MobileFace: 3D Face Reconstruction with Efficient CNN Regression

Nikolai Chinaev[1], Alexander Chigorin[1], and Ivan Laptev[1,2]

[1] VisionLabs, Amsterdam, The Netherlands
{n.chinaev, a.chigorin}@visionlabs.ru
[2] Inria, WILLOW, Departement d'Informatique de l'Ecole Normale Superieure, PSL Research University, ENS/INRIA/CNRS UMR 8548, Paris, France
ivan.laptev@inria.fr

Abstract. Estimation of facial shapes plays a central role for face transfer and animation. Accurate 3D face reconstruction, however, often deploys iterative and costly methods preventing real-time applications. In this work we design a compact and fast CNN model enabling real-time face reconstruction on mobile devices. For this purpose, we first study more traditional but slow morphable face models and use them to automatically annotate a large set of images for CNN training. We then investigate a class of efficient MobileNet CNNs and adapt such models for the task of shape regression. Our evaluation on three datasets demonstrates significant improvements in the speed and the size of our model while maintaining state-of-the-art reconstruction accuracy.

Keywords: 3d face reconstruction · morphable model · CNN

1 Introduction

3D face reconstruction from monocular images is a long-standing goal in computer vision with applications in face recognition, film industry, animation and other areas. Earlier efforts date back to late nineties and introduce morphable face models [1]. Traditional methods address this task with optimization-based techniques and analysis-through-synthesis methods [2–6]. More recently, regression-based methods started to emerge [7–10]. In particular, the task has seen an increasing interest from the CNN community over the past few years [9–13]. However, the applicability of neural networks remains difficult due to the lack of large-scale training data. Possible solutions include the use of synthetic data [8, 12], incorporation of unsupervised training criteria [10], or combination of both [14]. Another option is to produce semi-synthetic data by applying an optimization-based algorithm with proven accuracy to a database of faces [9, 11, 13].

Optimization-based methods for morphable model fitting vary in many respects. Some design choices include image formation model, regularization and optimization strategy. Another source of variation is the kind of face attributes being used. Traditional formulation employs face texture [1]. It uses morphable

model to generate a synthetic face image and optimizes for parameters that would minimize the difference between the synthetic image and the target. However, this formulation also relies on a sparse set of facial landmarks used for initialization. Earlier methods used manually annotated landmarks [4]. The user was required to annotate a few facial points by hand. Recent explosion of facial landmarking methods [15–18] made this process automatic and the set of landmarks became richer. This posed the question if morphable model fitting could be done based purely on landmarks [19]. It is especially desirable because algorithms based on landmarks are much faster and suitable for real-time performance while texture-based algorithms are quite slow (on the order of 1 minute per image).

Unfortunately existing literature reports only few quantitative evaluations of optimization-based fitting algorithms. Some works assume that landmark-based fitting provides satisfactory accuracy [20, 21] while others demonstrate its limitations [19, 22]. Some use texture-based algorithms at the cost of higher computational demands, but the advantage in accuracy is not quantified [5,6,23]. The situation is further complicated by the lack of standard benchmarks with reliable ground truth and well-defined evaluation procedures.

We implement a morphable model fitting algorithm and tune its parameters in two scenarios: relying solely on landmarks and using landmarks in combination with the texture. We test this algorithm on images from BU4DFE dataset [24] and demonstrate that incorporation of texture significantly improves the accuracy.

It is desirable to enjoy both the accuracy of texture-based reconstruction algorithms and the high processing speed enabled by network-based methods. To this end, we use the fitting algorithm to process 300W database of faces [25] and train a neural network to predict facial geometry on the resulting semi-synthetic dataset. It is important to keep in mind that the applicability of the fitting algorithm is limited by the expressive power of the morphable model. In particular, it doesn't handle large occlusions and extreme lighting conditions very well. To rule the failures out, we visually inspect the processed dataset and delete failed examples. We compare our dataset with a similarly produced 300W-3D [9] and show that our dataset allows to learn more accurate models. We make our dataset publicly available [3].

An important consideration for CNN training is the loss function. Standard losses become problematic when predicting parameters of morphable face models due to the different nature and scales of individual parameters. To resolve this issue, the MSE loss needs to be reweighted and some ad-hoc weighting schemes have been used in the past [9]. We present a loss function that accounts for individual contributions of morphable model parameters in a clear and intuitive manner by constructing a 3D model and directly comparing it to the ground truth in the 3D space and in the 2D projected space.

This work provides the following contributions: (i) we evaluate variants of the fitting algorithm on a database of facial scans providing quantitative evidence

[3] https://github.com/nchinaev/MobileFace

of texture-based algorithms superiority; (ii) we train a MobileNet-based neural network that allows for fast facial shape reconstruction even on mobile devices; (iii) we propose an intuitive loss function for CNN training; (iv) we make our evaluation code and datasets publicly available.

1.1 Related Work

Algorithms for monocular 3d face shape reconstruction may be broadly classified into two following categories: optimization-based and regression-based. Optimization-based approaches make assumptions about the nature of image formation and express them in the form of energy functions. This is possible because faces represent a set of objects that one can collect some strong priors about. One popular form of such prior is a morphable model. Another way to model image formation is shape from shading technique [26–28]. This class of algorithms has a drawback of high computational complexity. Regression-based methods learn from data. The absence of large datasets for this task is a limitation that can be addressed in several ways outlined below.

Learning From Synthetic Data. Synthetic data may be produced by rendering facial scans [8] or by rendering images from a morphable model [12]. Corresponding ground truth 3d models are readily available in this case because they were used for rendering. These approaches have two limitations: first, the variability in facial shapes is only limited to the subjects participating in acquisition, and second, the image formation is limited by the exact illumination model used for rendering.

Unsupervised Learning. Tewari et al. [10] incorporate rendering process into their learning framework. This rendering layer is implemented in a way that it can be back-propagated through. This allows to circumvent the necessity of having ground truth 3d models for images and makes it possible to learn from datasets containing face images alone. In the follow up work Tewari et al. [29] go further and learn corrections to the morphable model. Richardson et al. [14] incorporate shape from shading into learning process to learn finer details.

Fitting + Learning. Most closely related to our work are works of Zhu et al. [9] and Tran et al. [13]. They both use fitting algorithms to generate datasets for neural network training. However, accuracies of the respective fitting algorithms [2] and [3] in the context of evaluation on datasets of facial scans are not reported by their authors. This raises two questions: what is the maximum accuracy attainable by learning from the results of these fitting methods and what are the gaps between the fitting methods and the respective learned networks? We evaluate accuracies of our fitting methods and networks on images from BU4DFE dataset in our work.

2 MobileFace

Our main objective is to create fast and compact face shape predictor suitable for real-time inference on mobile devices. To achieve this goal we train a network

to predict morphable model parameters (to be introduced in Sec. 2.2). Those include parameters related to 3d shape $\boldsymbol{\alpha_{id}}$ and $\boldsymbol{\alpha_{exp}}$, as well as those related to projection of the model from 3d space to the image plane: translation $\boldsymbol{t}$, three angles ϕ, γ, θ and projection f, P_x, P_y. Vector $\boldsymbol{p} \in \mathbb{R}^{118}$ is a concatenation of all the morphable model parameters predicted by the network:

$$\boldsymbol{p} = \begin{pmatrix} \boldsymbol{\alpha_{id}}^T & \boldsymbol{\alpha_{exp}}^T & \boldsymbol{t}^T & \phi & \gamma & \theta & f & Px & Py \end{pmatrix}^T \tag{1}$$

2.1 Loss Functions

We experiment with two losses in this work. The first MSE loss can be defined as

$$\texttt{Loss}_{\texttt{MSE}} = \sum_i ||\boldsymbol{p^i} - \boldsymbol{p^i}_{gt}||_2^2. \tag{2}$$

Such a loss, however, is likely to be sub-optimal as it treats parameters $\boldsymbol{p}$ of different nature and scales equally. They impact the $3d$ reconstruction accuracy and the projection accuracy differently. One way to overcome this is to use the outputs of the network to construct 3d meshes $\boldsymbol{S}(\boldsymbol{p^i})$ and compare them with ground truth $\boldsymbol{S}_{gt}$ during training [30]. However, such a loss alone would only allow to learn parameters related to the 3d shape: $\boldsymbol{\alpha_{id}}$ and $\boldsymbol{\alpha_{exp}}$. To allow the network to learn other parameters, we propose to augment this loss by an additional term on model projections $P(\boldsymbol{p^i})$:

$$\texttt{Loss}_{\texttt{2d + 3d, } l_2} = \sum_i \left(||\boldsymbol{S}(\boldsymbol{p^i}) - \boldsymbol{S_{gt}})||_2^2 + ||P(\boldsymbol{p^i}) - P_{gt}||_2^2 \right) \tag{3}$$

Subscript l_2 indicates that this loss uses l_2 norm for individual vertices. Likewise, we define

$$\texttt{Loss}_{\texttt{2d + 3d, } l_1} = \sum_i \left(||\boldsymbol{S}(\boldsymbol{p^i}) - \boldsymbol{S_{gt}})||_1 + ||P(\boldsymbol{p^i}) - P_{gt}||_1 \right) \tag{4}$$

We provide details of $\boldsymbol{S}(\boldsymbol{p^i})$ construction and $P(\boldsymbol{p^i})$ projection in the next subsection.

2.2 Morphable Model

Geometry Model. Facial geometries are represented as meshes. Morphable models allow to generate variability in both face identity and expression. This is done by adding parametrized displacements to a template face model called the mean shape. We use the mean shape and 80 modes from Basel Face Model [1] to generate identities and 29 modes obtained from Face Warehouse dataset [31] to generate expressions. The meshes are controlled by two parameter vectors $\boldsymbol{\alpha}_{id} \in \mathbb{R}^{80}$ and $\boldsymbol{\alpha}_{exp} \in \mathbb{R}^{29}$:

$$\boldsymbol{S} = \boldsymbol{M} + A_{id} \cdot \boldsymbol{\alpha}_{id} + A_{exp} \cdot \boldsymbol{\alpha}_{exp}. \tag{5}$$

Vector $\boldsymbol{S} \in \mathbb{R}^{3 \cdot N}$ stores the coordinates of N mesh vertices. $\boldsymbol{M}$ is the mean shape. Matrices $A_{id} \in \mathbb{R}^{3 \cdot N \times 80}$, $A_{exp} \in \mathbb{R}^{3 \cdot N \times 29}$ are the modes of variation.

Projection Model. Projection model translates face mesh from the 3d space to a 2d plane. Rotation matrix R and translation vector $\boldsymbol{t}$ apply a rigid transformation to the mesh. Projection matrix with three parameters f, Px, Py transforms mesh coordinates to the homogeneous space. For a vertex $\boldsymbol{v} = (x_m, y_m, z_m)^T$ the transformation is defined as:

$$\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} = \Pi \cdot \begin{pmatrix} R & \boldsymbol{t} \end{pmatrix} \cdot \begin{pmatrix} x_m \\ y_m \\ z_m \\ 1 \end{pmatrix}, \quad \Pi = \begin{pmatrix} f & 0 & P_x \\ 0 & f & P_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{6}$$

and the final projection of a vertex to the image plane is defined by u and v as:

$$u = \frac{x_t}{z_t}, \quad v = \frac{y_t}{z_t}. \tag{7}$$

The projection is defined by 9 parameters including three rotation angles, three translations and three parameters of the projection matrix Π. We denote projected coordinates by:

$$P(\Pi, R, \boldsymbol{t}, \boldsymbol{S}) = \begin{pmatrix} u_1 \; u_2 \; \ldots \; u_N \\ v_1 \; v_2 \; \ldots \; v_N \end{pmatrix}^T \tag{8}$$

2.3 Data Preparation

Our objective here is to produce a dataset of image-model pairs for neural network training. We use the fitting algorithm detailed in Sec. 3.3 to process the 300W database of annotated face images [25]. Despite its accuracy reported in Sec. 4.3 this algorithm has two limitations. First, the expressive power of the morphable model is inherently limited due to laboratory conditions in which the model was obtained and due to the lighting model being used. Hence, the model can't generate occlusions and extreme lighting conditions. Second, the hyperparameters of the algorithm have been tuned for a dataset taken under controlled conditions. Due to these limitations, the algorithm inevitably fails on some of the in-the-wild photos. To overcome this shortcoming, we visually inspect the results and delete failed photos. Note that we do not use any specific criteria and this deletion is guided by the visual appeal of the models, hence it may be performed by an untrained individual. This leaves us with an even smaller amount of images than has initially been in the 300W dataset, namely 2300 images.
This necessitates data augmentation. We randomly add blur and noise in both RGB and HSV spaces. Since some of the images with large occlusions have been deleted during visual inspection, we compensate for this and randomly occlude images with black rectangles of varied sizes [32]. Fig. 1 shows some examples of our training images.

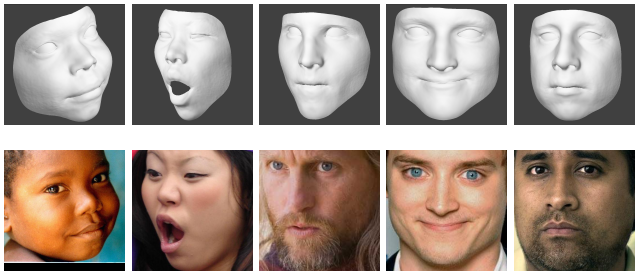


Fig. 1: Example images and corresponding curated ground truth from our training set.

2.4 Network Architecture

Architecture of our network is based on MobileNet [33]. It consists of interleaving convolution and depth-wise convolution [34] layers followed by average pooling and one fully connected layer. Each convolution layer is followed by a batch normalization step [35] and a ReLU activation. Input images are resized to 96×96. The final fully-connected layer generates the outputs vector $\boldsymbol{p}$ eq. (1). Main changes compared to the original architecture in [33] include the decreased input image size $96 \times 96 \times 3$, the first convolution filter is resized to $3 \times 3 \times 3 \times 10$, the following filters are scaled accordingly, global average pooling is performed over 2×2 region, and the shape of the FC layer is 320×118.

3 Morphable Model Fitting

We use morphable model fitting to generate 3d models of real-world faces to be used for neural network training. Our implementation follows standard practices [5, 6]. Geometry and projection models have been defined in (Sec. 2.2). Texture model and lighting allow to generate face images. Morphable model fitting aims to revert the process of image formation by finding the combination of parameters that will result in a synthetic image resembling the target image as closely as possible.

3.1 Image Formation

Texture Model. Face texture is modeled similarly to eq. (5). Each vertex of the mesh is assigned three RGB values generated from a linear model controlled by a parameter vector $\boldsymbol{\beta}$:

$$\boldsymbol{T} = \boldsymbol{T}_0 + B \cdot \boldsymbol{\beta}. \tag{9}$$

We use texture mean and modes from BFM [1].

Lighting Model. We use the Spherical Harmonics basis [36,37] for light computation. The illumination of a vertex having albedo ρ and normal $\boldsymbol{n}$ is computed as

$$I = \rho \cdot \left(\boldsymbol{n}^T \; 1 \right) \cdot M \cdot \begin{pmatrix} \boldsymbol{n} \\ 1 \end{pmatrix}, \tag{10}$$

M is as in [37] having 9 controllable parameters per channel. RGB intensities are computed separately thus giving overall $9 \cdot 3 = 27$ lighting parameters, $\boldsymbol{l} \in \mathbb{R}^{27}$ is the parameter vector. Albedo ρ is dependent on $\boldsymbol{\beta}$ and computed as in eq. (9).

3.2 Energy Function

Energy function expresses the discrepancy between the original attributes of an image and the ones generated from the morphable model:

$$E = E_{\texttt{tex}} + c_{\texttt{lands}} \cdot E_{\texttt{lands}} + r_{beta,2} \cdot E_{\texttt{reg,tex}} + r_{exp,2} \cdot E_{\texttt{reg,exp}}. \tag{11}$$

We describe individual terms of this energy function below.

Texture. The texture term $E_{\texttt{tex}}$ measures the difference between the target image and the one rendered from the model. We translate both rendered and target images to a standardized UV frame as in [2] to unify all the image resolutions. Visibility mask $\mathscr{M}$ cancels out the invisible pixels.

$$E_{\texttt{tex}} = \frac{||\mathscr{M} \cdot (I_{\texttt{target}} - I_{\texttt{rendered}})||}{|\mathscr{M}|}. \tag{12}$$

We produce $I_{\texttt{rendered}}$ by applying eq. (10) and $I_{\texttt{target}}$ by sampling from the target image at the positions of projected vertices P eq. (8). Visibility mask $\mathscr{M}$ is computed based on the orientations of vertex normals. We test three alternative norms in place of $||\cdot||$: l_1, l_2 and $l_{2,1}$ norm [5] that sums $l2$ norms computed for individual pixels.

Landmarks. We use the landmark detector of [15]. Row indices $\mathscr{L} = \{k_i\}_{i=1}^{68}$ for matrix P eq. (8) correspond to the 68 landmarks. Detected landmarks are $L \in \mathbb{R}^{68 \times 2}$. The landmark term is defined as:

$$E_{\texttt{lands}} = ||L - P_{\mathscr{L},:}||_2^2. \tag{13}$$

One problem with this term is that indices $\mathscr{L}$ are view-dependent due to the landmark marching. We adopt a solution similar to that of [20] and annotate parallel lines of vertices for the landmarks on the border.

Regularization. We assume multivariate Gaussian priors on morphable model parameters as defined below and use σ_{id} and σ_{tex} provided by [1].

$$E_{\texttt{reg,id}} = \sum_{i=1}^{80} \frac{\alpha_{id,i}^2}{\sigma_{id,i}^2}, \quad E_{\texttt{reg,exp}} = \sum_{i=1}^{29} \frac{\alpha_{exp,i}^2}{\sigma_{exp,i}^2}, \quad E_{\texttt{reg,tex}} = \sum_{i=1}^{80} \frac{\beta_i^2}{\sigma_{tex,i}^2} \tag{14}$$

We regularize neither lighting nor projection parameters.

3.3 Optimization

Optimization process is divided into two major steps: First, we minimize the landmark term:

$$E_1 = E_{\texttt{lands}} + r_{id,1} \cdot E_{\texttt{reg,id}} + r_{exp,1} \cdot E_{\texttt{reg,exp}}. \tag{15}$$

We then minimize the full energy function eq. (11). These two steps are also divided into sub-steps minimizing the energy function with respect to specific parameters similarly to [6]. We minimize the energy function with respect to only one type of parameters at any moment. We do not include identity regularization into eq. (11) because it did not improve accuracy in our experiments.

4 Experiments

We carry out three sets of experiments. First, we study the effect of different settings for the fitting of the morphable model used in this paper. Second, we experiment with different losses and datasets for neural network training. Finally, we present a comparison of our method with other recent approaches.

Unfortunately current research in 3d face reconstruction is lacking standardized benchmarks and evaluation protocols. As a result, evaluations presented in research papers vary in the type of error metrics and datasets used (see Table 1). This makes the results from many works difficult to compare. We hope to contribute towards filling this gap by providing the standard evaluation code and a testing set of images[4].

BU4DFE Selection. Tulyakov et al. [38] provide annotations for a total of 3000 selected scans from BU4DFE. We divide this selection into two equally sized subsets BU4DFE-test and BU4DFE-val. We report final results on the former and experiment with hyperparameters on the latter. For the purpose of evaluation we use annotations to initialize the ICP alignment.

4.1 Implementation Details

We trained networks for the total of $3 \cdot 10^5$ iterations with the batches of size 128. We added l_2 weight decay with coefficient of 10^{-4} for regularization. We used Adam optimizer [39] with learning rate of 10^{-4} for iterations before $2 \cdot 10^5$-th and 10^{-5} after. Other settings for the optimizer are standard. Coefficients for morphable model fitting are $r_{id,1} = 0.001$, $r_{exp,1} = 0.1$, $r_{beta,2} = 0.001$, $c_{\texttt{lands}} = 10$, $r_{exp,2} = 10$.

4.2 Accuracy Evaluation

Accuracy of 3D reconstruction is estimated by comparing the resulting 3D model to the ground truth facial scan. To compare the models, we first perform ICP

[4] https://github.com/nchinaev/MobileFace

alignment. Having reconstructed facial mesh $\boldsymbol{S}$ and the ground truth scan $\boldsymbol{S_{gt}}$, we project vertices of $\boldsymbol{S}$ on $\boldsymbol{S_{gt}}$ and Procrustes-align $\boldsymbol{S}$ to the projections. These two steps are iterated until convergence.

Error Measure. To account for variations in scan sizes, we use a normalization term

$$C(\boldsymbol{S_{gt}}) = ||\boldsymbol{S^0_{gt}}||_2^2, \tag{16}$$

where $\boldsymbol{S^0_{gt}}$ is $\boldsymbol{S_{gt}}$ with the mean of each x, y, z coordinate subtracted. The dissimilarity measure between $\boldsymbol{S}$ and $\boldsymbol{S_{gt}}$ is

$$d(\boldsymbol{S}, \boldsymbol{S_{gt}}) = c_s \cdot \frac{||\boldsymbol{S} - \boldsymbol{S_{gt}}||_2^2}{C(\boldsymbol{S_{gt}})} \tag{17}$$

The scaling factor $c_s = 100$ is included for convenience.

Table 1: Methods and their corresponding testsets.

Work	Testset
Jackson et al. [11]	AFLW2000-3D, renders from BU4DFE and MICC
Tran et al. [13]	MICC video frames
Tewari et al. [10]	synthetic data; Face Warehouse
Dou et al. [30]	UHDB31, FRGC2, BU-3DFE
Roth et al. [28]	renders from BU4DFE

4.3 Morphable Model Fitting

We compare the accuracy of the fitting algorithm in two major settings: using only landmarks and using landmarks in combination with texture. To put the numbers in a context, we establish two baselines. First baseline is attained by computing the reconstruction error for the mean shape. This demonstrates the performance of a hypothetical dummy algorithm that always outputs the mean shape for any input. Second baseline is computed by registering the morphable model to the scans in 3d. It demonstrates the performance of a hypothetical best method that is only bounded by the descriptive power of the morphable model. Landmark-based fitting is done by optimizing eq. (15) from sec. 3.3. Texture-based fitting is done by optimizing both eq. (15) and eq. (11). Fig. 2 shows cumulative error distributions. It is clear from the graph that texture-based fitting significantly outperforms landmark-based fitting which is only as accurate as the meanshape baseline. However, there is still a wide gap between the performance of the texture-based fitting and the theoretical limit. Figs. 3a, 3b

show the performance of texture-based fitting algorithm with different settings. The settings differ in the type of norm being used for texture term computation and the amount of regularization. In particular, Fig. 3a demonstrates that the choice of the norm plays an important role with $l_{2,1}$ and l_1 norms outperforming l_2. Fig. 3b shows that the algorithm is quite sensitive to the regularization, hence the regularization coefficients need to be carefully tuned.

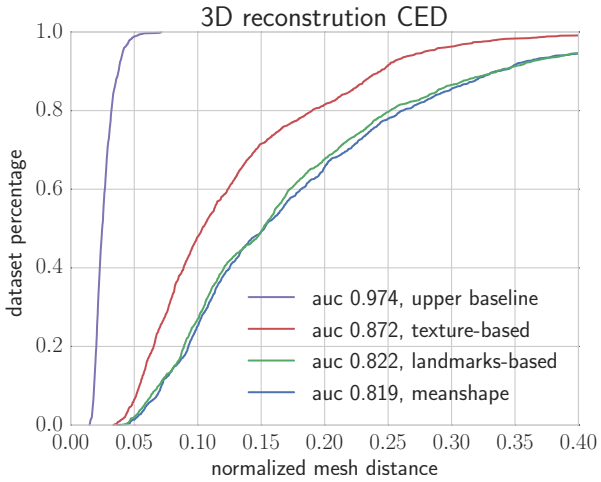


Fig. 2: Evaluation of fitting methods on BU4DFE-test. Areas under curve are computed for normalized mesh distances ranging from 0 to 1. Shorter span of x-axis is used for visual clarity.

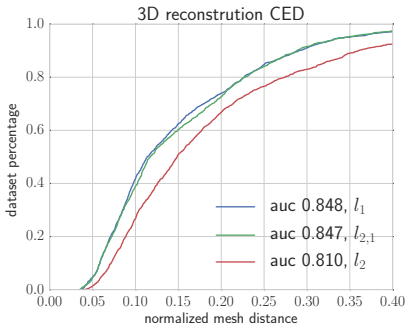
4.4 Neural Network

We train the network on our dataset of image-model pairs. For the sake of comparison, we also train it on 300W-3D [9]. The training is performed in different settings: using different loss functions and using manually cleaned version of the dataset versus non-cleaned. The tests are performed on BU4DFE-val. Figs. 4a, 4b show cumulative error distributions. These experiments support following claims:

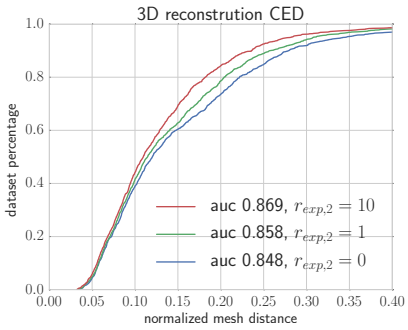
- Learning from our dataset gives better results than learning from 300W-3D,
- Our loss function improves results compared to baseline MSE loss function,
- Manual deletion of failed photos by an untrained individual improves results.

4.5 Comparison with the State of the Art

Quantitative Results. Fig. 5 presents evaluations of our network and a few recent methods on BU4DFE-test. Error metric is as in eq. (17). The work of Tran

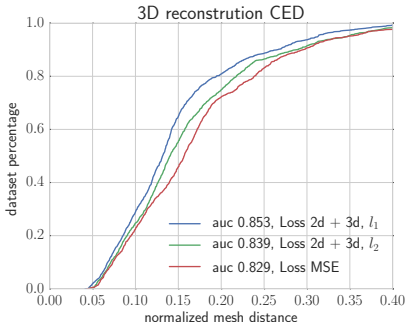


(a) Evaluations for different norms. $r_{exp,2} = 0$

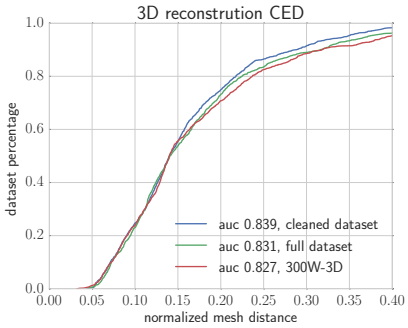


(b) Evaluations with different regularizations. $l_{2,1}$ norm is used in all cases.

Fig. 3: Evaluation of texture-based fitting algorithm on BU4DFE-val with different settings.



(a) Comparison of networks trained on our cleaned dataset with different losses.



(b) Comparison of networks trained with $\texttt{Loss}_{\texttt{2d + 3d}},\ l_2$ on different datasets.

Fig. 4: Evaluation on BU4DFE-val for networks trained with different losses on different datasets.

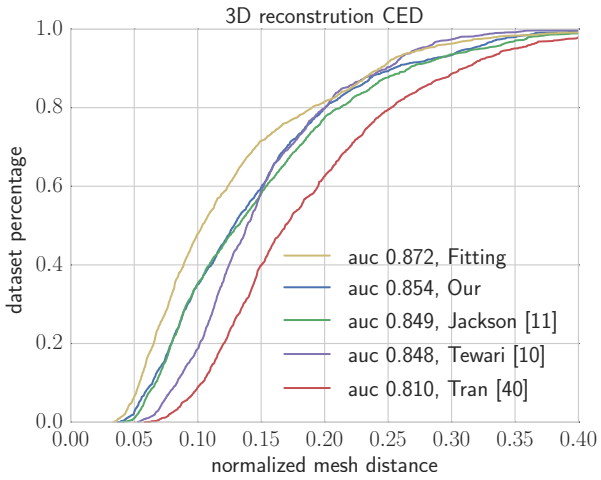


Fig. 5: Comparison of methods on BU4DFE-test.

et al. [40] is based on [13] and their code allows to produce non-neutral models, therefore we present an evaluation of [40] and not [13]. We do not present an evaluation of 3DDFA [9] because Jackson et al. [11] have already demonstrated that 3DDFA is inferior to their method. We do not include the work of Sela et al. [41] into comparison because we were not able to reproduce their results. For Jackson et al. [11] we were able to reproduce their error on AFLW2000-3D. We used MATLAB implementation of isosurface algorithm to transform their volumes into meshes. Tran et. al [40] do not present an evaluation on 3d scans, however we were able to roughly reproduce an error for MICC dataset [42] from their earlier work [13]. We noticed that their method is sensitive to the exact selection of frames from MICC videos. For an optimal selection of frames the error equals 1.43 which is less than 1.57 reported in [13]. In the worst case error equals 2.34. Tewari et al. [10] did not open-source their implementation of MOFA, but authors kindly provided their reconstructed models for our testset.

It is seen from the graph that our network performs on a par with other recent methods being slightly ahead of the second-best method. Additionally, the size of our model is orders of magnitude smaller, see Table 4 for a comparison. We used Intel Core i5-4460 for CPU experiments, NVIDIA GeForce GTX 1080 for GPU experiments (except for Tewari [10], they used NVIDIA Titan X Pascal) and Samsung Galaxy S7 for ARM experiments.

Table 2 presents a comparison with Tran et al. [13] on MICC dataset [42]. This is a dataset of 53 subjects. For each of the subjects it provides three videos and a neutral facial scan. It is therefore crucial that a method being evaluated on this dataset should output neutral models for any input. Method of Tran et al. [13] is specifically designed for this purpose. We adapt our method to this scenario by setting $\boldsymbol{\alpha}_{exp}$ to zero. We randomly select 23 frames per individual

and form 23 corresponding testsets. We compute errors over these testsets and average those. One important aspect affecting the errors is the use of scaling during ICP alignment: Tran et al. [13] did not allow models to scale during the alignment. We present evaluations in both settings.

Table 3 presents a comparison with Tewari et al. [10] and Garrido et al. [6] on a selection of 9 subjects from Face Warehouse [31] dataset. Version of Tewari et al. network with surrogate loss has been used for this and previous evaluations.

Table 2: Comparison with Tran et al. [13] on MICC dataset [42]. All numbers are in mm.

Method	w.o. scale	w. scale
Tran [13]	1.83 ± 0.04	1.46 ± 0.03
Our	**1.70** ± 0.02	**1.33** ± 0.02

Table 3: Comparisons on a selection from Face Warehouse dataset. All numbers are in mm.

Method	Our	Tewari [10]	Garrido [6]
Error	1.8 ± 0.07	1.7	1.4

Qualitative Results. Fig. 6 shows a comparison with Jackson et al. [11], Tewari er al. [10] and Tran et al. [40] on a few images from BU4DFE-test.

5 Conclusions

We have presented an evaluation of monocular morphable model fitting algorithms and a learning framework. It is demonstrated that incorporation of texture term into the energy function significantly improves fitting accuracy. Gains in the accuracy are quantified. We have trained a neural network using the outputs of the fitting algorithms as training data. Our trained network is shown to perform on a par with existing approaches for the task of monocular 3d face reconstruction while showing faster speed and smaller model size. Running time of our network on a mobile devise is shown to be 3.6 milliseconds enabling real-time applications. Our datasets and code for evaluation are made publicly available.

Table 4: Model size and running time comparison.

Method	Model size	GPU	CPU	ARM	AUC
Jackson [11]	1.4 GB	-	2.7 s	-	0.849
Tewari [10]	0.27 GB	4 ms	-	-	0.848
Tran [40]	0.35 GB	40 ms	-	-	0.810
Our	**1.5 MB**	**1 ms**	**1.8 ms**	3.6 ms	**0.854**

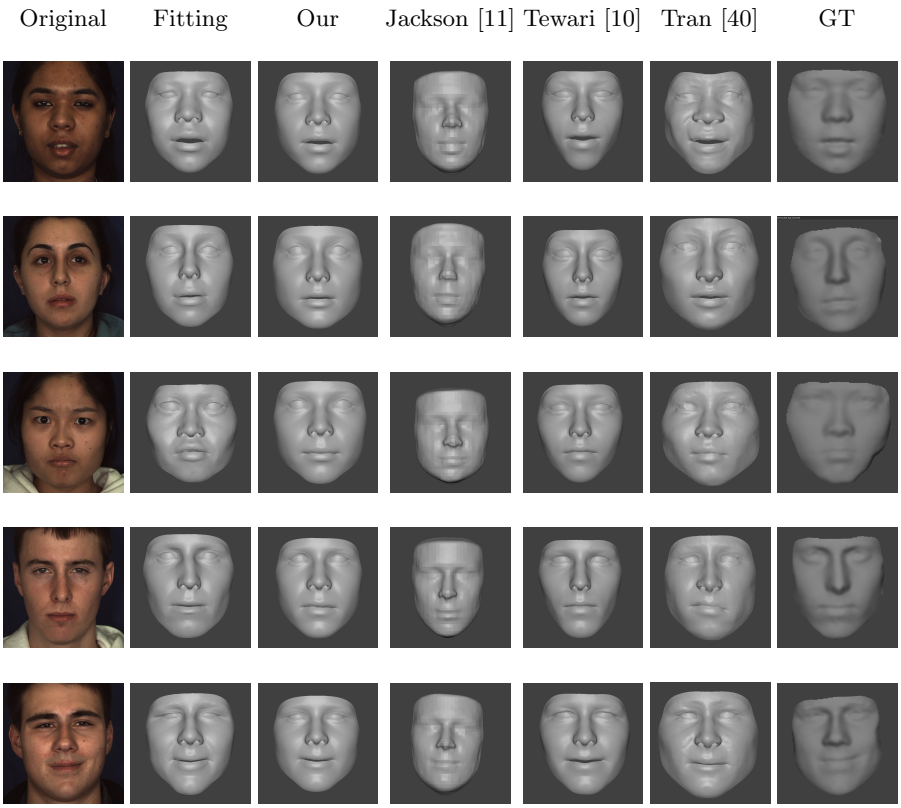


Fig. 6: Qualitative results. Images are from BU4DFE-test. Our implementation of fitting was used for the second column.

References

1. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH Conference Proceedings. (1999)
2. Romdhani, S., Vetter, T.: Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In: Computer Vision and Pattern Recognition (CVPR). (2005)
3. Piotraschke, M., Blanz, V.: Automated 3d face reconstruction from multiple images using quality measures. Computer Vision and Pattern Recognition (CVPR) (2016)
4. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. IEEE Trans. Pattern Anal. Mach. Intell. (2003)
5. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: Real-time face capture and reenactment of rgb videos. In: Computer Vision and Pattern Recognition (CVPR). (2016)
6. Garrido, P., Zollhoefer, M., Casas, D., Valgaerts, L., Varanasi, K., Perez, P., Theobalt, C.: Reconstruction of personalized 3d face rigs from monocular video. In: ACM Trans. Graph. (Presented at SIGGRAPH 2016). (2016)
7. Zhu, X., Yan, J., Yi, D., Lei, Z., Li, S.Z.: Discriminative 3d morphable model fitting. In: Automatic Face and Gesture Recognition (FG). (2015)
8. Jeni, L.A., Cohn, J.F., Kanade, T.: Dense 3d face alignment from 2d videos in real-time. In: Automatic Face and Gesture Recognition (FG). (2015)
9. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.Z.: Face alignment across large poses: A 3d solution. In: Computer Vision and Pattern Recognition (CVPR). (2016)
10. Tewari, A., Zollhöfer, M., Kim, H., Garrido, P., Bernard, F., Perez, P., Theobalt, C.: MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In: International Conference on Computer Vision (ICCV). (2017)
11. Jackson, A., Bulat, A., Argyriou, V., Tzimiropoulos, G.: Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In: International Conference on Computer Vision (ICCV). (2017)
12. Richardson, E., Sela, M., Kimmel, R.: 3d face reconstruction by learning from synthetic data. In: Fourth International Conference on 3D Vision (3DV). (2016)
13. Tran, A., Hassner, T., Masi, I., Medioni, G.: Regressing robust and discriminative 3d morphable models with a very deep neural network. In: Computer Vision and Pattern Recognition (CVPR). (2017)
14. Richardson, E., Sela, M., Or-El, R., Kimmel, R.: Learning detailed face reconstruction from a single image. In: Computer Vision and Pattern Recognition (CVPR). (2017)
15. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: Computer Vision and Pattern Recognition (CVPR). (2014)
16. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: Computer Vision and Pattern Recognition (CVPR). (2013)
17. Bulat, A., Tzimiropoulos, G.: How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In: International Conference on Computer Vision (ICCV). (2017)
18. Trigeorgis, G., Snape, P., Nicolaou, M.A., Antonakos, E., Zafeiriou, S.: Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In: Computer Vision and Pattern Recognition (CVPR). (2016)
19. Bas, A., Smith, W.A.P.: What does 2d geometric information really tell us about 3d face shape? Arxiv preprint (2017)

20. Zhu, X., Lei, Z., Yan, J., Yi, D., Li, S.Z.: High-fidelity pose and expression normalization for face recognition in the wild. In: Computer Vision and Pattern Recognition (CVPR). (2015)
21. Fried, O., Shechtman, E., Goldman, D.B., Finkelstein, A.: Perspective-aware manipulation of portrait photos. ACM Transactions on Graphics (Proc. SIGGRAPH) (2016)
22. Keller, M., Knothe, R., Vetter, T.: 3d reconstruction of human faces from occluding contours. In: Computer Vision/Computer Graphics Collaboration Techniques. (2007)
23. Saito, S., Wei, L., Hu, L., Nagano, K., Li, H.: Photorealistic facial texture inference using deep neural networks. Computer Vision and Pattern Recognition (CVPR) (2017)
24. Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M.: A 3d facial expression database for facial behavior research. In: Automatic Face and Gesture Recognition (FG). (2006)
25. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: 300 faces in-the-wild challenge: The first facial landmark localization challenge. In: International Conference on Computer Vision (ICCV). (2013)
26. Kemelmacher-Shlizerman, I., Basri, R.: 3d face reconstruction from a single image using a single reference face shape. IEEE Trans. Pattern Anal. Mach. Intell. (2011)
27. Suwajanakorn, S., Kemelmacher-Shlizerman, I., Seitz, S.M.: Total moving face reconstruction. In: European Conference on Computer Vision (ECCV). (2014)
28. Roth, J., Tong, Y., Liu, X.: Adaptive 3d face reconstruction from unconstrained photo collections. In: Computer Vision and Pattern Recognition (CVPR). (2016)
29. Tewari, A., Zollhöfer, M., Garrido, P., Bernard, F., Kim, H., Perez, P., Theobalt, C.: Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In: arxiv preprint. (2017)
30. Dou, P., Shah, S.K., Kakadiaris, I.A.: End-to-end 3d face reconstruction with deep neural networks. In: Computer Vision and Pattern Recognition (CVPR). Volume 5. (2017)
31. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: a 3d facial expression database for visual computing. IEEE Transactions on Visualization and Computer Graphics (2014)
32. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
33. G. Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. (04 2017)
34. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. Computer Vision and Pattern Recognition (CVPR) (2017)
35. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML). (2015)
36. Ramamoorthi, R., Hanrahan, P.: A signal-processing framework for inverse rendering. In: SIGGRAPH. (2001)
37. Ramamoorthi, R., Hanrahan, P.: An efficient representation for irradiance environment maps. In: 28th annual conference on Computer graphics and interactive techniques. (2001)
38. Tulyakov, S., Sebe, N.: Regressing a 3d face shape from a single image. In: International Conference on Computer Vision (ICCV). (2015)

39. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
40. Tran, A.T., Hassner, T., Iacopo, M., Paz, E., Nirkin, Y., Medioni, G.: Extreme 3d face reconstruction: Looking past occlusions. In: arxiv preprint. (2017)
41. Sela, M., Richardson, E., Kimmel, R.: Unrestricted facial geometry reconstruction using image-to-image translation. In: International Conference on Computer Vision (ICCV). (2017)
42. Bagdanov, A.D., Del Bimbo, A., Masi, I.: The florence 2d/3d hybrid face dataset. In: Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding. (2011)