Faster, Better and More Detailed: 3D Face Reconstruction with Graph Convolutional Networks

Shiyang Cheng[1], Georgios Tzimiropoulos[1,3], Jie Shen[2,*], and Maja Pantic[2]

[1] Samsung AI Center, Cambridge. {shiyang.c,georgios.t}@samsung.com
[2] Imperial College London. {js1907,m.pantic}@imperial.ac.uk
[3] Queen Mary University of London.

Abstract. This paper addresses the problem of 3D face reconstruction from a single image. While available solutions for addressing this problem do exist, to our knowledge, we propose the very first approach which is robust, lightweight and detailed i.e. it can reconstruct fine facial details. Our method is extremely simple and consists of 3 key components: (a) a lightweight non-parametric decoder based on Graph Convolutional Networks (GCNs) trained in a supervised manner to reconstruct *coarse* facial geometry from image-based ResNet features. (b) An extremely lightweight (35K parameters) subnetwork – also based on GCNs – which is trained in an unsupervised manner to refine the output of the first network. (c) A novel feature-sampling mechanism and adaptation layer which injects fine details from the ResNet features of the first network into the second one. Overall, our method is the first one (to our knowledge) to reconstruct detailed facial geometry relying solely on GCNs. We exhaustively compare our method with 7 state-of-the-art methods on 3 datasets reporting state-of-the-art results for all of our experiments, both qualitatively and quantitatively, with our approach being, at the same time, significantly faster.

1 Introduction

3D face reconstruction is the problem of recovering the 3D geometry (3D shape in terms of X,Y,Z coordinates) of a face from one or more 2D images. 3D face reconstruction from a single image has recently witnessed great progress thanks to the advent of end-to-end training of deep neural networks for supervised learning. However, it is still considered a difficult open problem in face analysis as existing solutions are far from being perfect.

In particular, a complete solution for 3D face reconstruction must possess at least the following 3 features: (a) Being robust: it should work for arbitrary facial poses, illumination conditions, facial expressions, and occlusions. (b) Being

* Corresponding author.

efficient: it should reconstruct a large number of 3D vertices without using excessive computational resources. (c) Being detailed: it should capture fine facial details (e.g. wrinkles). To our knowledge, there is no available solution having all aforementioned features. For example, VRN [1] is robust but it is neither efficient nor detailed. PRNet [2] is both robust and efficient but it is not detailed. CMD [3] is lightweight but not detailed. The seminal work of [4] and the very recent DF^{2}Net [5] produce detailed reconstructions but it is not robust. Our goal in this paper is to make a step forward towards solving all three aforementioned problems.

To address this challenge, we propose a method which effectively combines the favourable properties of many of the methods mentioned above. In particular, our framework – consisting of two connected subnetworks as shown in Fig. 1 – innovates in the following 4 ways:

1. Our first subnetwork is a non-parametric method, like [1], which is trained to perform direct regression of the 3D coordinates in a supervised manner and works robustly for in-the-wild facial images in large poses, expressions and arbitrary illuminations. Contrary to [1] though, we use Graph Convolutional Networks (GCN) to perform regression in a very lightweight manner.
2. Our method also has a second subnetwork, like [4] and [5], which is trained in an unsupervised manner – using a Shape-from-Shading (SfS) loss – to refine the output of the first subnetwork by adding missing facial details. Contrary to [4] and [5] though, we implemented this subnetwork in a extremely lightweight manner using a second GCN, the vertices of which are in full correspondence with the vertices of our first subnetwork.
3. We further improve the ability of our method to reconstruct fine facial details by introducing a novel feature-sampling mechanism and adaptation layer which injects fine details from the mid-level features of the encoder of the first subnetwork into the decoder of the second subnetwork one.
4. We extensively compare our method with 7 state-of-the-art methods on 3 datasets and report better results for all of our experiments, both quantitatively and quantitatively, with our approach being, at the same time, significantly faster than most.

2 Related Work

Dense 3D face reconstruction from a single image is a heavily studied topic in the area of face analysis. In the following section, we will briefly review related works from the Deep Learning era.

Parametric (3DMM) methods. A large number of methods for 3D reconstruction build upon 3D Morphable Models (3DMMs) [6, 7] which was the method of choice for 3D face modelling prior to the advent of Deep Learning.

Early approaches focused on supervised learning for 3DMM parameter estimation using ground truth 3D scans or synthesized data. 3DDFA [8] iteratively applies a CNN to estimate the 3DMM parameters using the 2D image and a 3D representation produced at the previous iteration as input. The authors in [9]

fit a 3DMM into a 2D image using a very deep CNN, and describe a method to construct a large dataset with 3D pseudo-groundtruth. A similar 3DMM fitting approach was proposed in [10]. Parameter estimation in 3DMM space is, in general, a difficult optimization problem for CNNs. As a result, these methods (a) fail to produce good results for difficult facial images with large poses, expressions and occlusions while (b) in many cases the reconstructions fail to capture the shape characteristics of the face properly. We avoid both obstacles by using a non-parametric model for our first subnetwork which uses a GCN to learn directly to regress the 3D coordinates of the facial geometry without requiring to perform any 3DMM parameter estimation.

Beyond supervised learning, several methods also attempt to fit or even learn a 3DMM from in-the-wild images in an unsupervised manner (i.e. without 3D ground truth data) via image reconstruction. MOFA [11] combines a CNN encoder with an hand-crafted differentiable model-based decoder that analytically implements image formation which is then used for learning from in-the-wild images. This idea was further extended in [12] which proposed an improved multi-level model that uses a 3DMM for regularization and a learned – in a self-supervised manner – corrective space for out-of-space generalization which goes beyond the predefined low-dimensional 3DMM face prior. A similar idea was also proposed in [13] with a different network and loss design. The authors in [14] propose to learn a non-linear 3DMM, where texture and shape reconstruction is performed with neural network decoders (rather than linear operations as in 3DMM) learned directly from data. This work was extended in [15] which proposes to learn coarse shape and albedo for ameliorating the effect of strong regularisations as well as promoting high-fidelity facial details. The last two methods do not use a linear model for shape and texture however. They are trained in a semi-supervised fashion where 3DMM results for the 300W-LP dataset [8] are used to constrain the learning.

All the aforementioned methods employ at some point a 3DMM (either explicitly or as regularisation), and, as such, inevitably the reconstruction result does not capture well identity-related shape characteristics and is biased towards the mean face. Furthermore, image reconstruction losses provide an indirect way to learn a model which has not been shown effective for completely unconstrained images in arbitrary poses. Our method bypasses these problems by using non-parametric supervised learning to reconstruct coarse facial geometry (notably without a 3DMM) and non-parametric unsupervised learning via image reconstruction to recover the missing facial details.

Non parametric methods. There are also a few methods which avoid the use and thus the limitations of parametric models for 3D face reconstruction. By performing direct volumetric regression, VRN [1] is shown to work well for facial images of arbitrary poses. Nonetheless, the method regresses a large volume which is redundant, memory intensive and does not scale well with the number of vertices. Our method avoids these problems by using a GCN to perform direct regression of surface vertices. GCNs for 3D body reconstruction were used in [16]. But this method can capture only coarse geometry and cannot be applied for

detailed face reconstruction. Moreover, we used a different GCN formulation based on spiral convolutions. The semi-parametric method of [3] combines GCNs with an unsupervised image reconstruction loss for model training. Owing to the use of GCNs the method is lightweight but not able to capture fine details.

Shape-from-Shading (SfS) based methods. Shape-from-Shading (SfS) is a classical technique for decomposing shape, reflectance and illumination from a single image. SfS methods have been demonstrated to be capable of reconstructing facial details beyond the space of 3DMMs [17, 18, 4, 5, 19–25]. SfS is a highly ill-posed problem and as such SfS methods require regularisation. For example, in Pix2vertex [18], a smoothness constraint was applied on the predicted depth map. The seminal work of [4] was the first one to incorporate an unsupervised image reconstruction loss based on SfS principles for end-to-end detailed 3D face reconstruction. It used a subnetwork trained in a supervised manner to firstly estimate a coarse face (sometimes also called *proxy face*) using a 3DMM and then another subnetwork trained in a unsupervised manner using SfS principles to refine reconstruction. A notable follow-up work is the multi-stage DF^{2}Net [5] which predicts depth maps in a supervised manner and then refines the result in two more SfS-based stages, where all stages are trained with progressively more detailed datasets. Notably, our method is inspired by [4], but it is based on non-parametric estimation. In addition, ours is based on GCNs, and thus is simpler, faster, and more robust compared to both [4] and [5].

Graph Convolution Networks (GCNs) based methods. Graph Convolution Networks (GCNs) are a set of methods that try to define various convolution operations on graphs. They include but are not limited to spectral methods [26–28], local spatial methods [29, 30] and soft attention [31–33]. As 3D face mesh is also a graph, applications of GCNs on 3D face modeling [34–37] are emerging. The work of [35] was the first one to build a 3D-to-3D face autoencoder based on spectral convolutions. More recently, the work of [34] employed the spiral convolution network of [29] to build another 3D-to-3D face autoencoder. These works focus on a 3D-to-3D setting. To our knowledge, we are the first to employ spiral convolutions for 3D face reconstruction from a single 2D image. More importantly, we are the first to show how to integrate GCNs with SfS-based losses for recovering facial details in an unsupervised manner.

3 Method

3.1 Overview of our framework

The proposed framework is illustrated in Fig. 1, it consists of two connected subnetworks: the first one is an encoder-decoder designed to reconstruct coarse 3D facial geometry. It takes advantage of a simple and light-weight spiral convolution decoder to directly regress the 3D coordinates of a face mesh with arbitrary pose (i.e. in an non-parametric fashion). This mesh will be used to sample and provide features for the second network. Our second network is another GCN that utilises the normals of the coarse face and the per vertex RGB values sampled from the

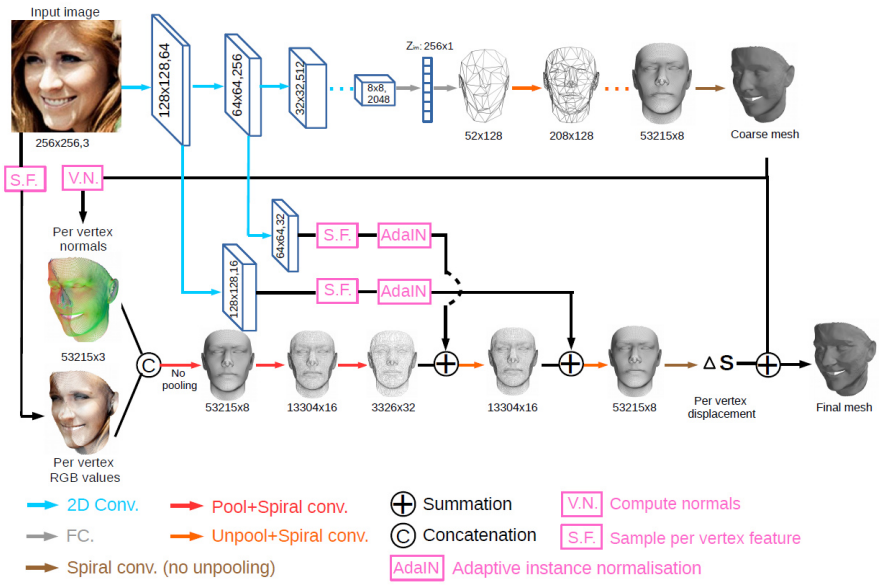


Fig. 1. Overview of our framework. It consists of two connected sub-networks: (1) a coarse 3D facial mesh reconstruction network with a CNN encoder and a GCN decoder; (2) a GCN-based mesh refinement network for recovering the fine facial details. We also device a feature sampling and adaptation layer which injects fine details from the CNN encoder to the refinement network.

input image to estimate per vertex shape displacements (i.e. again in a non-parametric fashion) that are used to synthesise facial geometric details. We then simply superimpose the predicted shape displacement on the coarse mesh to obtain our final 3D face. We also propose a novel feature-sampling mechanism and adaptation layer which injects fine details from the features of the first network into the layers of the second one.

3.2 Coarse 3D face reconstruction with GCN

We design an encoder-decoder network trained to reconstruct the coarse 3D facial geometry from a single image in a fully supervised manner. Note that the reconstructed face at this stage is coarse primarily because of the dataset employed to train the network (300W-LP [8]), and not because of some limitation of our decoder. We emphasize that the network is trained to directly regress the 3D coordinates and does not perform any parameter estimation. This is in contrary to many existing non-linear 3DMMs fitting strategies [14, 38, 11, 39, 40], where the decoder is trained to regress 3DMMs parameters. To our knowledge, we are

the first to leverage a GCN, and in particular, based on spiral convolutions [29] to directly regress a 3D facial mesh from in-the-wild 2D images .

As shown in the upper half of Fig. 1, given an input image $\mathbf{I}$, we first employ a CNN encoder (ResNet [41] or MobileNetV2 [42]) to encode this image into a feature vector $\mathbf{z}_{im} = E(\mathbf{I})$. We then employ a mesh decoder D built using the spiral convolution operator of [29] described below. The feature vector $\mathbf{z}_{im}$ is firstly transformed into a mesh-like structure (each node represents a 128-d feature) using a FC layer. Then, it is unpooled and convolved five times until reaching the full resolution of the target mesh. Lastly, another spiral convolution is performed to generate the coarse mesh.

Spiral convolution and mesh pooling. We define the face mesh as a graph $\mathcal{M} = (\mathcal{V}, \mathcal{E})$, in which $\mathcal{V} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, and $\mathcal{E}$ denote the sets of vertices and edges, respectively. We further denote the vertex feature as $f(\mathbf{x}) \in \mathcal{R}^C$. We built our GCN using the spiral convolution of [29] due to its simplicity: to perform a convolution-like operation over the graph, a local vertex ordering is defined using the spiral operator proposed in [29]. Specifically, for each vertex $\mathbf{x} \in \mathcal{V}$, the operator outputs a spiral S which is an ordered sequence of fixed length of L neighbouring vertices as shown in Fig. 2. Since the order and length is fixed, one can concatenate the features from all vertices in S into a vector $f_S \in \mathcal{R}^{(C \times L) \times 1}$ and define the output of a set of C_{out} filters stored as rows in matrix $\mathcal{W} \in \mathcal{R}^{C_{out} \times (C \times L)}$ as $f_{out} = \mathcal{W} f_S$. This is equivalent to applying a set of filters on a local image window. Furthermore, since the vertices $\mathcal{V}$ of the facial mesh are ordered, this process can be applied sequentially for all $\mathbf{x} \in \mathcal{V}$. This defines a convolution over the graph. Finally, for mesh pooling and unpooling, we follow the practice introduced in [35].

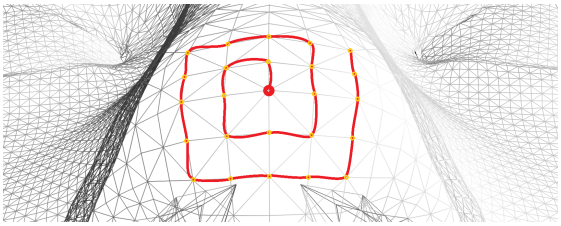


Fig. 2. An example of a spiral neighborhood around a vertex on the facial mesh.

Loss function. We use the $\mathcal{L}_1$ reconstruction error between the groundtruth 3D mesh $\mathbf{S}_{gt} \in \mathcal{R}^{n \times 3}$ and our predicted mesh $\mathbf{S}_{coarse} = D(E(\mathbf{I}))$:

$$\mathcal{L}_{coarse} = \sum_{i=1}^{N} |D(E(\mathbf{I})) - \mathbf{S}_{gt}|, \tag{1}$$

The method of [3] is semi-parametric as it tries to recover 22 parameters for pose and lighting

where N is the total number of training examples. Note that we do not define any additional scale, pose nor expression parameters in our network. We also observe that the spiral mesh decoder tends to produce smooth results, thus there is no need to define an extra smoothness loss.

3.3 Unsupervised detailed reconstruction

Spiral mesh refinement network. As depicted in the lower half of Fig. 1, we devise a mesh refinement network for synthesising fine details over the coarse mesh. Again, our network is fully based on spiral convolution networks. There are two inputs, the first one is the per vertex RGB values sampled from the input image. Specifically, we project the coarse mesh back to the image space and sample the corresponding RGB values using bilinear interpolation. Here, orthographic projection is chosen for simplicity. The second input is the vertex normals of the coarse mesh which provides a strong prior for the detailed 3D shape of the target face. Note that we prefer vertex normals over xyz coordinates because: (1) vertex normals are scale and translation invariant; (2) vertex normals have a fix range of value (*i.e.*, $[-1, 1]$). Both properties lower the training difficulty of our refinement network. These two inputs are concatenated and then convolved and pooled 3 and 2 times respectively until reaching $\sim 1/16$ of the full mesh resolution. Following this, the feature mesh is unpooled and convolved twice. During this process, we adapt and inject intermediate features from the 2D image encoder to the refinement network (we will elaborate this module in the next paragraph). Finally, after another spiral convolution is applied, we obtain the facial details in the form of per vertex shape displacement values $\Delta\mathbf{S}$. We apply the displacement over the coarse mesh to obtain the final reconstruction result, $\mathbf{S}_{final} = \mathbf{S}_{coarse} + \Delta\mathbf{S}$.

Image-level feature sampling and adaptation layer. One of the main contributions of our paper is the utilization of fine CNN features from the image encoder into our refinement GCN. More specifically, and as can be seen in Fig. 1, in our framework the coarse and fine networks are bridged by injecting intermediate features from the 2D image encoder into the spiral mesh refinement network. Although this idea is simple, we found it is non-trivial to design an appropriate module for this purpose, because the features from these two networks are coming from different domains (RGB and 3D mesh). We therefore introduce a novel feature-sampling mechanism and adaptation layer to address this problem which we describe here with a concrete example: Given an image feature $\mathbf{f}_{im} \in \mathbb{R}^{128\times128\times64}$ returned by the first convolution block, we first perform a 1x1 convolution to ensure it has the same number of channels as the target feature $\mathbf{f}_{mesh} \in \mathbb{R}^{13304\times16}$ in the refinement network. Next, we sample the feature using the predicted mesh from the coarse reconstruction network. Specifically, we downsample the coarse mesh $\mathbf{S}_{coarse} \in \mathbb{R}^{53215\times3}$ to obtain a new mesh $\mathbf{S}_{new} \in \mathbb{R}^{13304\times3}$ with identical number of vertices as $\mathbf{f}_{mesh}$, after which, we project (and resize) $\mathbf{S}_{new}$ onto the 128×128 image plane to sample from feature tensor $\mathbf{f}_{im}$ using bilinear interpolation. Nevertheless, the extracted feature $\tilde{\mathbf{f}}_{im} \in \mathbb{R}^{13304\times16}$ cannot be used directly, as it comes from another domain, so we

design an extra layer to adapt this feature to the target domain. Adpative Instance Normalisation (AdaIN) [43] is chosen for this purpose. Essentially, AdaIN aligns the channel-wise mean and variance of the source features with those of the target feature (this simple approach has been shown effective in the task of style transfer). We normalise the extracted feature $\tilde{\mathbf{f}}_{im}$ as:

$$AdaIN(\tilde{\mathbf{f}}_{im}, \mathbf{f}_{mesh}) = \sigma(\mathbf{f}_{mesh}) \left(\frac{\tilde{\mathbf{f}}_{im} - \mu(\tilde{\mathbf{f}}_{im})}{\sigma(\tilde{\mathbf{f}}_{im})} \right) + \mu(\mathbf{f}_{mesh}), \tag{2}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are the channel-wise mean and variance, respectively. Note that we also tried to replace AdaIN with batch normalization [44], unfortunately, our networks fail to produce sensible results with it. Finally, we add the two features together and feed them into the next spiral convolution layer.

Loss function. As there does not exist detailed ground truth shape for images in-the-wild, we train the refinement network in an unsupervised manner using Shape-from-Shading (SfS) loss. SfS loss is defined as the $\mathcal{L}_2$ norm of the difference between the original intensity image and the reflected irradiance $\tilde{\mathbf{I}}$. According to [45, 5], $\tilde{\mathbf{I}}$ can be computed as:

$$\tilde{\mathbf{I}}(\mathbf{c}^*, \mathbf{N}, \mathbf{A}) = \mathbf{A} \sum_{i=1}^{9} \mathbf{c}_i^* \mathbf{Y}_i(\mathbf{N}), \tag{3}$$

where $\mathbf{N}$ is the unit normals of the predicted depth image and $\mathbf{A}$ is the albedo map of the target image (estimated using SfSNet [46]), $\mathbf{Y}_i(\mathbf{N})$ are the Spherical Harmonics (SH) basis functions computed from the predicted unit normals $\mathbf{N}$, and $\mathbf{c}^*$ are the second-order SH coefficients that can be precomputed using the original image intensity $\mathbf{I}$ and depth $\mathbf{N}_{gt}$ image:

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \|\mathbf{A} \sum_{i=1}^{9} \mathbf{c}_i \mathbf{Y}_i(\mathbf{N}_{gt}) - \mathbf{I}\|_2^2. \tag{4}$$

In practice, we calculate $\mathbf{N}_{gt}$ from the fitted coarse mesh. Different from [5, 18], our model predicts a 3D mesh rather than a depth image, therefore we need to render our final mesh to obtain the unit normals in image space. To achieve this, we first compute the vertex normals of our predicted mesh $\mathbf{S}_{final}$, and then we render the normals to the image using a differentiable renderer [47] to get the normal image $\mathbf{N}$. Our SfS refinement loss function can be written as:

$$\mathcal{L}_{refine} = \|\tilde{\mathbf{I}}(\mathbf{c}^*, \mathbf{N}, \mathbf{A}) - \mathbf{I}\|_2. \tag{5}$$

Our refinement loss accounts for the difference between the target and reconstructed image using shape-from-shading, and drives the refinement network to reconstruct fine geometric details.

3.4 Network architecture and training details

This section describes the training data and procedure. More details about the network architectures used are provided in the supplementary material.

Training data and pre-processing. We train the proposed networks using only 300W-LP database [8] which contains over 60K large-pose facial images synthesized from 300W database [48]. Although the *ground truth* 3D meshes of 300W-LP come from a conventional optimisation-based 3DMM fitting method, they can be used to provide a reliable estimation of the coarse target face, which is then refined by our unsupervised refinement network. Note that we randomly leave out around 10% of the data for validation purposes, and the rest of the data (around 55K images and meshes) are all used for training. For each image, we compute the face bounding box using the ground truth 3D mesh, and then use the bounding box to crop and resize the image to 256x256. During training, we apply several data augmentation techniques that are proven useful in [2]. They include random scaling from 0.85 to 1.15, random in-plane rotation from -45 to 45 degrees, and random 10% translation w.r.t image width and height.
Training procedure. Because the refinement network requires a reasonable estimation of the coarse mesh, the training of our model consists of two stages. Note that we always use the same training data. The first stage is to train the coarse face reconstruction network only. For this, we use SGD with momentum [49] with an initial learning rate of 0.05 and momentum value of 0.9. We train the coarse network for 120 epochs, and for every two epochs, we decay the learning rate by a ratio of 0.9. The second stage is to jointly train the coarse networks and refinement networks. We do not freeze any layers during this stage, and as our pipeline is fully differentiable, the encoder and the decoder of the coarse network can also adapt and improve with the extra SfS loss. During the second stage training, both $\mathcal{L}_{coarse}$ and $\mathcal{L}_{refine}$ are used to drive the network training. We found that no additional weight balance is needed between them. The second stage is also trained with SGD with momentum equal to 0.9, but the initial learning rate is 0.008. We train the whole network for 100 epochs. Similarly, for every two epochs, we decay the learning rate by a ratio of 0.9. All the models are trained using two 12GB NVidia GeForce RTX 2080 GPUs with Tensorflow [50].

4 Experiments

4.1 Evaluation databases

We evaluated the accuracy of our method on the following databases.
Florence. Florence [51] is a widely used database to evaluate face reconstruction quality. It contains 53 high-resolution recordings of different subjects, and the subjects only show neutral expression in the controlled environment recording.
BU3DFE. BU3DFE [52] is the first large scale 3D facial expression database. It contains a neutral face and 6 articulated expressions captured from 100 adults. Since there are 4 different intensities per expression per subject, a total of 2,500 meshes are provided. These 3D faces have been cropped and aligned beforehand.
4DFAB. 4DFAB [53] is the largest dynamic 3D facial expression database. It contains 1.8M 3D meshes captured from 180 individuals. The recordings capture

rich posed and spontaneous expressions. We used a subset of 1,482 meshes that display either neutral or spontaneous expressions from different subjects.
AFLW2000-3D. AFLW2000-3D [8] contains 68 3D landmarks of the first 2,000 examples from the AFLW database [54]. We used this database to evaluate our method on the task of sparse 3D face alignment.

4.2 Evaluation protocol

For each database, we generated test data by rendering ground truth textured mesh with different poses, *i.e.*, [-20°, 0°, 20°] for pitch, and [-80°, -40°, 0°, 40°, 80°] for yaw angles. Orthographic projection was used to project the rotated mesh. Each mesh produced 15 different facial renderings for testing. For each rendering, we also cast arbitrary light from a random direction with a random intensity to make it challenging. We selected the Normalised Mean Error(NME) to measure the accuracy of 3D reconstruction. It is defined as:

$$NME(\mathbf{S}_{pred}, \mathbf{S}_{gt}) = \frac{1}{n} \sum_{i \in \mathcal{S}_{gt}} \frac{\|\mathbf{S}^i_{pred} - \mathbf{S}^i_{gt}\|^2}{d_{occ}}, \tag{6}$$

where $\mathbf{S}_{pred}$ and $\mathbf{S}_{gt}$ are the predicted and ground truth 3D meshes correspondingly, n is the number of vertices, and d_{occ} is the outer interoccular distance. To provide a fair comparison for all the methods, we only use the visible vertices of $\mathbf{S}_{gt}$ to calculate the errors. We denote this set of vertices as $\mathcal{S}_{gt}$. Z-buffering is employed to determine the visibility. Since there is no point-to-point correspondence between $\mathbf{S}_{pred}$ and $\mathbf{S}_{gt}$, we apply Iterative Closest Points (ICP) [55, 56] to align $\mathbf{S}_{pred}$ to $\mathbf{S}_{gt}$ to retrieve the correspondence for each visible vertex in $\mathbf{S}_{gt}$. Note that we do not apply the full optimal transform estimated by ICP to the predicted mesh. This is because it is important to test whether each method can correctly predict the target's global pose.

4.3 Ablation study

For our ablation study, we trained different variants of our method and tested them on the Florence dataset [51]. The results are shown in Fig. 3 in the form of cumulative error curves (CEDs) and NMEs. We start by training a variant of our method that only contains the first GCN for coarse mesh reconstruction. We then train a second variant by adding the second GCN for detailed reconstruction (which we dubbed "SfS" in the figure), and, finally, we add the image-level feature sampling and adaptation layers for additional facial detail injection ("skip" in the figure). The latter represents the full version of our method.

As illustrated in Fig. 3 (left), adding each new component enables the model to achieve higher accuracy. In Fig. 3 (right), we also show examples of the mesh reconstructed by the aforementioned variants and demonstrate that each component in our method can indeed boost the model's capability in reconstructing fine details (notice the differences in wrinkles). Last but not least, as shown in

Fig. 3, switching the image encoder backbone to MobleNet V2 resulted in a small drop in accuracy but it can also drastically decrease the model size and inference time, as shown in Tab. 2.

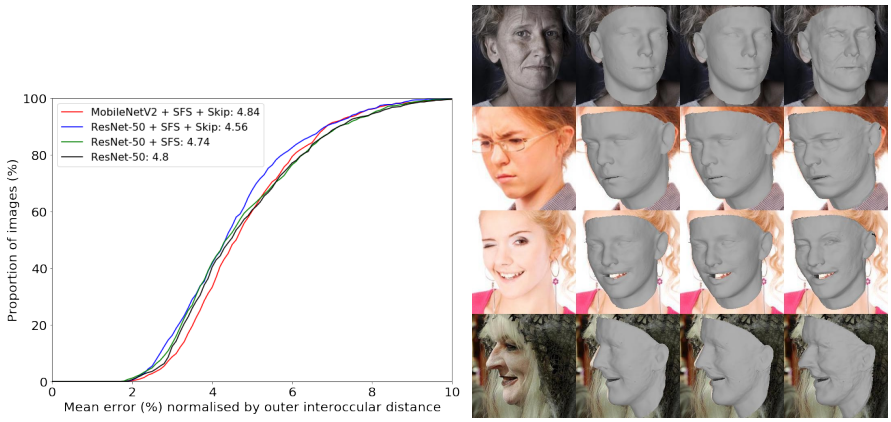


Fig. 3. Ablation study on the reconstruction performance of our proposed method, performed on Florence database [51]. Left shows the CED curves and NMEs of the variants. Right shows some examples they produced. From left to right, we show: the input images and outputs given by the baseline GCN model with ResNet50 image encoder, ResNet50+SfS, and ResNet50+SfS+Skip (the full model), respectively.

4.4 3D face alignment results

Our results on the AFLW2000-3D [8] dataset are shown in Tab. 1. Our approach, when using ResNet50 as the image encoder, significantly outperformed all other methods. Even after switching to the much lighter MobileNet-v2 as the encoder backbone, our method still achieved very good accuracy which is only slightly worse than that of PRNet [2], the next best-performing method.

Table 1. Face alignment results on the AFLW2000-3D dataset, we reported the average mean error (%) normalised by the face bounding box.

Method	N3DMM[38]	3DDFA[8]	PRNet[2]	CMD[3]	Ours (MobileNetV2)	Ours (ResNet50)
NME	4.12	3.79	3.62	3.98	3.65	**3.39**

4.5 3D face reconstruction results

Our results on Florence [51], BU3DFE [52], and 4DFAB [53] are shown (from left to right) in Fig. 4. On each dataset, we show both the CED curves com-

puted from all test examples (top) and the pose-specific NMEs (bottom). As the figure shows, our method (with ResNet50) performed the best in all three test datasets. When switching to MobileNet-v2, our method still outperformed all other methods. The NME of our method is also consistently low across all poses, demonstrating the robustness of our approach. This is in contrast to Pix2Vertex [18] and DF^{2}Net [5], which performed relatively well when the face is at a frontal pose but significant worse for large-pose cases. For PRNet [2], 3DDFA [8], and VRN[1], although they achieved decent quantitative results (in terms of CED and NMEs), they lack the ability to reconstruct fine facial details.

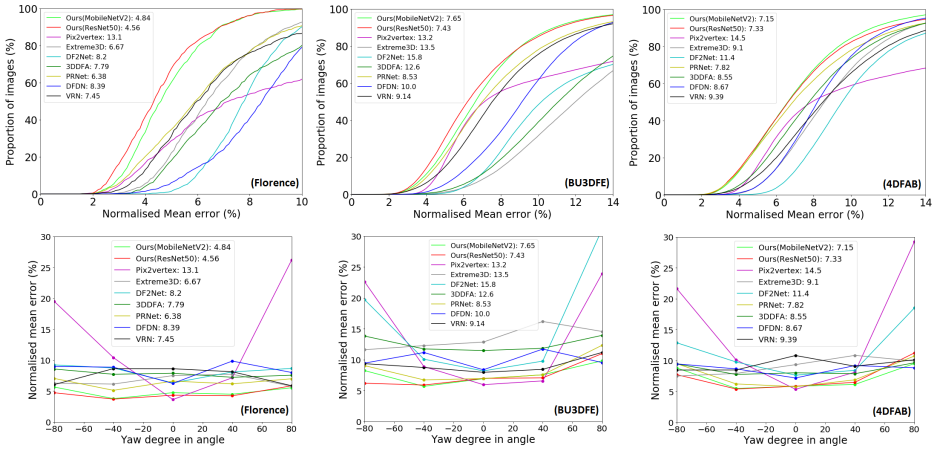


Fig. 4. 3D face reconstruction results on, from left to right Florence [51], BU3DFE [52], and 4DFAB [53] datasets. In each column, the top row shows the CED curves and NMEs of various methods, while the bottom row shows the pose-wise NMEs.

4.6 Qualitative evaluation

Fig. 5 shows qualitative reconstruction results produced by our method (with a ResNet50 encoder) and other competitive methods. In particular, we compare against Extreme3D, PRNet and DF^{2}Net (comparisons with more methods are provided in the supplementary material). The first two methods are among the best performing in our quantitative evaluations while the latter is one of the best methods for reconstructing fine details. From the figure, we observe that our method is the best being both robust and able to capture fine facial details at the same time.

4.7 Comparisons of inference speed and model size

We compare the inference speed and model size of our approach to previous methods. The tests were conducted on a machine with an Intel Core i7-7820X

CPU @3.6GHz, a GeForce GTX 1080 graphics card, and 96GB of main memory. For all methods (for CMD [3], no available implementation exists, so we used the result from their paper), we used the implementation provided by the original authors. For more details see supplementary material. As most methods consist of multiple stages involving more than one model, for a fair comparison, we report the *end-to-end* inference time and *total size* of all models (i.e., weights of networks, basis of 3DMMs, etc.) that are needed to estimate the face mesh from an input facial image. As shown in Tab. 2, our approach is among the fastest, taking only 10.8 ms / 6.2 ms (when using ResNet50 / MobileNet v2, respectively) to reconstruct a 3D face. Our method also has the smallest model size when using MobileNet-v2 as the image encoder.

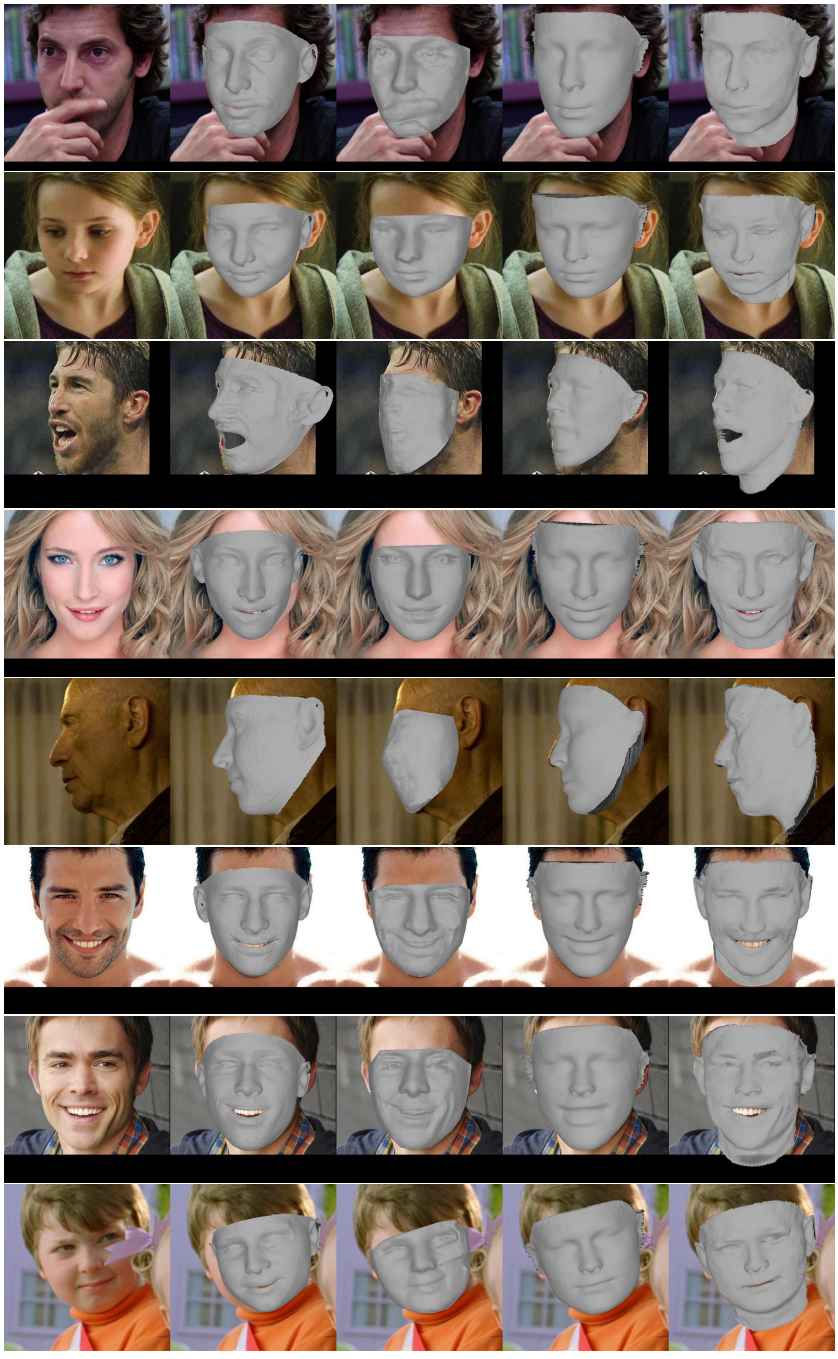
Table 2. Inference speed and model size comparison.

Method	Inference speed (ms per sample)		Total model size (MB)
	Main model(s)	Post-processing	
DF^{2}Net[5]	40.4		222
Extreme3D[23]	230.9	14328.9	503
DFDN[19]	38762.7		1982
3DDFA[8, 57]	6.7		45
PRNet[2]	19.4		153
VRN[1]	16.4	220.2	1415
Pix2vertex[18]	40.0	248016.5	1663
CMD[3]	**3.1**		93
Ours (ResNet50)	10.8		209
Ours (MobileNetV2)	6.2		**37**

5 Conclusions

We presented a robust, lightweight and detailed 3D face reconstruction method. Our framework consists of 3 key components: (a) a lightweight non-parametric GCNs decoder to reconstruct coarse facial geometry from image encoder; (b) a lightweight GCNs model to refine the output of the first network in an unsupervised manner; (c) a novel feature-sampling mechanism and adaptation layer which injects fine details from the image encoder into the refinement network. To our knowledge, we are the first to reconstruct high-fidelity facial geometry relying solely on GCNs. We compared our method with 7 state-of-the-art methods on Florence, BU3DFE and 4DFAB datasets, and reported state-of-the-art results for the experiments, both quantitatively and quantitatively. We also compared the speed and model size of our method against other methods, and showed that it can run faster than real-time, while at the same time, being extremely lightweight (with MobileNet-V2 as backbone, our model size is 37MB).

For CMD [3], we just show the values reported in their paper as the authors did not release their code and model.



(1) Image (2) Extreme3D (3) DF^{2}Net (4) PRNet (5) Ours

Fig. 5. Qualitative comparisons.

References

1. Jackson, A.S., Bulat, A., Argyriou, V., Tzimiropoulos, G.: Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 1031–1039
2. Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X.: Joint 3d face reconstruction and dense alignment with position map regression network. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 534–551
3. Zhou, Y., Deng, J., Kotsia, I., Zafeiriou, S.: Dense 3d face decoding over 2500fps: Joint texture & shape convolutional mesh decoders. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 1097–1106
4. Richardson, E., Sela, M., Or-El, R., Kimmel, R.: Learning detailed face reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 1259–1268
5. Zeng, X., Peng, X., Qiao, Y.: Df2net: A dense-fine-finer network for detailed 3d face reconstruction. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 2315–2324
6. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. (1999) 187–194
7. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D face model for pose and illumination invariant face recognition. In: IEEE AVSS. (2009)
8. Zhu, X., Lei, Z., Li, S.Z., et al.: Face alignment in full pose range: A 3d total solution. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
9. Tuan Tran, A., Hassner, T., Masi, I., Medioni, G.: Regressing robust and discriminative 3d morphable models with a very deep neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 5163–5172
10. Dou, P., Shah, S.K., Kakadiaris, I.A.: End-to-end 3d face reconstruction with deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 5908–5917
11. Tewari, A., Zollhofer, M., Kim, H., Garrido, P., Bernard, F., Perez, P., Theobalt, C.: Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. (2017) 1274–1283
12. Tewari, A., Zollhöfer, M., Garrido, P., Bernard, F., Kim, H., Pérez, P., Theobalt, C.: Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 2549–2559
13. Genova, K., Cole, F., Maschinot, A., Sarna, A., Vlasic, D., Freeman, W.T.: Unsupervised training for 3d morphable model regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 8377–8386
14. Tran, L., Liu, X.: Nonlinear 3d face morphable model. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7346–7355
15. Tran, L., Liu, F., Liu, X.: Towards high-fidelity nonlinear 3d face morphable model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 1126–1135
16. Kolotouros, N., Pavlakos, G., Daniilidis, K.: Convolutional mesh regression for single-image human shape reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 4501–4510

17. Patel, A., Smith, W.A.: Driving 3d morphable models using shading cues. Pattern Recognition **45** (2012) 1993–2004
18. Sela, M., Richardson, E., Kimmel, R.: Unrestricted facial geometry reconstruction using image-to-image translation. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 1576–1585
19. Chen, A., Chen, Z., Zhang, G., Mitchell, K., Yu, J.: Photo-realistic facial details synthesis from single image. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 9429–9439
20. Garrido, P., Valgaerts, L., Wu, C., Theobalt, C.: Reconstructing detailed dynamic face geometry from monocular video. ACM Trans. Graph. **32** (2013) 158–1
21. Li, Y., Ma, L., Fan, H., Mitchell, K.: Feature-preserving detailed 3d face reconstruction from a single image. In: Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production. (2018) 1–9
22. Roth, J., Tong, Y., Liu, X.: Unconstrained 3d face reconstruction. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 2606–2615
23. Tran, A.T., Hassner, T., Masi, I., Paz, E., Nirkin, Y., Medioni, G.G.: Extreme 3d face reconstruction: Seeing through occlusions. In: CVPR. (2018) 3935–3944
24. Jiang, L., Zhang, J., Deng, B., Li, H., Liu, L.: 3d face reconstruction with geometry details from a single image. IEEE Transactions on Image Processing **27** (2018) 4756–4770
25. Abrevaya, V.F., Boukhayma, A., Torr, P.H., Boyer, E.: Cross-modal deep face normals with deactivable skip connections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 4979–4989
26. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: NIPS. (2016)
27. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR. (2017)
28. Klicpera, J., Weißenberger, S., Günnemann, S.: Diffusion improves graph learning. In: Conference on Neural Information Processing Systems (NeurIPS). (2019)
29. Lim, I., Dielen, A., Campen, M., Kobbelt, L.: A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 0–0
30. Fey, M., Lenssen, J.E., Weichert, F., Müller, H.: Splinecnn: Fast geometric deep learning with continuous B-spline kernels. In: CVPR. (2018)
31. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
32. Bai, S., Zhang, F., Torr, P.H.: Hypergraph convolution and hypergraph attention. arXiv preprint arXiv:1901.08150 (2019)
33. Verma, N., Boyer, E., Verbeek, J.: Feastnet: Feature-steered graph convolutions for 3d shape analysis. In: CVPR. (2018)
34. Bouritsas, G., Bokhnyak, S., Ploumpis, S., Bronstein, M., Zafeiriou, S.: Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 7213–7222
35. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3d faces using convolutional mesh autoencoders. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 704–720
36. Litany, O., Bronstein, A., Bronstein, M., Makadia, A.: Deformable shape completion with graph convolutional autoencoders. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 1886–1895

37. Cheng, S., Bronstein, M., Zhou, Y., Kotsia, I., Pantic, M., Zafeiriou, S.: Meshgan: Non-linear 3d morphable models of faces. arXiv preprint arXiv:1903.10384 (2019)
38. Tran, L., Liu, X.: On learning 3d face morphable model from in-the-wild images. IEEE transactions on pattern analysis and machine intelligence (2019)
39. Sanyal, S., Bolkart, T., Feng, H., Black, M.J.: Learning to regress 3d face shape and expression from an image without 3d supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 7763–7772
40. Tewari, A., Bernard, F., Garrido, P., Bharaj, G., Elgharib, M., Seidel, H.P., Pérez, P., Zollhofer, M., Theobalt, C.: Fml: Face model learning from videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 10812–10822
41. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
42. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. arXiv preprint arXiv:1801.04381 (2018)
43. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 1501–1510
44. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
45. Ramamoorthi, R., Hanrahan, P.: An efficient representation for irradiance environment maps. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. (2001) 497–500
46. Sengupta, S., Kanazawa, A., Castillo, C.D., Jacobs, D.W.: Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 6296–6305
47. Henderson, P., Ferrari, V.: Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. International Journal of Computer Vision (2019) 1–20
48. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: A semi-automatic methodology for facial landmark annotation. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. (2013) 896–903
49. Qian, N.: On the momentum term in gradient descent learning algorithms. Neural networks **12** (1999) 145–151
50. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). (2016) 265–283
51. Bagdanov, A.D., Masi, I., Del Bimbo, A.: The florence 2d/3d hybrid face datset. In: Proc. of ACM Multimedia Int.'l Workshop on Multimedia access to 3D Human Objects (MA3HO'11). (2011)
52. Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M.J.: A 3d facial expression database for facial behavior research. In: 7th international conference on automatic face and gesture recognition (FGR06), IEEE (2006) 211–216
53. Cheng, S., Kotsia, I., Pantic, M., Zafeiriou, S.: 4dfab: A large scale 4d database for facial expression analysis and biometric applications. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 5117–5126

54. Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: 2011 IEEE international conference on computer vision workshops (ICCV workshops), IEEE (2011) 2144–2151
55. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. International journal of computer vision **13** (1994) 119–152
56. Cheng, S., Marras, I., Zafeiriou, S., Pantic, M.: Statistical non-rigid icp algorithm and its application to 3d face alignment. Image and Vision Computing **58** (2017) 3–12
57. Jianzhu Guo, X.Z., Lei, Z.: 3ddfa. https://github.com/cleardusk/3DDFA (2018)