i3DMM: Deep Implicit 3D Morphable Model of Human Heads

Tarun Yenamandra[1,2], Ayush Tewari[2], Florian Bernard[1,2], Hans-Peter Seidel[2], Mohamed Elgharib[2], Daniel Cremers[1], and Christian Theobalt[2]
[1]TU Munich, [2]MPI Informatics, Saarland Informatics Campus

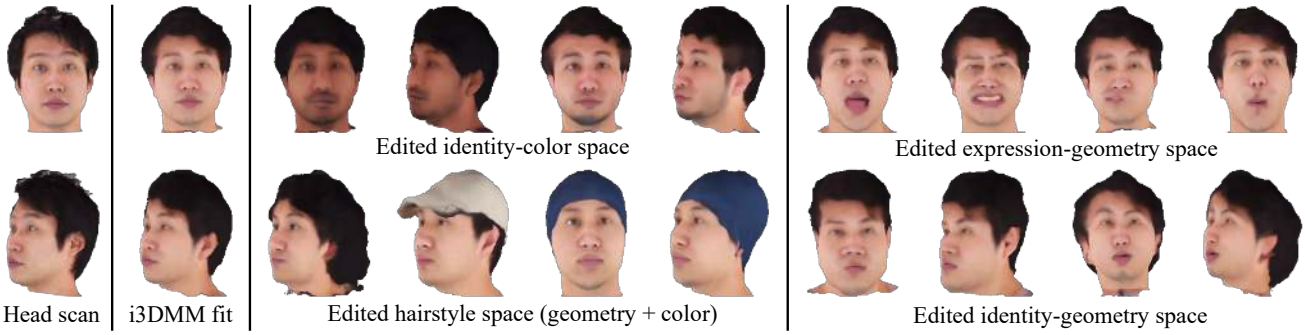

Figure 1. Our deep implicit 3D morphable model (i3DMM) of human heads includes semantically disentangled spaces for color (identity and hairstyle), and geometry (identity, expression, and hairstyle). We can fit i3DMM to head scans, and edit the different components.

Abstract

We present the first deep implicit 3D morphable model (i3DMM) of full heads. Unlike earlier morphable face models it not only captures identity-specific geometry, texture, and expressions of the frontal face, but also models the entire head, including hair. We collect a new dataset consisting of 64 people with different expressions and hairstyles to train i3DMM. Our approach has the following favorable properties: (i) It is the first full head morphable model that includes hair. (ii) In contrast to mesh-based models it can be trained on merely rigidly aligned scans, without requiring difficult non-rigid registration. (iii) We design a novel architecture to decouple the shape model into an implicit reference shape and a deformation of this reference shape. With that, dense correspondences between shapes can be learned implicitly. (iv) This architecture allows us to semantically disentangle the geometry and color components, as color is learned in the reference space. Geometry is further disentangled as identity, expressions, and hairstyle, while color is disentangled as identity and hairstyle components. We show the merits of i3DMM using ablation studies, comparisons to state-of-the-art models, and applications such as semantic head editing and texture transfer. We will make our model publicly available[1].

[1]http://gvv.mpi-inf.mpg.de/projects/i3DMM/

1. Introduction

3D morphable models (3DMMs) are parametric models of geometry and appearance of human faces, with widespread use in applications such as image and video editing, face recognition and cognitive science [20]. These models are trained using 3D scans of humans, e.g., laser scans [4], depth sensor-based scans [9], or photometric multi-view scans [28]. As 3DMMs usually learn a deformation space of a fixed template mesh, the training shapes commonly need to be brought into dense surface correspondence with each other. Computing such correspondence for the face region alone is already hard and often requires a challenging non-convex optimizaton problem [20]; computing dense correspondence for the rest of the head and the hair is close to impossible. Therefore, as well as due to the lack of large 3D scan datasets with hair, most 3DMMs only capture the face region. While some recent approaches aim to model the complete head [15, 28, 41, 40], they do not capture the hair region, and the appearance in the head region (if captured) is very limited.

In this work, we present i3DMM, the first implicit 3D morphable model which captures the full head region, including hair deformations and appearance. We capture a new dataset of full photogrammetric head scans of 64 people for training. Each subject performs several expressions and natural hair is captured. Since such scans are

noisy, especially in the hair region, our training algorithm uses an adaptive sampling strategy based on the quality of reconstructions in different regions of the head, allowing us to effectively handle noise without smoothing out details. In contrast to existing 3DMMs, which use a mesh-representation with a fixed template, we implicitly represent surfaces using signed distance functions. This allows us to capture large deformations easily, which is particularly convenient for the hair region. The implicit representation also allows us to avoid computing dense correspondence between training scans. Our method is inspired by recent works on deep implicit surface modeling [36, 31, 32], which demonstrate the advantages of using an implicit representation compared to voxel grids, point clouds or meshes. Our main technical innovation compared to these works is that we use a novel neural network architecture which *decouples the learning process*, separating it into learning a reference shape, learning geometry deformations with respect to this reference, and learning a color network. This allows us to automatically compute dense correspondences between any two shapes with only sparse supervision on salient face landmarks. This decoupling is also essential for disentangling the learned space into semantically meaningful parametric components, an important feature of 3DMMs.

Most classical models disentangle identity geometry, expression geometry, and appearance of the face. This is sufficient for several applications in face editing [52, 51, 48], but does not allow the explicit control of head parts other than the facial region. Our model allows for unprecedented control over the different semantic modes of full heads. To this end, we learn to disentangle the geometry and color, where geometry is further separated into identity, expression, and hairstyle components, and color is further separated into identity and hairstyle components.

In summary, our main contributions are:

1. A method for learning full head 3D morphable models directly from rigidly aligned real-world noisy 3D scans without dense ground truth correspondences.
2. A novel network architecture which can compute dense correspondences between 3D shapes represented using implicit functions. This network is trained only with sparse supervision.
3. A training method for disentangling
   (a) the color and geometry components, and
   (b) the identity, expression, and hairstyle components of the geometry; and the identity and hairstyle components of the color.

We compare our model to several state-of-the-art 3DMMs by fitting to 3D scans. We also demonstrate the quality of our learned dense correspondences by showing texture transfer between scans, and show that i3DMM benefits applications such as semantic head editing, and one-shot computation of segmentation masks and landmarks on 3D scans.

2. Related Work

Most approaches for face and head modeling are mesh-based. We summarize them here and refer the reader to Egger *et al*. [20] for a more comprehensive report. As our model is based on implicit representations, we also discuss methods which learn implicit shape and color.

Face and Head Morphable Models. The work of Blanz and Vetter [5] showed the possibility of representing faces using a 3D Morphable Model (3DMM). The model is learned by transforming the shape and texture of examples into low-dimensional vector representations using principal components analysis (PCA). Further improvements were proposed [23, 38, 8, 6] by using better registration and higher quality scans. Multilinear models have also been proposed to model identity-dependent expressions [10, 21].

Li *et al*. [29] presented FLAME, a head model that combines a linear shape space with an articulated jaw, neck and eyeballs, pose-dependent corrective expression blendshapes, and global expressions. The model is learned from a large 3D dataset, including data from D3DFACS [12]. Ranjan *et al*. [42] learned a head model using an autoencoder based on spectral graph convolutions [17]. The LYHM head model [16, 14] uses a hierarchical parts-based template morphing framework to process the head shape, and uses optical flow to refine the texture. The model is built using 1,200 different identities. Ploumpis *et al*. [40, 39] combined the LYHM and LSFM [7] models as the Universal Head model (UHM), with a focus on face geometry. Note that all existing full head models only model the cranium geometry without hair.

Implicit Representation. Implicit representations have a long history in computer vision for inferring 3D shape [27], temporally evolving shape [13, 26] and textured shape [47]. Park *et al*. [36] presented DeepSDF to represent a class of shapes using signed distance fields with an autodecoder. Chen *et al*. [11] and Mescheder *et al*. [31] learned a generative model of a class of shapes by classifying a point in space as inside or outside a shape. Recent approaches [53, 18, 22] have proposed to represent shapes using a collection of local implicit patches for higher quality. Oechsle *et al*. [35] extended OccupancyNets [31] to represent texture along with the geometry using monocular inputs. Niemeyer *et al*. [34] presented an approach for handling time-varying shapes by learning to predict motion vectors for each point in 3D space using OccupancyNets. PIFu [44, 45] allows for 3D reconstruction of humans from monocular inputs. Pixel-aligned implicit functions estimate a continuous field that determines whether a pixel is inside or outside the surface of the human subject, as well as the color on the surface. Several recent methods [46, 32, 49] have presented ways to achieve higher-quality implicit representations by using periodic functions as activations. Saito *et al*. [43] presented an approach for

3D hair modeling. Their technique learns a manifold for 3D hairstyles represented as implicit surfaces using a volumetric variational autoencoder. Concurrent to us, some methods [30, 19, 54] also learn dense correspondences between shapes represented using implicit functions.

3. Method

In this section, we describe how we obtain our deep implicit 3D morphable model (i3DMM). Our overall processing pipeline is illustrated in Fig. 2. In the following we will describe our data acquisition, the training data preparation, and the neural network architecture.

3.1. Data Acquisition

For training, we have scanned 64 subjects (46 male, 18 female; 22 Caucasian, 9 Asian, 25 Indian Sub-continent, 3 Hispanic, 3 Middle Eastern, 2 undisclosed; with an age ranging from 19 to 69 with average 26). For each subject we have recorded 10 facial expressions (a subset of which are chosen from paGAN [33]), including neutral expressions with 3-4 different "hairstyles" (open hair, tied hair for subjects with long hair, and two different caps), as shown in Fig. 3. Our data was acquired with the *Treedys*[2] multi-view scanning system, that comprises of 137 calibrated cameras, see Fig. 4. The total capture time per person amounts to about 20 minutes. Photogrammetry-based 3D reconstruction on the multi-view data was applied to obtain the final textured meshes (see Fig. 4), where each mesh comprises of around 50,000 vertices.

3.2. Training Data

Given our recorded textured mesh data, we apply several preprocessing steps. This includes a semi-automated landmark annotation, cropping of the head area to discard the shoulders and parts of the upper body, rigidly aligning the meshes, and closing the hole at the neck to make the mesh watertight. In the following we elaborate on these steps.

Landmark Annotation. First, we apply the automated face landmark detector from *dlib* [24] on the front-facing image obtained from the multi-view camera acquisition setup, which produces a total of 66 face landmarks. We select a subset of 8 landmarks, i.e. the four corner points of the eyes, the tip of the nose, the two corner points of the mouth, and the chin, and then transfer these 2D landmarks to the 3D mesh. We manually correct the inaccurately detected landmarks, and additionally annotate 8 landmarks for the corners of ears (top, bottom, left, and right) directly in 3D.

Head Cropping. In order to remove the shoulders and parts of the upper body, we crop the head meshes based on the 3D landmarks. To this end, we first compute the vector $\mathbf{v}$ from the centroid of the four eye cornerpoints to the tip of the nose. Then, for $\mathbf{p}$ being the chin landmark in 3D space, we define a virtual plane that goes through the point $\mathbf{p}+\frac{1}{2}\mathbf{v}$ and has the normal $\mathbf{v}$. We only keep the side of the plane that contains the facial landmarks.

Rigid Alignment. Based on the 3D facial landmarks, we perform a rigid alignment of all head scans in order to ensure that they are oriented consistently in 3D space. We use a numerically stable implementation [3] of the transformation synchronization method [2], which solves the generalized Procrustes problem in an initialization-free and unbiased way. To this end, we first rigidly align all pairs of heads based on Singular Value Decomposition (SVD), i.e. we solve all pairwise Procrustes problems, and subsequently use transformation synchronization to establish cycle-consistency in the set of pairwise transformations.

Hole Closing. Since our full head model is based on a signed distance representation, we require that the meshes are watertight. After rigid alignment, we assume that the longitudinal axis of the head aligns with the y-axis. With that, we use a flat patch to close the hole at the neck by extruding the respective boundary vertices to coincide with the smallest y-coordinate. Finally, we scale all meshes with the same value such that all of them fit in the unit cube.

3.3. Training

We learn a vector-valued function $f_\theta(\mathbf{x}, \mathbf{z})$, where θ are the neural network weights, $\mathbf{x} \in \mathbb{R}^3$ is the query point, and $\mathbf{z}$ is a code vector that encodes the head instance. The output $f_\theta(\mathbf{x}, \mathbf{z}) = (s, \mathbf{c})$ includes a scalar s that represents the signed distance to the surface, as well as the color $\mathbf{c} \in \mathbb{R}^3$ at the closest surface point from $\mathbf{x}$. The shape boundary is represented as the zero level-set of the signed distance function (SDF), while the interior parts of the shapes have a negative signed distance value, and the exterior parts a positive value. We use an autodecoder network architecture [36], where the weights of the network θ and the input latent codes $\mathbf{z}$ for all shapes are learned jointly.

Mesh Sampling. We require $(\mathbf{x}, s, \mathbf{c})$-triplets (query point, signed distance value, color) for training. We use a combination of two strategies for sampling these triplets. First, we sample points on the mesh surface based on uniform random sampling. However, in order to also account for higher accuracy in high-detail facial regions, i.e. the eyes, nose and mouth, we additionally sample more points in these areas. We take the center of each eye, the tip of the nose, and the center of the mouth, and place a sphere around them that covers the respective region of interest. Then, we sample points on the mesh surface that lie within each sphere. Eventually we mix the uniformly sampled points with the landmark-based sampled points in a 1 to 3 ratio.

After sampling, the points are perturbed by a uniform 3D Gaussian with standard deviation 0.005 times the length

[2] https://www.treedys.com/

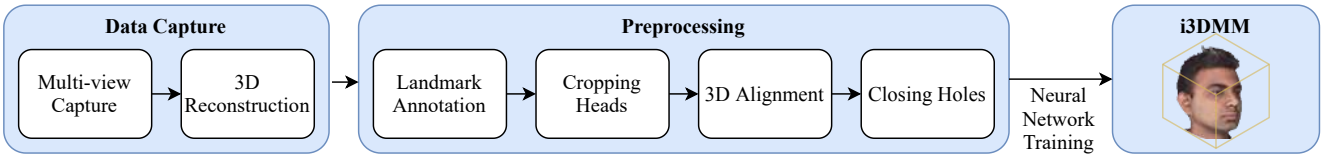

Figure 2. Overview of our approach.

Figure 3. Overview of the captured expressions (red box: test set; blue box: hairstyles) for each subject.

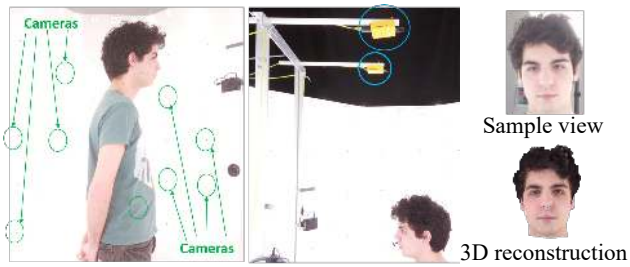

Figure 4. Multi-view scanning system with 135 cameras around the person (green), and 2 cameras (blue) on top. Right: front view image, and reconstruction.

of the bounding box, such that not only surface points but also points in the interior and exterior are used, see [36]. The color value for each sample is obtained by finding the closest point on the mesh surface, and then looking up its color in the texture map.

Latent Codes and Disentanglement. As mentioned earlier, latent codes for each object (head scan) are also learned during training. Existing approaches use a single object latent code to describe the shape. In contrast, we design several separate latent spaces for our objects in order to learn a semantically disentangled model. We use two separate latent vector spaces for geometry and color, $\mathbf{z}_{\text{geo}}$ and $\mathbf{z}_{\text{col}}$, respectively. The geometry space includes three code vectors for identity, expression, and hairstyle, and the color space includes two code vectors for identity and hairstyle. Thus, $\mathbf{z}_{\text{geo}} = (\mathbf{z}_{\text{geoId}}, \mathbf{z}_{\text{geoEx}}, \mathbf{z}_{\text{geoH}})$ and $\mathbf{z}_{\text{col}} = (\mathbf{z}_{\text{colId}}, \mathbf{z}_{\text{colH}})$. During training, the number of different identity code vectors is equal to the number of training identities, 58. The number of different expression vectors is fixed to 10 (cf. Fig. 3 for the training expressions) and hairstyle to 4 (*short*, *long*, *cap1* or *cap2*) for geometry, and to 3 (*nocap*, *cap1* or *cap2*) for color. For all scans of the i-th subject we use the same latent variables for the geometry identity and color identity, $\mathbf{z}^i_{\text{geoId}}$ and $\mathbf{z}^i_{\text{colId}}$. Similarly, for each expression and hairstyle, the same variables $\mathbf{z}_{\text{geoEx}}$, $\mathbf{z}_{\text{geoH}}$ and $\mathbf{z}_{\text{colH}}$ are used across all identities. By doing so, we are able to learn disentangled latent variables without imposing any explicit constraints. At test time, we can control each latent space individually, leading to semantically meaningful editing.

Network Architecture. Our network comprises three components, the *Reference Shape Network* (*RefNet*), the *Shape Deformation Network* (*DeformNet*) and the *Color Network* (*ColorNet*), see Fig. 5. All networks are composed of fully-connected layers with a *ReLU* non-linearity after every layer, except the output layer. The inputs to all our networks are encoded using sinusoidal positional encoding [32].

Our *Reference Shape Network* encodes a single reference shape, such that all individual head shapes can be obtained by deforming this shape. The output at a query point $\mathbf{x}$ is the signed distance value $f^r_\theta(\mathbf{x}) \in \mathbb{R}$. Note that for the reference shape there is no latent code input since only a *single* reference shape is learned. This can be seen analogously to the mean (or neutral) shape used in classical 3DMMs [5]. We use 3 fully connected layers for this network, where each hidden layer has dimensionality 512.

The *Shape Deformation Network* deforms the reference shape to represent the shape of an individual head. The network takes the geometry latent code $\mathbf{z}^i_{\text{geo}}$ for an object i, and a query point $\mathbf{x}$ as input, and produces the output,

$$f^s_\theta(\mathbf{x}, \mathbf{z}^i_{\text{geo}}) = \delta \in \mathbb{R}^3\,, \tag{1}$$

which is a displacement vector that deforms the query point to the reference shape. Thus, the signed distance value at any point $\mathbf{x}$ of the object i with geometry latent $\mathbf{z}^i_{\text{geo}}$ is

$$s(\mathbf{x}, \mathbf{z}^i_{\text{geo}}) = f^r_\theta(\mathbf{x} + \delta)\,, \tag{2}$$

where $s(\cdot)$ is the scalar signed distance value, and δ is the deformation from Eq. (1). This formulation allows us to compute dense correspondences between any input head scan and the reference head shape. The separation between a reference shape and deformations with respect to this reference shape is also common in classic morphable models [20]. However, these models require dense correspondences before training, which is not necessary in our approach. We use 8 fully connected layers for this network, where each hidden layer has dimensionality 1024.

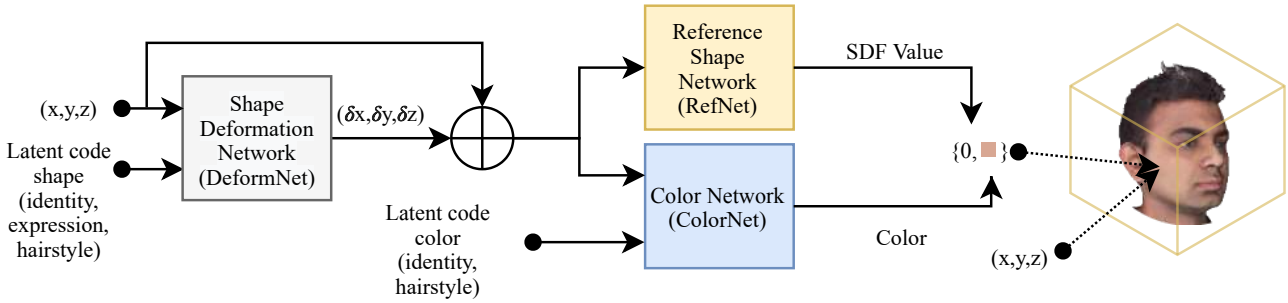

Figure 5. Overview of our network architecture. We learn weights of three network components, a *Shape Deformation* component, a *Reference Shape* component, and a *Color* component. Moreover, the latent codes for each object are also optimized for. The input of the network is a 3D query point, and the output is a signed distance value along with the corresponding color.

The introduction of the reference shape makes it possible to disentangle the geometry and color components. The *Color Network* learns the color of the query point in the reference space. Given a query point $\mathbf{x}$, deformation δ from Eq. (1), and color latent vector $\mathbf{z}^i_{\text{col}}$ for the object i, the output is represented as $f^c_\theta(\mathbf{x}+\delta, \mathbf{z}^i_{\text{col}}) \in \mathbb{R}^3$, which is the color at point $\mathbf{x}$. Note that without this separation of a reference space, the *ColorNet* would also have to take into account information about object geometry, thus not being able to disentangle shape and color. Note that the latent code for *ColorNet*, for a given identity and hairstyle, does not change with expressions. *DeformNet* finds the right colors and geometry for different expressions by achieving dense correspondences. We use 9 fully connected layers for this network, where each hidden layer has dimensionality 1024.

Loss Functions. We define the loss function to train our network as

$$\begin{aligned}\mathcal{L}_\theta(\mathbf{x}, \mathbf{z}_{\text{geo}}, \mathbf{z}_{\text{col}}) = \sum_{i=1}^{K} (&\mathcal{L}^{\text{geo}}_\theta(\mathbf{x}, \mathbf{z}^i_{\text{geo}}) + \mathcal{L}^{\text{def}}_\theta(\mathbf{x}, \mathbf{z}^i_{\text{geo}}) \\ &+ \mathcal{L}^{\text{col}}_\theta(\mathbf{x}, \mathbf{z}^i_{\text{col}}) + \mathcal{L}^{\text{reg}}(\mathbf{z}^i_{\text{geo}}, \mathbf{z}^i_{\text{col}})) \\ &+ \sum_{i \neq j} \mathcal{L}^{\text{lm}}_\theta(\mathbf{z}^i_{\text{geo}}, \mathbf{z}^j_{\text{geo}}) \,, \end{aligned} \tag{3}$$

where, $i, j = \{1, \ldots, K\}$, K is the number of scans in the batch, $\mathbf{z}_{\text{geo}} = \{\mathbf{z}^1_{\text{geo}}, \ldots, \mathbf{z}^K_{\text{geo}}\}$, and $\mathbf{z}_{\text{col}} = \{\mathbf{z}^1_{\text{col}}, \ldots, \mathbf{z}^K_{\text{col}}\}$. The latent vectors of head i are represented as $\mathbf{z}^i_{\text{geo}}$ and $\mathbf{z}^i_{\text{col}}$. Here, $\mathcal{L}^{\text{geo}}_\theta(\cdot)$ enforces good geometry reconstructions, $\mathcal{L}^{\text{def}}_\theta(\cdot)$ regularizes the deformation field, $\mathcal{L}^{\text{col}}_\theta(\cdot)$ is used to train the *ColorNet*, $\mathcal{L}^{\text{reg}}(\cdot)$ is a regularizer on the latent vectors, and $\mathcal{L}^{\text{lm}}_\theta(\cdot)$ is a sparse pairwise landmark supervision loss. For the geometry term that we impose upon the signed distance values, we use the ℓ_1-loss

$$\mathcal{L}^{\text{geo}}_\theta(\mathbf{x}, \mathbf{z}^i_{\text{geo}}) = w_g \left\| \text{cl}(s(\mathbf{x}, \mathbf{z}^i_{\text{geo}}), t) - \text{cl}(s_{\text{gt}}(\mathbf{x}), t) \right\|_1 , \tag{4}$$

where $s_{\text{gt}}(\mathbf{x})$ is the ground truth signed distance value at $\mathbf{x}$, and $s(.)$ is from Eq. (2). These values are (symmetrically) clamped at t=0.1, for which we define $\text{cl}(x, t) := \min(t, \max(x, -t))$. We use a similar ℓ_1-loss for the color component, i.e.

$$\mathcal{L}^{\text{col}}_\theta(\mathbf{x}, \mathbf{z}^i_{\text{col}}) = w_c \left\| f^c_\theta(\mathbf{x} + \delta, \mathbf{z}^i_{\text{col}}) - c_{\text{gt}}(\mathbf{x}) \right\|_1 , \tag{5}$$

where $c_{\text{gt}}(\mathbf{x})$ is the ground truth color at $\mathbf{x}$. We also enforce that the 16 landmarks $\{\mathbf{x}^\ell_i\}$, as described in Sec. 3.2, of each shape i, deform to the same points in the reference space using the pairwise loss

$$\mathcal{L}^{\text{lm}}_\theta(\mathbf{z}^i_{\text{geo}}, \mathbf{z}^j_{\text{geo}}) = w_{lm} \sum_{\ell=1}^{16} \left\| (\mathbf{x}^\ell_i + \delta^\ell_i) - (\mathbf{x}^\ell_j + \delta^\ell_j) \right\|_2 , \tag{6}$$

where, $\delta^\ell_i = f^s_\theta(\mathbf{x}^\ell_i, \mathbf{z}^i_{\text{geo}})$ as in Eq. (1). For scans with the ears covered by hair, we do not have any ear annotations. We would like to compress the ear region in the reference shape to a single point in the reconstruction for these shapes. Thus, we additionally optimize for one point for each ear. We enforce pairwise constraints between the learnable ear points and the annotated ear points for the other shapes in the batch using Eq. (6).

Further, to ensure regularized deformations to the reference shape, we impose a loss on the amount of deformation. We use $\mathcal{L}^{\text{def}}_\theta(\mathbf{x}, \mathbf{z}^i_{\text{geo}}) = w_s ||f^s_\theta(\mathbf{x}, \mathbf{z}^i_{\text{geo}})||_2$. Finally, we use an ℓ_2-regularizer on the latent vectors assuming a Gaussian prior distribution, $\mathcal{L}^{\text{reg}}(\mathbf{z}^i_{\text{geo}}, \mathbf{z}^i_{\text{col}}) = w_r(||\mathbf{z}^i_{\text{geo}}||_2 + ||\mathbf{z}^i_{\text{col}}||_2)$.

Optimization. Given N batches with K objects per batch, we optimize for the network weights and the latent vectors by solving the optimization problem

$$\underset{\theta, \{\mathbf{z}^b_{\text{geo}}, \mathbf{z}^b_{\text{col}}\}_{b=1}^N}{\text{argmin}} \quad \sum_{b=1}^{N} \sum_{\mathbf{x}} \mathcal{L}_\theta(\mathbf{x}, \mathbf{z}^b_{\text{geo}}, \mathbf{z}^b_{\text{col}}) \,, \tag{7}$$

where, the inner sum takes into account all sampled points, as explained above, and we abuse the notation $\mathbf{z}^b_{\text{geo}}, \mathbf{z}^b_{\text{col}}$ to now represent latent codes of all the scans in batch b.

4. Experiments

In this section, we present an experimental evaluation of our head model. We demonstrate that the model can be used

for reconstructing scan data. We present an ablation study to carefully analyze our design choices, and compare i3DMM to state-of-the-art head models. We also show dense correspondence results and applications of our model. Before we present these results, we provide additional information on the neural network training and testing.

Training Details. We train our networks using PyTorch [37], where we use the Adam [25] solver with mini-batches of size 64. We train for 1000 epochs with a learning rate of 0.0005, which decays by a factor of 2 every 250 epochs. We initialize *RefNet* by pretraining it using only one mouth-open (top row, third from left in Fig. 3) training scan. Our network takes about 2 days to train on 2 NVIDIA RTX8000 GPUs.

Test Data. We collect a separate test set consisting of scans of 6 identities, which are not part of the training data. We capture each identity in 5 novel expressions that are not part of the training expressions, see Fig. 3. Each scan is preprocessed analogously to the training scans.

4.1. Reconstruction

For a given test scan, we fit our learned model to it. This is done by optimizing for the latent vector that can best reproduce the scan, i.e. by finding the latent variables that minimize the problem

$$\underset{\mathbf{z}^i_{\text{geo}},\mathbf{z}^i_{\text{col}}}{\operatorname{argmin}} \sum_{\mathbf{x}} (\mathcal{L}^{\text{geo}}_\theta(\mathbf{x},\mathbf{z}^i_{\text{geo}}) + \mathcal{L}^{\text{def}}_\theta(\mathbf{x},\mathbf{z}^i_{\text{geo}}) + \mathcal{L}^{\text{col}}_\theta(\mathbf{x},\mathbf{z}^i_{\text{col}}) + \mathcal{L}^{\text{reg}}(\mathbf{z}^i_{\text{geo}},\mathbf{z}^i_{\text{col}}))\,, \quad (8)$$

where, $\mathbf{z}^i_{\text{geo}}, \mathbf{z}^i_{\text{col}}$ is the latent code for test scan i. This equation is similar to Eq. (7), with the difference that the network weights are fixed here and the pairwise landmark supervision loss in Eq. (6) is not enforced. We use Adam [25] with a step size of 0.0005 to solve this problem. We show several reconstruction results on the test data in Fig. 6. We can generalize to unseen identities and expressions, even though our training data only consists of 58 people with 10 expressions. Note that while we can generally preserve the detailed face region of the scans, we also smooth the scans in the noisy hair area.

4.2. Correspondences

As mentioned in Sec. 3.3, due to the particular design of our network, where the reference shape and the deformations are separated, our approach also establishes dense correspondences between shapes, with extremely sparse landmark supervision. We demonstrate these correspondences in Fig. 6, where the color is transferred from one scan to the other. The correspondences are also used in the applications of segmentation and landmark transfer in Sec. 4.6. Our model can reliably find correspondences across different subjects, expressions and hairstyles, including long and short hair. Please refer to the supplementary material for more evaluations.

4.3. Ablative Analysis

We design several experiments to evaluate the most important design choices for our approach: landmark-based sampling described in Sec. 3.3, sparse pairwise landmarks loss in Eq. (6), and jointly training *ColorNet*, *DeformNet*, and *RefNet*. We evaluate the qualitative impact of these choices by excluding them one at a time while training our model, see Fig. 7. Without landmarks-based sampling, the model focuses more on the noisy hair region and ignores the details in the face region (first column in Fig. 7). Without landmark supervision, the network creates small (fake) ear regions in the hair for the long-haired scans (second column in Fig. 7). It also leads to poor texture transfer around the ear. Finally, for evaluating the importance of joint training of the color and geometry networks, we first train *RefNet* and DeformNet. After convergence, we train *ColorNet*. The correspondences in this case are learned only from geometry reconstructions, which leads to artifacts as shown in the third column of Fig. 7.

4.4. Comparisons to Existing Models

We compare our model with the full head FLAME [29], face only FLAME, and Basel Face Model (BFM) [38] which only models the frontal face. We fit each model to the test set by optimizing for their model parameters. Please refer to the supplemental for more details on the fitting.

We show qualitative results in Fig. 8. FLAME only models the cranium without hair, thus fitting to the full head region results in incorrect head shapes. We also evaluate FLAME by only using the the face region for fitting. This leads to higher quality results. We obtain higher quality geometry both for the face and head regions. In addition, we can also reconstruct the color of the head, while FLAME can only model the geometry. We combine the BFM [38] identity geometry and appearance models with the expression model used in Tewari *et al*. [50], and use this to fit to our test scans. The expression model is a combination of two blendshape models [1, 9]. Since BFM also includes color, we jointly optimize to minimize the geometry as well as color alignment errors. As can be seen in Fig. 8 (middle), this model can fit to the expression as well as color of the scans. However, it is limited to only the face region, while i3DMM can reconstruct the full head. In addition, our reconstructions are more personalized, with higher quality nose geometry and face colors.

We show the quantitative evaluations of the different models in terms of the symmetric Chamfer distance and F-score metrics in Table 1. Chamfer distance is the mean distance of points from one shape to their closest points in another. We compute the symmetric Chamfer distance as

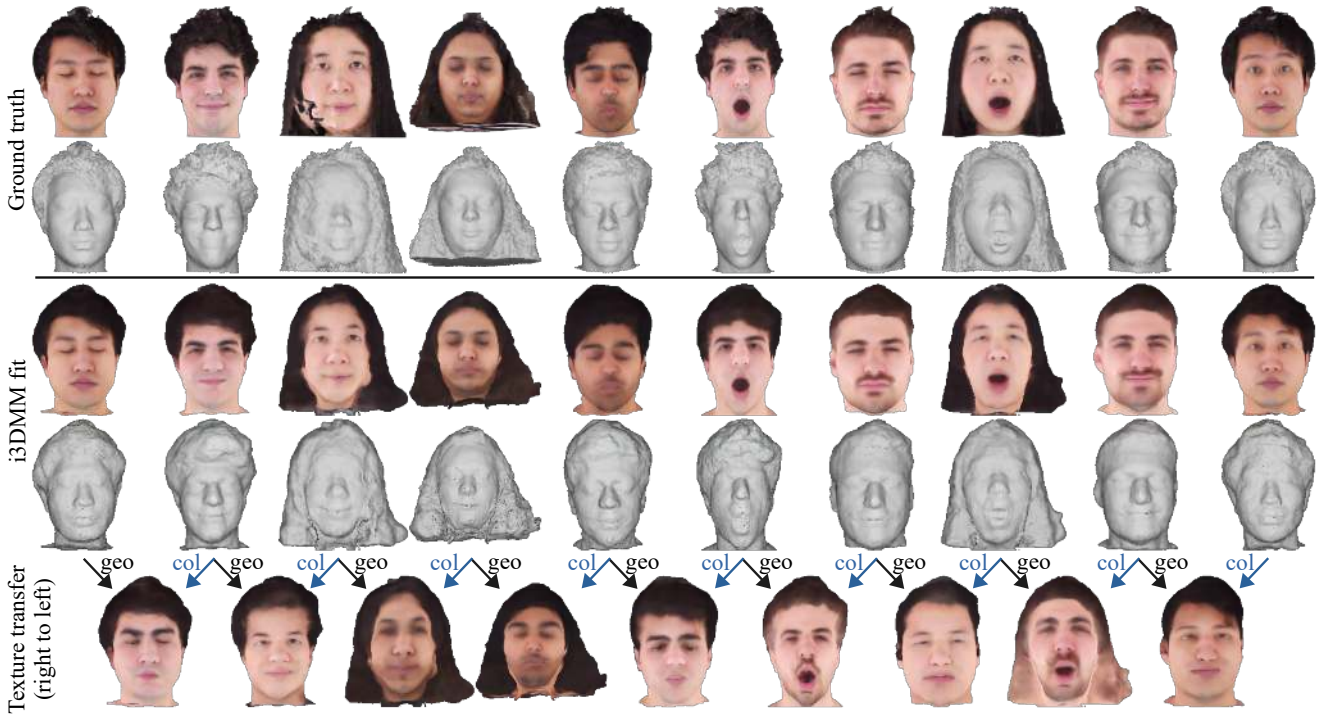

Figure 6. Reconstruction quality of i3DMM on test data. The top part shows texture and geometry of ground truth scans. The middle part shows texture and geometry of i3DMM fits. The last part shows texture transfer from the scan on right to the scan on left of the image. In column 3, noise in ground truth does not transfer to the i3DMM fit, showing the robustness of our method.

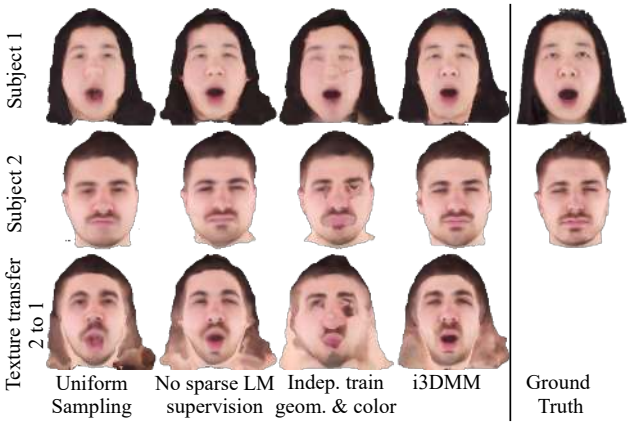

Figure 7. Ablation study. Top and middle row show reconstructions of two different test scans. Last row shows texture transfer from the scans in the top row to those in the middle row.

the mean of Chamfer distance from the ground truth to the fits and the Chamfer distance in the opposite direction. F-score is (100x) the harmonic mean of these two distances after applying a threshold. We use a similar metric for color. First, the mean error in color at points in the ground truth and the color at their nearest points in the fits is computed, along with the mean error in the opposite direction Finally, the color error is reported as their average. Please refer to the supplemental for details on the metrics used. We sample 150,000 points for computing the metrics. We achieve similar geometric quality as FLAME and BFM in the face region, but significantly outperform FLAME for the full head reconstructions, see Table 1. Moreover, in terms of color, our reconstructions are more realistic than BFM and also outperform it quantitatively.

Region	**Full Head**		**Face**		
Metric	Ours	FLAME	Ours	BFM	FLAME
Chamfer ↓ (mm)	**3.31**	7.83	1.02	0.96	**0.88**
F-score ↑	**63.38**	45.96	99.31	97.66	**99.6**
Color ↓	0.11	–	**0.07**	0.09	–

Table 1. Quantitative comparison of head models. Symmetric Chamfer distance is reported here. F-score is computed with a threshold of 0.01.

4.5. Application: Semantic Head Editing

Due to the semantic disentanglement in our model, we can selectively change semantic components of a head. In Fig. 1, we edit one feature of a test scan while keeping the other features fixed. To obtain the edited latent codes, we first find the principle components of variation in each latent-subspace by running PCA on the training la-

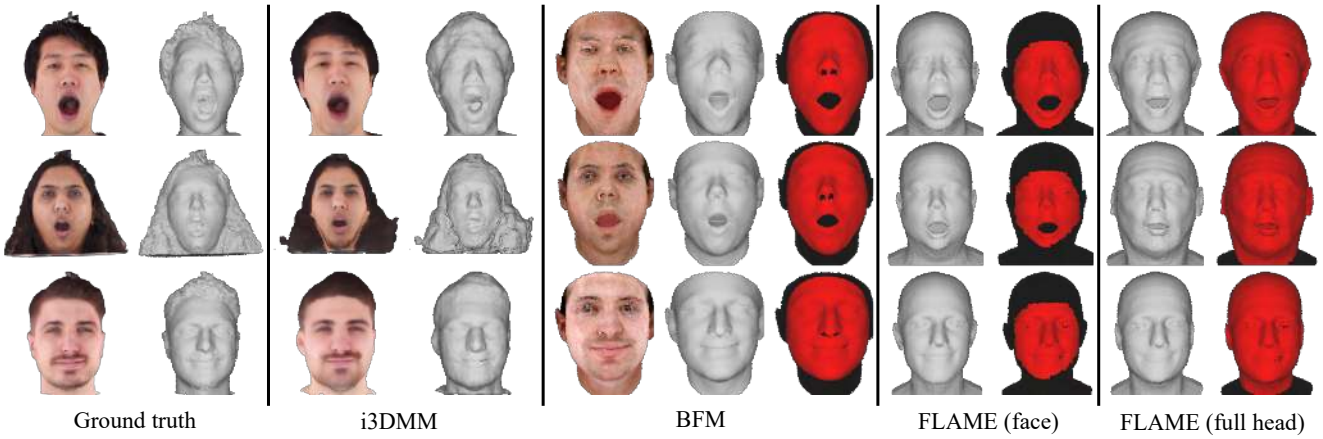

Figure 8. Comparison of fitting various models to scan data (left). The three considered models are our i3DMM (middle-left), BFM [38] (middle), and FLAME [29] (middle-right and right). FLAME does not provide color, and is evaluated in two settings, face-only (middle-right) and full head (right). We visualize both color and geometry for BFM and our results, and only geometry for FLAME. The masks used for fitting are also visualized.

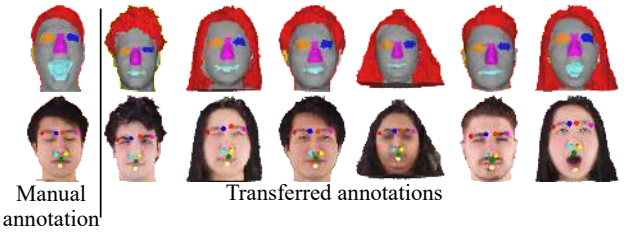

Figure 9. Application: One-shot annotations on real-world scans. Coarsely drawn annotations (top row: semantic segmentation, bottom row: landmarks, not used during training) on one scan (left) can be automatically transferred to other real-world head scans (right).

tent spaces. We move along the principal components to pick the latent codes to edit the identity, and we pick others from the trained latent codes. Unlike existing models, we can also edit hairstyles and add caps while keeping other components fixed. We also model the mouth interior. These features allow for semantic head editing applications much beyond the capabilities of the existing methods.

4.6. Application: One-shot Annotation Transfer

As i3DMM can predict dense correspondences among head scans, it also enables us to transfer annotations across different reconstructions. As our model has low reconstruction errors, it can also be used to annotate real-world 3D scans, see Fig. 9. We first transfer the annotations from one manually annotated scan to the reference shape. We can then transfer them to different i3DMM fits, and also to the real-world scans using nearest neighbors from the fits to the scans. As i3DMM can be used to even annotate the hair region, it can be useful to curate scan datasets. Please refer to the supplemental for more results.

5. Discussion and Future Work

Although i3DMM disentangles and models an unprecedented amount of variety in head features, and enables reconstructions with very high quality, it still has some limitations. The hair geometry is a coarse approximation of the physical nature of hair comprising of individual hair strands. One challenge here is that our scans are noisy in the hair region, partly due to the photometric multi-view reconstruction techniques used. In addition, the learned space of hairstyles is a smooth approximation for discrete hairstyles such as caps. There is room for extending i3DMM for more fine-grained control over hairstyles. Finally, we model hairstyles that cover the ear by trying to collapse the ear to a single point (Eq. (6)). Ideally, the layered geometry of ears behind the hair should be correctly represented.

6. Conclusion

We have presented the first unified 3D morphable model of geometry and appearance of full heads, which includes different identities, expressions and hairstyles. The head geometry and color is represented using implicit functions parameterized with neural networks. By explicitly separating the network into a reference shape, and shape and color deformations, our model can disentangle the different semantic components. It also allows us to obtain dense correspondences between different scans represented using our model, enabling several interesting applications.

Acknowledgements: This work has been supported by the ERC Consolidator Grant 4DReply (770784), the ERC Advanced Grant SIMULACRON (884679), and the Munich Center for Machine Learning. We thank Garvita Tiwari, Navami Kairanda, and Neng Qian for their support in capturing the dataset. We thank all the subjects in the dataset for lending their data for research.

References

[1] Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. The Digital Emily Project: photoreal facial modeling and animation. In *ACM SIGGRAPH Courses*, pages 12:1–12:15. ACM, 2009. 6

[2] Florian Bernard, Johan Thunberg, Peter Gemmar, Frank Hertel, Andreas Husch, and Jorge Goncalves. A solution for multi-alignment by transformation synchronisation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 3

[3] Florian Bernard, Johan Thunberg, Andreas Husch, Luis Salamanca, Peter Gemmar, Frank Hertel, and Jorge Goncalves. Transitively consistent and unbiased multi-image registration using numerically stable transformation synchronisation. In *The MIDAS Journal - Spectral Analysis in Medical Imaging*, October 2015. 3

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proc. SIGGRAPH*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999. 1

[5] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 2, 4

[6] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3d morphable models. *Int. J. Comput. Vision*, 126(2–4):233–254, Apr. 2018. 2

[7] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[8] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniahy, and D. Dunaway. A 3d morphable model learnt from 10,000 faces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5543–5552, June 2016. 2

[9] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3D facial expression database for visual computing. *IEEE TVCG*, 20(3):413–425, 2014. 1, 6

[10] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, March 2014. 2

[11] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[12] D. Cosker, E. Krumhuber, and A. Hilton. A facs valid 3d dynamic action unit database with applications to 3d dynamic morphable facial modeling. In *2011 International Conference on Computer Vision*, pages 2296–2303, Nov 2011. 2

[13] D. Cremers. Dynamical statistical shape priors for level set based tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1262–1273, Aug. 2006. 2

[14] H. Dai, N. Pears, W. Smith, and C. Duncan. A 3d morphable model of craniofacial shape and texture variation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3104–3112, Oct 2017. 2

[15] Hang Dai, Nick Pears, William Smith, and Christian Duncan. Statistical modeling of craniofacial shape and texture. *International Journal of Computer Vision*, 128(2):547–571, Nov 2019. 1

[16] Hang Dai, Nicholas Edwin Pears, William Alfred Peter Smith, and Christian Duncan. Statistical modeling of craniofacial shape and texture. *International Journal of Computer Vision*, November 2019. 2

[17] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3844–3852, Red Hook, NY, USA, 2016. Curran Associates Inc. 2

[18] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31–44, 2020. 2

[19] Yu Deng, Jiaolong Yang, and Xin Tong. Deformed implicit field: Modeling 3d shapes with learned dense correspondence. *arXiv preprint arXiv:2011.13650*, 2020. 3

[20] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3d morphable face models—past, present, and future. *ACM Trans. Graph.*, 39(5), June 2020. 1, 2, 4

[21] V. Fernández Abrevaya, S. Wuhrer, and E. Boyer. Multilinear autoencoder for 3d face model learning. In *Applications of Computer Vision (WACV), 2018 IEEE Winter Conference on*, 2018. 2

[22] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 2

[23] Thomas Gerig, Andreas Forster, Clemens Blumer, Bernhard Egger, Marcel Lüthi, Sandro Schönborn, and Thomas Vetter. Morphable face models - an open framework. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82, 2017. 2

[24] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 3

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 6

[26] T. Kohlberger, D. Cremers, M. Rousson, and R. Ramaraj. 4d shape priors for level set segmentation of the left myocardium in SPECT sequences. In *Medical Image Computing and Computer Assisted Intervention*, volume 4190 of *LNCS*, pages 92–100, October 2006. 2

[27] M. Leventon, W. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Conf. Comput. Vis. Pattern Recog.*, volume 1, pages 316–323, Hilton Head Island, SC, 2000. 2

[28] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194:1–194:17, 2017. 1

[29] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 2, 6, 8

[30] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4823–4834. Curran Associates, Inc., 2020. 3

[31] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[32] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 4

[33] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, Hao Li, Richard Roberts, et al. pagan: real-time avatars using dynamic textures. *ACM Trans. Graph.*, 37(6):258–1, 2018. 3

[34] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *International Conference on Computer Vision*, Oct. 2019. 2

[35] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *International Conference on Computer Vision*, Oct. 2019. 2

[36] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 4

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 6

[38] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, Sep. 2009. 2, 6, 8

[39] Stylianos Ploumpis, Evangelos Ververas, Eimear O' Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick Pears, William AP Smith, Baris Gecer, and Stefanos Zafeiriou. Towards a complete 3d morphable model of the human head. *arXiv preprint arXiv:1911.08008*, 2019. 2

[40] Stylianos Ploumpis, Haoyang Wang, Nick Pears, William A. P. Smith, and Stefanos Zafeiriou. Combining 3d morphable models: A large scale face-and-head model. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[41] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3d faces using convolutional mesh autoencoders. In *ECCV '18*, volume 11207 of *Lecture Notes in Computer Science*, pages 725–741. Springer, 2018. 1

[42] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pages 725–741. Springer International Publishing, 2018. 2

[43] Shunsuke Saito, Liwen Hu, Chongyang Ma, Hikaru Ibayashi, Linjie Luo, and Hao Li. 3d hair synthesis using volumetric variational autoencoders. *ACM Trans. Graph.*, 37(6), Dec. 2018. 2

[44] 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 *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[45] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020. 2

[46] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 2

[47] Jürgen Sturm, Erik Bylow, Fredrik Kahl, and Daniel Cremers. Copyme3d: Scanning and printing persons in 3d. In *Pattern Recognition*, pages 405–414, 2013. 2

[48] Qianru Sun, Ayush Tewari, Weipeng Xu, Mario Fritz, Christian Theobalt, and Bernt Schiele. A hybrid model for identity obfuscation by face replacement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 553–569, 2018. 2

[49] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 2

[50] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed Elgharib, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Fml: Face model learning from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10812–10822, 2019. 6

[51] Ayush Tewari, Mohamed Elgharib, Mallikarjun B R, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. *ACM Trans. Graph.*, 39(6), Nov. 2020. 2

[52] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2Face: Real-time

face capture and reenactment of RGB videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[53] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. PatchNets: Patch-Based Generalizable Deep Implicit 3D Shape Representations. *European Conference on Computer Vision (ECCV)*, 2020. 2

[54] Zerong Zheng, Tao Yu, Qionghai Dai, and Yebin Liu. Deep implicit templates for 3d shape representation. *arXiv preprint arXiv:2011.14565*, 2020. 3