Modeling Facial Geometry using Compositional VAEs

Timur Bagautdinov*[1], Chenglei Wu[2], Jason Saragih[2], Pascal Fua[1], Yaser Sheikh[2]
[1]École Polytechnique Fédérale de Lausanne
[2]Facebook Reality Labs, Pittsburgh
{firstname.lastname}@epfl.ch, {firstname.lastname}@fb.com

Abstract

We propose a method for learning non-linear face geometry representations using deep generative models. Our model is a variational autoencoder with multiple levels of hidden variables where lower layers capture global geometry and higher ones encode more local deformations. Based on that, we propose a new parameterization of facial geometry that naturally decomposes the structure of the human face into a set of semantically meaningful levels of detail. This parameterization enables us to do model fitting while capturing varying level of detail under different types of geometrical constraints.

1. Introduction

Building robust and expressive face models is challenging because they must be able to capture deformations at many different scales. These range from large ones to represent the overall shape of specific person's face to small ones to capture subtle expressions such as a smirk or a frown.

Most existing methods can be roughly split into two categories depending on whether they use global linear models [3, 18, 9] or local ones [34, 39]. While the former are simple to use and usually robust to noise and mismatches, the underlying linear space is over-constrained and does not provide sufficient flexibility to represent high-frequency deformations. By contrast local models bring flexibility by separately modeling local deformations. However, they are also more vulnerable to noise and outliers, and can easily produce non-face shapes. Even recent hybrid methods that enforce global anatomical constraints [39] remain limited to person-specific settings and it is not clear how to extend them to capture facial features across multiple identities.

With the advent of Deep Learning, there have been several attempts at using deep nets for data-driven face reconstruction [35, 11, 29]. However, these methods still rely on global linear models, which precludes from performing required multi-scale modeling.

In this work, we propose a novel method to model multi-scale face geometry that learns the facial geometry from the data without making any restrictive linear assumptions. Our approach starts with the observation that both global and local linear models can be viewed as specific instances of autoencoders. They can therefore both be incorporated into a generic compositional architecture that combines the strengths of both local and global models, while being completely data-driven. In particular, our approach features a new Variational Autoencoder (VAE) with multiple layers of hidden variables that capture various level of geometrical details. In effect, some network layers capture the low-frequency geometry while others represent high-frequency details.

In the experimental evaluation we demonstrate our model's effectiveness on a variety of fitting tasks, including dense depth data, sparse 2D and 3D correspondences, as well as shape-from-shading reconstruction. We show that it can capture high-quality face geometry even when trained using a database featuring only 16 different people.

In short, our main contribution is a model that encodes facial geometry over a range of scales and generalizes to new identities and arbitrary expressions, while being learned from a small number of different people. The last point is important because creating databases of high-quality meshes that cover a wide range of human expressions and a large number of different identities is both expensive and time-consuming.

2. Related Work

One of the main motivations of our work is to demonstrate that it is possible to use deep generative models to learn meaningful geometric representations directly from the data. In this section, we therefore first review existing face models and several recent efforts on applying deep learning to data-driven face reconstruction. We then give a very brief introduction into deep generative models with a focus on VAEs.

* Work done during an internship at Facebook Reality Labs, Pittsburgh.

2.1. Parametric Face Models

Many different global 3D face parameterizations have been proposed over the years. They include Active Appearance Models (AAM) [7], blendshapes [23], principal components analysis (PCA) derived from a set of training shapes [22, 3], and multilinear models [37]. They have been successfully used to overcome the ambiguities associated with monocular face tracking [24, 2, 8, 14, 15, 9, 32]. However, because they are designed to model the *whole* face at once, it is difficult to use them to represent small details without making them exceedingly large and unwieldy.

Local or region-based shape models have therefore also been proposed to remedy this problem. For example Joshi et al. [18] use a region-based blendshape model for keyframe facial animation and automatically determine the best segmentation using a physical model. Na and Jung [25] use local blendshapes for motion capture retargeting and devise a method for choosing the local regions and their corresponding weighting factors automatically. Tena et al. [34] learn a region-based PCA model based on motion capture data, which allows direct local manipulation of the face. Neumann et al. [26] extract sparse localized deformation components from an animated mesh sequence, for the purpose of intuitive editing as well as statistical processing of the face. Brunton et al. [4] rely on many localized multilinear models to reconstruct faces from noisy or occluded point cloud data. All these approaches offer more flexibility than the globals models but at the cost of being less constrained to realistically represent human faces.

Wu et al. [39] propose a hybrid approach that combines a local 3D model made of many overlapping patches, which can be locally deformed, and a global model in the form of anatomical constraints that simulate the existence of a skull and jaw bone. This is effective, but it has to be tailored to each individual, and only considers bone structure, while ignoring other types of constraints.

2.2. Deep Learning for 3D Face Reconstruction.

Deep models have been successfully used for 3D face reconstruction. In [35], the authors propose a weakly-supervised approach to learning a CNN-based regressor from the space of images into a pre-defined semantic space, which includes global pose and facial expressions, as well as illumination and texture. Similarly, in [28], used a large dataset of artificially rendered face images to train a CNN that maps images into the space of facial geometry. Both these approaches, however, rely on a pre-defined geometry space based on a variation of a bilinear AAM model [7].

By contrast, applying deep generative models to learning a geometric representation has been largely overlooked. The approach of [13] is an exception that relies on deep restricted Boltzmann machines to model the shape of the face. However, that approach does not model the entire facial geometry, but is restricted to represent a sparse set of facial landmarks.

2.3. Deep Generative Models

Deep Generative Models, including Variational Autoencoders (VAEs) [20] and Generative Adversarial Networks (GANs) [16, 12, 10], are highly effective at learning complex high-dimensional distributions and have been put to good use for image synthesis and unsupervised learning. However, GANs are notoriously hard to train, which we noticed empirically in preliminary experiments. We therefore chose to rely on VAEs. We provide the basics of VAE below and will use the same formalism in the next section to describe how we use it for our purposes.

Let $\mathcal{M} = \{\mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(M)}\}$ be a set of observations $\mathbf{M}^{(i)}$ which are distributed according to the generative distribution $p(\mathbf{M}^{(i)}, \mathbf{z}^{(i)}; \boldsymbol{\theta}_d) = p(\mathbf{M}^{(i)}|\mathbf{z}^{(i)}; \boldsymbol{\theta}_d) \cdot p(\mathbf{z}^{(i)}; \boldsymbol{\theta}_d)$, where $\mathbf{z}^{(i)}$ is a vector of latent (hidden) variables, and $\boldsymbol{\theta}_d$ are the parameters of the distribution. In theory, these parameters can be learned by maximizing the log-likelihood of the observed data

$$\log p(\mathbf{M}^{1:M}; \boldsymbol{\theta}_d) = \sum_{i=1}^{M} \log p(\mathbf{M}^{(i)}; \boldsymbol{\theta}_d) \ . \qquad (1)$$

In practice, computing the actual log-likelihood is intractable for non-trivial generative models. As a result, a number of approximations have been introduced, including Variational Bayes methods which instead maximize the following lower-bound:

$$\mathcal{L} = \langle \log p(\mathbf{M}, \mathbf{z}; \boldsymbol{\theta}_d) - \log q(\mathbf{z}|\mathbf{M}; \boldsymbol{\theta}_e) \rangle_{q(\mathbf{z}|\mathbf{M};\boldsymbol{\theta}_e)} \ , \qquad (2)$$

where we dropped the indices (i) for clarity and $\langle \cdot \rangle_q$ denotes expectation with respect to the variational distribution q defined over hidden variables $\mathbf{z}$ and parameterized by $\boldsymbol{\theta}_e$. The fact that $\mathcal{L}$ is a lower bound follows directly from Jensen's inequality and Eq. 2 can be rewritten as

$$\mathcal{L} = \langle \log p(\mathbf{M}|\mathbf{z}; \boldsymbol{\theta}_d) \rangle_q - \langle \log \frac{q(\mathbf{z}|\mathbf{M}; \boldsymbol{\theta}_e)}{p(\mathbf{z}; \boldsymbol{\theta}_d)} \rangle_q \ , \qquad (3)$$

where the left-hand term can be understood as a negative reconstruction error of the generative model (decoder) $p(\mathbf{M}|\mathbf{z})$ and the right-hand term is the KL divergence between the approximate posterior (encoder) $q(\mathbf{z}|\mathbf{M})$ and the prior $p(\mathbf{z})$, which acts as a reguralizer. Without this term, there would be no incentive to learn a smooth and meaningful representation for $\mathbf{z}$, which is crucial if we want to then traverse this space when doing model fitting. In the context of deep generative models, both the generative model $p(\mathbf{M}|\mathbf{z})$ and the approximate posterior $q(\mathbf{z}|\mathbf{M})$ are parameterized using deep neural networks. Distribution q is usually taken to be a diagonal Gaussian, but more sophisticated distributions have been investigated in [27, 20, 36].

3. Method

In this section, we first describe the mesh parameterization that enables us to efficiently apply CNNs to the face geometry. We then discuss an important insight of this paper, which is that both global and local linear models that are central to most state-of-the-art approaches to modeling 3D faces can be expressed as shallow auto-encoders. A natural way to increase their flexibility would therefore be to simply replace the linear encoders and decoders by non-linear ones. However, in practice, this would not be enough because model fitting requires a well-behaved parameter space that is well suited for optimization. We therefore show that the convolutional VAEs can be used for this purpose in the global case. Finally, since this results in a model that is more flexible than the original ones but still suffers from the limitations of all global ones, we introduce a compositional version of VAEs, which combines the strength of local and global models by explicitly representing various deformation levels.

3.1. Mesh Representation

Typically, face geometry is represented as a triangular mesh, or, more formally, as a pair $(\mathcal{V}, \mathcal{T})$, where $\mathcal{V} \in \mathcal{R}^{N\times 3}$ is a collection of 3D vertices and $\mathcal{T}$ is a set of triangles that defines the topology. In this work, we keep the same triangulation for all the faces and assume the shape variations are all captured by the $\mathcal{V}$ coordinates. Details on how to perform mesh registration are given in Section 5.1. Further, these coordinates are represented as a 3-channel image $\mathbf{M} \in \mathcal{R}^{H\times W\times 3}$ and the triangles in $\mathcal{T}$ by triplets of the vertex indices of the form $\{(i,j),(i+1,j),(i+1,j+1)\}$ and $\{(i,j),(i,j+1),(i+1,j+1)\}$, as shown in Figure 1. Importantly, this means that pixels that are neighbors in terms of pixel coordinates are also topological neighbors. This makes it natural to perform 2D convolutions on meshes and efficiently use the deep learning machinery.

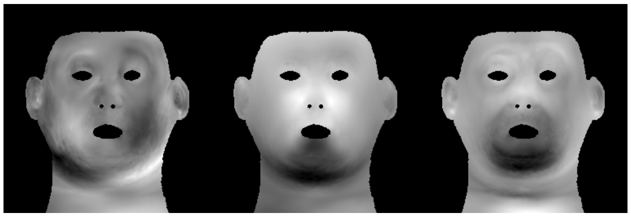


Figure 1. Example of (mean-subtracted) UV parameterization of a face. From left-to-right: x, y, z coordinates.

3.2. Linear Face Models as Autoencoders

A global linear model such as the one of [3] represents all possible face shapes as linear combinations in a set of basis vectors. In [3], it was obtained by performing principal component analysis on a training database.

Formally, we can write

$$\mathbf{h} = \mathbf{W}_e \cdot \mathbf{M} \ , \ \hat{\mathbf{M}} = \mathbf{W}_d \cdot \mathbf{h} \ , \tag{4}$$

where $\mathbf{W}_e \in \mathcal{R}^{k\times 3N}$, $\mathbf{W}_d \in \mathcal{R}^{3N\times k}$ are respectively encoding and decoding matrices, and $\mathbf{h} \in \mathcal{R}^k$ is a set of k linear coefficients, such that $\|\mathbf{M} - \hat{\mathbf{M}}(\mathbf{h})\|$ is mimimized in the space spanned my $\mathbf{W}_e$. The transformations of Eq. 4 can be implemented by a shallow linear auto-encoder, as shown in Figure 2 (a). Given the observations such as depth maps or the 2D positions of sparse landmarks, which we will denote $\mathbf{X}$, fitting a model to it can then be expressed as finding a set of parameters $\hat{\mathbf{h}}$ that maximizes the data likelihood $p(\mathbf{X}|\mathbf{W}_d \cdot \mathbf{h})$.

Local linear models such as [34] give more flexibility than global ones by decoupling the parameters between different parts of the mesh. In practice, this means that $\mathbf{h}$ is factored into independent sets of parameters $\mathbf{h}_\rho$ for distinct patches $\mathbf{M}_\rho$ of the mesh. Assuming that all these parameters are expressed in the same bases $\boldsymbol{\theta}_e, \boldsymbol{\theta}_d$, these local models can be seen as shallow *convolutional* auto-encoders, whose space of potential deformations is captured by a convolutional feature map $\mathbf{h}$, as shown in Figure 2 (b). Bases $\boldsymbol{\theta}_e$ and $\boldsymbol{\theta}_d$ are then the parameters of the convolutional layers of respectively encoder and decoder, which are shared among all the patches.

3.3. Convolutional Mesh VAE

Given that linear models can be viewed as linear auto-encoders, a natural way to extend them and potentially solve the problems discussed in Section 2, is to use non-linear versions of the encoders and decoders.

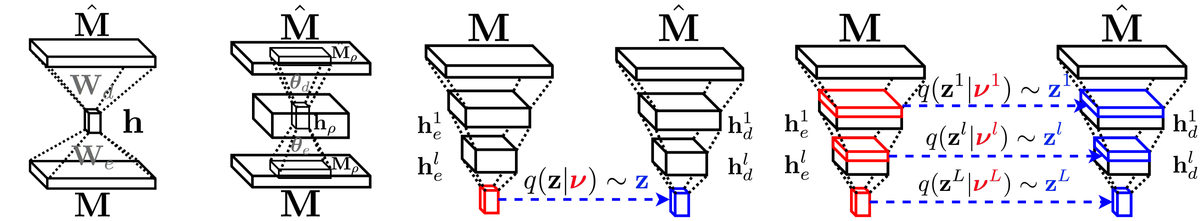
For global models, we therefore write

$$\mathbf{h} = E(\mathbf{M}; \boldsymbol{\theta}_e) \ , \ \hat{\mathbf{M}} = D(\mathbf{h}; \boldsymbol{\theta}_d) \ , \tag{5}$$

where $E(\cdot; \boldsymbol{\theta}_e)$ and $D(\cdot; \boldsymbol{\theta}_d)$ are multi-layer convolutional encoders and decoders, parameterized by weights $\boldsymbol{\theta}_e$ and $\boldsymbol{\theta}_d$ respectively, similarly to architectures in Figure 2 (c)-(d). In a similar manner as for the linear case, we can estimate $\boldsymbol{\theta}_e$ and $\boldsymbol{\theta}_d$ from the training data and then do model fitting by finding the parameter vector $\hat{\mathbf{h}}$ that maximizes $p(\mathbf{X}|D(\mathbf{h}; \boldsymbol{\theta}_d))$.

The non-linear parameterization of Eq. 5 is more flexible than the one of Eq. 4. Unfortunately, it does not guarantee anymore that even small differences in the value of $\mathbf{h}$ from the values observed during training will not result in estimated shapes $\hat{\mathbf{M}} = D(\mathbf{h}; \boldsymbol{\theta}_d)$ which are not representative of the true posterior, or, in other words, which are *not* face-like. To remedy this, we replace the simple auto-encoder of Eq. 5 by a *variational* auto-encoder based on the formalism described in Section 2.3, which ensures the smoothness of the learned space by enforcing a prior on the posterior q.

Namely, we parameterize the distribution over latent variables $q(\mathbf{z}|\mathbf{M}; \boldsymbol{\theta}_e)$ and the generative model $p(\mathbf{M}|\mathbf{z}; \boldsymbol{\theta}_d)$ in terms of a deep net encoder $E(\cdot)$ and decoder $D(\cdot)$ respectively. This yields a variational reformulation of Eq. 5:



(a) Global model (b) Local model (c) Convolutional VAE (d) Compositional VAE

Figure 2. Autoencoding architectures for face geometry.

$$\boldsymbol{\nu} = E(\mathbf{M}; \boldsymbol{\theta}_e)\ ,\ \mathbf{z} \sim q(\mathbf{z}|\boldsymbol{\nu})\ ,\ \hat{\mathbf{M}} = D(\mathbf{z}; \boldsymbol{\theta}_d)\ , \tag{6}$$

where $\boldsymbol{\nu}$ are the parameters of the approximate posterior, which is assumed to be a diagonal Gaussian. In practice, evaluating $\hat{\mathbf{M}}$ now requires sampling from $q(\mathbf{z}|\boldsymbol{\nu})$, which is not a differentiable operation. This was addressed in [21] by representing $\mathbf{z}$ as a deterministic variable that depends on $\boldsymbol{\nu}$ and auxiliary noise, which makes it possible to minimize the lower bound $\mathcal{L}$ of Eq. 2 and Eq. 3 using stochastic gradient descent.

3.4. Compositional Mesh VAE

The non-linear parameterization of Eq. 6 is more flexible than the linear one of Eq. 4 while still providing a latent space that is smooth and easy to optimize over. However, both formulations still depend on a single low-dimensional vector, namely $\mathbf{h}$ in Eq. 4 and $\mathbf{z} \sim q(\cdot|\boldsymbol{\nu})$ in Eq. 6, to represent the shape, which makes it difficult to capture high-frequency deformations.

In this section, we propose a solution to this difficulty by introducing *multiple* layers of hidden variables $\mathbf{z}^{1:L}$, where each individual layer models a separate level of detail. Intuitively, the goal of the encoder is then to gradually *decompose* the input mesh $\mathbf{M}$ into those variables, such that the decoder can then *compose* those individual representations back into a final reconstruction $\hat{\mathbf{M}}$. The higher-level layers, that is, those corresponding to lower l-s, have more degrees of freedom and more local control with smaller receptive field, are therefore well suited to represent the high-frequency geometric components, whereas the lower-level layers have more control over the global shape. This will be demonstrated at the end of the evaluation section.

Formally, the joint distribution for the observed meshes $\mathbf{M}$ and latent variables $\mathbf{z}^{1:L}$ can now be written as

$$p(\mathbf{M}, \mathbf{z}^{1:L}) = p(\mathbf{M}|\hat{\mathbf{M}}(\mathbf{z}^{1:L})) \cdot \prod_{l=1}^{L} p(\mathbf{z}^l|\boldsymbol{\xi}^l)\ , \tag{7}$$

where $\boldsymbol{\xi}^l$ are the parameters of the prior, and the approximate posterior q is factorized over layers l as

$$q(\mathbf{z}^{1:L}|\mathbf{M}; \boldsymbol{\theta}_e) = \prod_{l=1}^{L} q(\mathbf{z}^l|\boldsymbol{\nu}^l)\ . \tag{8}$$

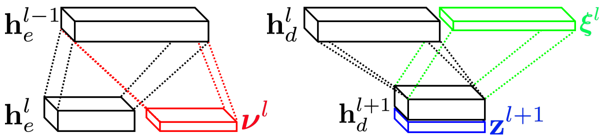


Figure 3. Compositional VAE layers. Encoder (left): given activations $\mathbf{h}_e^{l-1}$ we output the lower-dimensional activations $\mathbf{h}_e^l$ along with the *posterior* parameters $\boldsymbol{\nu}^l$. Decoder (right): given activations $\mathbf{h}_d^{l+1}$ and a sample $\mathbf{z}^{l+1}$ we output the higher-dimensional activation $\mathbf{h}_d^l$ along with the *prior* parameters $\boldsymbol{\xi}^l$.

To account for the new factorized structure of our latent space, we expand the formulation of Eq. 6 and write

$$\begin{aligned} \mathbf{h}_e^l, \boldsymbol{\nu}^l &= E^l(\mathbf{h}_e^{l-1}; \boldsymbol{\theta}_e^l)\ , \\ \mathbf{z}^l &\sim q(\mathbf{z}^l|\boldsymbol{\nu}^l)\ , \\ \mathbf{h}_d^l, \boldsymbol{\xi}^l &= D^l(\mathbf{h}_d^{l+1}, \mathbf{z}^{l+1}; \boldsymbol{\theta}_d^l)\ , \end{aligned} \tag{9}$$

where $\boldsymbol{\nu}^l \in \mathcal{R}^{H^l \times W^l \times C^l}$ and $\boldsymbol{\xi}^l \in \mathcal{R}^{H^l \times W^l \times C^l}$ are parameters of the approximate posterior $q(\mathbf{z}^l|\boldsymbol{\nu}^l)$ and prior $p(\mathbf{z}^l|\boldsymbol{\xi}^l)$, respectively, which we take to be diagonal Gaussians as in the original VAE [21]. During training, the KL term of Eq. 3 ensures that $q(\mathbf{z}^l|\boldsymbol{\nu}^l)$ stays close to the prior $p(\mathbf{z}^l|\boldsymbol{\xi}^l)$, which encourages the model to learn a more well-behaved representation for $\mathbf{z}^l$. Note that, for $\mathbf{z}^L$, we do not have to predict $\mathbf{h}_e^L$, and the corresponding prior is set to zero-mean unit-variance $\boldsymbol{\xi}^L = (\mathbf{0}, \mathbf{I})$. Figure 3 shows a graphical illustration of Eq. 9, and Figure 2 (d) depicts the whole architecture. Note that, the overall architecture is quite similar to the U-Net [30], which is widely used for semantic segmentation, with an important difference that in our model the skip connections are probabilistic.

Finally, substituting Eqs. 7 and 8 into the lower bound of Eq. 3 gives us the training objective that we can optimize given a training set using SGD. We give additional details on this procedure in Section 5.2.

4. Model Fitting

The compositional VAE model described above is designed to effectively encode the facial deformations in different layers of its hidden variables. An important property that is a result of the factorized structure and the variational nature of the model is its ability to extrapolate, which is especially useful for face model fitting given 3D or 2D constraints. In what follows, we describe the model fitting procedure in different application scenarios, ranging from depth map-based face fitting to shading-based face reconstruction from just a single image.

Namely, given generic image data $\mathbf{X}$ and the pre-trained decoder D, our goal is to find parameter vectors $\mathbf{z}^{1:L}$ such that decoded mesh whose shape is given by $\hat{\mathbf{M}} = D(\mathbf{z}^{1:L})$ fits the data as well as possible. Formally, this is equivalent to solving a MAP problem, that is, maximizing

$$\log p(\mathbf{X}|\hat{\mathbf{M}}(\mathbf{z}^{1:L})) + \sum_{l=1}^{L} \log p(\mathbf{z}^l|\boldsymbol{\xi}^l(\mathbf{z}^{l+1})) \ , \qquad (10)$$

wrt $\mathbf{z}^{1:L}$, where $p(\mathbf{X}|\hat{\mathbf{M}})$ is the probability of observing $\mathbf{X}$ if the mesh shape is given by $\hat{\mathbf{M}}$. Note that the prior probability terms act as regularizers that prevent the model parameters from straying too far away from values observed in the training data. While this may be advantageous in the presence of noise, it also limits the ability of the model to extrapolate. In the results section, we will therefore compare results with different combinations of these terms across various types of constraints and noise levels. In practice, we use gradient descent to iteratively optimize Eq. 10. Below, we describe the formulation of the data term $p(\mathbf{X}|\hat{\mathbf{M}})$ for different types of input data.

3D to 3D correspondences. The simplest case is when we know the position $\mathbf{M}_i$ of a subset $\mathcal{I}$ of vertices up to some precision, for example obtained from a multi-view setup. Assuming a Gaussian error distribution with unit variance and conditional independence of individual observations, we write

$$\sum_{i \in \mathcal{I}} \log p(\mathbf{M}_i|\hat{\mathbf{M}}_i) \propto -\sum_{i \in \mathcal{I}} ||\mathbf{M}_i - \hat{\mathbf{M}}_i||_2^2 \ . \qquad (11)$$

2D to 3D correspondences. In realistic scenarios, 3D to 3D correspondences are rarely available but 2D to 3D ones can be established by matching sparse facial landmarks in an image. Therefore, let $\mathcal{I}$ now be the set of vertices $\mathbf{M}_i$ for which we have 2D projections $\mathbf{P}_i \in \mathcal{R}^2$. Given camera intrinsic $\mathbf{K} \in \mathcal{R}^{3\times3}$ and extrinsic $\mathbf{R}|\mathbf{t} \in \mathcal{R}^{3\times4}$ parameters, and making the same Gaussian IID assumptions about the observations, we can write:

$$\sum_{i \in \mathcal{I}} \log p(\mathbf{P}_i|\hat{\mathbf{M}}_i) \propto -\sum_{i \in \mathcal{I}} ||\mathbf{P}_i - \Pi_{\mathbf{K},\mathbf{R}|\mathbf{t}} \hat{\mathbf{M}}_i||_2^2 \ , \qquad (12)$$

where $\Pi_{\mathbf{K},\mathbf{R}|\mathbf{t}} \hat{\mathbf{M}}_i$ are the 2D projections of the model vertices.

Depth maps. Depth cameras have now become an inexpensive and widely available means for face capture. Furthermore, high-quality depth maps can be obtained by stereo matching of high-resolution RGB images. Let $\mathbf{D} \in \mathcal{R}^{H_D \times W_D}$ be such a depth map. We now need to define $p(\mathbf{D}|\hat{\mathbf{M}})$. Ignoring differentiability for a moment, we consider the set of vertices visible from the depth camera point of view $\mathcal{I}_V \subset H \times W$, compute their image coordinates $(\hat{u}_i, \hat{v}_i)$ in the depth map coordinate frame defined by $\mathbf{K}, \mathbf{R}|\mathbf{t}$. Then, we evaluate the difference between the depth value stored at those coordinates $\mathbf{D}_i = \mathbf{D}_i(\hat{u}_i, \hat{v}_i)$ and the one that projected from the 3D vertex position using camera extrinsics $\hat{\mathbf{D}}_i = (\mathbf{R} \cdot \hat{\mathbf{M}}_i + \mathbf{t})_z$. Under the same Gaussian assumptions as before, this allows us to write

$$\sum_{i \in \mathcal{I}_V} \log p(\mathbf{D}_i|\hat{\mathbf{M}}_i) \propto -\sum_{i \in \mathcal{I}_V} ||\mathbf{D}_i - \hat{\mathbf{D}}_i||_2^2 \ . \qquad (13)$$

Unfortunately, self-occlusions make visibility non-differentiable. To overcome this difficulty, we compute $\mathcal{I}_V$ by rendering the mask of visible vertex indices using OpenGL during forward passes and keep $\mathcal{I}_V$ fixed during the backward passes. Furthermore, in order for us to be able to propagate gradients not only through the values of depth, but also through the image coordinates $(\hat{u}, \hat{v})$, we employ a bilinear kernel

$$\mathbf{D}_i = \sum_{u,v} \mathbf{D}(u,v) \max(0, 1-|u-\hat{u}|) \max(0, 1-|v-\hat{v}|) \ , \qquad (14)$$

to perform the differentiable sampling, as in [17].

Shape from Shading Constraints. Another compelling but very challenging application is to fit face model to a single RGB image. Whereas the rough expression can be estimated using sparse 2D-3D correspondences, they are not sufficient to capture identity-specific high-frequency detail. One approach to overcome this is using image formation models. Let $\mathbf{I} \in \mathcal{R}^{H_I \times W_I \times 3}$ be an RGB image. Our goal is now to define $p(\mathbf{I}|\hat{\mathbf{M}})$. We assume a simple Lambertian model, with a single 3-channel light source parameterized by $\mathbf{L} \in \mathcal{R}^{3\times3}$. Further, we use the mesh $\hat{\mathbf{M}}$ to compute vertex normals $\hat{\mathbf{N}}$, which amounts to computing a cross product between two sets of vectors. Then, the model intensity can be computed as $\hat{\mathbf{I}}_i = \mathbf{T}_i \cdot \mathbf{L} \cdot \hat{\mathbf{N}}_i$, given the texture $\mathbf{T}_i$. Computing the texture is a highly non-trivial task, and here we simply set it to be uniform white, assuming that to some extent the albedo can be captured by $\mathbf{L}$. We now can write

$$\sum_{i \in \mathcal{I}_V} \log p(\mathbf{I}_i|\hat{\mathbf{M}}_i) \propto -\sum_{i \in \mathcal{I}_V} ||\mathbf{I}_i - \hat{\mathbf{I}}_i||_2^2 \qquad (15)$$

where we used same approach for sampling and computing $\mathcal{I}_V$ as for the depth maps. Moreover, we also use a similar trick for computing $\mathbf{L}$: at every forward pass, we use the current estimate of $\hat{\mathbf{N}}$ to solve Eq. 15 for $\mathbf{L}$, and then keep it fixed during the backward pass.

5. Evaluation

We start with a description of our face geometry dataset and give some implementation details. We then present quantitative results on several benchmarks and demonstrate qualitatively that our model can be used both to fit noisy depth maps and to perform shape-from-shading. Finally, we present experiments designed to explore the learned latent space and showcase its decompositional power.

5.1. Dataset

A face geometry dataset aligned with a reference topology is required to train and evaluate our model. However, none of the publicly available face shape datasets [6, 5] offer truly high-resolution models, which would not allow us to fully test descriptive power of our compositional model. We thus built a new one that comprises high-quality face geometries using a multi-view camera setup similar to [1] and performing stereo-based 3D face reconstruction. We captured 20 different people, each performing a set of expressions similar to those of blendshapes of [18]. This resulted in 2140 high-quality meshes. To create a uniform face topology, we first defined a generic neutral face template mesh with a precomputed UV map. This generic mesh was then aligned to the mesh for each subject with their expression being neutral. To this end, we performed non-rigid mesh deformation [33] with facial landmark constraints, which were detected on the corresponding RGB images from the multi-view setup [38].

Given those topologically aligned neutral meshes for each individual, we further aligned them to identity-specific peak expression scans using facial landmarks, geometrical constraints, and optical flow-based constraints. This produced fully-aligned meshes, which are all registered to the same topology represented as a UV map of size $H \times W = 256 \times 256$. Finally, we removed from all mesh coordinates the global rotation and translation of the head. Figure 4 depicts some of the fully-registered meshes.

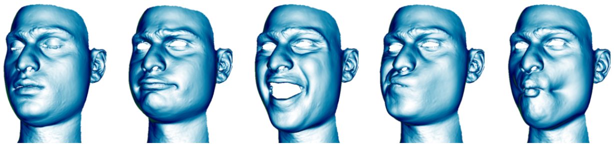


Figure 4. Samples from the dataset.

In all of our numerical experiments, we use a total of 1712 meshes of 16 randomly chosen subjects for training, and 428 meshes of the remaining 4 subjects for testing.

5.2. Implementation Details

All the models are trained using stochastic gradient descent with ADAM [19] optimizer with step size 1×10^{-4} and the hyperparameters $\beta_1 = 0.9, \beta_2 = 0.999$. For the convolutional models, we use identical architecture with 5 residual blocks, with down(up)-sampling after each block of the encoder(decoder). Each block consists of two 4x4 convolutional layers with ELU non-linearities, with weights initialized from a zero-mean Gaussian distribution with standard deviation 0.001. The final 8×8 convolutional representation is mapped to the bottleneck representation using a fully connected layer. Both linear and convolutional VAE models use 128-dimensional bottleneck. For the compositional VAE, we use 64-dimensional bottleneck $\mathbf{z}^6$, all the remaining convolutional maps $\mathbf{z}^5, \ldots, \mathbf{z}^1$-s have 16 channels and the size of the corresponding activation layers. When training variational models, we employ the free-bits technique of [20] with $\lambda = 4$, as we found that it leads to better convergence.

Given a pre-trained model, the model fitting is done by optimizing Eq. 10 with one of the data terms from Section 4 using gradient descent with ADAM optimizer. For noisy depth maps from Section 5.4, we found that using a more robust L1 loss leads to better results. For the 2D-3D and 3D-3D fitting results presented in the following section, the optimization takes around 3-4s per image, and for depth fitting it takes around 8s, on a single NVidia P100 GPU.

5.3. Quantitative Evaluation

In this section, we evaluate quantitatively the behavior of our model and compare it to that of baselines on synthetically generated 3D to 3D correspondences, 2D to 3D correspondences, and depth maps. In all three cases, we perform the fitting as described in Section 4 and will demonstrate in the following section that our approach works equally well on real stereo and shape-from-shading data.

Our baselines include the traditional linear model, introduced in Section 3.2, as well as a the deep convolutional VAE from Section 3.3. We will refer to them as VAE and LINEAR in the result tables below.

We also compare multiple variants of our approach depending on how we handle the prior terms $\log p(\mathbf{z}^l | \boldsymbol{\xi}^l(\mathbf{z}^{l+1}))$ of Eq. 10. We denote them as $\mathbf{z}^l$ for simplicity and can either use them or ignore them. More specifically, we report our results that range from using only $\mathbf{z}^1$ (less priors) to $\mathbf{z}^{1:4}$ (more priors). Recall from Section 3.4 that the lower values of l denote layers that influence most the overall shape and the higher values the fine details. This means that we progressively make constraints more and more global.

Method	0.2%	0.5%	2%	10%
LINEAR	2.795	1.309	1.016	0.980
VAE	1.678	1.317	1.176	1.139
OURS $\mathbf{z}^1$	1.470	1.079	**0.596**	**0.247**
OURS $\mathbf{z}^{1:2}$	1.468	1.121	0.609	0.336
OURS $\mathbf{z}^{1:3}$	1.396	1.020	0.616	0.467
OURS $\mathbf{z}^{1:4}$	**1.320**	**0.986**	0.775	0.717

Table 1. Model fitting with 3D-3D correspondences. RMSE in mm for different proportions of constrained vertices.

3D to 3D correspondences. In Table 1, we report the average RMSE in mm when constraining the 3D position of a subset of mesh vertices, as a function of the proportion of vertices being fixed. While these are chosen randomly for each subsampling level, the error is measured for *all* mesh vertices. All variants of our full compositional model outperform LINEAR and VAE, even when constraining as few as 0.2% of the vertices, which amounts to about 60 3D to 3D correspondences. This suggests that the performance boost is not only attributable to the increased flexibility of our representation but also to the fact it captures the right priors about face geometry. Unsurprisingly, the fewer correspondences we have, the more important the global shape constraints become, as evidenced by the fact that we get the best results when using the priors for all the layers in the 0.2% case but only the ones on the fine details in the 2% and 10% cases.

Method	0.2%	0.5%	2%	10%
LINEAR	4.381	3.691	3.394	3.302
VAE	3.606	3.183	3.114	3.077
OURS $\mathbf{z}^1$	2.690	2.521	**2.390**	**2.330**
OURS $\mathbf{z}^{1:2}$	2.660	2.521	2.396	2.343
OURS $\mathbf{z}^{1:3}$	2.606	**2.512**	2.431	2.396
OURS $\mathbf{z}^{1:4}$	**2.586**	2.545	2.472	2.453

Table 2. Model fitting with 2D-3D correspondences. RMSE in mm for different proportions of constrained vertices.

2D to 3D correspondences. In Table 2, we present fitting results obtained by constraining some mesh vertices to project at the right location in one of the camera views. As before, we report results obtained by constraining in this fashion from 0.2% to 10% of the vertices. Due to 2D-3D ambiguities, this is a more difficult that exploiting 3D to 3D correspondences and the accuracies for all methods are worse than those reported in Table 2. Nevertheless all variants of our approach still outperform the baselines and we observe again that, the sparser the data is, the more important it is to account for the priors at all four levels of our architecture.

Method	$\sigma^2 = 1$	$\sigma^2 = 2$	$\sigma^2 = 3$
LINEAR	3.908	3.924	3.953
VAE	3.167	3.199	3.249
OURS $\mathbf{z}^1$	3.032	3.142	3.252
OURS $\mathbf{z}^{1:2}$	**3.020**	**3.114**	3.215
OURS $\mathbf{z}^{1:3}$	3.079	3.127	**3.191**
OURS $\mathbf{z}^{1:4}$	3.110	3.150	3.226

Table 3. Model fitting with depth data. RMSE in mm for different noise levels.

Depth maps. We generate synthetic depth maps from the ground truth data and corrupt them by adding different levels of IID Gaussian noise. We report our results in Table 3. Since the correspondences must be established and computing visibility is a non-differentiable operation as discussed in Section 4, fitting is more difficult than before. As a result, our method still outperforms the baselines but by a smaller margin. In this case, the best variants of our model are those that enforce priors up to $\mathbf{z}^3$. In other words, in the presence of noisy but dense data, over-constraining the model can be less beneficial.

5.4. Qualitative Results

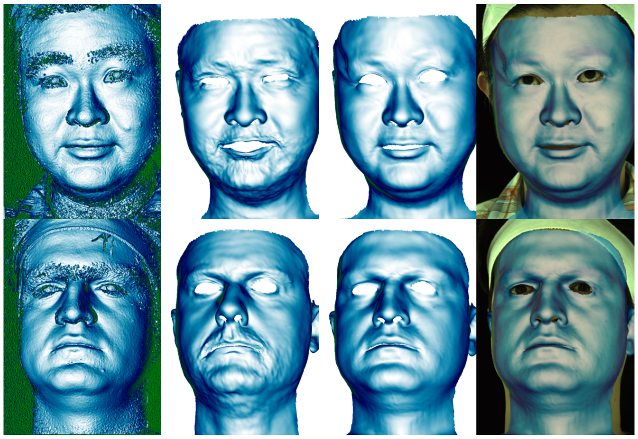


Figure 5. Visual results for fitting noisy depth maps. From left-to-right: input depth map, rendered mesh (LINEAR), rendered mesh (OURS), rendered mesh (OURS) overlaid with the image.

We now turn to more realistic image data to demonstrate the power of our model. To this end, we first captured two additional subjects using a small 3-camera setup, and used stereo to compute noisy depth maps that are representative of what can expect in a real world environment. Figure 5 depicts our results alongside those of LINEAR. Our method correctly captures not only the overall head shape but also fine details whereas LINEAR introduces numerous artifacts instead.

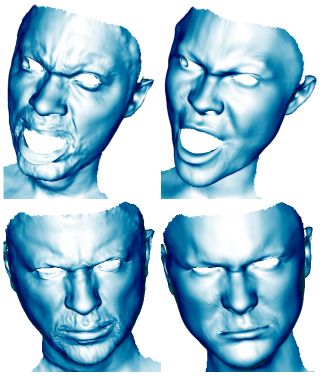


Figure 6. Visual results for shape-from-shading for images from [31]. From left-to-right: rendered mesh (LINEAR), rendered mesh (OURS). Note: for original images, please refer to [31].

In Figure 6, we demonstrate the ability of our model to capture an unusual expression—that of the woman of the top—or face—that of the man at the bottom—using images from 300-W dataset [31]. We initialize the process by using the 2D landmarks provided by [31], to compute the head pose and general expression, and then solve the MAP of Eq. 10 with the data term of Eq. 15. For comparison purposes, we also used LINEAR, which again produced unwanted artifacts.

5.5. Exploring the Latent Space

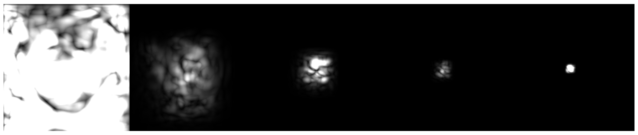


Figure 7. Visualizing the receptive field: how changing the value of a single variable affects the output. Heatmaps represent the MSE between the deformed mesh and the original in the UV space. From left-to-right: $\mathbf{z}^6$ - $\mathbf{z}^2$.

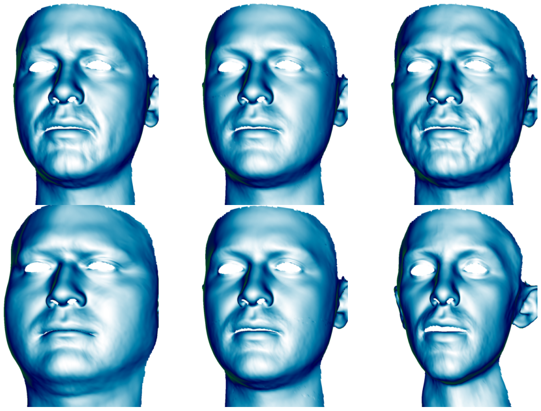


Figure 8. Visualizing the effect of varying the first PCA component of $\mathbf{z}^{1:2}$ (top) and $\mathbf{z}^{4:5}$ (bottom) representations.

We start with an experiment that demonstrates the spatial extent the changes in a single hidden variable at different levels have on the output. For that, we first fix all the variables $\mathbf{z}^{1:L}$ to the values corresponding to the mean face, and then vary a single location in $\mathbf{z}^l$ from the minimum to the maximum value for that variable across the dataset. The results of those variations are shown in Figure 7. Naturally, the variables from the layers which are closer to the bottleneck have global influence on the mesh, and as we go closer to the output, their effective receptive field gradually shrinks.

Further, we explore the learned space by looking at the kind of details that different subsets of variables $\mathbf{z}^{1:L}$ are capturing. PCA is a classical approach for this kind of exploratory analysis. Namely, we first compute the projections $\hat{\mathbf{z}}^{1:L}$ for all the meshes in the dataset by optimizing the posterior of Eq. 10, and then compute the PCA basis via SVD for a subset of variables of interest. We report visual results of varying first principal components of $\mathbf{z}^{1,2}$ and $\mathbf{z}^{5,6}$ in Figure 8. As can be seen from this illustration, the higher layers $\mathbf{z}^{1,2}$, which have smaller receptive field size and more degrees of freedom, capture high-frequency deformations, such as beards and wrinkles. On the other hand, the lower layers $\mathbf{z}^{5,6}$ evidently capture global details, such as the general shape of the head.

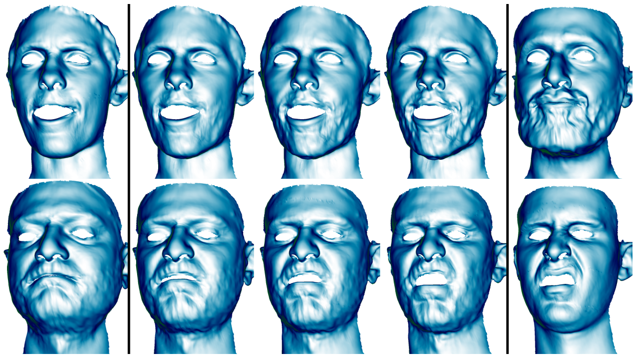


Figure 9. Detail transfer. The leftmost and rightmost columns are the two original meshes. Top: interpolating $\mathbf{z}^{1,2}$ while keeping $\mathbf{z}^{5,6}$ details fixed. Bottom: interpolating $\mathbf{z}^{5,6}$ while keeping $\mathbf{z}^{1,2}$ fixed.

An alternative way to explore the latent space, which is usually employed in deep generative model literature, is to directly traverse the space between the projections of the data samples. To do that we select several random pairs of meshes and find the corresponding values of $\hat{\mathbf{z}}^{1:L}$ by optimizing Eq. 10. Given those, we then can interpolate the values of a certain subset of variables between two projections, while keeping all the others fixed. The visual demonstration of this process for $\mathbf{z}^{1,2}$ and $\mathbf{z}^{5,6}$ is shown in Figure 9. We see that higher layers $\mathbf{z}^{1,2}$ are capturing higher-frequency details, e.g. beards and small variations in eyelids and lips, whereas the lower layers $\mathbf{z}^{5,6}$ are capturing the overall shape of the head and the general expression. This indicates that the model indeed separates the geometrical details into different semantically meaningful layers of representation.

6. Conclusion

We proposed a novel data-driven parameterization for face geometry, and demonstrated its versatility on a variety of model fitting tasks. An exciting direction for future work is investigating alternative architectures for the decoders, such as PixelRNN, and learning to predict hidden representations directly from the images, without a need for optimization. We believe that applying modern generative modeling techniques to geometry data is a very promising field, especially since, unlike for natural images, there exist more straightforward ways to evaluate the quality of the latent space.

Acknowledgements. This work was supported in part by the Swiss National Science Foundation.

References

[1] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. ACM Trans. Graph., 29(4):40:1–40:9, July 2010.

[2] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In ICCV, pages 374–381, 1995.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In Proc. SIGGRAPH, pages 187–194, 1999.

[4] A. Brunton, T. Bolkart, and S. Wuhrer. Multilinear wavelets: A statistical shape space for human faces. In ECCV, 2014.

[5] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). arXiv preprint arXiv:1703.07332, 2017.

[6] 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.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. IEEE TPAMI, 23(6):681–685, 2001.

[8] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In CVPR, page 231, 1996.

[9] M. Dimitrijevic, S. Ilic, and P. Fua. Accurate face models from uncalibrated and ill-lit video sequences. In Computer Vision and Pattern Recognition, volume 2, pages II–II. IEEE, 2004.

[10] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. arXiv preprint arXiv:1605.09782, 2016.

[11] P. Dou, S. K. Shah, and I. A. Kakadiaris. End-to-end 3d face reconstruction with deep neural networks. arXiv preprint arXiv:1704.05020, 2017.

[12] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. arXiv preprint arXiv:1606.00704, 2016.

[13] C. N. Duong, K. Luu, K. G. Quach, and T. D. Bui. Deep appearance models: A deep boltzmann machine approach for face modeling. arXiv preprint arXiv:1607.06871, 2016.

[14] I. Essa, S. Basu, T. Darrell, and A. Pentland. Modeling, tracking and interactive animation of faces and heads using input from video. In Proc. of Computer Animation, page 68, 1996.

[15] P. Fua. Regularized Bundle-Adjustment to Model Heads from Image Sequences Without Calibration Data. International Journal of Computer Vision, 38(2):153–171, July 2000.

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

[17] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In Advances in Neural Information Processing Systems, pages 2017–2025, 2015.

[18] P. Joshi, W. C. Tien, M. Desbrun, and F. Pighin. Learning controls for blend shape based realistic facial animation. In SCA, pages 187–192, 2003.

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

[20] D. P. Kingma, T. Salimans, and M. Welling. Improving variational inference with inverse autoregressive flow. arXiv preprint arXiv:1606.04934, 2016.

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

[22] M. Lau, J. Chai, Y.-Q. Xu, and H.-Y. Shum. Face poser: Interactive modeling of 3d facial expressions using facial priors. ACM Trans. Graph., 29(1):3:1–3:17, Dec. 2009.

[23] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng. Practice and Theory of Blendshape Facial Models. In S. Lefebvre and M. Spagnuolo, editors, Eurographics 2014 - State of the Art Reports. The Eurographics Association, 2014.

[24] H. Li, P. Roivainen, and R. Forcheimer. 3-d motion estimation in model-based facial image coding. IEEE TPAMI, 15(6):545–555, 1993.

[25] K.-G. Na and M.-R. Jung. Local shape blending using coherent weighted regions. The Vis. Comp., 27(6-8):575–584, 2011.

[26] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt. Sparse localized deformation components. ACM Trans. Graphics (Proc. SIGGRAPH Asia), 32(6):179:1–179:10, 2013.

[27] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. arXiv preprint arXiv:1505.05770, 2015.

[28] E. Richardson, M. Sela, and R. Kimmel. 3d face reconstruction by learning from synthetic data. In 3D Vision (3DV), 2016 Fourth International Conference on, pages 460–469. IEEE, 2016.

[29] E. Richardson, M. Sela, R. Or-El, and R. Kimmel. Learning detailed face reconstruction from a single image. arXiv preprint arXiv:1611.05053, 2016.

[30] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 234–241. Springer, 2015.

[31] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 896–903, 2013.

[32] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. IJCV, 91(2):200–215, 2011.

[33] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In Proceedings of the Fifth Eurographics Symposium on Geometry Processing, SGP '07, pages 109–116, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[34] J. R. Tena, F. De la Torre, and I. Matthews. Interactive region-based linear 3d face models. ACM Trans. Graphics (Proc. SIGGRAPH), 30(4):76:1–76:10, 2011.

[35] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez, and C. Theobalt. Mofa: Model-based deep con-

volutional face autoencoder for unsupervised monocular reconstruction. arXiv preprint arXiv:1703.10580, 2017.

[36] J. M. Tomczak and M. Welling. Vae with a vampprior. arXiv preprint arXiv:1705.07120, 2017.

[37] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. ACM Trans. Graphics (Proc. SIGGRAPH), 24(3):426–433, 2005.

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

[39] C. Wu, D. Bradley, M. Gross, and T. Beeler. An anatomically-constrained local deformation model for monocular face capture. ACM Trans. Graph., 35(4):115:1–115:12, July 2016.