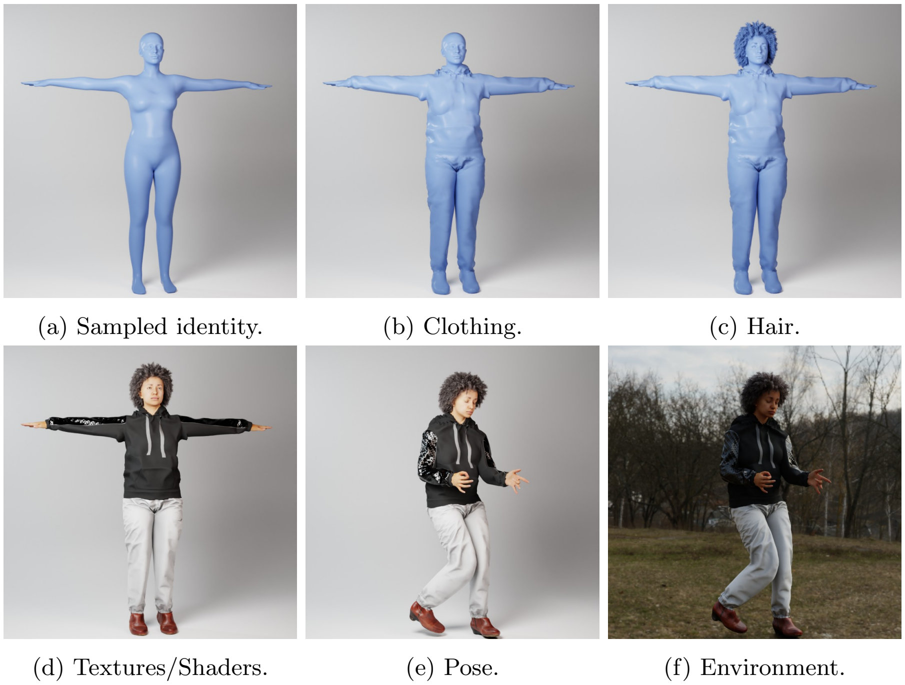


Figure 5: Example images generated using our human synthetics pipeline.


(a) Sampled identity.

(b) Clothing.

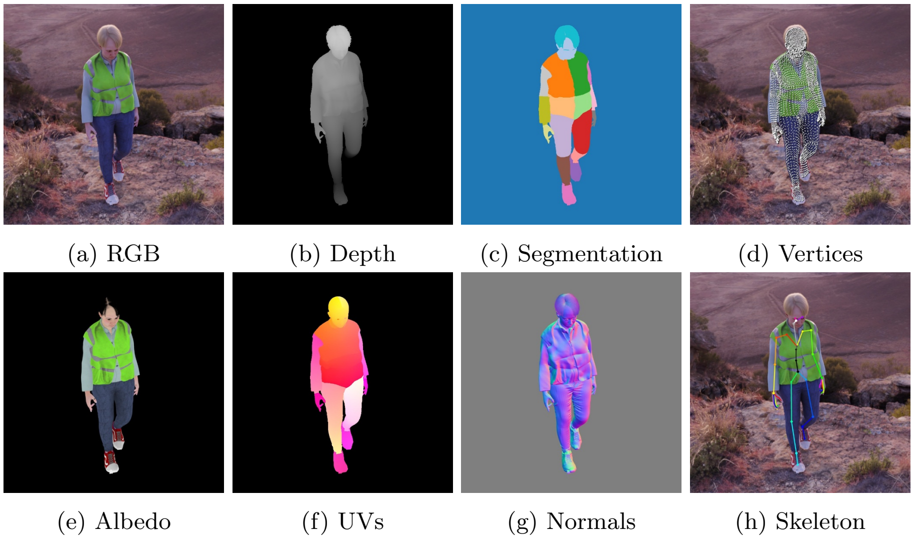
(c) Hair.

(d) Textures/Shaders.

(e) Pose.

(f) Environment.

Figure 6: Stages of our pipeline to construct a synthetic human.


(a) RGB

(b) Depth

(c) Segmentation

(d) Vertices

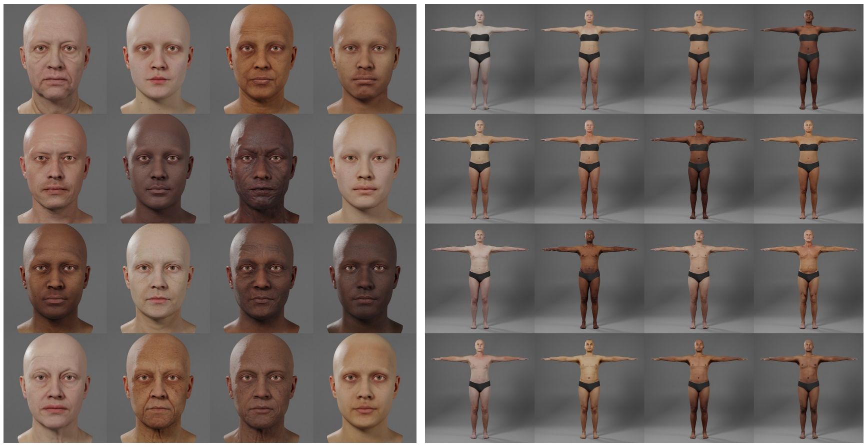
(e) Albedo

(f) UVs

(g) Normals

(h) Skeleton

Figure 7: Many ground truth label types can be generated using our pipeline.



(a) Face [20]. (b) Body.

Figure 8: Examples of skin textures from our library.

match in skin tone (see Figure 9a).

To address this we first sample a head texture from the library as our head texture library has greater diversity, then select a random body texture with average skin tone within some bound of perceptual similarity to that of the face using Equation 1 to determine perceptual color difference from input RGB values.

$$\Delta C = \sqrt{\left(2 + \frac{\bar{r}}{256}\right) + \Delta R^2 + 4 \times \Delta G^2 + \left(2 + \frac{255 + \bar{r}}{256}\right) \times \Delta B^2} \tag{1}$$

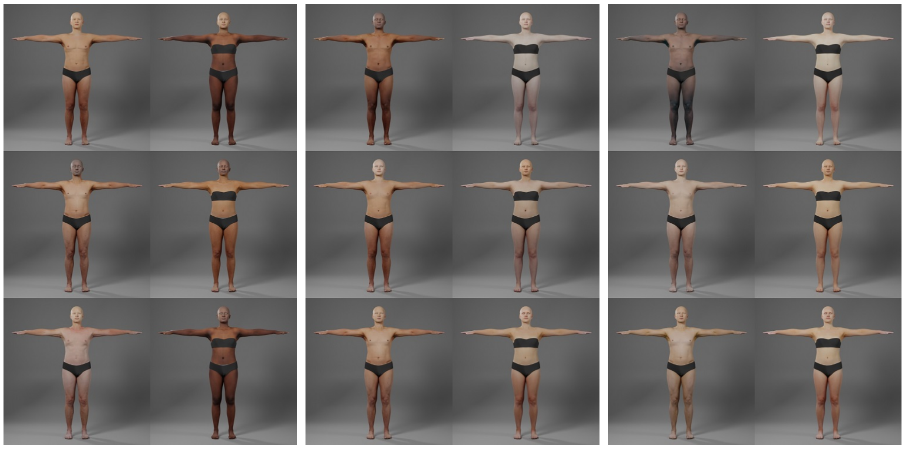
Where $\bar{r}$ is the average red component of the two colors.

In general this provides quite close matches in skin tone between face and body, though there are still minor mismatches (see Figure 9b). To address this we adjust the mean and variance of pixel values in the body texture to match that of the face texture, ensuring a quite precise match in skin tone of the body with the face (see Figure 9c).

2.2 Clothing

Wood et al. [20] use mesh based assets for adding clothing and accessories to face. For headwear, facewear (masks, eye-patches) and glasses the same technique can be used, simply parenting these assets to the head bone of the full body. But other clothing items must now adapt to the dynamic pose of the body. As such, we use displacement maps to model clothing items [10] We split these assets into tops and bottoms (including shoes), as well as using this technique to model some further accessories such as gloves, watches, bracelets and rings.

Dynamic subdivision lets us produce very high fidelity results using this technique. For each asset we author normal, roughness and metallic maps in



(a) Random sampling. (b) Filtered sampling. (c) Color adjustment.

Figure 9: Skin color matching process. We filter to achieve approximate matches when sampling then apply a further color correction.

addition to albedo and displacement, providing a high level of realism in terms of shading.

The clothing items are authored using Marvelous Designer [6] and displacement maps baked using Marmoset Toolbag [7]. Manual cleanup and material detail is authored in Substance Painter [5]. Examples of some of the assets in our displacement map clothing library can be seen in Figure 10.

There are a few significant shortcomings of this method, most obviously it is not possible to represent loose clothing using displacement maps. Furthermore, simulation is impossible; the clothing must directly follow the body mesh underneath it when animated. We find that displacement maps can give surprisingly compelling results for more than just very tight-fitting clothing as one might expect, but cannot be used for items such as dresses and skirts, and items like ties or jackets do not behave realistically in certain poses. To address these issues we plan to incorporate mesh based clothing in the future, along with cloth simulation.

2.3 Pose Library

For the face, the expression library of Wood et al. [20] is used. For the body we use the AMASS dataset [11] as an initial pose library. To this we add data collected using a motion-capture stage and processed using MoSh [8]. Some of this motion-capture data is targeted specifically to fill gaps in the existing library such as poses with crossed legs. In total our body pose library contains over 2 million frames at 30 fps, so approximately 19 hours of motion data.

In some cases the pose data is captured including articulated hand pose, but in many cases this is missing. For frames without hand pose we randomly sample

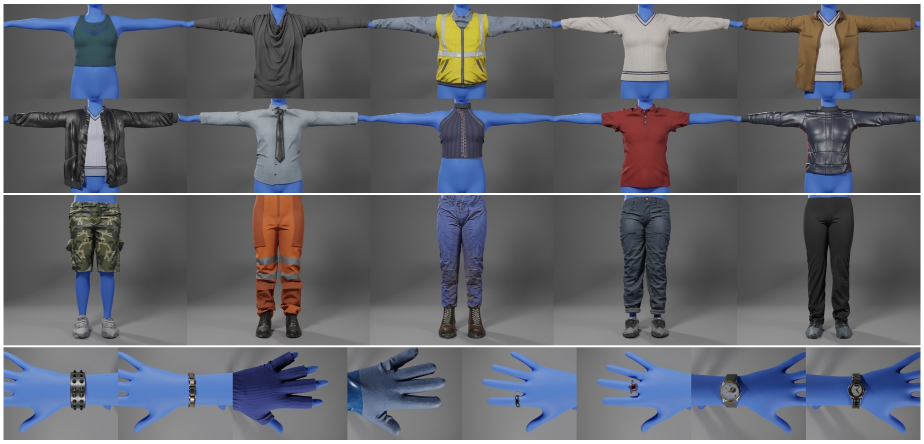


Figure 10: Examples of displacement clothing from our library including tops, bottoms, bracelets, gloves, ring and watches.

poses for each hand from the MANO dataset [14] and splice these on. Face expression (and eye pose) and body pose are sampled independently and spliced together. In general we find this produces very plausible results, particularly for single-frame (i.e., non-sequential) data.

When collecting motion-capture data it is common to start (and end) the motion sequence in a canonical pose, often T-pose. As a result we found that when sampling poses uniformly we had a very high occurrence of these T-poses, as such we use a Gaussian mixture model (GMM) to classify poses into a set of coarse classes one of which is T-pose. This allows us to significantly down-weight T-poses in our resulting samples.

In addition, we found relatively neutral, standing poses were common and typically not useful when it comes to training DNNs for downstream tasks such as landmark detection. Consequently, we also up-weight frames with higher mean absolute joint angles, i.e., frames which we consider to have more 'interesting' poses. We employ a similar approach for sampling facial expressions from the expression library of Wood et al. [20], weighted by mean blendshape activation. Finally, we randomly mirror body poses to effectively double the number of unique poses.

Examples of poses sampled from our library using the above technique are shown in Figure 11.

3 Landmark Regression

Perhaps one of the most common use-cases for this kind of human-centred visual data is detection and tracking of people within images. As such, we define landmark definitions corresponding to vertices of our body model defined in section 1. A sparse definition of just 36 landmarks (Figure 12a) which is used

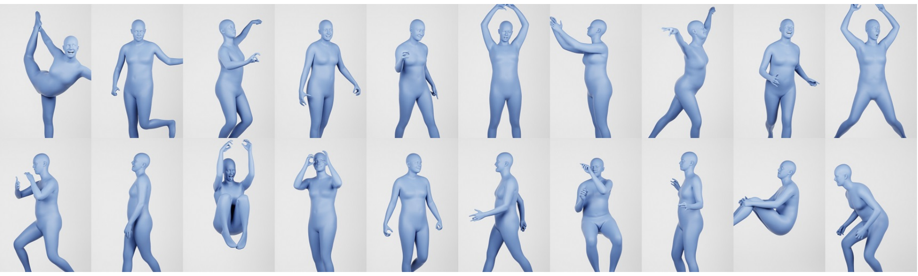
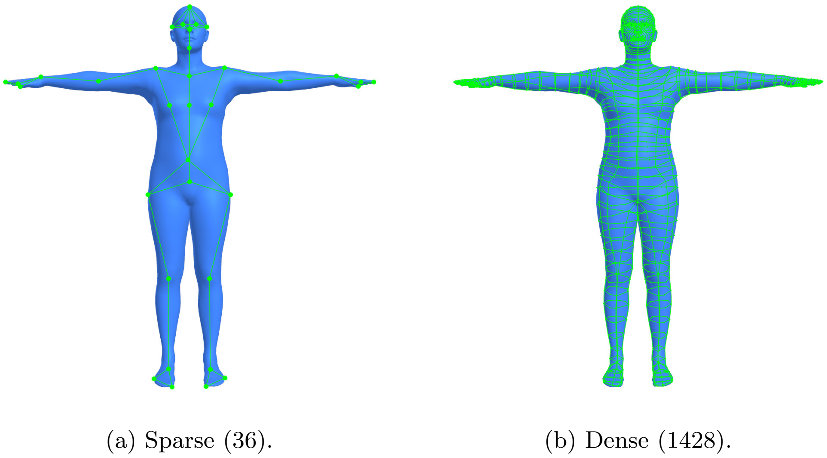


Figure 11: Example poses sampled from our body pose library with spliced facial expressions from Wood et al. [20] and hand poses from MANO [14] in some frames.



(a) Sparse (36).

(b) Dense (1428).

Figure 12: Full-body landmark sets used for (a) detection and (b) dense tracking.

for detection and tracking with the sliding window approach outlined by Wood et al. [19]. As well as a dense definition of 1428 landmarks (Figure 12b) used for model fitting, see section 4. Using these definitions it is trivial to generate 2D landmark annotations for our synthetic data using the vertex location outputs (Figure 7d).

We render a dataset of 100,000 images containing a single person, with 20,000 identities and 5 frames per identity, using the pipeline outlined in section 2. Each frame contains different pose and environmental lighting to increase the diversity of the data. An example of such an input image used for training is shown in Figure 7a above.

To regress sparse landmarks we train a MobileNetV2 [15] model, for dense landmarks we train a ResNet101 [4] model, both with 256 × 256 pixel input image size. In both cases we train using the procedure of Wood et al. [19]

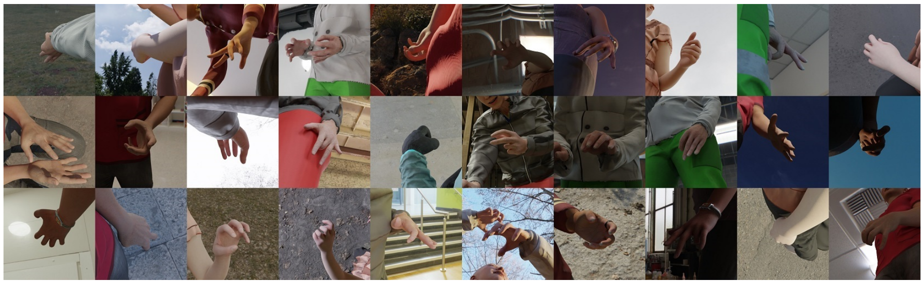


Figure 13: Sample images from our hand dataset.

with Gaussian Negative Log Likelihood (GNLL) loss and heavy use of data augmentation techniques. The models therefore predict 2D landmark positions as well as per-landmark uncertainty values.

3.1 Hand and face sub-networks

When predicting landmarks for the full body as described above we find performance for the hands is poor. This is not surprising given how small the hand is in the 256 pixel ROI. The shape of the hand and how it moves also result in high levels of self-occlusion, making this task especially challenging. Consequently, we train dedicated DNNs for hand landmark prediction using a ROI including just the hand as input.

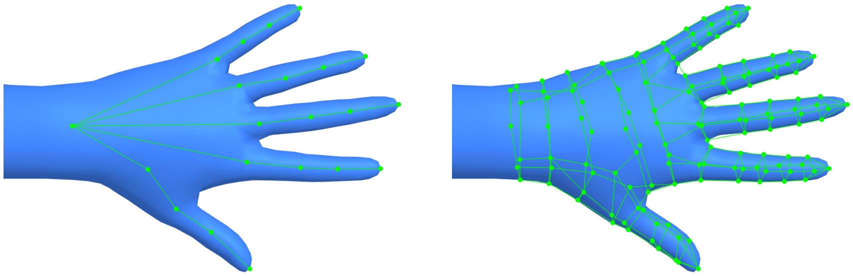
We generate a dataset of 100,000 synthetic images cropped to include just the left hand, examples are shown in Figure 13. We also define sparse and dense landmark definitions for just the hands shown in Figure 14. In the dense case the hand landmarks are a subset of the full-body definition in Figure 12b meaning they have direct correspondence.

Again we train using the procedure of Wood et al. [19], using MobileNetV2 [15] in the sparse case and ResNet18 [4] in the dense case with 128 × 128 pixel input image size. We increase the amount of rotation augmentations used to further increase data diversity. We also increase the frequency of motion blur augmentation to match observations in real data due to the typically faster motion of the hands than other body parts.

At run-time we first predict full-body landmarks and use these to extract and approximate ROI around the hand. We then use the sparse DNN to iteratively refine the ROI and finally run the dense DNN to get output landmarks. Due to the direct correspondence in the dense definition we can overwrite the hand landmarks from the initial prediction, interpolating at the wrist.

As our network has only seen left hands, and our initial full-body prediction disambiguates the left and right hands, we simply mirror the ROI for the right hand and input it to our left hand landmark DNNs. The returned landmarks are then mirrored back before use.

Similar to hands, we find that face landmarks are also not predicted accurately when regressing full-body landmarks with a single DNN. This is again



(a) Sparse hand (21).

(b) Dense hand (141) .

Figure 14: Hand landmark sets used for (a) detection and (b) dense tracking.

likely to be due to the small size of the face in the 256 pixel ROI used in those models. As such, we also take a dedicated face ROI and use the DNN of Wood et al. [19] to regress accurate face landmarks. As the face model used is the same, the face landmarks also retain a direct correspondence, so we can again overwrite those of the initial prediction.

3.2 Results

Some results of our dense full body landmark prediction method (including hand sub-networks) are shown in Figure 15. We are able to deal with a large range of pose, shape, appearance and environment. In some cases we are even able to deal with loose clothing, children and prosthetic limbs despite these not being modelled in our synthetic data. Particularly useful is the ability of the network to predict plausible landmarks with high uncertainty in cases of partial occlusion, like when one arm is totally hidden by the body.

Examples of failures of our method are shown in Figure 16. We particularly struggle with extreme poses, heavy (self-)occlusion, loose clothing, missing and prosthetic limbs. This is likely because many of these elements are not currently modelled explicitly in our synthetics data generation pipeline. In fact, missing limbs cannot even be represented by our parametric body model described in section 1. However, due to our use of GNLL loss and resulting prediction of uncertainties, we typically get an accurate indication of where these errors are occurring, as demonstrated in many of these examples. Future work on our shape model and rendering pipeline will aim to fill these gaps in representation of real world data.

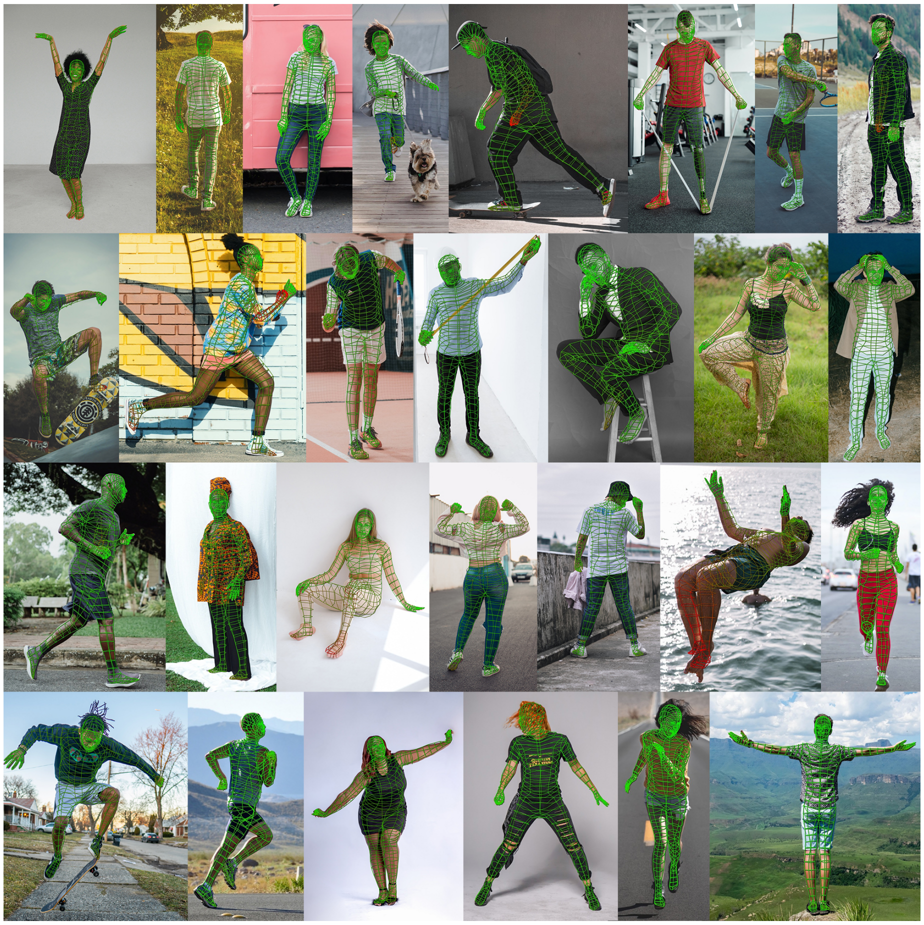


Figure 15: Dense landmark tracking results. Confidence is colour coded, with green being high and red being low. Images collected from https://pexels.com.

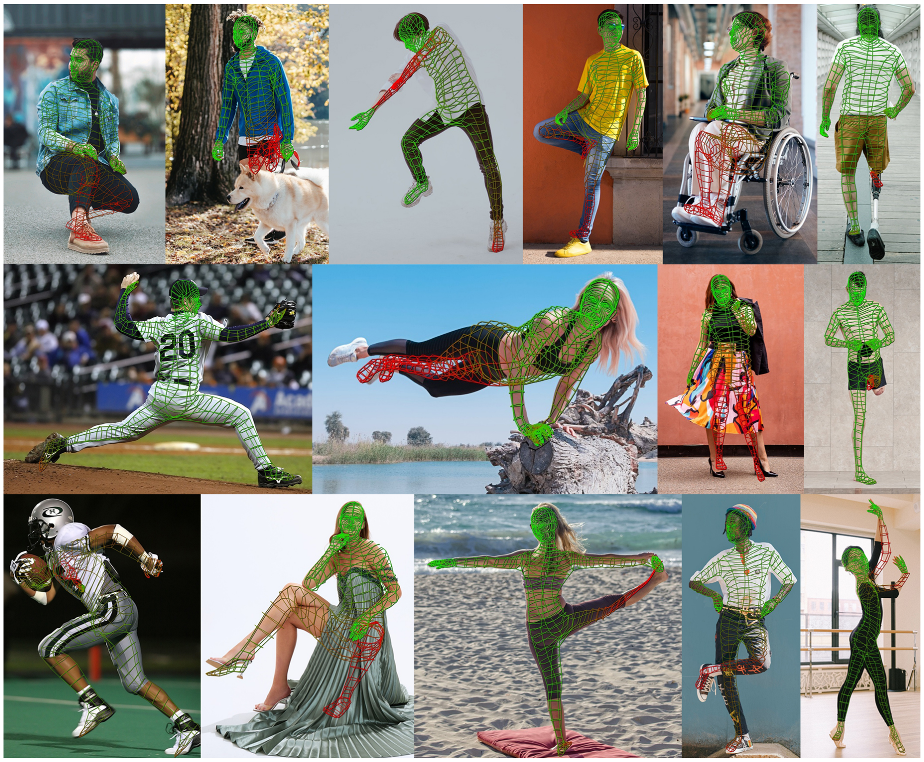


Figure 16: Examples of failures of our landmark prediction method. Confidence is colour coded, with green being high and red being low. Images collected from https://pexels.com.

4 Model fitting

It is often useful not just to have landmarks, but to have a parameterized representation of a person's shape and motion . We extend the approach of Wood et al. [19] from face reconstruction to fit our complete body model described in section 1 to the dense landmarks output by the pipeline of section 3.

So, given probabilistic dense 2D landmarks L, our goal is to find optimal model parameters $\mathbf{\Phi}^*$ that minimize the following energy:

$$E(\mathbf{\Phi}; L) = \underbrace{E_{\text{landmarks}}}_{\text{Data term}} +$$
$$\underbrace{E_{\text{face\_identity}} + E_{\text{body\_identity}} + E_{\text{expression}} + E_{\text{pose}} + E_{\text{temporal}} + E_{\text{intersect}}}_{\text{Regularizers}}$$

Where we are optimizing $\mathbf{\Phi}$, that is face identity $\vec{\gamma}$, body identity $\vec{\beta}$, expression for each frame $\mathbf{\Psi}$, pose for each frame $\mathbf{\Theta}$, and camera rotations $\mathbf{R}$ and positions $\mathbf{T}$.

$$\mathbf{\Phi} = \{\underbrace{\vec{\gamma}, \vec{\beta}, \mathbf{\Psi}_{F\times|\vec{\psi}|}, \mathbf{\Theta}_{F\times|\vec{\theta}|}}_{\text{Human}}; \underbrace{\mathbf{R}_{C\times 3}, \mathbf{T}_{C\times 3}}_{\text{Cameras}}\}$$

Where F is the number of frames in the given sequence and C is the number of cameras. $E_{\text{landmarks}}$ takes the same form as in Wood et al. [19]. $E_{\text{face\_identity}}$, $E_{\text{expression}}$, E_{temporal} and $E_{\text{intersect}}$ also follow the implementation of Wood et al. [19].

For $E_{\text{body\_identity}}$ we use an L2 prior given that SMPL-H uses a variance-scaled PCA basis for identity. It may be beneficial to use a GMM body identity prior (as we do for the face) to promote more plausible body shape, we leave this is a potential direction for future work.

For pose, instead of using an L2 prior as in Wood et al. [19], we use three GMM priors. One for body pose (excluding hand and eye pose) with a GMM fit to a subset of our pose library, and one for each hand with the GMMs fit to the MANO dataset [14] for each hand respectively. This helps to promote plausible poses in a data-driven way. More advanced pose priors (e.g., DNN) could provide better results [13], again we leave this for future work.

In cases of 2D-to-3D lifting such as this, bodies provide a much harder challenge than faces. After projection there is much more ambiguity in limb position, for example, due to the high range of motion of some body parts compared to the face. Further, self-occlusion is much more common for bodies, while for faces symmetry provides a very strong prior when limited self-occlusion does occur. As such, we observe that effective model fitting is much more difficult than for faces, and the monocular case is often ill-posed. We find that multiple camera views ($C \geq 3$) are required and results improve significantly for higher numbers of cameras.

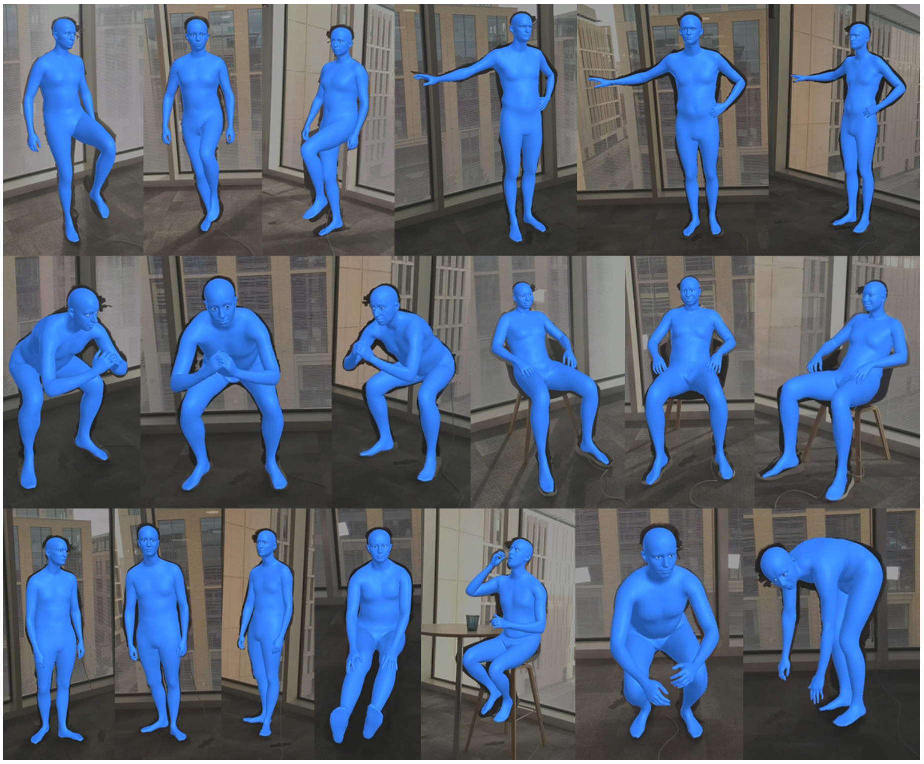


Figure 17: Example model fitting results on data collected using three Azure Kinect RGB cameras. Showing three viewpoints for various subjects and poses, and a single viewpoint for four diverse poses in the bottom right.

We also find reasonable initialization to be more important than for faces, and significantly harder. For faces simple 6-DoF alignment using PnP is sufficient to get a very good starting point for the optimizer, but in the case of the full body this gives quite poor results. Particularly for fine details like hand pose which will struggle to converge without good initialization, even if the landmarks are highly accurate.

To address this we initialize the pose using a machine learning approach to predict pose directly from one of the available views [3], providing the optimizer with a very good starting point. The multi-view landmarks are then used by the optimizer to achieve highly accurate 3D consistency across views, and so a very precise registration in world-space. Many machine learning approaches do not take multi-view data as input and, even when they do, struggle to achieve very precise alignment and consistency between views as required here. Future work might attempt to improve this initialization step to use multi-view data and reduce the need for the secondary optimization step.

Some results of our model fitting approach are shown in Figure 17.

Acknowledgements

Thanks to Rodney Brunet and his team for invaluable help with clothing assets and artistic input throughout, Darren Cosker for assistance in collecting motion-capture data for the pose library and Kendall Robertson for assistance with processing parts of the pose library.

References

[1] Gwangbin Bae, Martin de La Gorce, Tadas Baltrušaitis, Charlie Hewitt, Dong Chen, Julien Valentin, Roberto Cipolla, and Jingjing Shen. "DigiFace-1M: 1 Million Digital Face Images for Face Recognition". In: WACV. IEEE. 2023.

[2] Blender Foundation. Cycles Renderer. https://www.cycles-renderer.org/. 2021.

[3] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. "Monocular Expressive Body Regression through Body-Driven Attention". In: ECCV. 2020. URL: https://expose.is.tue.mpg.de.

[4] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: CVPR. 2016.

[5] Adobe Inc. Substance Painter. https://www.adobe.com/products/substance3d-painter.html. 2022.

[6] CLO Virtual Fashion Inc. Marvelous Designer. https://marvelousdesigner.com/. 2022.

[7] Marmoset LLC. Marmoset Toolbag. https://marmoset.co/Toolbag/. 2022.

[8] Matthew Loper, Naureen Mahmood, and Michael J Black. "MoSh: Motion and shape capture from sparse markers". In: ACM ToG 33.6 (2014), pp. 1–13.

[9] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. "SMPL: A Skinned Multi-Person Linear Model". In: ACM ToG 34.6 (Oct. 2015), 248:1–248:16.

[10] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. "Learning to dress 3d people in generative clothing". In: CVPR. 2020, pp. 6469–6478.

[11] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. "AMASS: Archive of Motion Capture as Surface Shapes". In: ICCV. Oct. 2019, pp. 5442–5451.

[12] Daniel McDuff, Xin Liu, Javier Hernandez, Erroll Wood, and Tadas Baltrusaitis. "Synthetic Data for Multi-Parameter Camera-Based Physiological Sensing". In: EMBC. IEEE. 2021, pp. 3742–3748.

[13] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. “Expressive body capture: 3d hands, face, and body from a single image”. In: CVPR. 2019, pp. 10975–10985.

[14] Javier Romero, Dimitrios Tzionas, and Michael J. Black. “Embodied Hands: Modeling and Capturing Hands and Bodies Together”. In: ACM ToG. 245:1–245:17 36.6 (Nov. 2017).

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: CVPR. 2018, pp. 4510–4520.

[16] Ten24 Animation Ready Body Scans. Ten24 Media LTD. https://www.3dscanstore.com/retopologised-body-models/animation-ready-body-scans. 2022.

[17] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. “SHIFT: A Synthetic Driving Dataset for Continuous Multi-Task Domain Adaptation”. In: CVPR. 2022, pp. 21371–21382.

[18] Phillip Thomas, Lars Pandikow, Alex Kim, Michael Stanley, and James Grieve. Open Synthetic Dataset for Improving Cyclist Detection. Nov. 2021. URL: https://paralleldomain.com/.

[19] Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Matthew Johnson, Jingjing Shen, Nikola Milosavljevic, Daniel Wilde, Stephan Garbin, Chirag Raman, Jamie Shotton, Toby Sharp, Ivan Stojiljkovic, Tom Cashman, and Julien Valentin. 3D face reconstruction with dense landmarks. 2022. DOI: 10.48550/ARXIV.2204.02776. URL: https://arxiv.org/abs/2204.02776.

[20] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Matthew Johnson, Virginia Estellers, Thomas J. Cashman, and Jamie Shotton. Fake It Till You Make It: Face analysis in the wild using synthetic data alone. 2021. arXiv: 2109.15102 [cs.CV].