High-Fidelity Monocular Face Reconstruction based on an Unsupervised Model-based Face Autoencoder

Ayush Tewari, Michael Zollhöfer, Florian Bernard, Pablo Garrido,
Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt

(Invited Paper)

Abstract—
In this work we propose a novel model-based deep convolutional autoencoder that addresses the highly challenging problem of reconstructing a 3D human face from a single in-the-wild color image. To this end, we combine a convolutional encoder network with an expert-designed generative model that serves as decoder. The core innovation is the differentiable parametric decoder that encapsulates image formation analytically based on a generative model. Our decoder takes as input a code vector with exactly defined semantic meaning that encodes detailed face pose, shape, expression, skin reflectance and scene illumination. Due to this new way of combining CNN-based with model-based face reconstruction, the CNN-based encoder learns to extract semantically meaningful parameters from a single monocular input image. For the first time, a CNN encoder and an expert-designed generative model can be trained end-to-end in an unsupervised manner, which renders training on very large (unlabeled) real world datasets feasible. The obtained reconstructions compare favorably to current state-of-the-art approaches in terms of quality and richness of representation. This work is an extended version of [1], where we additionally present a stochastic vertex sampling technique for faster training of our networks, and moreover, we propose and evaluate analysis-by-synthesis and shape-from-shading refinement approaches to achieve a high-fidelity reconstruction.

1 Introduction

Detailed, dense 3D reconstruction of the human face from image data is a longstanding problem in computer vision and computer graphics. Previous approaches have tackled this challenging problem using calibrated multi-view data or uncalibrated photo collections [2], [3]. Robust and detailed three-dimensional face reconstruction from a single arbitrary in-the-wild image, e.g., downloaded from the Internet, is still an open research problem due to the high degree of variability of uncalibrated photos in terms of resolution and employed imaging device. In addition, in unconstrained photos, faces show a high variability in global pose, facial expression, and are captured under diverse and difficult lighting. Detailed 3D face reconstruction is the foundation for a broad scope of applications, which range from robust face recognition, over emotion estimation, to complex image manipulation tasks. In many applications, faces should ideally be reconstructed in terms of meaningful low-dimensional model parameters, which facilitates interpretation and manipulation of reconstructions (cf. [4]).

Recent monocular reconstruction methods broadly fall into two categories: Generative and regression-based. Generative approaches fit a parametric face model to image and video data, e.g., [5], [6], [7], by optimizing the alignment between the projected model and the image [4], [8], [9], [10], [11]. State-of-the-art generative approaches capture very detailed and complete 3D face models on the basis of semantically meaningful low-dimensional parameterizations [4], [8]. Unfortunately, the fitting energies are usually highly non-convex. Good results thus require an initialization close to the global optimum, which is only possible with some level of control during image capture or additional input data, e.g., detected landmarks.

Only recently, the first regression-based approaches for dense 3D face reconstruction based on deep convolutional neural networks were proposed. Richardson et al. [12] use iterative regression to obtain a high quality estimate of pose, shape and expression, and finer scale surface detail [13] of a face model. The expression-invariant regression approach of Tran et al. [14] obtains high-quality estimates of shape and skin reflectance. Based on an image-to-image translation network, Sela et al. [15] obtain the facial geometry from a single image by translating the input image to a depth map. Unfortunately, these approaches can only be trained in a supervised fashion on corpora of densely annotated facial images whose creation poses a major obstacle in practice. In particular, the creation of a training corpus of photo-realistic synthetic facial images that include facial hair, parts of the upper body and a consistent background is challenging. While the refinement network of Richardson et al. [13] can be trained in an unsupervised manner, their coarse shape regression network requires synthetic ground truth data for training. Also, the quality and richness of representation (e.g., illumination and colored reflectance in addition to geometry) of these methods does not match the best generative ones. However, trained networks are efficient to evaluate and can be trained to achieve remarkable robustness under difficult real world conditions.

This paper contributes a new type of model-based face autoencoder (MoFA) that joins forces of state-of-the-art generative and CNN-based regression approaches for dense 3D face reconstruction via a deep integration of the two on an architectural level. Our network architecture is inspired by recent progress on deep convolutional autoencoders, which, in their original form, couple a CNN encoder and a CNN

- *A. Tewari, H. Kim, F. Bernard, and C. Theobalt are with the Max-Planck-Institute for Informatics*
- *M. Zollhöfer is with Stanford University*
- *P. Garrido and P. Pérez are with Technicolor*

decoder through a code-layer of reduced dimensionality [16], [17], [18]. Unlike previously used CNN-based decoders, our convolutional autoencoder deeply integrates an expert-designed decoder. This layer implements, in closed form, an elaborate generative analytically-differentiable image formation model on the basis of a detailed parametric 3D face model [5]. Some previous fully CNN-based autoencoders tried to disentangle [19], [20], but could not fully guarantee the semantic meaning of code layer parameters. In our new network, exact semantic meaning of the code vector, i.e., the input to the decoder, is ensured by design. Moreover, our decoder is compact and does not need training of enormous sets of unintuitive CNN weights.

Unlike previous CNN regression-based approaches for face reconstruction, a single forward pass of our network estimates a much more complete face model, including pose, shape, expression, skin reflectance, and illumination, at a high quality. Our new network architecture allows, for the first time, combined end-to-end training of a sophisticated model-based (generative) decoder and a CNN encoder, with error backpropagation through all layers. It also allows, for the first time, unsupervised training of a network that reconstructs dense and semantically meaningful faces on unlabeled in-the-wild images via a dense photometric training loss. In consequence, our network generalizes better to real world data compared to networks trained on synthetic face data [12], [13]. This article builds upon the work of Tewari et al. [1]. In this version, we give more implementation details, introduce a stochastic vertex sampling strategy to train the networks faster, and also provide an extended evaluation of the original approach. Since learning-based approaches have limited capacity, they have to trade-off the quality of individual reconstructions in order to work on a diverse range of images. Therefore, we also present and evaluate two additional optimization-based techniques that can be added as refinement steps to further improve the quality of the results. Our focus is on a fast data-parallel implementation of these two additional steps.

2 Related Work

In this section, we summarize previous works that are related to our approach. We focus on parametric model fitting and CNN approaches in the context of monocular face reconstruction. For further work on general template-based mesh tracking, please refer to [2], [9], [10], [11], [21].

Parametric Face Models: Active Appearance Models (AAMs) use a linear model for jointly capturing shape and texture variation [22]. Matching an AAM to an image is a registration problem, usually tackled via energy optimization. A closely related approach to AAMs is the 3D morphable model of faces (3DMM) [5], which has been used for learning facial animations from 3D scans [6]. In [7], a parametric head model has been employed to modify the relative head pose and camera parameters of portrait photos. Recently, a 3DMM that incorporates a feature-based texture model that is obtained from in-the-wild images has been proposed [23].

Monocular Optimization-based Reconstruction: Many monocular reconstruction approaches solve an optimization problem to fit a model to a given image. For example, the 3DMM has been used for monocular reconstruction [24] and image collection-based reconstruction [3]. In [8], high-quality 3D face rigs are obtained from monocular RGB video based on a multi-layer model. Even real-time facial reconstruction and reenactment has been achieved [4], [25]. Compared to optimization-based approaches, ours differs in two main regards. First, our network efficiently *regresses model parameters* without requiring iterative optimization. Second, given a cropped face image, our method does *not require an initialization* of the model parameters, which is a significant advantage over optimization-based techniques.

Deep Learning for Coarse Face Reconstruction: The detection of facial landmarks in images is an active area of research [26], [27]. Various approaches are based on deep learning, including CNN cascades [28], [29], a deep face shape model based on Restricted Boltzmann Machines [30], a recurrent network with long-short term memory [18], a recurrent encoder-decoder network for real-time facial landmark detection in video [31], or a two-stage convolutional part heatmap regression approach [32]. In [33], a multi-task CNN is trained to predict several face-related parameters (e.g. pose, gender, age), in addition to facial landmarks. These deep learning approaches share common limitations: They are trained in a *supervised* manner and *predict only sparse information*. In contrast, our approach works *unsupervised* and obtains a *dense* reconstruction by regressing generative model parameters.

Deep Learning for Dense Face Reconstruction: Apart from the approaches mentioned above, there exist several dense deep learning approaches. A multilayer generative model based on deep belief networks for the generation of images under new lighting has been introduced in [34]. A face identity-preserving (FIP) descriptor has been proposed for reconstructing a face image in a canonical view [35]. The *Multi-View Perceptron* approach for face recognition learns disentangled view and facial identity parameters based on a training corpus that provides annotations of these dimensions [36]. The generation of faces from attributes [37] and dense shape regression [38] have also been studied. Non-linear variants of AAMs based on Deep Boltzmann Machines have been presented in [39], [40]. In [12], a CNN is trained using synthetic data for extracting the face geometry from a single image. Unsupervised refinement of these reconstructions has been proposed in [13]. In [15], the authors propose an image-to-image translation network that converts the input image into a depth image along with a facial correspondence map. Tuan Tran et al. [14] used photo collections to obtain the ground truth parameters from which a CNN is trained for regressing facial identity. In [41], a CNN is trained under controlled conditions in a supervised fashion for facial animation tasks. A framework for face hallucination from low-resolution face images has been proposed in [42]. Recently, data-driven approaches for the synthesis of photorealistic texture maps [43] and fine-scale geometric skin detail [44] have been proposed. Jackson et al. [45] train a CNN that directly regresses a volumetric 3D face representation from a single image. In [46], a CNN is used to estimate surface normals from a given input image. Dou et al. [47] trained a network to regress the 3DMM parameters using synthetic images, akin to [12]. All the discussed approaches require annotated training data. Since the annotation of a large image body is extremely expensive,

some approaches (e.g. [12], [13]) resort to synthetic data. However, synthetic renderings usually lack realistic features, which has a negative impact on the reconstruction accuracy. In contrast, our approach uses *real data* and does not require ground truth model parameters.

Autoencoders: Autoencoders approximate the identity mapping by coupling an encoding stage with a decoding stage to learn a *compact intermediate description*, the so-called *code vector*. They have been used for nonlinear dimensionality reduction [16] and to extract biologically plausible image features [17]. An appealing characteristic is that these architectures are in general *unsupervised*, i.e., no labeled data is required. Closely related are approaches that consider the encoding or decoding stage individually, such as inverting a generative model [48], or generating images from code vectors [49]. Autoencoders have been used to tackle a wide range of face-related tasks, including stacked progressive autoencoders for face recognition [50], real-time face alignment [51], face recognition using a supervised autoencoder [52], learning of face representations with a stacked autoencoder [53], or face de-occlusion [18]. The *Deep Convolutional Inverse Graphics Network* (DC-IGN) learns *interpretable* graphics codes that allow the reproduction of images under different conditions (e.g. pose and lighting) [19]. This is achieved by using mini-batches where only a single scene parameter is known to vary. The disentanglement of code variables, such as shape and scene-related transformations has been considered in [20]. Our proposed approach stands out from existing techniques, since we consider the *full set of meaningful parameters* and do not need to group images according to known variations.

Deep Integration of Expert Layers: Inspired by *Spatial Transformer Networks* [54], the *gvnn* library implements low-level geometric computer vision layers [55]. Unsupervised volumetric 3D object reconstruction from a single-view by *Perspective Transformer Nets* has been demonstrated in [56]. Unlike these approaches, we tackle a higher level computer vision task, namely the monocular reconstruction of semantically meaningful parameters for facial geometry, expression, illumination, and camera extrinsics. Recently, using a 3DMM as a spatial transformer within an unsupervised learning framework to normalize images in terms of 3D head pose and self-occlusion was proposed in [57]. Bhagavatula et al. [58] use a 3D spatial transformer to learn 3D pose and subject-specific shape even in unconstrained poses.

3 Overview

Our novel deep convolutional model-based face autoencoder enables unsupervised end-to-end learning of meaningful semantic face and rendering parameters, see Fig. 1. To this end, we combine convolutional encoders with an expert-designed differentiable model-based decoder that analytically implements image formation. The decoder generates a realistic synthetic image of a face and enforces semantic meaning by design. Rendering is based on an image formation model that enforces full semantic meaning via a parametric face prior. More specifically, we independently parameterize pose, shape, expression, skin reflectance and illumination. The synthesized image is compared to the input image using a robust photometric loss E_{loss} that includes statistical regularization of the face. In combination, this enables unsupervised end-to-end training of our networks. 2D facial landmark locations can be optionally provided to add a surrogate loss for faster convergence and improved reconstructions, see Sec. 6. Note, both scenarios require no supervision of the semantic parameters. After training, the encoder part of the network enables regression of a dense face model and illumination from a single monocular image, without requiring any other input, such as landmarks.

4 Semantic Code Vector

The semantic code vector $\mathbf{x} \in \mathbb{R}^{257}$ parameterizes the facial expression $\boldsymbol{\delta} \in \mathbb{R}^{64}$, shape $\boldsymbol{\alpha} \in \mathbb{R}^{80}$, skin reflectance $\boldsymbol{\beta} \in \mathbb{R}^{80}$, camera rotation $\mathbf{T} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$, and the scene illumination $\boldsymbol{\gamma} \in \mathbb{R}^{27}$ in a unified manner:

$$\mathbf{x} = (\underbrace{\boldsymbol{\alpha},\ \boldsymbol{\delta},\ \boldsymbol{\beta}}_{\text{face}},\ \underbrace{\mathbf{T},\ \mathbf{t},\ \boldsymbol{\gamma}}_{\text{scene}})\ . \tag{1}$$

In the following, we describe the parameters that are associated with the employed face model. The parameters that govern image formation are described in Sec. 5.

The face is represented as a manifold triangle mesh with $N = 24$k vertices $\mathbf{V} = \{\mathbf{v}_i \in \mathbb{R}^3 | 1 \leq i \leq N\}$. The associated vertex normals $\mathbf{N} = \{\mathbf{n}_i \in \mathbb{R}^3 | 1 \leq i \leq N\}$ are computed using a local one-ring neighborhood. The spatial embedding $\mathbf{V}$ is parameterized by an affine face model:

$$\mathbf{V} = \hat{\mathbf{V}}(\boldsymbol{\alpha}, \boldsymbol{\delta}) = \mathbf{A}_s + \mathbf{E}_s\boldsymbol{\alpha} + \mathbf{E}_e\boldsymbol{\delta}\ . \tag{2}$$

Note that, by abuse of notation, here we represent the point-set $\mathbf{V}$ as $3N$-dimensional vector. Here, the average face shape $\mathbf{A}_s$ has been computed based on 200 (100 male, 100 female) high-quality face scans [5]. The linear PCA bases $\mathbf{E}_s \in \mathbb{R}^{3N\times 80}$ and $\mathbf{E}_e \in \mathbb{R}^{3N\times 64}$ encode the modes with the highest shape and expression variation, respectively. We obtain the expression basis by applying PCA to the combined set of blendshapes of [59] and [60], which have been re-targeted to the face topology of [5] using deformation transfer [61]. The PCA basis covers more than 99% of the variance of the original blendshapes.

In addition to facial geometry, we also parameterize per-vertex skin reflectance $\mathbf{R} = \{\mathbf{r}_i \in \mathbb{R}^3 | 1 \leq i \leq N\}$ based on an affine parametric model:

$$\mathbf{R} = \hat{\mathbf{R}}(\boldsymbol{\beta}) = \mathbf{A}_r + \mathbf{E}_r\boldsymbol{\beta}\ . \tag{3}$$

Here, the average skin reflectance $\mathbf{A}_r$ has been computed based on [5] and the orthogonal PCA basis $\mathbf{E}_r \in \mathbb{R}^{3N\times 80}$ captures the modes of highest variation. Note, all basis vectors are already scaled with the appropriate standard deviations $\sigma_k^\bullet$ such that $\mathbf{E}_\bullet^T\mathbf{E}_\bullet = \text{diag}(\cdots, [\sigma_k^\bullet]^2, \cdots)$.

5 Parametric Model-based Decoder

Given a scene description in the form of a semantic code vector $\mathbf{x}$, our parametric decoder generates a realistic synthetic image of the corresponding face. Since our image formation model is fully analytical and differentiable, we also implement an efficient backward pass that inverts image formation via standard backpropagation. This enables unsupervised end-to-end training of our network. The image formation model that we employ is described in the following.

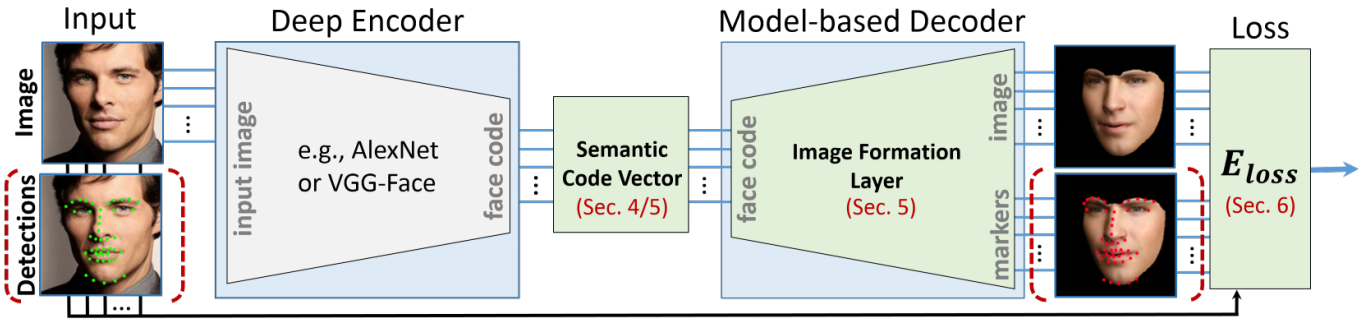

Fig. 1. Our deep model-based face autoencoder enables unsupervised end-to-end learning of semantic parameters, such as pose, shape, expression, skin reflectance and illumination. An optional landmark-based surrogate loss enables faster convergence and improved reconstruction results, see Sec. 6. Both scenarios require no supervision of the semantic parameters during training.

Perspective Camera: We render realistic facial imagery using a pinhole camera model under a full perspective projection $\Pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ that maps camera space coordinates onto screen space coordinates. The position and orientation of the camera in world space is given by a rigid transformation, which we parameterize based on a rotation $\mathbf{T} \in \mathbf{SO}(3)$ and a global translation $\mathbf{t} \in \mathbb{R}^3$. Hence, the functions $\Phi_{\mathbf{T},\mathbf{t}}(\mathbf{v}) = \mathbf{T}^{-1}(\mathbf{v}-\mathbf{t})$ and $\Pi \circ \Phi_{\mathbf{T},\mathbf{t}}(\mathbf{v})$ transform an arbitrary point $\mathbf{v}$ from world space into camera space and further into screen space, respectively.

Illumination Model: We represent scene illumination using Spherical Harmonics (SH) [62]. Here, we assume distant low-frequency illumination and a purely *Lambertian* surface reflectance. Thus, we evaluate the radiosity at vertex $\mathbf{v}_i$ with surface normal $\mathbf{n}_i$ and skin reflectance $\mathbf{r}_i$ as follows:

$$C(\mathbf{r}_i, \mathbf{n}_i, \boldsymbol{\gamma}) = \mathbf{r}_i \cdot \sum_{b=1}^{B^2} \boldsymbol{\gamma}_b \mathbf{H}_b(\mathbf{n}_i) \ . \tag{4}$$

The $\mathbf{H}_b : \mathbb{R}^3 \rightarrow \mathbb{R}$ are SH basis functions and the $B^2 = 9$ coefficients $\boldsymbol{\gamma}_b \in \mathbb{R}^3$ ($B = 3$ bands) parameterize colored illumination using the red, green and blue channel.

Image Formation: We render realistic images of the face using the presented camera and illumination model. To this end, in the forward pass $\mathcal{F}$, we compute the screen space position $\mathbf{u}_i(\mathbf{x})$ and associated pixel color $\mathbf{c}_i(\mathbf{x})$ for each $\mathbf{v}_i$:

$$\begin{aligned} \mathcal{F}_i(\mathbf{x}) &= [\mathbf{u}_i(\mathbf{x}), \mathbf{c}_i(\mathbf{x})]^T \in \mathbb{R}^5 \ , \\ \mathbf{u}_i(\mathbf{x}) &= \Pi \circ \Phi_{\mathbf{T},\mathbf{t}}\big(\hat{\mathbf{V}}_i(\boldsymbol{\alpha}, \boldsymbol{\delta})\big) \ , \\ \mathbf{c}_i(\mathbf{x}) &= C\big(\hat{\mathbf{R}}_i(\boldsymbol{\beta}), \mathbf{T}\mathbf{n}_i(\boldsymbol{\alpha}, \boldsymbol{\delta}), \boldsymbol{\gamma}\big) \ . \end{aligned} \tag{5}$$

Here, $\mathbf{T}\mathbf{n}_i$ transforms the world space normals into camera space and γ models illumination in camera space.

Backpropagation: To enable training, we implement a backward pass that inverts image formation:

$$\mathcal{B}_i(\mathbf{x}) = \frac{\mathrm{d}\mathcal{F}_i(\mathbf{x})}{\mathrm{d}(\boldsymbol{\alpha},\ \boldsymbol{\delta},\ \boldsymbol{\beta},\ \mathbf{T},\ \mathbf{t},\ \boldsymbol{\gamma})} \in \mathbb{R}^{5 \times 257} \ . \tag{6}$$

This requires the computation of the gradients of the image formation model (see Eq. (5)) with respect to the face and scene parameters. For high efficiency during training, we evaluate the gradients in a data-parallel manner, see Sec. 6.

6 Loss Layer

We employ a robust dense photometric loss function that enables efficient end-to-end training of our networks. The loss is inspired by recent optimization-based approaches [4], [8] and combines three terms:

$$E_{\text{loss}}(\mathbf{x}) = \underbrace{w_{\text{land}} E_{\text{land}}(\mathbf{x}) + w_{\text{photo}} E_{\text{photo}}(\mathbf{x})}_{\text{data term}} + \underbrace{w_{\text{reg}} E_{\text{reg}}(\mathbf{x})}_{\text{regularizer}} \ . \tag{7}$$

Here, we enforce sparse landmark alignment E_{land}, dense photometric alignment E_{photo} and statistical plausibility E_{reg} of the modeled faces. Note, E_{land} is optional and implements a surrogate loss that can be used to speed up convergence, see Sec. 8. The binary weight $w_{\text{land}} \in \{0, 1\}$ toggles this constraint. The constant weights $w_{\text{photo}} = 1.92$ and $w_{\text{reg}} = 2.9 \times 10^{-5}$ balance the contributions of the objectives.

Dense Photometric Alignment: The goal of the encoder is to predict model parameters that lead to a synthetic face image that matches the provided monocular input image. To this end, we employ dense photometric alignment, similar to [4], on a per-vertex level using a robust $\ell_{2,1}$-norm:

$$E_{\text{photo}}(\mathbf{x}) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \Big\| \mathcal{I}\big(\mathbf{u}_i(\mathbf{x})\big) - \mathbf{c}_i(\mathbf{x}) \Big\|_2 \ . \tag{8}$$

Here, $\mathcal{I}$ is an image of the training corpus and for occlusion awareness we iterate over all visible vertices, which we approximate as the set of front facing vertices $\mathcal{V}$.

Sparse Landmark Alignment: In addition to dense photometric alignment, we propose an optional surrogate loss based on detected facial landmarks [63]. We use a subset of 46 landmarks (out of 66), see Fig. 1. Given the subset $\mathcal{L} = \{(\mathbf{s}_j, c_j, k_j)\}_{j=1}^{46}$ of detected 2D landmarks $\mathbf{s}_j \in \mathbb{R}^2$, with confidence $c_j \in [0, 1]$ (1 confident) and corresponding model vertex index $k_j \in \{1, ..., N\}$, we enforce the projected 3D vertices to be close to the 2D detections:

$$E_{\text{land}}(\mathbf{x}) = \sum_{j=1}^{46} c_j \cdot \Big\| \mathbf{u}_{k_j}(\mathbf{x}) - \mathbf{s}_j \Big\|_2^2 \ . \tag{9}$$

Please note, this surrogate loss is optional. Our networks can be trained fully unsupervised without supplying these sparse constraints. After training, landmarks are never required.

Statistical Regularization: During training, we further constrain the optimization problem using statistical regularization [5] on the model parameters:

$$E_{\text{reg}}(\mathbf{x}) = \sum_{k=1}^{80} \boldsymbol{\alpha}_k^2 + w_{\boldsymbol{\beta}} \sum_{k=1}^{80} \boldsymbol{\beta}_k^2 + w_{\boldsymbol{\delta}} \sum_{k=1}^{64} \boldsymbol{\delta}_k^2 \ . \tag{10}$$

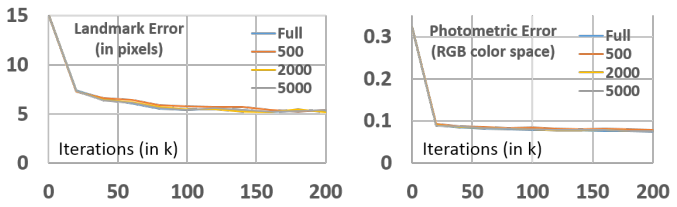

Fig. 2. Quantitative evaluation of stochastic sampling on real data. Even drastic sampling of $\approx 2\%$ of vertices only marginally reduces the quality of the reconstruction results.

This constraint enforces plausible facial shape $\boldsymbol{\alpha}$, expression $\boldsymbol{\delta}$ and skin reflectance $\boldsymbol{\beta}$ by preferring values close to the average (the basis of the linear face model is already scaled by the standard deviations). The parameters $w_{\boldsymbol{\beta}} = 1.7 \times 10^{-3}$ and $w_{\boldsymbol{\delta}} = 0.8$ balance the importance of the terms. Note, we do not regularize pose $(\mathbf{T}, \mathbf{t})$ and illumination $\boldsymbol{\gamma}$.

Backpropagation: To enable training via stochastic gradient descent during backpropagation, the gradient of the robust loss is passed backward to our model-based decoder and is combined with $\mathcal{B}_i(\mathbf{x})$ using the chain rule.

Data-parallel GPU Implementation: We implement Eq. (8) in an iteratively reweighted fashion as follows:

$$E_{\text{photo}}(\mathbf{x}) = \frac{1}{N} \sum_{i \in \mathcal{V}} \frac{1}{C_i} \left\| \mathcal{I}\big(\mathbf{u}_i(\mathbf{x})\big) - \mathbf{c}_i(\mathbf{x}) \right\|_2^2, \qquad (11)$$

where $C_i = \|\mathcal{I}(\mathbf{u}_i(\mathbf{x}^{\text{old}})) - \mathbf{c}_i(\mathbf{x}^{\text{old}})\|_2$. Here, $\mathbf{x}^{\text{old}}$ is the estimate for the code vector in the current iteration. Moreover, since the computation of the number of visible vertices $|\mathcal{V}|$ is expensive (since it would require an additional pass over the vertices), here we approximate it with N. Our loss function can now be represented as a sum of squares of individual residuals, i.e., $E_{\text{loss}}(\mathbf{x}) = \mathbf{F}^T(\mathbf{x})\mathbf{F}(\mathbf{x})$, where $\mathbf{F} : \mathbb{R}^{257} \rightarrow \mathbb{R}^M$ is a vector-valued function such that $\mathbf{F}(\mathbf{x})$ contains all the $M = |V|+46+80+80+64$ residuals of the energy (7). For obtaining high performance, we parallelize the computation of $\mathbf{F}$ to exploit the data-parallel computing power of modern graphics cards, i.e., all elements of the vector $\mathbf{F}$ are computed fully in parallel (each entry by a dedicated thread). In the forward pass, we compute $E_{\text{loss}} = \mathbf{F}^T\mathbf{F}$ using block reductions. The local dot product in each block is computed using shared memory and thread synchronization. Results from different blocks are added on the CPU. In the backward pass, the gradients of E_{loss} can be calculated as

$$\frac{\mathrm{d}E_{\text{loss}}(\mathbf{x})}{\mathrm{d}\mathbf{x}} = 2\mathbf{J}^T(\mathbf{x})\mathbf{F}(\mathbf{x}) \ , \qquad (12)$$

where $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{M \times 257}$ is the Jacobian of $\mathbf{F}$ at $\mathbf{x}$. $\mathbf{J}$ is computed similarly to $\mathbf{F}$ by using one thread per entry of the matrix. The dense matrix-vector multiplication can be interpreted as computing a dot product for each element of $\mathbf{x}$, which is done similarly to the forward pass. The updated mesh (geometry and albedo) for the next forward-backward pass is computed based on a matrix-vector multiplication and we use one thread per entry of the output vector.

7 Stochastic Sampling

Since our MoFA depends on several parameters (the weighting of the individual energy terms, the relative learning rates

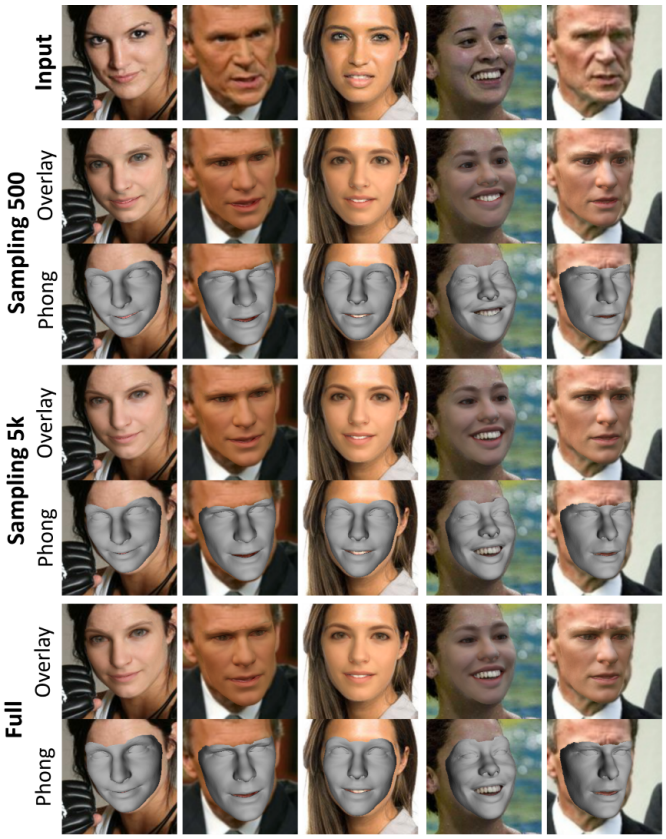

Fig. 3. Qualitative comparison of MoFA with and without stochastic sampling. Stochastic sampling of vertices lets us train networks much faster with comparable results to networks trained using all vertices.

for different output parameters, and other network hyperparameters), finding a good configuration is a repetitive task that requires several (user-guided) iterations. The bottleneck of this procedure is the relatively long training time of the network. In order to speed-up this process, we make use of a stochastic sampling strategy. The basic idea is to randomly sample a small subset of vertices for each input image and then merely backpropagate the error for this small set. To be more specific, we define the energy E_{photo} in (7) for a subset of sampled vertices $\mathbf{S} \subseteq \mathbf{V}$. Let $\mathcal{S} \subseteq \mathbf{S}$ be the subset of visible vertices. The loss is then defined as the sum of model-vertex-specific energy terms E^i_{photo}, i.e.,

$$E_{\text{photo}} = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} E^i_{\text{photo}} \, , \qquad (13)$$

where

$$E^i_{\text{photo}} := \|\mathcal{I}\big(\mathbf{u}_i(\mathbf{x})\big) - \mathbf{c}_i(\mathbf{x})\|_2. \qquad (14)$$

We implement this energy similarly as done in (11). Using this sampling strategy for training can be interpreted as stochastic gradient descent not only over the set of images in the training set, but also over the face vertices. Note that since our semantically defined code vector has global influence, i.e., each vertex of the reconstruction influences all parameters of the network through E^i_{photo}, this is a valid sampling strategy.

Evaluation of the Stochastic Sampling: We quantitatively evaluate the sampling strategy (Fig. 2) for different numbers of samples used while training. As can be seen,

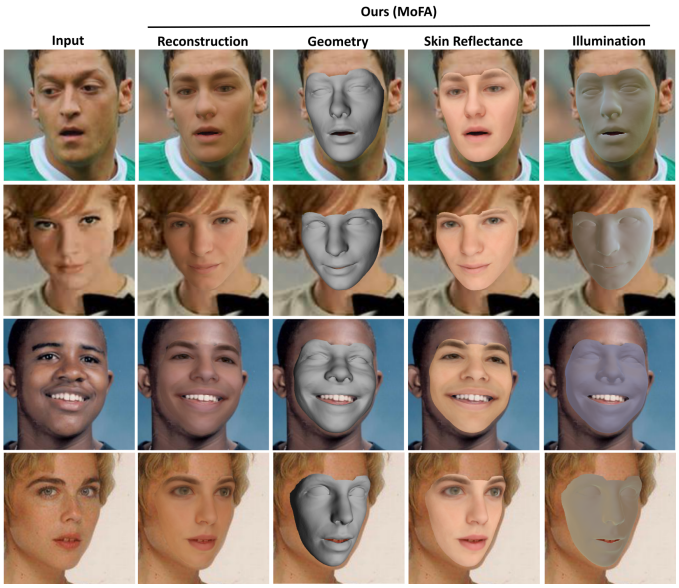

Fig. 4. Our approach enables the regression of high quality pose, shape, expression, skin reflectance and illumination from just a single monocular image (images from CelebA [64]).

Fig. 5. Sample images of our real world training corpus.

sampling fewer vertices only marginally reduces the quality of the results, while enabling us to train the networks much faster (time taken to train the network with 500, 2000, 5000 and all the vertex samples are 4, 5.2, 7.7 and 23.7 hours, respectively, using a GeForce TitanX graphics card). Qualitative results are shown in Fig. 3, where it can be seen that the resulting rendered images have similar visual quality.

8 Results of MoFA

In this section we demonstrate unsupervised learning of our model-based autoencoder in-the-wild, and we show that a surrogate loss during training improves accuracy. We test encoders based on AlexNet [69] and VGG-Face [70], where we modified the last fully connected layer to output our 257 model parameters. The reported results have been obtained using AlexNet [69] as encoder. Note that we do not employ the surrogate loss and use all the vertices of the mesh (i.e., no stochastic sampling) unless stated otherwise. After training, the encoder regresses pose, shape, expression, skin reflectance and illumination at once from a single image, see Fig. 4. For training we use an image corpus (see Fig. 5), which is a combination of four datasets: CelebA [64], LFW [68], Facewarehouse [71], and 300-VW [65], [66], [67]. The corpus is automatically annotated using facial landmark detection (see Sec. 6) and cropped to a bounding box using Haar Cascade Face Detection [72]. We prune frames with bad detections. The crops are scaled to a resolution of 240×240 pixels. In total, we collect 147k images, which we randomize and split into 142k for training and 5k for evaluation. We train our network using the *Caffe* [73] deep learning framework. For efficiency, we implement our model-based decoder and the robust photometric loss in a single CUDA [74] layer. We train our networks using *AdaDelta* and perform 200k batch iterations (batch size of 5). The base learning rate is 0.1 for all parameters, except for the Z-translation that was set to 0.0005. At test time, regressing all parameters using a TitanX Pascal graphics card is fast and takes only 4ms (AlexNet) or 14ms (VGG-Face). Training takes 13 hours (AlexNet) or 20 hours (VGG-Face). The encoder is initialized based on the provided pre-trained weights. All weights in the last fully connected layer are initialized to zero. This guarantees that the initial prediction is the average face placed in the middle of the screen and lit by ambient light, which is a good initialization. Note, the ambient coefficients of our renderer have an offset of 0.7 to guarantee that the scene is initially lit. Next, we compare to state-of-the-art optimization- and learning-based monocular reconstruction approaches, and evaluate all components of our approach.

Comparison to Richardson et al. [12], [13]: We compare our approach to the CNN-based iterative regressor of Richardson et al. [12], [13]. Our results are compared qualitatively (Fig. 6) and quantitatively (Fig. 15) to their coarse regression network. Note, the refinement layer of [13] is orthogonal to our approach. Unlike [12], [13], our network is trained completely unsupervised on real images, while they use a synthetic training corpus that lacks realistic features. In contrast to [12], [13], we also regress colored skin reflectance and illumination, which is critical for many applications, e.g., relighting. Note, the grayscale reflectance of [12], [13] is not regressed, but obtained via optimization.

Comparison to Tran et al. [14]: We compare qualitatively (Fig. 7) to the CNN-based identity regression approach of Tran et al. [14]. Our reconstructions are of visually similar quality; however, we additionally obtain high quality estimates of the facial expression and illumination. We also performed a face verification test on LFW. Our approach obtains an accuracy of 77%, which is higher than the monocular 3DMM baseline [75] (75%). Tran et al. [14] report an accuracy of 92%. Our approach is not designed for this scenario, since it is trained unsupervised on in-the-wild images. Tran et al. [14] require more supervision (photo collection) to train their network.

Comparison to Thies et al. [4]: We compare our approach qualitatively (Fig. 8) and quantitatively (Fig. 15) to the state-of-the-art optimization-based monocular reconstruction approach of Thies et al. [4]. Our approach obtains similar or even higher quality, while being orders of magnitude faster (4ms vs. $\approx$ 500ms). Note, while [4] tracks at real-time after identity estimation, it requires half a second to fit all parameters starting from the average model. While our approach only requires face detection at test time, Thies et al. [4] require detected landmarks.

Comparison to Garrido et al. [8]: We compare to our own implementation (no detail refinement and shape correctives, photometric + landmark + regularization terms, 50 Gauss-Newton steps) of the high quality off-line monocular reconstruction approach of [8], which requires landmarks as input. Our approach obtains comparable quality, while requiring no landmarks, see Fig. 9 and Fig. 15. Without sparse constraints as input, optimization-based approaches often get stuck in a local minimum.

We also compare to the monocular CNN-based approach

Fig. 6. Comparison to Richardson et al. [12], [13] on 300-VW [65], [66], [67] (left) and LFW [68] (right). Our approach obtains higher reconstruction quality and provides estimates of colored reflectance and illumination. Note, in [12], [13] the grayscale reflectance is not regressed but obtained via optimization. We on the other hand regress all parameters (including reflectance) at once.

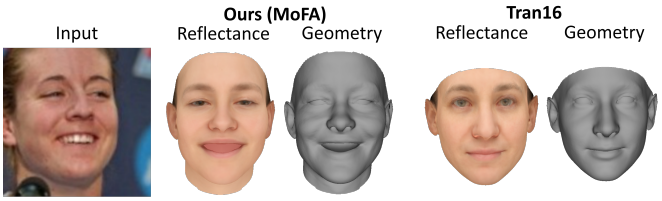

Fig. 7. Comparison to Tran et al. [14] on LFW [68]. Our approach obtains visually similar quality. Here, we show the full face model, but training only uses the frontal part (cf. Fig 1, right).

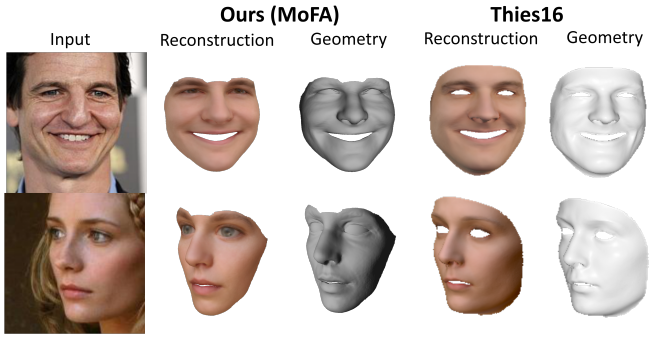

Fig. 8. Comparison to the monocular reconstruction approach of [4] on CelebA [64]. Our approach obtains similar or higher quality, while being orders of magnitude faster (4ms vs. ≈ 500ms).

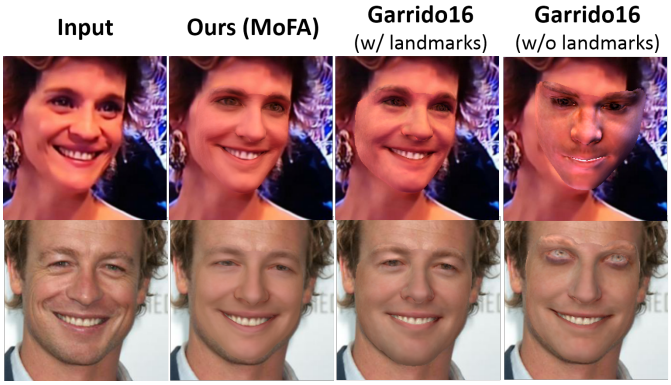

Fig. 9. We compare to our implementation of the high quality off-line monocular reconstruction approach of [8]. We obtain similar quality without requiring landmarks as input. Without landmarks, [8] often gets stuck in a local minimum.

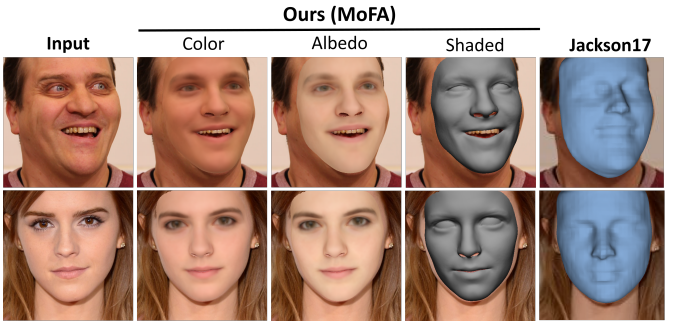

Fig. 10. Comparison to Jackson et al. [45]. Our approach obtains higher quality reconstructions while also estimating the reflectance and incident scene illumination.

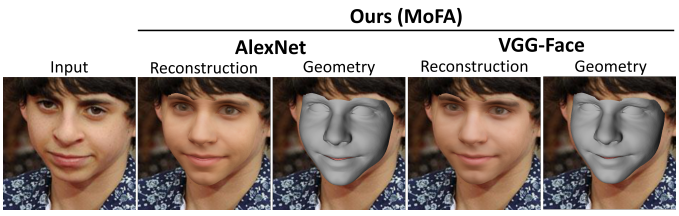

Fig. 11. We evaluate different encoders in combination with our model-based decoder. Overall, VGG-Face [70] leads to slightly better results than AlexNet [69], though the results are comparable.

of Jackson et al. [45] (Fig. 10). Our approach obtains qualitatively better alignments and higher quality results.

Evaluation of Different Encoders: We evaluate the impact of different encoders. VGG-Face [70] leads to slightly better results than AlexNet [69], see Fig. 11. On average, VGG-Face [70] has a slightly lower landmark (4.9 pixels vs. 5.3 pixels) and photometric error (0.073 vs. 0.075, color distance in RGB space, each channel in $[0, 1]$), see Fig. 12.

Quantitative Evaluation of Unsupervised Training: Unsupervised training decreases the dense photometric and landmark error (on a validation set of 5k real images), even when landmark alignment is not part of the loss function, see Fig. 12. The landmark error is computed based on 46 detected landmarks [63]. Training with our surrogate loss improves landmark alignment (AlexNet: 3.7 pixels vs. 5.3 pixels, VGG-Face: 3.4 pixels vs. 4.9) and leads to a similar photometric error (AlexNet: 0.078 vs. 0.075, VGG-Face: 0.078 vs. 0.073, color distance in RGB space, each channel in $[0, 1]$). We also evaluate the influence of our landmark-based surrogate loss qualitatively, see Fig. 13. Training with landmarks helps to improve robustness to occlusions and the quality of

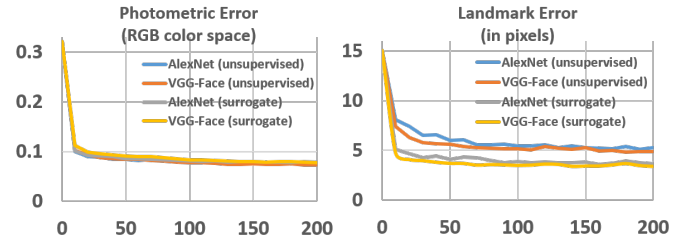

Fig. 12. Quantitative evaluation of MoFA on real data: Both landmark and photometric error are decreased during unsupervised training, even though landmark alignment is not part of the loss function.

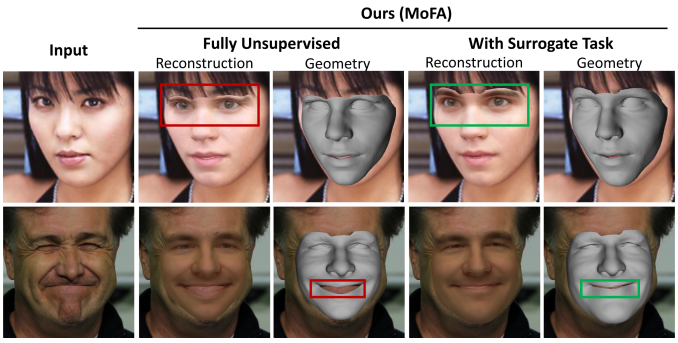

Fig. 13. We evaluate the influence of the proposed surrogate task. The surrogate task leads to improved reconstruction quality and increases robustness to occlusions and strong expressions.

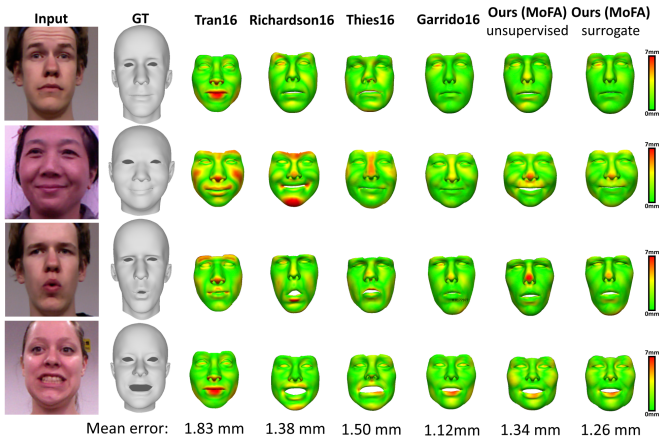

Fig. 15. Quantitative evaluation on Facewarehouse [71]: We obtain a low error that is comparable to optimization-based approaches. For this test, we trained our network using the intrinsics of the Kinect.

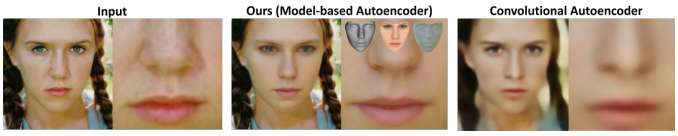

Fig. 16. Our model-based autoencoder gives results of higher quality than convolutional autoencoders. In addition, it provides access to dense geometry, reflectance, and illumination.

the predicted expressions. Note that both scenarios do not require landmarks at test time.

Quantitative Evaluation: We perform a ground truth evaluation based on 5k rendered images with known parameters. Our model-based autoencoder (AlexNet, unsupervised) is trained on a corpus of 100k synthetic images with background augmentation (cf. Fig. 14). We measure the geometric error as the point-to-point 3D distance (including the estimated rotation, we compensate for translation and isotropic scale) between the estimate and the ground truth mesh. This error drops from 21.6mm to 4.5mm. The photometric error in RGB space also decreases (0.33 to 0.05) and so does the landmark error (31.6 pixels to 3.9 pixels). Overall, we obtain good fits. We also performed a quantitative comparison for 9 identities (180 images) on Facewarehouse, see Table 1 and Fig. 15. Our approach obtains low errors and on par with optimization-based techniques in terms of mean closest point distance, but it is much faster (4ms vs. a few minutes) and requires no landmarks at test time. This error metric does not penalize misalignments in the tangent plane (surface sliding). To also quantitatively evaluate the reconstructions in terms of surface drift, we precompute a dense correspondence map between the employed test set and our mesh topology using a non-rigid registration approach. The correspondences are computed based on two almost neutral meshes with a slightly open mouth (to not erroneously bring the upper lip of one topology into correspondence with the lower lip of the other mesh). Based on this fixed set of correspondences, we performed an additional evaluation of the surface-to-surface error (including surface sliding) on the same test set, see Table. 2. Our results are comparable to the very recent coarse-level results of Tewari et al. [76] and Kim et al. [77]. Our refined results, see Sec. 9, outperform these two other state-of-the-art learning-based techniques on the coarse level. The results of Garrido et al. [8] are still slightly better, but our approach runs orders of magnitude faster.

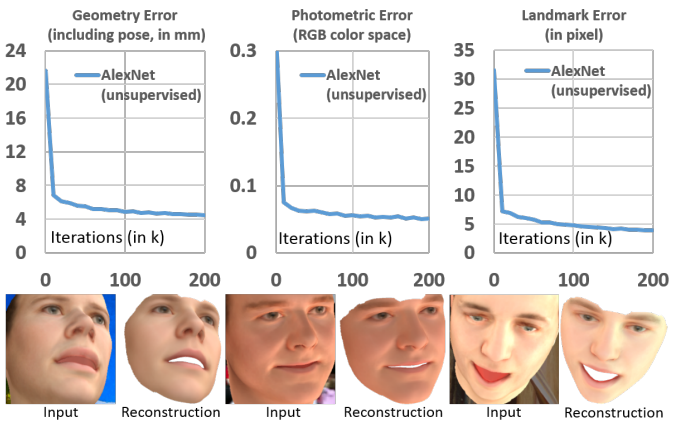

Fig. 14. Quantitative evaluation of MoFA on synthetic ground truth data: Training decreases the geometric, photometric and landmark error.

TABLE 1
Quantitative evaluation on real data. Average closest point distance to the ground truth for different approaches.

	Geometry	Photometric	Landmark
Ours (MoFA w/o surrog.)	1.9mm	0.065	5.0px
Ours (MoFA w/ surrog.)	1.7mm	0.068	3.2px
Garrido et al. [8]	1.4mm	0.052	2.6px

Comparison to Autoencoders and Learned Decoders: We compare our model-based with a convolutional autoencoder, as shown in Fig. 16. The autoencoder uses four 3×3 convolution layers (64, 96, 128, 256 channels),

TABLE 2
Geometric error on 180 meshes of the FaceWarehouse [71] dataset. Surface-to-surface error (including sliding) based on a precomputed dense correspondence map between the employed test set and our mesh topology.

	Ours		Others			
	MoFA (surrogate)	Opt	Tewari et al. [76] (Fine)	Tewari et al. [76] (Coarse)	Kim et al. [77]	Garrido et al. [8] (Coarse)
Mean	2.19 mm	1.87 mm	1.84 mm	2.03 mm	2.11 mm	1.59 mm
SD	0.54 mm	0.42 mm	0.38 mm	0.52 mm	0.46 mm	0.30 mm
Time	4 ms	110 ms	4 ms	4 ms	4 ms	>1 min

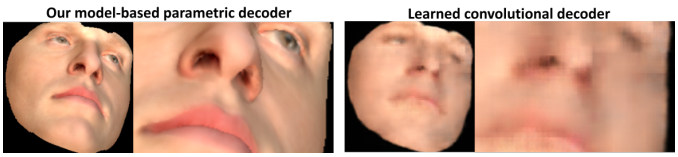

Fig. 17. Our model-based decoder provides higher fidelity than a learned convolutional decoder in terms of image quality.

a fully connected layer (257 outputs, same as number of model parameters), and four 4×4 deconvolution layers (128, 96, 64, 3 channels). Our model-based approach obtains sharper reconstruction results and provides fine granular semantic parameters, allowing access to dense geometry, reflectance and illumination, see Fig. 16 (middle). Explicit disentanglement [19], [20] of a convolutional autoencoder requires labeled ground truth data. We also compare to image formation based on a trained decoder. To this end, we train the decoder (similar parameters as above) based on synthetic imagery generated by our model to learn the parameter-to-image mapping. Our model-based decoder obtains renderings of higher fidelity compared to the learned decoder, see Fig. 17.

9 Optimization-based Refinement

Similar to other data-driven techniques, neural networks have a limited capacity and might not generalize well to inputs outside the span of the employed training corpus. Finding the right balance between under- and over-fitting is a highly challenging problem on its own. Under-fitting leads to a loss of reconstruction quality and over-smoothed results, while over-fitting leads to bad generalization to unseen images. On the other hand, standard optimization-based approaches (without the guidance of discriminative detected landmarks) often get stuck in a bad local minimum, which leads to low reconstruction quality, as shown in Fig. 18. In this section, we demonstrate that the combination of a coarse discriminative estimate with an optimization-based analysis-by-synthesis approach and a shading-based surface refinement step can significantly improve the quality of the obtained reconstructions. First, we describe a local minimization of the energy E_{loss} in Eq. (7) based on the Gauss-Newton method, which leads to an improved reconstruction that remains within the span of the employed model (Sec. 4). Moreover, in order to explain fine-scale details on a wrinkle-level, we (locally) optimize a modified energy function over per-vertex displacements. These displacements are able to represent faces that are outside the (restricted) model-subspace.

9.1 Analysis-by-synthesis Optimization

Since our trained network has limited capacity, it has to trade-off the quality of individual reconstructions in order to work on a diverse range of images. We show that running an analysis-by-synthesis optimizer on the output of MoFA can significantly improve the results. Our optimizer minimizes the energy E_{loss} in (7) as used to train our network. Starting from the MoFA output as initialization, we run Gauss-Newton optimization. Since the Gauss-Newton method requires the energy to be represented as a sum of squares, we implement our photometric term in (8) as explained in (11). Additionally, we have implemented our optimizer in a data-parallel fashion on the GPU, as explained next.

Data-parallel GPU Implementation: Our face reconstruction energy is in a general non-linear least-squares form:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\, \mathbf{E}_{\text{loss}}(\mathbf{x}), \text{ where} \tag{15}$$

$$\mathbf{E}_{\text{loss}}(\mathbf{x}) = \sum_i \left(\mathbf{F}_i(\mathbf{x})\right)^2 \ . \tag{16}$$

Thus, we find the (local) optimum $\mathbf{x}^*$ using the Gauss-Newton algorithm. In each iteration step, we linearize the problem based on Taylor expansion and solve the resulting normal equations:

$$\mathbf{J}^T\mathbf{J}\delta = \mathbf{J}^T\mathbf{F} \ . \tag{17}$$

$\mathbf{J}$ and $\mathbf{F}$ are the same as defined in Sec. 6 and are computed in the same manner. δ is the optimal update of the parameters. We use a data-parallel implementation [78] of dense matrix-matrix and matrix-vector multiplication to compute the system matrix $\mathbf{J}^T\mathbf{J}$ and right hand side $\mathbf{J}^T\mathbf{F}$ of Eq. (17), respectively. Afterwards, we copy the resulting small linear system to the CPU and solve the system via Cholesky factorization to compute the optimal update δ. We iterate this process for 5 Gauss-Newton steps. The runtime to obtain our final reconstructions (network inference + optimizer) is 110 ms for one image, orders of magnitude faster compared to a few mins per image for [8].

Results: The combination of our discriminative approach with this analysis-by-synthesis fitting strategy (referred to as "Opt") leads to higher quality results, as shown in Figs. 18, 20, 21 and Table 2. Purely optimization-based approaches are highly sensitive to the initialization and often fall into local minima in the absence of the landmark alignment term. The parameter regression result of MoFA provides a good initialization that can reliably be refined by the local optimizer such that good reconstructions can be obtained even without landmarks, cf. Fig. 18. Note, all results obtained with the optimizer other than Fig. 18 use our MoFA network with the surrogate loss and the landmark alignment term for higher quality results. The weights used for the optimizer are $w_{\text{photo}} = 0.44$, $w_{\text{reg}} = 0.01$, $w_{\boldsymbol{\beta}} = 0.11$,

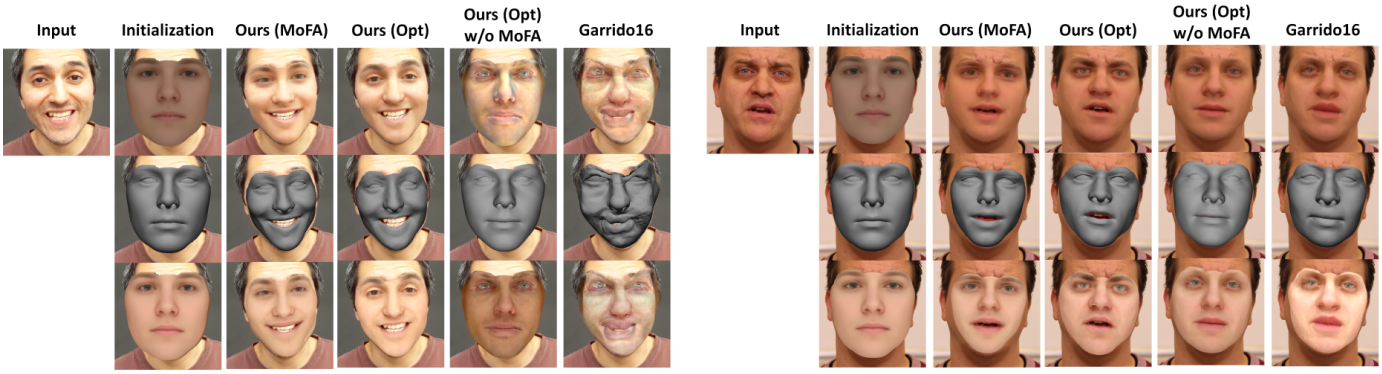

Fig. 18. Qualitative comparison between MoFA and MoFA with analysis-by-synthesis optimization (Opt) without the landmark term. Opt improves the MoFA estimates while Garrido et al. [8], which starts from a neutral initialization (second column), often ends up in local minima in the absence of landmarks. Opt, when starting from a neutral initialization also fails to estimate plausible reconstructions.

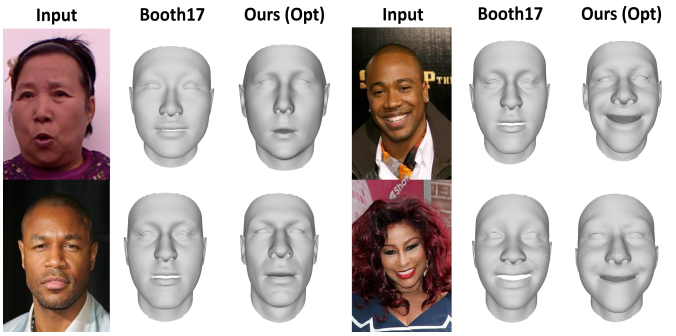

Fig. 19. Comparison between Opt and the approach by Booth et al. [23], which learns an in-the-wild texture model from images to improve the reconstruction of geometry. We obtain similar or better quality results only using the reflectance model of [5].

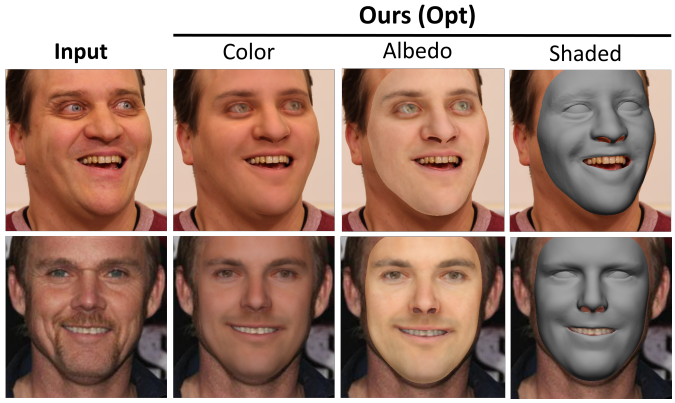

Fig. 20. MoFA with analysis-by-synthesis optimization allows for high-quality geometry and appearance reconstructions.

$w_{\boldsymbol{\delta}} = 0.01$. We have found that using only 5 Gauss-Newton iterations leads to significant improvements over MoFA. In Table 2 we provide quantitative results comparing various methods, where it can be seen that Opt is able to reduce the MoFA reconstruction error further. Although we still are not able to outperform the results achieved by Garrido et al. [8], we point out that [8] runs for 50 iterations (with the landmark alignment term, starting from a neutral face), thus requiring significantly more time. We compare our Opt approach with the optimization-based approach of Booth et al. [23] (Fig. 19) where our approach obtains comparable or

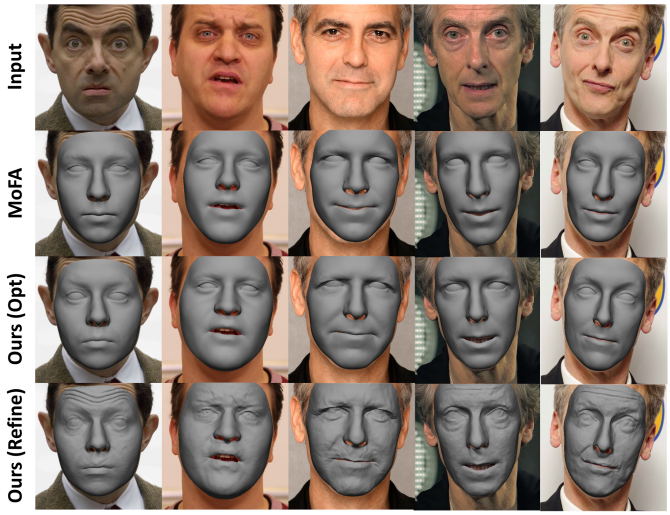

Fig. 21. Qualitative comparison of MoFA with and without refinement. While MoFA provides good reconstructions, the analysis-by-synthesis optimization (Opt) significantly improves reconstruction quality. Shading-based-refinement (Refine) further adds high-frequency details on the surface, leading to high-fidelity reconstructions.

better results. Our method also provides individual estimates for the reflectance and illumination channels while [23] only estimates the combined texture.

9.2 Shading-based Surface Refinement

While we obtain high quality reconstructions on in-the-wild monocular images, our results are limited to the subspace spanned by the underlying low-dimensional affine model (Sec. 4). This limits the ability of our method to capture fine-scale wrinkle-level details. Hence, we further refine the output of Opt by allowing the mesh to go outside of this restricted low-dimensional deformation space.

Recovering fine-scale surface structure is a long standing and well researched problem in computer vision. Refinement techniques for general surfaces [79], [80], [81], [82], [83] are normally based on multi-view imagery. In the context of facial detail estimation a variety of techniques exist. Data-driven approaches [13], [44], [84] learn a mapping from the input image to the fine-scale geometric structure. While these approaches are in general fast, the recovered detail does not necessarily match the input. Some approaches produce

Fig. 22. Comparison between our method with shading-based surface refinement (Refine), Richardson et al. [13] and Sela et al. [15]. In [13], only the refined depth maps are estimated while in [15], an expensive non-rigid template alignment step is needed to compute the final reconstructions. We obtain similar or higher quality reconstructions by directly optimizing for the surface details on the mesh.

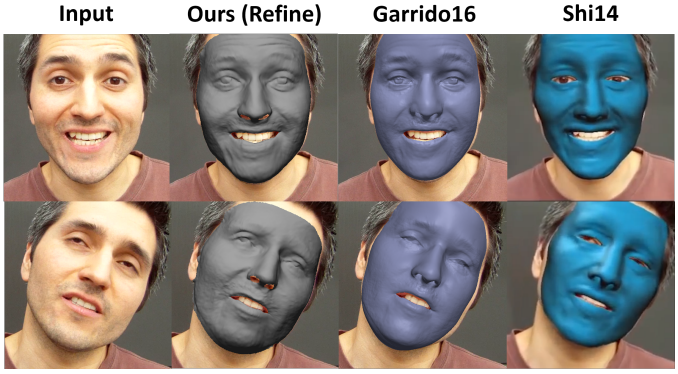

Fig. 23. Comparison between our method with shading-based surface refinement (Refine), Garrido et al. [8] and Shi et al. [88]. We obtain similar results, while being significantly faster.

details directly based on intensity variation [15], [85], [86]. While the obtained results look visually plausible, they are not physically accurate. Optimization-based refinement techniques [8], [87], [88] try to invert physical image formation models. Although the recovered detail in general matches the input, these approaches are computationally quite expensive, normally requiring multiple minutes to process a single frame. We leverage the data-parallel power of modern graphics cards to accelerate optimization-based mesh refinement.

Consider the vertex positions of the low-dimensional coarse reconstruction as $\mathbf{V}^{\mathrm{C}} = \{\mathbf{v}_i^{\mathrm{C}} \in \mathbb{R}^3 | 1 \leq i \leq N\}$. We model out-of-subspace deformations using per-vertex displacements $\mathbf{D} = \{\mathbf{d}_i \in \mathbb{R}^3 | 1 \leq i \leq N\}$, such that the final vertex positions $\mathbf{V}^{\mathrm{F}} = \{\mathbf{v}_i^{\mathrm{F}} = \mathbf{v}_i^{\mathrm{C}} + \mathbf{d}_i | 1 \leq i \leq N\}$ align well to the input image $\mathcal{I}$. The optimal displacements are determined as

$$\mathbf{D}^* = \underset{\mathbf{D}}{\operatorname{argmin}}\, E_{\mathrm{ref}}(\mathbf{D}) \ , \tag{18}$$

where

$$E_{\mathrm{ref}}(\mathbf{D}) = \underbrace{E_{\mathrm{photo}}(\mathbf{D}) + w_{\mathrm{grad}} E_{\mathrm{grad}}(\mathbf{D})}_{\text{data term}} + \underbrace{w_{\mathrm{reg}} E_{\mathrm{reg}}(\mathbf{D})}_{\text{regularizer}} . \tag{19}$$

Dense Photometric Alignment: Similar to (8), we use a dense photometric alignment term

$$E_{\mathrm{photo}}(\mathbf{D}) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left\| \mathcal{I}(\mathbf{u}_i(\mathbf{D})) - \mathbf{c}_i(\mathbf{D}) \right\|_2 \ , \tag{20}$$

where $\mathcal{V}$ is the set of visible vertices (we approximate vertex visibility by the set of front-facing vertices), and $\mathbf{u}_i(\mathbf{D})$ and $\mathbf{c}_i(\mathbf{D})$ are the screen space position and color of vertex i, respectively. They are computed analogously to Eq. (5). We implement the photometric term in an iteratively reweighted fashion, as in Eq. (11).

Gradient Alignment Term: We also consider high-frequency shading details, similarly as proposed in [81]. More precisely, we introduce a gradient alignment term that tries to match the color gradients between the input and a synthetic rendering of the model, as follows:

$$E_{\mathrm{grad}}(\mathbf{D}) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \left\| (\mathbf{c}_i(\mathbf{D}) - \mathbf{c}_j(\mathbf{D})) - (\mathcal{I}(\mathbf{u}_i(\mathbf{D})) - \mathcal{I}(\mathbf{u}_j(\mathbf{D}))) \right\|_2^2 , \tag{21}$$

where $\mathcal{N}_i$ is the one-ring neighborhood of vertex i. Finite differences efficiently approximate image gradients based on mesh gradients.

Regularization Term: Additionally, we use a Laplacian regularizer on the displacements, as follows:

$$E_{\mathrm{reg}}(\mathcal{D}) = \frac{1}{N} \sum_{i \in \mathbf{V}} \left\| \sum_{j \in \mathcal{N}_i} (\mathbf{d}_i - \mathbf{d}_j) \right\|_2^2 . \tag{22}$$

Note that Eq. (22) enforces smoothness of the reconstructions and stability of the optimizer.

Mesh Topology: We use the topology of [5] for both our MoFA and for the analysis-by-synthesis optimization as described in Sec. 9.1. In order to ensure numerical stability for shading-based refinement, one has to take care of near-degenerate mesh faces in the topology of [5]. To this end, we remeshed the neutral face $\mathbf{V}_N^{T1}$ from the topology $T1$ of [5] to a face $\mathbf{V}_N^{T2}$ represented by a more uniform topology $T2$. The transformation of vertices of topology $T1$ to vertices of topology $T2$ can be represented by the linear map $L\colon \mathbf{V}_N^{T1} \rightarrow \mathbf{V}_N^{T2}$. After analysis-by-synthesis optimization (Sec. 9.1), we transfer the results from topology $T1$ to our topology $T2$ using L, and then optimize over per-vertex displacements using the topology $T2$.

Optimization: Since the number of unknowns is much larger than for the problem in Sec. 9.1, we use gradient descent to optimize for the displacements. Similarly as before, we approximate $|\mathcal{V}|$ in the individual energy terms using N. The weights used in the energy term are $w_{\mathrm{grad}} = 1.0$, $w_{\mathrm{reg}} = 133.3$. We use 250 iterations with a step-size of 0.008, which we have found sufficient to achieve convergence.

Data-parallel GPU Implementation: We have also implemented the per-vertex displacement optimization in a data-parallel fashion on the GPU. Since the Jacobian matrix here is much bigger and sparse, we do not use the approach from Sec. 9.1. Instead, to compute the gradients, we launch one dedicated thread for each element of $\mathbf{F}$, where thread i computes $\frac{\mathrm{d}(\mathbf{F}_i)^2}{\mathrm{d}\mathbf{D}}$. The gradients for each variable coming from different threads are integrated using global memory

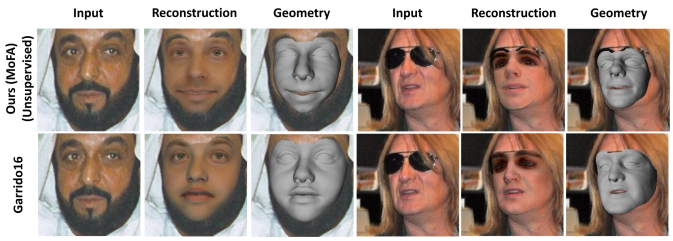

Fig. 24. Limitations: Facial hair and occlusions are challenging to handle.

atomics. Using this optimized parallel implementation results in a processing time of 450 ms for one image. Thus, the overall time to obtain a high-quality reconstruction including fine-scale details is 450 + 110 = 560 ms.

Results: We initialize our refinement approach with the results from Opt. We show that our approach with refinement ("Refine") recovers high-frequency geometry details from images (Fig. 21). We compare to two high-quality face reconstruction approaches [13], [15], as shown in Fig. 22. We obtain details directly on the mesh in contrast to Richardson et al. [13] that obtain only refined depth maps. Sela et al. [15] reconstruct details on the mesh but at the cost of an expensive non-rigid template alignment step. Our approach obtains similar or higher quality while directly optimizing for the details on the mesh topology. We additionally estimate the reflectance and illumination channels. Our approach also obtains similar results compared to [8] and [88], see Fig. 23. However, both [8] and [88] are orders of magnitude slower.

10 Limitations

We have demonstrated compelling monocular reconstructions using a novel model-based autoencoder that is trained in an unsupervised manner. Similar to other regression approaches, implausible reconstructions are possible with MoFA when the regressed parameters are outside the span of the training data. This can be alleviated by enlarging the training corpus, which is easy to achieve in our unsupervised setting. Since we employ a face model, MoFA reconstructions are limited to the modeled subspace. Similar to optimization-based approaches, strong occlusions, e.g., by facial hair or external objects, cause our approach to fail, see Fig. 24. Even with the refinement strategies, our approach can fail in such cases. Unsupervised occlusion-aware training is an interesting open research problem. Similar to related approaches, strong head rotations are challenging. Since we do not model the background, our reconstructions can slightly shrink. Shrinking is discussed and addressed in [89].

11 Conclusion

We have presented a deep convolutional model-based face autoencoder that can be trained in an unsupervised manner and learns meaningful semantic parameters. Semantic meaning in the code vector is enforced by a parametric model that encodes variation along the pose, shape, expression, skin reflectance and illumination dimensions. Our model-based decoder is fully differentiable and allows end-to-end learning of our network. We have additionally shown a stochastic vertex sampling strategy in the loss function for faster training, and analysis-by-synthesis optimization and shape-from-shading refinement methods for high-fidelity reconstruction. We believe that the fundamental technical concepts of our approach go far beyond the context of monocular face reconstruction and will inspire future work.

Acknowledgments

We thank True-VisionSolutions Pty Ltd for kindly providing the 2D face tracker, Anh Tuan Tran and Aaron S. Jackson for publishing their source code, and Justus Thies, Elad Richardson, Matan Sela and Stefanos Zafeiriou for the comparisons. This work was supported by the ERC Starting Grant CapReal (335545), the Max Planck Center for Visual Computing and Communications (MPC-VCC), and by Technicolor.

References

[1] A. Tewari, M. Zollöfer, H. Kim, P. Garrido, F. Bernard, P. Perez, and T. Christian, "MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction," in *ICCV*, 2017.
[2] I. Kemelmacher-Shlizerman and S. M. Seitz, "Face reconstruction in the wild," in *ICCV*, 2011.
[3] J. Roth, Y. Tong, and X. Liu, "Adaptive 3d face reconstruction from unconstrained photo collections," December 2016.
[4] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *CVPR*, 2016.
[5] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proc. SIGGRAPH*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.
[6] V. Blanz, C. Basso, T. Poggio, and T. Vetter, "Reanimating faces in images and video," in *Computer graphics forum*. Wiley Online Library, 2003, pp. 641–650.
[7] O. Fried, E. Shechtman, D. B. Goldman, and A. Finkelstein, "Perspective-aware manipulation of portrait photos," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016.
[8] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt, "Reconstruction of personalized 3D face rigs from monocular video," *ACM Transactions on Graphics*, vol. 35, no. 3, pp. 28:1–15, Jun. 2016.
[9] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz, "Total moving face reconstruction," in *ECCV*, 2014.
[10] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "What makes tom hanks look like tom hanks," in *ICCV*, 2015.
[11] I. Kemelmacher-Shlizerman, A. Sankar, E. Shechtman, and S. M. Seitz, "Being john malkovich," in *ECCV*, 2010.
[12] E. Richardson, M. Sela, and R. Kimmel, "3D face reconstruction by learning from synthetic data," in *3DV*, 2016.
[13] E. Richardson, M. Sela, R. Or-El, and R. Kimmel, "Learning detailed face reconstruction from a single image," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[14] A. Tuan Tran, T. Hassner, I. Masi, and G. Medioni, "Regressing robust and discriminative 3d morphable models with a very deep neural network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[15] M. Sela, E. Richardson, and R. Kimmel, "Unrestricted facial geometry reconstruction using image-to-image translation," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
[16] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[17] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*, 2011.
[18] F. Zhao, J. Feng, J. Zhao, W. Yang, and S. Yan, "Robust lstm-autoencoders for face de-occlusion in the wild," 2016, arXiv:1612.08534.
[19] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum, "Deep convolutional inverse graphics network," in *NIPS*, 2015.
[20] E. Grant, P. Kohli, and M. van Gerven, "Deep disentangled representations for volumetric reconstruction," in *ECCVW*, 2016.
[21] F. Liu, D. Zeng, J. Li, and Q. Zhao, "3D face reconstruction via cascaded regression in shape space," 2016, arXiv:1509.06161.

[22] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[23] J. Booth, E. Antonakos, S. Ploumpis, G. Trigeorgis, Y. Panagakis, and S. Zafeiriou, "3d face morphable models "in-the-wild"," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[24] S. Romdhani and T. Vetter, "Estimating 3D Shape and Texture Using Pixel Intensity, Edges, Specular Highlights, Texture Constraints and a Prior," *CVPR*, 2005.

[25] P. Huber, P. Kopp, M. Rätsch, W. Christmas, and J. Kittler, "3D face tracking and texture fusion in the wild," May 2016, arXiv:1605.06764.

[26] N. Wang, X. Gao, D. Tao, and X. Li, "Facial Feature Point Detection: A Comprehensive Survey," Oct. 2014, arXiv:1410.1037.

[27] X. Jin and X. Tan, "Face alignment in-the-wild: A survey," Aug. 2016, arXiv:1608.04188.

[28] Y. Sun, X. Wang, and X. Tang, "Deep Convolutional Network Cascade for Facial Point Detection." 2013.

[29] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive Facial Landmark Localization with Coarse-to-Fine Convolutional Network Cascade," in *CVPRW*, 2013.

[30] Y. Wu and Q. Ji, "Discriminative deep face shape model for facial point detection," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 37–53, 2015.

[31] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas, "A recurrent encoder-decoder network for sequential face alignment," in *ECCV*, 2016.

[32] A. Bulat and G. Tzimiropoulos, "Two-stage convolutional part heatmap regression for the 1st 3D face alignment in the wild (3DFAW) challenge," in *ECCVW*, 2016.

[33] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," 2016, arXiv:1611.00851.

[34] Y. Tang, R. Salakhutdinov, and G. E. Hinton, "Deep lambertian networks," 2012.

[35] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep Learning Identity-Preserving Face Space." 2013.

[36] ——, "Multi-view perceptron: a deep model for learning face identity and view representations," 2014.

[37] M. Li, W. Zuo, and D. Zhang, "Convolutional network for attribute-driven and identity-preserving human face generation," Aug. 2016, arXiv:1608.06434.

[38] R. A. Güler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos, "Densereg: Fully convolutional dense shape regression in-the-wild," Dec. 2016, arXiv:1612.01202.

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

[40] K. G. Quach, C. N. Duong, K. Luu, and T. D. Bui, "Robust deep appearance models," Jul. 2016, arXiv:1607.00659.

[41] S. Laine, T. Karras, T. Aila, A. Herva, and J. Lehtinen, "Facial performance capture with deep neural networks," 2016, arXiv:1609.06536.

[42] S. Zhu, S. Liu, C. C. Loy, and X. Tang, "Deep cascaded bi-network for face hallucination," Jul. 2016, arXiv:1607.05046.

[43] S. Saito, L. Wei, L. Hu, K. Nagano, and H. Li, "Photorealistic facial texture inference using deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 2326–2335.

[44] L. Huynh, W. Chen, S. Saito, J. Xing, K. Nagano, A. Jones, P. Debevec, and H. Li, "Mesoscopic facial geometry inference using deep neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[45] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, "Large pose 3d face reconstruction from a single image via direct volumetric cnn regression," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[46] G. Trigeorgis, P. Snape, I. Kokkinos, and S. Zafeiriou, "Face normals "in-the-wild" using fully convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[47] P. Dou, S. K. Shah, and I. A. Kakadiaris, "End-to-end 3d face reconstruction with deep neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[48] V. Nair, J. Susskind, and G. E. Hinton, "Analysis-by-synthesis by learning to invert generative black boxes," in *International Conference on Artificial Neural Networks (ICANN)*, 2008.

[49] A. Zhmoginov and M. Sandler, "Inverting face embeddings with convolutional neural networks," Jun. 2016, arXiv:1606.04189.

[50] M. Kan, S. Shan, H. Chang, and X. Chen, "Stacked Progressive Auto-Encoders (SPAE) for Face Recognition Across Poses." 2014.

[51] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-Fine Auto-Encoder Networks (CFAN) for Real-Time Face Alignment." 2014.

[52] S. Gao, Y. Zhang, K. Jia, J. Lu, and Y. Zhang, "Single Sample Face Recognition via Learning Deep Supervised Autoencoders," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2108–2118, 2015.

[53] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2049–2058, 2015.

[54] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *NIPS*, 2015.

[55] A. Handa, M. Bloesch, V. Patraucean, S. Stent, J. McCormac, and A. J. Davison, "gvnn: Neural network library for geometric computer vision," in *ECCV*, 2016.

[56] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," Dec. 2016, arXiv:1612.00814.

[57] A. Bas, P. Huber, W. A. Smith, M. Awais, and J. Kittler, "3d morphable models as spatial transformer networks," Oct 2017.

[58] C. Bhagavatula, C. Zhu, K. Luu, and M. Savvides, "Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[59] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec, "The Digital Emily Project: photoreal facial modeling and animation," in *ACM SIGGRAPH Courses*. ACM, 2009, pp. 12:1–12:15.

[60] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "Facewarehouse: A 3D facial expression database for visual computing," *IEEE TVCG*, vol. 20, no. 3, pp. 413–425, 2014.

[61] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM TOG*, vol. 23, no. 3, pp. 399–405, 2004.

[62] C. Müller, *Spherical harmonics*. Springer, 1966.

[63] J. M. Saragih, S. Lucey, and J. F. Cohn, "Deformable model fitting by regularized landmark mean-shift," vol. 91, no. 2, pp. 200–215, 2011.

[64] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[65] G. G. Chrysos, E. Antonakos, S. Zafeiriou, and P. Snape, "Offline deformable face tracking in arbitrary videos," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.

[66] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results," in *ICCVW*, December 2015.

[67] G. Tzimiropoulos, "Project-out cascaded regression with an application to face alignment," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[68] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[69] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012.

[70] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.

[71] 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*, vol. 20, no. 3, pp. 413–425, Mar. 2014.

[72] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[73] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[74] NVIDIA, *NVIDIA CUDA Programming Guide 2.0*, 2008.

[75] S. Romdhani and T. Vetter, "Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior," in *CVPR*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 986–993.

[76] A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez, and C. Theobalt, "Self-supervised multi-level face model learning

for monocular reconstruction at over 250 hz," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[77] H. Kim, M. Zollöfer, A. Tewari, J. Thies, C. Richardt, and C. Theobalt, "Inversefacenet: Deep single-shot inverse face rendering from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
[78] nVidia, *CUBLAS Library User Guide*, v5.0 ed., nVidia, Oct. 2012. [Online]. Available: http://docs.nvidia.com/cublas/index.html
[79] S. Li, S. Y. Siu, T. Fang, and L. Quan, "Efficient multi-view surface refinement with adaptive resolution control," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, 2016, pp. 349–364.
[80] A. Delaunoy and E. Prados, "Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility," *Int. J. Comput. Vision*, vol. 95, no. 2, pp. 100–123, Nov. 2011.
[81] C. Wu, B. Wilburn, Y. Matsushita, and C. Theobalt, "High-quality shape from multi-view stereo and shading under general illumination," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 969–976.
[82] H. H. Vu, P. Labatut, J. P. Pons, and R. Keriven, "High accuracy and visibility-consistent dense multiview stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 889–901, May 2012.
[83] R. Tylecek and R. Sara, "Refinement of surface mesh for accurate multiview reconstruction," *International Journal of Virtual Reality*, vol. 9, no. 1, pp. 45–54, 2010.
[84] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time high-fidelity facial performance capture," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 46:1–46:9, Jul. 2015.
[85] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-quality single-shot capture of facial geometry," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 40:1–40:9, Jul. 2010.
[86] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross, "High-quality passive facial performance capture using anchor frames," in *ACM SIGGRAPH 2011 Papers*, ser. SIGGRAPH '11. New York, NY, USA: ACM, 2011, pp. 75:1–75:10.
[87] P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt, "Reconstructing detailed dynamic face geometry from monocular video," in *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2013)*, vol. 32, no. 6, November 2013, pp. 158:1–158:10.
[88] F. Shi, H.-T. Wu, X. Tong, and J. Chai, "Automatic acquisition of high-fidelity facial performances using monocular videos," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 222:1–222:13, Nov. 2014.
[89] S. Schönborn, B. Egger, A. Forster, and T. Vetter, "Background modeling for generative image models," *Comput. Vis. Image Underst.*, vol. 136, no. C, pp. 117–127, Jul. 2015.

Ayush Tewari is a Ph.D. student in the 'Graphics, Vision & Video' group at the Max Planck Institute for Informatics in Saarbrücken, Germany. Before, he received his M.Sc. in Computer Science from Grenoble INP, and B.Tech. in Computer Science and Engineering from IIIT Hyderabad. His research interests are in computer vision, computer graphics, and machine learning, with a focus on 3D reconstruction problems. He currently works on monocular face reconstruction.

Michael Zollhöfer is a Visiting Assistant Professor at Stanford University. His stay at Stanford is funded by a postdoctoral fellowship of the Max Planck Center for Visual Computing and Communication (MPC-VCC), which he received for his work in the fields of computer vision, computer graphics, and machine learning. Before, Michael was a Postdoctoral Researcher in the 'Graphics, Vision & Video' group at the Max Planck Institute for Informatics in Saarbrücken, Germany. He received his PhD in 2014 from the University of Erlangen-Nuremberg for his work on real-time static and dynamic scene reconstruction. His research is focused on teaching computers to reconstruct and analyze our world at frame rate based on visual input.

Florian Bernard is a postdoctoral researcher at the Max-Planck-Institute for Informatics. In 2016 he received the Ph.D. degree for his work on multi-shape analysis from the University of Luxembourg. His research interests are computer vision, pattern recognition and machine learning, with a focus on optimization methods relevant for shape analysis. He has received the Abertay University Prize (best postgraduate student, 2011), the German Association for Medical Informatics (GMDS) award for his Master's thesis (2013), an AFR PhD Grant by the National Research Fund Luxembourg (2014–2016), the Siemens/Sicas SHAPE Award for the best conference paper (2015), and the BVM Award by for his Ph.D. thesis (2017).

Pablo Garrido is a Postdoctoral Researcher at Technicolor. His research lies between Computer Vision and Computer Graphics and mainly focuses on monocular facial performance capture and facial animation, as well as video segmentation and editing. Before starting his doctoral studies, he worked as a research assistant at Federico Santa Maria University (2008-2010), where he also got his master degree in 2008. His research interests include performance capture, video cutout and editing, applied machine learning, and optimization in dynamic environments.

Hyeongwoo Kim is affiliated with the Computer Graphics group at the Max Planck Institute for Informatics. His research interests are in computer vision, computer graphics, machine learning and pattern recognition.

Patrick Pérez Patrick Pérez is Distinguished Scientist and Fellow at Technicolor where he leads exploratory research on machine learning and vision. He is currently on the Editorial Board of the International Journal of Computer Vision. Before joining Technicolor, Patrick Pérez has been researcher at Inria (1993-2000, 2004-2009) and at Microsoft Research Cambridge (2000-2004). His research interests include audio/video description, search and analysis, as well as photo/video editing and computational imaging.

Christian Theobalt Christian Theobalt is a Professor of Computer Science and the head of the research group "Graphics, Vision, & Video" at the Max-Planck-Institute for Informatics, Saarbruecken, Germany. He is also a professor at Saarland University. His research lies on the boundary between Computer Vision and Computer Graphics. For instance, he works on 4D scene reconstruction, marker-less motion and performance capture, machine learning for graphics and vision, and new sensors for 3D acquisition. Christian received several awards, for instance the Otto Hahn Medal of the Max-Planck Society (2007), the EUROGRAPHICS Young Researcher Award (2009), the German Pattern Recognition Award (2012), an ERC Starting Grant (2013), and an ERC Consolidator Grant (2017). In 2015, he was elected one of Germany's top 40 innovators under 40 by the magazine Capital. He is a co-founder of theCaptury (www.thecaptury.com).