Implicit Neural Deformation for Multi-View Face Reconstruction

Moran Li[*†] **Haibin Huang**[*] **Yi Zheng**[*] **Mengtian Li**[*] **Nong Sang**[†] **Chongyang Ma**[*]

[*] Kuaishou Technology [†] Huazhong University of Science and Technology

Abstract

In this work, we present a new method for 3D face reconstruction from multi-view RGB images. Unlike previous methods which are built upon 3D morphable models (3DMMs) with limited details, our method leverages an implicit representation to encode rich geometric features. Our overall pipeline consists of two major components, including a geometry network, which learns a deformable neural signed distance function (SDF) as the 3D face representation, and a rendering network, which learns to render on-surface points of the neural SDF to match the input images via self-supervised optimization. To handle in-the-wild sparse-view input of the same target with different expressions at test time, we further propose residual latent code to effectively expand the shape space of the learned implicit face representation, as well as a novel view-switch loss to enforce consistency among different views. Our experimental results on several benchmark datasets demonstrate that our approach outperforms alternative baselines and achieves superior face reconstruction results compared to state-of-the-art methods.

1 Introduction

In this paper, we tackle the problem of 3D face reconstruction given multi-view input, *i.e.*, to generate a textured face mesh based on a set of RGB images taken from different views. This problem is long-standing in both computer vision and computer graphics with many real-world applications, such as portrait manipulation and augmented/virtual reality.

Compared with reconstruction from a single RGB image or RGBD input, multi-view face reconstruction is a more practical setting with recent development of mobile devices, since it does not require additional depth senor but still provides rich information from different views about the target. Previous methods [4, 7] propose to reconstruct 3D faces under controlled environments, where the multi-view images are captured from well-calibrated camera arrays with fixed lighting. Although these methods can successfully produce high-fidelity 3D face models, their usage scenarios are quite limited due to the complex hardware tuning and their performances downgrade significantly for general inputs. To address these drawbacks, some recent approaches [2, 3, 39] exploit 3DMMs [6, 32, 55] together with multi-view algorithms to leverage cross-view geometry consistency and reveal promising improvements.However, those methods are built upon 3DMMs or its variants, where the number of vertices is limited and the topology is fixed.Therefore, it remains challenging to generate a faithful 3D face with high-quality details from multi-view input, especially in an uncontrolled setting or when the input views are of different expressions.

In this work, our focus is to improve the generalization performance as well as the quality of sparse-view 3D face reconstruction by learning an implicit neural representation. Our key insight is that, unlike 3DMMs that are limited by a pre-defined shape space, implicit functions such as SDFs can represent surfaces with arbitrary resolution and topology [22, 31, 49]. To this end, we propose to learn a geometry network that serves as a neural SDF for reconstruction of the target 3D face. Specifically, the proposed geometry network consists of two sub-modules, *i.e.*, a *reference network*

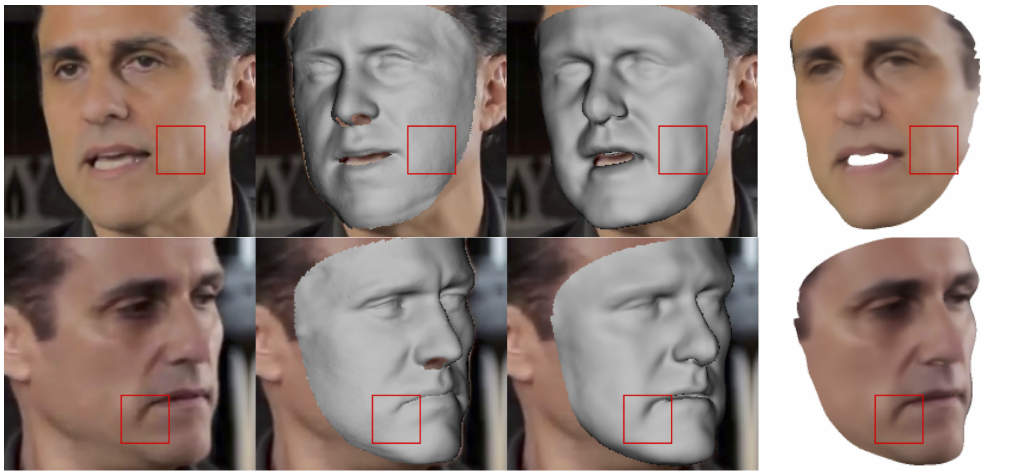


Figure 1: Face reconstruction results of DFNRMVS [2] and our method from two-view inputs. The last two columns are our reconstructed geometry and the rendering results. Our method captures more local details (red boxes).

and a *deformation network*. The reference network is trained offline to learn the SDF of a mean face given the training set and provides an initiation of SDFs for optimization at the test time. The deformation network is then applied and learns to deform the SDFs which generates local details and changes topology if necessary. Our experiments show that such a decomposition effectively leverages a 3D face prior to enhance the generalization capability of neural networks and prevents the neural SDFs from collapsing or distorting during optimization with limited views (*e.g.*, 2$\sim$4 views).

Inspired by [49], we further present a neural rendering network based on a self-supervised optimization procedure. This module learns to render on-surface points sampled from the implicit 3D face geometry. The self-supervision is achieved by minimizing the difference between rendered colors and the corresponding input images. Additionally, we exploit several latent codes (*i.e.*, geometry and color latent codes) to encode different geometry and texture information among different instances to enhance the generalization ability of the trained network. To expand the shape space of the learned neural SDF, we introduce *residual latent code* at test time. Furthermore, we design a *view-switch loss* via exchanging the latent code among different views and minimizing the rendering loss to enforce consistency across different views. As a result, our method can reconstruct 3D faces from sparse-view input with high-fidelity details. See Figure 1 as an example of face reconstruction from two in-the-wild images of the same person but with different expressions.

To summarize, the main contributions of this works are as follows:

- We present a novel pipeline for 3D face reconstruction from sparse-view input, including a geometry network to learn a deformable implicit neural representation for 3D shape and a rendering network to model the facial texture.
- We propose a novel view-switch loss as well as a newly designed latent code space of the implicit morphable model. These two terms help expand the underlying shape space and enforce cross-view consistency at test time.
- We conduct both qualitative and quantitative evaluations on benchmark datasets to demonstrate that our method outperforms baseline approaches and state-of-the-art face reconstruction algorithms.

2 Related Work

The literature on 3D face reconstruction is vast and the algorithm input ranges from depth map [24], single image [10, 13, 17, 20, 26, 34, 39, 46, 54], to multi-view images [2, 3, 12, 36, 47], or videos [19, 45]. Since our main focus is 3D face reconstruction from multi-view images using neural SDFs as the geometric representation, in this section we briefly review 3D morphable models, multi-view 3D face reconstruction methods, and the most related implicit neural representations.

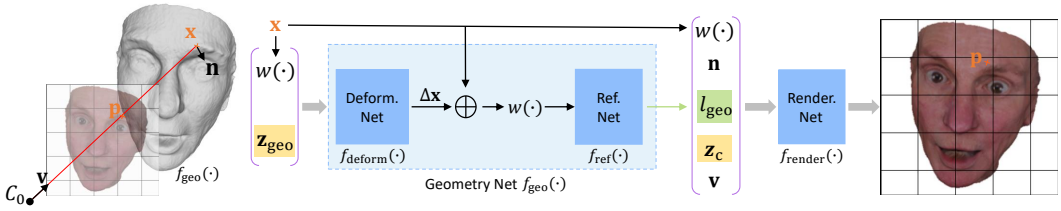


Figure 2: Overview of the proposed method. Our method consists of the *Geometry Network* and *Rendering Network*. And the *Geometry Network* can be decomposed into a *Deformation Network* and a *Reference Network*. $\mathbf{v}$ is the view direction from the camera center C_0 to the randomly sampled pixel $\mathbf{p}$. $\mathbf{x}$ is the corresponding on-surface point, and $\mathbf{n}$ is the normal vector. $\mathbf{z}_{\text{geo}}, \mathbf{z}_{\mathbf{c}}$ are the geometry and color latent codes, respectively. l_{geo} is the feature from Geometry Network (*i.e.*, $f_{\text{geo}}(\cdot)$). $w(\cdot)$ is the positional encoding function. $(\ )$ denotes the concatenate operation. For conciseness, we only present one view of the instance, while most of our experiments are multi-view inputs.

Face morphable models. The well-known 3D morphable model [6, 8, 32, 55] is a bilinear parametric method that decomposes the face geometry/texture into a template and a deformation component with respect to this template based on principal component analysis (PCA). Due to their simplicity and effectiveness, 3DMMs are widely used in faces reconstruction and animation. However, the capability of such models is limited by the basis of PCA. Even though several recent methods (*e.g.*, [8, 41, 48]) propose to extend the face basis with more 3D face scans from larger datasets, the geometry or texture space of those methods is still a subspace of real-world face space. For a complete report of 3D morphable face, we refer to [15, 56].

Multi-view face reconstruction. Existing learning-based algorithms of 3D face reconstruction from multi-view RGB images can be roughly categorized into supervised methods [2, 9, 19] and self-supervised methods [12, 36, 47]. [19] exploits parametric geometry prior information to learn a plausible coarse face mesh and fine-scale details are captured via shading-based refinement from videos. [2] proposes to expand the basis of 3DMMs via adaptive optimization to improve the representation of such parametric models and enforce multi-view consistency. To alleviate the requirement of large-scale 3D scan datasets, some researchers tackle this problem in a self-supervised manner. [12] uses aggregated complementary information among different images to achieve multi-view reconstruction. However, those models are built upon 3DMMs, where the mesh topology is fixed and cannot represent high-frequency details easily.

Implicit neural representation. In recent years, methods based on implicit neural representations are emerging for shapes [1, 11, 21, 22, 31, 43] and scenes [16, 23, 27, 40]. The seminal work DeepSDF [31] encodes a category of shapes into a neural network, and the specific features of each instance are encoded into a latent code. Based on DeepSDF, [14] proposes a curriculum architecture to enhance the quality of the reconstructed shape. Those methods are used to obtain the implicit neural representations of shape, objects, or scenes with 3D data (*e.g.*, point cloud) as the supervision. Inspired by those methods, [50] build implicit 3D morphable models for human heads with hair from a collected 3D scans dataset. [35] exploit pixel-aligned implicit function to estimate the surface of human subjects and the corresponding texture. Recently, [25, 30, 49, 51] are proposed for novel view synthesis from a set of images, where the key idea is to reconstruct the underlying 3D scene/object geometry and the neural radiance field at the same time.

3 Our Method

3.1 Face Representation and Problem Statement

In an implicit neural representation based on signed distance field (SDF), the geometry of a 3D face can be represented as the zero level set of a scalar valued network f:

$$S_\theta = \left\{\mathbf{x} \in \mathbb{R}^3 | f_\theta(\mathbf{x}) = 0\right\} \tag{1}$$

where the neural network $f_\theta(\mathbf{x})$ gives the signed shortest distance s of a 3D query point $\mathbf{x} \in \mathbb{R}^3$ to the face geometry S_θ and $\theta \in \mathbb{R}^m$ are learnable parameters of the network f. To model the facial texture, we extend the network output to include a vector value $\mathbf{c} \in \mathbb{R}^3$, which represents the RGB color of the closest point on the face to the query point.

To further model various faces of different identities and expressions, in our framework, we introduce a latent code $\mathbf{z}$ to represent the face instance in a portrait image. Following i3DMM [50], we denote the network as $f_\theta(\mathbf{x},\, \mathbf{z})$ to take this latent code as additional input.

In both the training and test stages, we jointly optimize the network parameters θ and the latent code $\mathbf{z}$ as described in detail below, in order to obtain the desired morphable model and the corresponding implicit representation of each face instance. To simplify the notation, we omit θ in the subscript and rewrite the network as $f(\mathbf{x},\, \mathbf{z}) = \{s,\, \mathbf{c}\}$.

3.2 Network Components

As illustrated in Figure 2, our overall framework consists of two network components, *i.e.*, a Geometry Network f_{geo} and a Rendering Network f_{render}. Accordingly, the latent code $\mathbf{z}$ of each face instance can be decomposed into two parts, *i.e.*, a geometry code $\mathbf{z}_{\text{geo}}$ and a color code $\mathbf{z}_\mathbf{c}$, which are used as input of f_{geo} and f_{render}, respectively.

Geometry network. Our Geometry Network f_{geo} is a scalar valued function to model the implicit 3D face shape. We follow i3DMM [50] to further decompose f_{geo} into two successive components, *i.e.*, a Reference Network f_{ref} to learn an implicit reference shape, and a Deformation Network f_{deform} to predict deformation offset $\Delta\mathbf{x}$ conditioned on the reference shape. The Reference Network can be considered as a neural version of the mean face in traditional 3DMM [5], while the Deformation Network models the per-instance variations from the mean face. As a result, the Geometry Network can be formulated as:

$$\begin{aligned} f_{\text{geo}}(\mathbf{x}, \mathbf{z}_{\text{geo}}) &= f_{\text{ref}}(\mathbf{x} + \Delta\mathbf{x}) \\ &= f_{\text{ref}}\big(\mathbf{x} + f_{\text{deform}}(\mathbf{x}, \mathbf{z}_{\text{geo}})\big) \end{aligned} \tag{2}$$

Rendering network. Our Rendering Network f_{render} is introduced to model the face texture in a self-supervised manner, where the texture information is encoded as the color latent code $\mathbf{z}_\mathbf{c}$ for each face instance. Therefore, for a given surface point $\mathbf{x}$ of a certain instance, the RGB value $\mathbf{c}(\mathbf{x},\, \mathbf{z}_\mathbf{c})$ can be modeled using our Rendering Network by taking several factors into account together:

$$\mathbf{c}(\mathbf{x},\, \mathbf{z}_\mathbf{c}) = f_{\text{render}}(\mathbf{x}, \mathbf{z}_\mathbf{c}, \mathbf{n}, \mathbf{v}, l_{\text{geo}}), \tag{3}$$

where $\mathbf{n}$ is the surface normal, $\mathbf{v}$ is the view direction, and $l_{\text{geo}} \in \mathbb{R}^{256}$ are geometric features computed as additional output by the Geometry Network f_{geo}. Note that the gradient of the signed distance computed by Geometry Network f_{geo} at a point $\mathbf{x}$ is the corresponding surface normal, *i.e.*, $\mathbf{n} = \nabla f_{\text{geo}}(\mathbf{x}, \mathbf{z}_{\text{geo}})$.

3.3 Network Training

Dataset. We use a training partition of the Stirling/ESRC [42] dataset to train our network, which contains more than 700 registered 3D face scans of about 95 subjects. To prepare training data for the Geometry Network f_{geo}, the 3D scans are scaled to fit into a unit bounding box and then aligned to the same orientation. We then randomly sample 860K on-surface points from each registered scan in a uniform distribution.We consider these points together with the corresponding normals as the zero level set of each SDF and use them to train the Geometry Network. To prepare training data for the Rendering Network f_{render}, we render about 40 RGB images of each 3D face scan from random view directions. For each rendered image, we also compute a binary mask to represent the face region.

Geometry loss function. Given a face instance i with a geometry latent code $\mathbf{z}^i_{\text{geo}}$ and a set of sample points Ω_I, the overall geometry loss function $\mathcal{L}_{\text{geo}}$ is computed as:

$$\mathcal{L}_{\text{geo}}(\mathbf{z}^i_{\text{geo}}) = \lambda_{\text{I}}\mathcal{L}_{\text{I}} + \lambda_{\text{d}}\mathcal{L}_{\text{d}} + \lambda_{\text{eik}}\mathcal{L}_{\text{eik}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} \tag{4}$$

where $\lambda_{\text{I}}, \lambda_{\text{d}}, \lambda_{\text{eik}}, \lambda_{\text{reg}}$ are hyperparameters to balance different loss terms. We set $\lambda_{\text{I}} = 1, \lambda_{\text{d}} =$ 1e-2, $\lambda_{\text{eik}} = 0.1$ and $\lambda_{\text{reg}} =$ 1e-4 in our experiments.

In Eq. (4), $\mathcal{L}_{\text{I}}$ is a reconstruction loss to enforce the signed distance values of sampled on-surface points are close to zero and the normals of those points are close to the ground truth values:

$$\begin{aligned}\mathcal{L}_{\text{I}} = \frac{1}{|\Omega_I|} \sum_{\mathbf{x}_j \in \Omega_I} (&|f_{\text{geo}}(\mathbf{x}_j, \mathbf{z}^i_{\text{geo}})| \\ &+ \lambda_{\text{n}} \left\| \nabla f_{\text{geo}}(\mathbf{x}_j, \mathbf{z}^i_{\text{geo}}) - \hat{\mathbf{n}}_j \right\|),\end{aligned} \tag{5}$$

where $|\cdot|$ is the L1 norm, $\|\cdot\|$ is the L2 norm, $\Omega_I = \{\mathbf{x}_j\}_{j \in I}$ is a randomly sampled set of the on-surface points, and $\hat{\mathbf{n}}_j$ is the ground truth surface normal of the on-surface point $\mathbf{x}_j$. We set $\lambda_{\text{n}} = 1$ in our experiments. $\mathcal{L}_{\text{d}}$ in Eq. (4) is the regularization of the deformation offset:

$$\mathcal{L}_{\text{d}} = \frac{1}{|\Omega_I|} \sum_{\mathbf{x}_j \in \Omega_I} \left\| \Delta \mathbf{x}^i_j \right\| = \frac{1}{|\Omega_I|} \sum_{\mathbf{x}_j \in \Omega_I} \left\| f_{\text{deform}}(\mathbf{x}_j, \mathbf{z}^i_{\text{geo}}) \right\|, \tag{6}$$

and $\mathcal{L}_{\text{reg}} = \|\mathbf{z}_{\text{geo}}\|$ is the regularization for the geometry latent code. Finally, $\mathcal{L}_{\text{eik}}$ is the Eikonal term to avoid universe zero and ensures that f_{geo} approximates valid SDFs [22, 49]:

$$\mathcal{L}_{\text{eik}} = \frac{1}{|\Omega_D|} \sum_{\mathbf{x}'_j \in \Omega_D} \left(\left\| \nabla f_{\text{geo}}(\mathbf{x}'_j, \mathbf{z}^i_{\text{geo}}) \right\| - 1 \right)^2, \tag{7}$$

where $\Omega_D = \{\mathbf{x}'_j\}_{j \in D}$ is a set of points sampled from a uniform distribution within a unit bounding box.

Given N face instances within a mini-batch, we can jointly optimize the parameters θ_{geo} of the Geometry Network f_{geo} and the geometry latent codes $\{\mathbf{z}^i_{\text{geo}}, i = 1, \ldots, N\}$ of these N instances by solving the optimization problem below:

$$\underset{\theta_{\text{geo}},\, \{\mathbf{z}^i_{\text{geo}}\}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{geo}}(\mathbf{z}^i_{\text{geo}}) \tag{8}$$

Rendering loss function. Given a pair of a rendered RGB image and the corresponding face mask, we randomly sample a subset of pixels $\mathcal{P}$ in the image plane and use the following rendering loss function $\mathcal{L}_{\text{render}}$ to train our Rendering Network f_{render}:

$$\mathcal{L}_{\text{render}} = \tau_{\text{rgb}} \mathcal{L}_{\text{rgb}} + \tau_{\text{mask}} \mathcal{L}_{\text{mask}} + \tau_{\text{d}} \mathcal{L}_{\text{d}} + \tau_{\text{eik}} \mathcal{L}_{\text{eik}} + \tau_{\text{reg}} \mathcal{L}'_{\text{reg}} \tag{9}$$

where τ_{rgb}, τ_{mask}, τ_{d}, τ_{eik} and τ_{reg} are set to 1, 100, 1e-4, 1e-2, and 1e-4 to balance different loss terms. $\mathcal{L}'_{\text{reg}} = \|\mathbf{z}_{\text{geo}}\| + \|\mathbf{z}_{\text{c}}\|$. In Eq. (9), the loss terms $\mathcal{L}_{\text{d}}$ and $\mathcal{L}_{\text{eik}}$ are similar to those in Eq. (4), while $\mathcal{L}_{\text{rgb}}$ is the RGB reconstruction loss and $\mathcal{L}_{\text{mask}}$ is the mask loss, respectively.

Specifically, the RGB reconstruction loss is computed as:

$$\mathcal{L}_{\text{rgb}} = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} |\mathbf{c}_{\mathbf{p}} - \hat{\mathbf{c}}_{\mathbf{p}}| \tag{10}$$

where $\mathbf{c}_{\mathbf{p}}$ is the RGB value at the pixel $\mathbf{p}$ predicted by the Rendering Network, and $\hat{\mathbf{c}}_{\mathbf{p}}$ is the corresponding ground truth RGB value. We use Cross-Entropy loss to compute the mask loss term $\mathcal{L}_{\text{mask}}$ as below:

$$\mathcal{L}_{\text{mask}} = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} \text{CE}(m_{\mathbf{p}}, \hat{m}_{\mathbf{p}}) \tag{11}$$

where $m_{\mathbf{p}}$ and $\hat{m}$ are the predicted and the ground truth mask values at the pixel $\mathbf{p}$, respectively. As in [49], we use a sigmoid function to obtain a differentiable mask rendering.

Training strategy. Reconstruction 3D face geometry from a sparse set of RGB inputs (*i.e.*, $2 \sim 4$ view) with various expressions is an ill-posed problem. Besides, the proposed self-supervision is achieved by enforcing a similarity between the rendered RGB values and the ground truth RGB values. This will make the Rendering Network tend to overfit the input RGB images and the implicit neural geometry will collapse. To alleviate this problem, we first optimize the Geometry Network with the geometry loss $\mathcal{L}_{\text{geo}}$ to obtain a good initialization. Then, we jointly optimize the Rendering Network and the Geometry Network via the rendering loss $\mathcal{L}_{\text{render}}$.

3.4 Test-Time Reconstruction

Estimation of camera parameters. To handle in-the-wild images at test time, we estimate camera parameters by optimizing the L1 distance between the projected on-surface point $\mathbf{x}$ and the ground-truth pixel location $\mathbf{p}$:

$$\arg\min_{\mathbf{K},\mathbf{R}} |\mathbf{K}\mathbf{R}\mathbf{x}_1^\top - \mathbf{p}| \tag{12}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the camera intrinsic parameter, and $\mathbf{R} \in \mathbb{R}^{3\times4}$ is the rotation matrix. $\mathbf{x}_1 = [\mathbf{x}, 1] \in \mathbb{R}^{1\times4}$ is the homogeneous coordinates of the on-surface point $\mathbf{x}$.

In our method, the on-surface point is defined as the first intersection point of the ray across the pixel $\mathbf{p}$ and the face geometry S_θ. The intersection point can be represented as a differentiable function of the implicit geometry and camera parameters. We use the differentiable sphere-tracing method [28] to find the on-surface point.

Residual latent code. Given a sparse set of RGB images with various expressions of an instance, it is hard to learn the latent code directly with the rendering loss described above. Hence, we use principal component analysis (PCA) on the learned latent codes of the training set and infer the weights of those PCA basis at test time. Specifically, the latent code of instance i (*i.e.*, $\mathbf{z}^i_{\text{geo}}, \mathbf{z}^i_{\mathbf{c}}$) can be represented as the weighted sum of the principal components, such as:

$$\begin{aligned} \mathbf{z}^i_{\text{geo}} &= \bar{\mathbf{z}}_{\text{geo}} + W^i_{\text{geo}} B_{\text{geo}}, \\ \mathbf{z}^i_{\mathbf{c}} &= \bar{\mathbf{z}}_{\mathbf{c}} + W^i_{\mathbf{c}} B_{\mathbf{c}} \end{aligned} \tag{13}$$

where $W^i_{\text{geo}} \in \mathbb{R}^{1\times m_{\text{geo}}}, W^i_{\mathbf{c}} \in \mathbb{R}^{1\times m_{\mathbf{c}}}$ are the weights to combine the basis of geometry and color latent codes, respectively. $\bar{\mathbf{z}}_{\text{geo}} \in \mathbb{R}^{d_{\text{geo}}}, \bar{\mathbf{z}}_{\mathbf{c}} \in \mathbb{R}^{d_{\mathbf{c}}}$ are the mean latent code for the identity and color, respectively. $B_{\text{geo}} \in \mathbb{R}^{m_{\text{geo}}\times d_{\text{geo}}}, B_{\mathbf{c}} \in \mathbb{R}^{m_{\mathbf{c}}\times d_{\mathbf{c}}}$ are the PCA basis of geometry and color latent code space, and $m_{\text{geo}}, m_{\mathbf{c}}$ are the number of the principal components. At test time, we can obtain the combination weights via solving the optimization problem:

$$\arg\min_{W^i_{\text{geo}}, W^i_{\mathbf{c}}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{render}}(\mathbf{z}^i_{\text{geo}}, \mathbf{z}^i_{\mathbf{c}}) \tag{14}$$

However, the representation space of such a weighted sum combination is limited by those basis. Hence, we introduce residual latent codes (*i.e.*, $r^i_{\text{geo}} \in \mathbb{R}^{d_{\text{geo}}}, r^i_{\mathbf{c}} \in \mathbb{R}^{d_{\mathbf{c}}}$) for each instance to expand the underlying representation space. In our method, the latent codes of instance i can be formulated as:

$$\begin{aligned} \tilde{\mathbf{z}}^i_{\text{geo}} &= \mathbf{z}^i_{\text{geo}} + r^i_{\text{geo}}, \\ \tilde{\mathbf{z}}^i_{\mathbf{c}} &= \mathbf{z}^i_{\mathbf{c}} + r^i_{\mathbf{c}} \end{aligned} \tag{15}$$

Hence, the optimization problem at test time is:

$$\arg\min_{W^i_{\text{geo}}, W^i_{\mathbf{c}}, r^i_{\text{geo}}, r^i_{\mathbf{c}}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{render}}(\tilde{\mathbf{z}}^i_{\text{geo}}, \tilde{\mathbf{z}}^i_{\mathbf{c}}) \tag{16}$$

To illustrate the effectiveness of the residual latent code, we conduct ablation studies on the test partition of the Stirling/ESRC dataset as shown in Table 3.

View-switch loss. A key problem in multi-view reconstruction is how to enforce view consistency to better leverage the multi-view information. In our method, the view consistency is enforced from two aspects: (i) we divide the geometry latent code into an identity component and an expression component (*i.e.*, $\mathbf{z}_{\text{geo}} = \{\mathbf{z}_{\text{id}}, \mathbf{z}_{\text{exp}}\}$). Then, we impose an implicit view consistency via enforcing different views of the same instance share the same identity and color latent codes; (ii) we propose a view-switch loss $\mathcal{L}_{\text{sw}}$ to further impose an explicit view consistency. Our view-switch loss is designed based on the intuition that the rendered images from different expression latent codes of the same identity under the same camera pose should be similar ignoring the local regions that are more likely to be influenced by the expression varieties (*e.g.*, mouth region).

Specifically, given two views i_m and i_n from a sparse set of images for an instance i, we replace the expression latent code of the view i_m with that of the view i_n (*i.e.*, $\mathbf{z}^{i_n}_{\text{exp}}$). Then, we use the switched

geometry latent code (*i.e.*, $\mathbf{z}^{i_m}_{\text{geo,sw}} = \{\mathbf{z}^{i}_{\text{id}}, \mathbf{z}^{i_n}_{\text{exp}}\}$) to calculate the RGB loss as Eq. (10) and the mask loss as Eq. (11). To reduce the impact of expression variations among different views, we use a face parsing network [52] to omit pixels in the mouse region. We denote the RGB and mask loss after switch views latent code as the view-switch loss $\mathcal{L}_{\text{sw}}$ to distinguish it from previous rendering loss. Hence, the total loss function $\mathcal{L}'_{\text{render}}$ at test time is:

$$\begin{aligned}\mathcal{L}'_{\text{render}} &= \mathcal{L}_{\text{render}} + \mu\mathcal{L}_{\text{sw}} \\ &= \mathcal{L}_{\text{render}} + \mu(\tau_{\text{rgb}}\mathcal{L}_{\text{rgb, sw}} + \tau_{\text{mask}}\mathcal{L}_{\text{mask, sw}})\end{aligned} \tag{17}$$

where μ, τ_{rgb}, and τ_{mask} are set to 0.1, 1, and 100 in our experiments.

Mesh recovery. To recover the mesh from our neural SDF, *i.e.*, S^i_θ for a face instance i, we use the marching cube algorithm [29] at a resolution of 250, which is a trade-off value to balance the output quality and the computational cost.

3.5 Implementation Details

Optimization. The optimization of the Geometry Network f_{geo} contains two steps. First, we optimize the Reference Network to represent the surface of one scan using the 3D data (*i.e.*, the 3D on-surface points and corresponding normals of this scan). Then, the 3D data of the training set are used to train the Deformation Network (f_{deform}) and finetune the Reference Network (f_{ref}). In the first step, we use the Adam optimizer with a learning rate of 1e-3. For the second step, the learning rate is 1e-4 with a mini-batch size of 32.

As for the Rendering Network optimization, the learning rate is set to 1e-4 with a mini-batch size of 32. During the test-time reconstruction process, the inputs RGB images are used as the supervision to find the corresponding latent codes. Hence, we use Adam optimizer with a learning rate of 1e-4 to minimize the loss function Eq. (17). The hyperparameters of Eq. (17) are similar to that of the rendering network optimization with $\mu = 0.1$ and anneal down to zero after 10 iterations.

For in-the-wild inputs, we obtain camera parameters $\mathbf{K}, \mathbf{R}$ by solving the optimization problem as in Eq.(12) using Adam optimizer with a learning rate of 1e-3.

Network architecture. We use fully connected (FC) layers with width 512 for our implicit neural networks (*i.e.*, f_{ref}, f_{deform}, f_{render}). As in many previous works of implicit neural representations [30, 49, 50], we use the Fourier positional encoding [44] for the inputs (*i.e.*, 3D coordinates) of the Reference Network and the Deformation Network to reduce the difficulty of learning high-frequency functions for the neural networks. As for one of the three coordinates (*i.e.*, x_i) of $\mathbf{x} = \{x_i\}^3_{i=1}$, the positional encoding is $w^K(x_i) = \sum_{k=0}^{K}(\cos(2^k\pi x_i) + \sin(2^k\pi x_i))$, where $K = 6$. For the view direction $\mathbf{v}$, we also use such a positional encoding as $w^4(\cdot)$.

Timing statistics. Our experiments are implemented on Pytorch with NVIDIA 2080Ti GPUs. The training time is about two days with 8 GPUs. As for test-time reconstruction, the inference time for one instance with 2$\sim$4 views as inputs is about two hours with one GPU.

4 Experiments

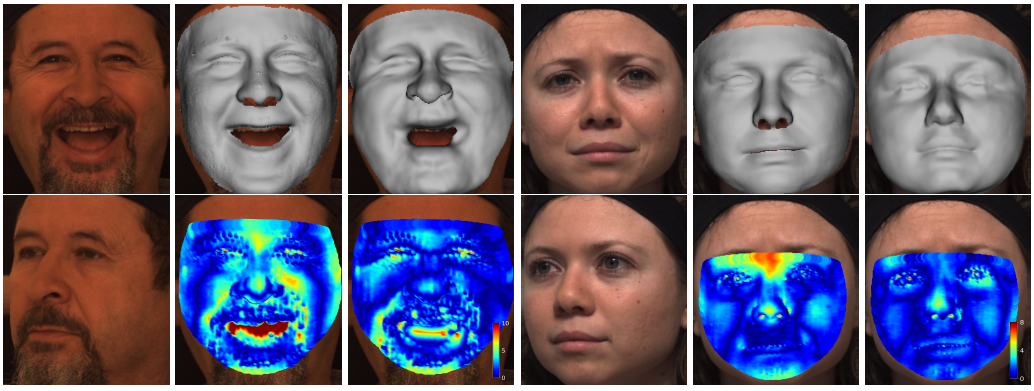
4.1 Experimental Setup

Datasets. We conduct experiments on two benchmarks for 3D face reconstruction as listed below.

- The Stirling/ESRC [42] dataset provides more than 1K high-quality 3D scans and is built upon more than 130 subjects with 8 different expressions. For each scan, a pair of RGB images taken from yaw angles $\pm 45^\circ$ are used as the texture. We split this dataset into training and testing sets containing 95 and 35 subjects, respectively, in the same way as [2].
- The Bosphorus [37] dataset contains 106 subjects with 35 expressions and 13 poses. For each subject, the images with different expressions are under the frontal view, while the images of neutral expression are under various poses. Following [2, 2, 3, 12], we adopt this dataset to evaluate the performance of our method under a two-view setting. Specifically, we select a non-neutral frontal

(a) Instance 1 (b) Instance 2

Figure 3: Qualitative comparison between DFNRMVS [2] and our method on the Stirling/ESRC test set. For each instance, from left to right, we show the two input views, the results of DFNRMVS [2], and the ones obtained via our method, respectively. For both methods, we show the reconstructed mesh in the first row and the corresponding error map in the second row. All the results are overlaid with the first input view. The unit of the color bar is millimeter.



(a) Instance 1 (b) Instance 2

Figure 4: Qualitative comparison between DFNRMVS [2] and our method on the Bosphorus dataset [37]. The images are arranged in the same way as Figure 3.

view image and another image of a neutral expression under a yaw angle of -30° for each instance. Hence, the overall test set contains 453 pairs of images.

Evaluation protocols. As in previous works [2, 3, 12], we use the Euclidean distance between the ground truth 3D face surface points and the aligned output mesh to evaluate the geometric error. The average geometric error (Mean, in mm) and the standard deviation (STD, in mm) among all test samples are computed and reported in our quantitative evaluations. The alignment contains two steps: (i) we first use the ground truth landmarks (*e.g.*, 7 landmarks provided in Bosphorus dataset [37]) and the landmark points in our reconstruction results to achieve rough alignment [38]; (ii) the rigid ICP algorithm [53] is used to further improve the alignment between our prediction and the ground truth. Besides, the ground truth is cropped to reduce the noise based on the corresponding 3D landmarks. Note that those strategies are the same as previous methods [2, 3, 12].

4.2 Qualitative Results

We present qualitative comparisons between DFNRMVS [2] and our method on both Stirling/ESRC and Bosphorus datasets in Figures 3 and 4 respectively. For each test instance in both figures, we

Method	Mean (mm)	STD (mm)
Deng et al. [12]	1.47	0.40
DFNRMVS [2]	1.44	0.38
Bai et al. [3]	**1.36**	0.38
Ours	1.39	**0.35**

Table 1: Quantitative results using two-view input on the Bosphorus dataset [37].

Method	Metric	2 views	3 views	4 views
DFNRMVS [2]	Mean (mm)	1.04	1.03	1.02
	STD (mm)	0.33	0.30	0.29
Ours	Mean (mm)	**0.995**	**0.991**	**0.981**
	STD (mm)	0.177	0.206	0.196

Table 2: Quantitative results on the Stirling/ESRC dataset [42] with different numbers of input views.

show the two input views on the left, and the result by DFNRMVS [2] in the middle, and our result on the right side. We provide the flat-shaded face mesh in the first row and the corresponding error map in the second row.

From these two figures, we can see that compared to DFNRMVS [2], the geometry of our method is closer to the ground truth with more local details, such as the forehead and the shape around the mouth. Also, the nose shape of our method is more similar to the input target than that from [2]. In Figure 4, the region of the error map is limited by the provided ground truth. More qualitative results are provided in our supplementary materials.

4.3 Quantitative Results

To illustrate the effectiveness of the proposed method, we compare with several state-of-the-art methods quantitatively in multi-view 3D face reconstruction, including [2, 3, 12]. As shown in Table 1, our approach achieves better performance in terms of mean errors compared with previous methods [2, 12] and is comparable to a more recent method [3] when evaluate on the Bosphorus dataset.

We also conduct quantitative comparisons on the test set of Stirling/ESRC dataset as shown in Table 2. Since the other methods [3, 12] have not provided the results on this test set, we only compare our results with DFNRMVS [2]. The results demonstrate that our method has lower mean errors consistently with different numbers of input views. Moreover, the performance improvement of our method becomes more significant when increasing the number of views.

4.4 Ablation Studies

We perform the ablation study on the test set of Stirling/ESRC dataset to investigate the impact of the proposed view-switch loss and residual latent code. The corresponding two-view reconstruction results are shown in Table 3. Our baseline is the reconstruction performance obtained via solving the optimization problem as Eq. (14), without using our residual latent code or view-switch loss.

Method	Mean (mm)	STD (mm)
DFNRMVS	1.040	0.33
DFNRMVS*	1.171	0.35
Baseline	1.152	0.351
Baseline + view-switch loss	1.108	0.163
Our full algorithm	**0.995**	0.177

Table 3: Ablation study on the Stirling/ESRC dataset [42]. * denotes that the results are obtained by testing the released model of DFNRMVS [2] on our selected samples for a fair comparison.

As shown in Table 3, our proposed view-switch loss leads to better reconstruction performance. The improvement of our full algorithm with the addition of residual latent code is also noticeable (the last row in Table 3). This fact demonstrates that our residual latent codes effectively extend the shape space of the morphable models. Since the Stirling/ESRC test partition of [2] is not available, we follow the same selection strategy as in [2] and present the test results of their released model on our selected test samples for fair comparison in Table 3.

5 Conclusions

In this work, we present a novel method for 3D face reconstruction from multi-view images via implicit neural deformation. By using a neural SDF based representation, we are able to reconstruct faces with high-fidelity details even from sparse-view input of diverse expressions. Different from previous 3DMM based methods, we propose residual latent code to extend the shape space of implicit morphable models. To further enforce consistency among different views of one instance at test time, we introduce a novel view-switch loss for joint optimization of the network and latent code. Besides, we design a training strategy for the implicit neural network to alleviate the collapse issue during self-supervised optimization by introducing prior information of face geometry and colors. Our results on the Stirling/ESRC dataset and the Bosphorus dataset demonstrate that our approach outperforms alternative baselines and state-of-the-art methods.

Our current implementation of test-time reconstruction takes about two hours for a single instance and the bottleneck is the ray-tracing process. We plan to integrate several recent approaches [18, 33] for accelerated rendering of neural SDFs to speed up the computation.

References

[1] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020.

[2] Ziqian Bai, Zhaopeng Cui, Jamal Ahmed Rahim, Xiaoming Liu, and Ping Tan. Deep facial non-rigid multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5850–5860, 2020.

[3] Ziqian Bai, Zhaopeng Cui, Xiaoming Liu, and Ping Tan. Riggable 3d face reconstruction via in-network optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6216–6225, 2021.

[4] Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. High-quality single-shot capture of facial geometry. *ACM Trans. Graph.*, 29(4), 2010.

[5] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[6] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003.

[7] Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. High resolution passive facial performance capture. *ACM Trans. Graph.*, 29(4), 2010.

[8] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2013.

[9] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):1–10, 2014.

[10] Anpei Chen, Zhang Chen, Guli Zhang, Kenny Mitchell, and Jingyi Yu. Photo-realistic facial details synthesis from single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9429–9439, 2019.

[11] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. Overfit neural networks as a compact shape representation. *arXiv preprint arXiv:2009.09808*, 2020.

[12] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[13] Pengfei Dou, Shishir K Shah, and Ioannis A Kakadiaris. End-to-end 3d face reconstruction with deep neural networks. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5908–5917, 2017.

[14] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. Curriculum deepsdf. In *European Conference on Computer Vision*, pages 51–67. Springer, 2020.

[15] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38, 2020.

[16] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[17] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 534–551, 2018.

[18] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14346–14355, October 2021.

[19] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Transactions on Graphics (TOG)*, 35(3):1–15, 2016.

[20] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018.

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

[22] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3789–3799, 2020.

[23] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.

[24] Vahid Kazemi, Cem Keskin, Jonathan Taylor, Pushmeet Kohli, and Shahram Izadi. Real-time face reconstruction from a single depth image. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 369–376. IEEE, 2014.

[25] Petr Kellnhofer, Lars C. Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4287–4297, 2021.

[26] Hyeongwoo Kim, Michael Zollhöfer, Ayush Tewari, Justus Thies, Christian Richardt, and Christian Theobalt. Inversefacenet: Deep monocular inverse face rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4625–4634, 2018.

[27] Amit Pal Singh Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. In *2020 International Conference on 3D Vision (3DV)*, pages 423–433. IEEE, 2020.

[28] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2019–2028, 2020.

[29] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[30] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[31] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[32] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *IEEE International Conference on Advanced Video and Signal-based Surveillance*, pages 296–301, 2009.

[33] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14153–14161, 2021.

[34] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1259–1268, 2017.

[35] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019.

[36] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7763–7772, 2019.

[37] Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Bosphorus database for 3d face analysis. In *European workshop on biometrics and identity management*, pages 47–56. Springer, 2008.

[38] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[39] Jiaxiang Shang, Tianwei Shen, Shiwei Li, Lei Zhou, Mingmin Zhen, Tian Fang, and Long Quan. Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 53–70, 2020.

[40] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019.

[41] William AP Smith, Alassane Seck, Hannah Dee, Bernard Tiddeman, Joshua B Tenenbaum, and Bernhard Egger. A morphable face albedo model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5011–5020, 2020.

[42] Stirling-ESRC. Stirling/esrc 3d face database, 2018. http://pics.stir.ac.uk/ESRC/.

[43] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11358–11367, 2021.

[44] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, pages 7537–7547, 2020.

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

[46] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5163–5172, 2017.

[47] Fanzi Wu, Linchao Bao, Yajing Chen, Yonggen Ling, Yibing Song, Songnan Li, King Ngi Ngan, and Wei Liu. Mvf-net: Multi-view 3d face morphable model regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 959–968, 2019.

[48] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 601–610, 2020.

[49] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems*, pages 2492–2502, 2020.

[50] Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. i3dmm: Deep implicit 3d morphable model of human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12803–12813, 2021.

[51] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2021.

[52] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 325–341, 2018.

[53] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.

[54] Wenbin Zhu, HsiangTao Wu, Zeyu Chen, Noranart Vesdapunt, and Baoyuan Wang. Reda: reinforced differentiable attribute for 3d face reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4958–4967, 2020.

[55] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z Li. High-fidelity pose and expression normalization for face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 787–796, 2015.

[56] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. *Computer Graphics Forum*, 37(2):523–550, 2018.