A Unified Multi-output Semi-supervised Network for 3D Face Reconstruction

Pengrui Wang[1,2], Yi Tian[1,3], Wujun Che[1], Bo Xu[1]
[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, China
[3] Hunan Normal University, China
wangpengrui2015@ia.ac.cn, tianyi@smail.hunnu.edu.cn, {wujun.che, xubo}@ia.ac.cn

***Abstract*—In this paper, we propose a method to reconstruct fine-grained 3D faces from single images base on a nearly unified multi-output regression network. The network estimates the facial shape, normal and appearance jointly in 2D UV map which preserves spatial adjacency relations among vertexes and provides semantic meaning of each vertex. Three contributions of the proposed method are: 1) we generate the UV map by as-rigid-as-possible parametrization to address the overlapping problem caused by cylindrical unwarp; 2) we directly estimate face normal rather than compute it from the estimated shape to let it catch geometric details from face texture; 3) we propose a post process strategy to generating more realistic faces and to employing the estimated normal. Experiments show that our network is able to learn a uniform appearance and predict more accurate shape from the proposed UV map. Additionally, the post process procedure can improve the quality of facial shapes and add geometric details from estimated normals.**

***Index Terms*—3D Face Reconstruction, Lambertian Reflection, Geometric Surface Parametrization, Position Map**

I. Introduction

3D face reconstruction serves as a fundamental task in computer vision and graphics [1], such as face recognition [2] and animation [3]–[5]. The data formats used for reconstruction include multi-view images [6], photo collections [7], videos [8] and single images [9]–[12], where the reconstructions of single images have a wide prospect of application due to its convenience.

But face reconstruction from a single image is very challenging due to the high variability in pose, expression, appearance, lighting environment and the loss of information during camera projection. Hence, researchers [11], [13], [14] usually constrain faces into low-dimensional subspace using statistical linear face models [15]–[17]. However, face models such as 3D Morphable Models (3DMM) exist limitations on shape and texture representation, because they are usually trained from a small number of 3D scan faces, which are defective for face representation [18].

Recently, some approaches extend the linear 3DMM via additional structures [12] or predict the unconstrained 3D geometry directly [19], [20]. However, those approaches have their defects as well. Tewari et al. [12] and Tran et al. [18] predict geometry with connected layers, which loss spatial relation among vertexes. Sela et al. [19] infers depth maps in image space, which is hard to correspond with geometry vertexes especially when self-occlusions happen, so it is necessary to do non-rigid registration. Jackson et al. [20] regresses a volume to represent facial shape, even though only part of the volume is needed. This limits the resolution of the recovered shape and discards the semantic meaning of points. Afterwards, Feng et al. [21] and Tran et al. [22] reconstruct 3D facial shape points in a UV position map, which is a 2D image recording the 3D coordinates of full facial point cloud. This 2D representation provides semantic meaning of each point in UV space, which is able to be modeled by a fully convolution neural network (CNN). However, the predicted points from the UV map usually produce noisy surface and do not utilize texture information to recover detailed geometry. As for face texture reconstruction, [18], [22] use cylindrical unwarp to present face texture in UV space, thus they can use CNN as texture decoder and apply adversarial training. However, [18] only recovers a mixture of appearance and illumination, and the cylindrical unwarp can cause overlapping problem for 3D face surfaces especially on the nose regions.

In this paper, we propose a nearly unified multi-output network for face reconstruction. The network estimates 3D face shape, normal, appearance and illumination coefficients simultaneously, and the former three are vertical features are all estimated on a UV map so that the spatial adjacency information among vertexes is preserved. Contributions are as follows. First, to avoid the crowding and overlapping cases among vertexes on the UV map, we adopt an as-rigid-as-possible parametrization (ARAP-P) method to project the face geometric surface into UV space. Next, differing from previous works [11], [22] computing normals from shape (denoted as CFN), our method estimates facial normal (denoted as EFN) according to spherical harmonics (SH) lighting function to better separate appearance and illumination. Meanwhile, the EFN can catch the shading information from the input face texture and hence helps to obtain shape details. For convenience, we call our network "normal auxiliary face reconstruction network" (NaNet). Then, under the assumption that we only have ground-truth shapes, we design suitable loss functions to train the NaNet in a semi-supervised manner. Finally, to ensure the predicted geometry is smooth and to merge the detailed information from the EFN, we propose a geometric deformation procedure as our post-processing to improve the quality of the estimated shape. Experiments

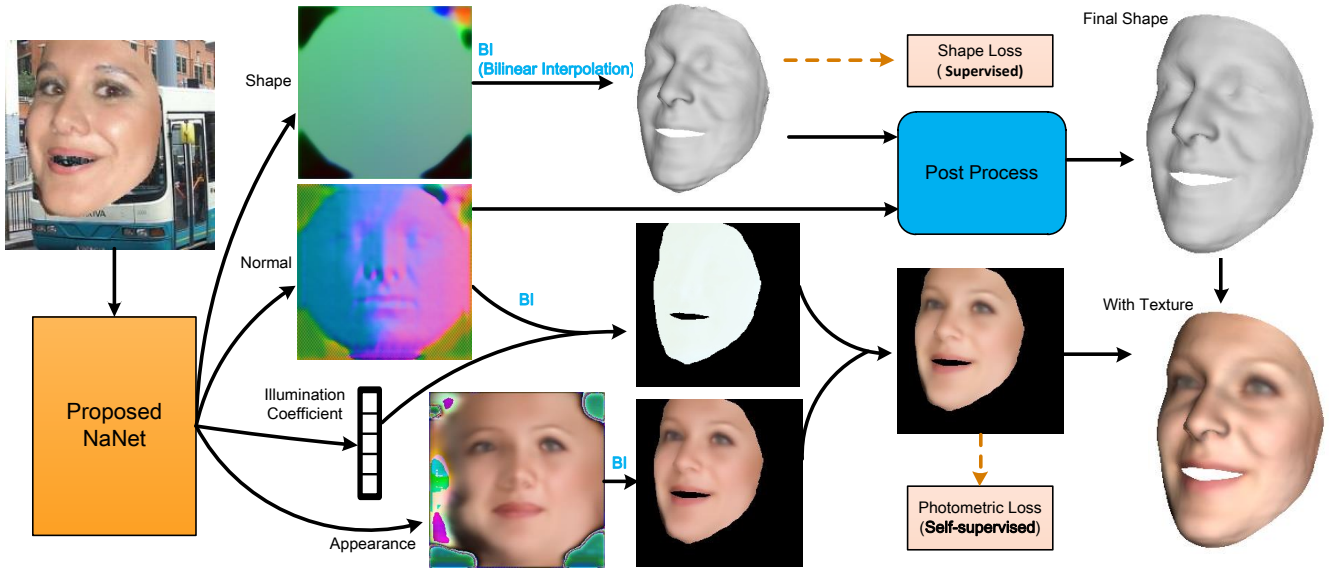

Fig. 1. Reconstruction workflow using the proposed NaNet. Dashed arrows are employed when training. Given an input face image, our approach first estimates illumination coefficient and facial features include shape, geometric normal and appearance on the UV map generated by ARAP-P. Afterwards, per-vertex features are derived by sampling from the UV map. In addition, we use a post process to make the shape more realistic.

show that our network is able to separate appearance and illumination and can create uniform appearance. Meanwhile, the post process strategy generates more realistic shapes than the shapes estimated directly and can add the geometric details from EFN. Quantitative experiments also show that the NaNet predicts more accurate shape from the proposed UV map.

In summary, the main contributions of this paper are:

- A more suitable UV map representation ARAP-P for face reconstruction to avoid crowding and overlapping problems of cylindrical unwarp.
- A nearly unified multi-output network (NaNet) estimates 3D facial features in UV space to preserve spatial adjacency relations among facial vertexes and a novel feature EFN to bring about geometric details.
- A post process procedure deforming the shape from NaNet to utilize the EFN for adding geometric details and to make the final shape more smooth and realistic.

II. Related Work

3D Face Reconstruction from A Single Image: It is a hot area to reconstruct a digital 3D face using only a single image. Approaches can be divided into optimization-based [10] and learning-based [13], [14], [18], [23]. Generally, learning-based methods exploiting the power of deep learning focus more on in-the-wild cases than optimization-based methods. Due to the scarce 3D scans database, learning-based methods train the network with large synthetic data or train in a weakly supervised fashion [11], [22] with a variety of real images. Learning-based methods can be further divided into multi-stage methods [9], [24] and nearly one-pass methods [12], [22], [23]. In multi-stage methods, face reconstruction is split into many subtasks and each network is assigned one. Whereas, one-pass methods usually estimate all the features together and are trained end-to-end. Our work belongs to the one-pass methods and is more like the work [22]. Tran et al. [22] estimates face shape in UV space and needs to estimate weak perspective projection parameters. They also estimate illumination coefficient based on SH function, but their normals are computed according to the vertical position. In contrast, we directly estimate position map [21] which reduces the estimated parameters for face poses and we adopt a more suitable UV map for the network. The normals we estimate according to image texture is not necessarily the vertex normal. They are more like the normal mapping in computer graphics.

Surface Normal Estimation in Face Reconstruction: Surface normals are related with face shape and the final visual texture. On the basis of Shape from Shading (SfS) [25], researches estimate normal directly or as an intermediate for fine detail shape reconstruction. Trigeorgis et al. [26] uses CNN to map image pixels to normals on pixel level directly. Then they integrate the recovered normals to recover 3D face. Other approaches estimate normal as a bridge to optimize z-coordinates of vertexes for template based reconstruction [27], [28] or detailed reconstruction [10], [29]. Our work uses CNN to estimate vertex normal rather than pixel normal which is different from most of previous works. Our normals together with the other facial features is estimated in a semi-supervised data-driven way, which is different from traditional optimization-based SfS. Normal is estimated for the purpose of geometric details added in our post process step.

III. Proposed Method

We first detail the framework of our work and introduce the architecture of NaNet. Next, we introduce the utilized UV map and its advantages. Then, we illustrate the procedure of our

post-processing. Finally, loss functions and training details for NaNet are introduced. In our work, we use two 3DMMs for analysis and experiment, which are denoted as BFM2017 from the work [17] and BFMclip from the work [30], respectively. An overview of our approach is shown in Fig. 1.

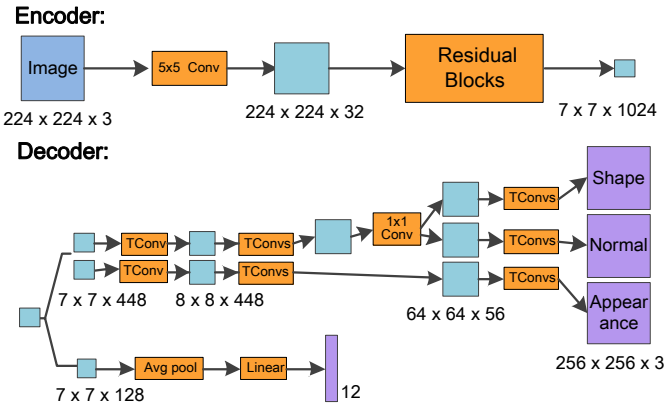

Fig. 2. The architecture of our NaNet. "Conv" represents the convolution layer. "TConv" represents the transposed convolution layer and "TConvs" means a stack of TConv. Numbers represent the sizes of features.

A. The Overall Framework

We propose a multi-output network NaNat for face reconstruction. NaNat can disentangle facial shape, appearance and illumination as is shown in Fig. 1. Facial shape, normal and appearance are predicted on 2D UV maps so that we can exploit a CNN based network and maintain the spatial relations among vertexes. Vertical features can be obtained from the corresponding UV maps by bilinear interpolation [18]. Our UV map are well-designed to avoid vertex crowding and overlapping problems, see Sec. III-C. Like most of the data-driven approaches [21], [23], our network uses the coarse 3D face geometries from synthetic data as ground-truth. Thus Our network is trained through semi-supervised methods and training losses will be introduced in Sec. III-E. Different from current researches [11], [22], we estimate vertex normal. The vertex normal is trained automatically according to the illumination model, see Sec. III-B, which we expect to be able to possess geometric details from shading information. Face shape estimated from the UV map has few restraints which can cause abnormal shapes especially on face boundaries. To smooth the estimated face shape and utilize the estimated normal, we use geometric deformation in our post-processing stage which produces the final face shape, see Sec. III-D.

Our NaNet's inputs and outputs are illustrated in Fig. 1, we now detail its architecture. Fig. 2 shows the main architecture of our NaNet. Based on the Image-to-Image encoder-decoder network PRN [21], we design our NaNet with substantial changes to achieve our desired goal. Except the kernel size of Conv marked in Fig. 2, we use the kernel size 3 for most of convolution layers. For the TConv with stride 2 we change the kernel size to 4. The encoder has 17 residual blocks and 1024 output channels in consideration of more features needed to be reconstructed than PRN. More changes are made in the decoder. It uses 128 channels from encoder to estimate light collections and transforms the other 896 channels to estimate UV maps. First, We split 896 channels into two parts to estimate appearance and shape using group TConvs (10 TConv). Next, we use one Conv with kernel size 1 to double the channels of shape. Finally, we use the three parts of channels to estimate shape, normal and appearance jointly with another group TConv (7 TConv). After each Conv and TConv, we add Batchnorm layer and ReLU activation layer, except the last two layers of decoder which only have eLU activation with alpha 2.0. The reason why we mix estimating shapes and normals at the beginning is that there are strong relations between normals and shapes, so they should share some shallow features.

B. Illumination Model

SH basis functions $H_b : \mathbb{R}^3 \rightarrow \mathbb{R}$ are commonly employed to represent the global illumination under the assumption of purely Lambertian surface [8], [12]. The final texture at vertex v_i with its normal n_i and appearance r_i is:

$$\mathrm{Tex}_i(\mathbf{n_i}, \mathbf{r_i}, \gamma) = \mathbf{r_i} \cdot \sum_{b=1}^{B^2} \gamma_\mathbf{b} H_b(\mathbf{n_i}), \tag{1}$$

where B is the number of SH bands and γ_b is coefficients to control the illumination. We use $B = 2$ which models a minimum of 87.5% of the lighting energy [30]. $\gamma_\mathbf{b} = (\gamma_b^r, \gamma_b^g, \gamma_b^b)^t$ is a 3D vector that controls the irradiance separately for each color channel. Therefore, in our work, the number of illumination coefficients is $3 * B^2 = 12$.

Through the above lighting model, we see that the normals are important elements to synthesize the texture. Inversely, normals are also decided by texture and have the function of separating the illumination and the appearance, which is the inspiration of our novel EFN. However, CFN computed from estimated facial shape may be inaccurate and has fewer relations with textures.

C. UV Map Representation

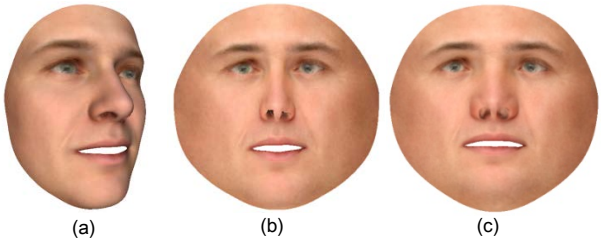

Fig. 3. Surface parametrization comparison. (a) Original face geometry. (b) Cylindrical unwarp [18]. (c) As-rigid-as-possible parametrization.

To keep the spatial relation among vertexes and leverage the modeling ability of CNN, we estimate the shape, appearance and normal in a 2D grid, called UV map. We denote as surface parametrization a mapping from the surface to UV space. Fig. 3 shows the results of parametrization for BFM2017 and

BFMclip respectively, in which (b) is cylindrical unwarp used in [18], [22] and (c) is ARAP-P used in this work. From Fig. 3(b), we see that the UV map created by cylindrical unwarp is crowded on height region (large absolute z-coordinates), especially in nose and mouth. Furthermore, cylindrical unwarp can't handle overlap region such as nostril. Tutte Embedding used in [21] still suffers from the problem of overlapping.

In consideration of the above problems, we propose adopting ARAP-P to generate the face UV map. ARAP-P computes a parametrization that strives to preserve distances, which is inspired by the ARAP surface deformation [31]. By observing Fig. 3, it's obvious that ARAP-P is more suitable for UV map representation than cylindrical unwarp. It not only retains the distances in surface (see nose region), but also overcomes the overlap problems. Our UV map is generated by means of Blender [1] and libigl [32].

D. Post-processing

Generally, facial shape estimated through UV map (also called position map) is noisy and has a grid-like surface and large irregular deformations on boundary regions. Besides, the EFN carrying details from texture needs to be considered on the final reconstructed shape. To solve these problems, we propose a geometric deformation method to deform a template face, actually a 3DMM in our work, to smooth the estimated shape and to utilize the normal information.

Because vertexes in our estimated shape and vertexes in the 3DMM are one-to-one correspondence, we firstly fit 3DMM to get a rapid initial shape. Next, the fitted face mesh is deformed to suit the estimated shape. Note that the output of the estimated shape, denoted as S, remains fixed during the processing. Details of our deformation method are:

1. We denote the identity and expression parameters controlling 3DMM as $\mathbf{p}_{\text{id}}$ and $\mathbf{p}_{\text{exp}}$, their linear bases as B_{id}, B_{exp} and a rigid transformation matrix as M which is consisted of scale, rotation and translation. The 3DMM fitting minimizes the following energy:

$$E = \|S - V\|_2^2 + w_{\text{id}} E_{\text{reg}}(\mathbf{p}_{\text{id}}) + w_{\text{id}} E_{\text{reg}}(\mathbf{p}_{\text{exp}}), \quad (2)$$

where $V = M(V_m + B_{\text{id}}\mathbf{p}_{\text{id}} + B_{\text{exp}}\mathbf{p}_{\text{exp}})$. The last two items in Eq. 2 are common statistical regularization on 3DMM parameters like [12], [13]. We solve M and parameters iteratively. Let $\mathbf{p} = [\mathbf{p}_{\text{id}}, \mathbf{p}_{\text{exp}}]$ and the initial $\mathbf{p}$ is zeros vector. Firstly, with fixed $\mathbf{p}$ we use Procrustes analysis to compute M between the point clouds. Then, with fixed M, we minimize Eq. 2 to get a new $\mathbf{p}$. The above two steps are iterated until the fitting loss is small enough. The resulting template face mesh V will be applied in the following step.

[1]Blender https://www.blender.org

2. We deform the template mesh V by minimizing the energy similar to [19]:

$$\begin{aligned} E =& \alpha_{p2point} \sum_{(v_i,s_i)\in J} w_i \|v_i - s_i\|_2^2 \\ &+ \alpha_{p2plane} \sum_{(v_i,s_i)\in J} |\vec{n}(s_i)(v_i - s_i)|^2 \\ &+ \alpha_{smooth} \sum_{i=1}^{\text{nV}} \|w_{i,i} v_i - \sum_{v_j \in \text{Neib}(v_i)} w_{i,j} v_j\|_2^2, \end{aligned} \quad (3)$$

where v_i and s_i are the vertexes in V and S respectively, J is the set of associated vertex pairs (v_i, s_i) whose Euclidean distance of v_i, s_i is not too large, and nV is the number of vertex used in the face model. Then w_i is position weight mask which will be introduced in Sec. III-E, $w_{i,j}$ is the item of the Laplace Beltrami operator on the mean shape of 3DMM and Neib(v_i) is the set of 1-ring neighboring vertexes about the v_i. The most important variable $\vec{n}(s_i)$ is the EFN from the NaNet.

E. Training Loss Function

We denote the outputs of shape, appearance, normal after bilinear interpolation and SH coefficients as $x = (S, R, N, \gamma)$. Note that the normal N should be normalized after interpolation. Assuming we only have the ground-truth 3D shapes like [21], our loss function consists of a data fitting term and a regularization term:

$$E_{\text{total}}(x) = E_{\text{data}}(x) + E_{\text{reg}}(x), \quad (4)$$

where,

$$\begin{aligned} E_{\text{data}}(x) =& w_{\text{shape}} E_{\text{shape}}(x) + w_{\text{photo}} E_{\text{photo}}(x) \\ E_{\text{reg}}(x) =& w_{\text{realN}} E_{\text{realN}}(x) + w_{\text{alb}} E_{\text{alb}}(x) + w_{\text{smo}} E_{\text{smo}}(x). \end{aligned}$$

We explain the details of the individual terms in the following.

Weight Mask Based Shape Loss: As [21] points out, mean square error (MSE) treats all points equally, but the central region of face has more discriminative features. Hence, we employ a weight mask W to our shape MSE loss:

$$E_{\text{shape}}(x) = \sum_{i=1}^{\text{nV}} W(i) \|S(i) - S^*(i)\|_2^2, \quad (5)$$

where S^* is the ground-truth shape. In our work, the weight mask has two regions and an example on BFM2017 is shown in Fig. 4.

Fig. 4. Two region weight mask on UV map. In the weight mask, the red region has higher weight.

Photometric Loss: According to the lighting model in Sec. III-B, we use vertex feature R, N and light info γ

to create synthetic image C. Each pixel in C is computed by interpolation of barycentric coordinate from ground-truth vertex textures. Let $\bar{V}$ be the set of visible pixel computed using backface culling and I is the input image, then the photometric loss is:

$$E_{\text{photo}}(x) = \sum_{p \in \bar{V}} \|I(p) - C(p)\|_2. \tag{6}$$

For robustness, we employ $l_{2,1}$-norm the same as [12]. This loss is also called self-supervised loss.

Real Vertex Normals Loss: In order to make the normal N not differ greatly from real vertex normals, we add a loss to minimise the angular distance between N and ground-truth vertex normals N^* and the loss is defined as follows:

$$E_{\text{realN}}(x) = \sum_{i=1}^{\text{nV}} 1 - N(i)^T N^*(i). \tag{7}$$

Shape Smoothness: Like post-processing, we impose smoothness on the vertex locations. The loss is,

$$E_{\text{smo}}(x) = \sum_{i=1}^{\text{nV}} \|w_{i,i} v_i - \sum_{v_j \in \text{Neib}(v_i)} w_{i,j} v_j\|_2^2,$$

where the symbols have been described in Sec. III-D. In fact, we increase weight on vertexes of face boundary during training .

Local Appearance Consistency: We add a loss with small weight to enforce local appearance consistency:

$$E_{\text{alb}}(x) = \sum_{i=1}^{\text{nV}} \sum_{v_j \in \text{Neib}(v_i)} \|R(i) - R(j)\|_2, \tag{8}$$

we also employ $l_{2,1}$-norm for this loss.

F. Training Data

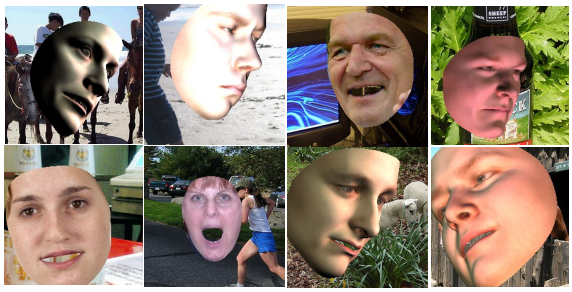

Fig. 5. Generated face image examples for training BFM2017. Faces varies in terms of shape and texture.

It is hard to collect enough 3D face scans to train deep neural network. As a result, we verify our method on a synthetic dataset, which avoids the difficulty of collecting authentic ground-truth face geometry. For experiments using BFMclip as face topological structure, we choose 300W-LP [33] to form our training sets. Specifically, we firstly find the corresponding vertex indexes using Nearest Neighbor Search between the 3DMM used in 300W-LP and BFMclip. Then, we use the coefficients in 300W-LP to recover the 3D vertexes. Finally, the corresponding vertexes are chosen as ground-truth vertexes. For the experiments using BFM2017, we create the training set by rendering face meshes with textures using Z-buffer on real images. Fig. 5 shows the generated examples. The face shapes are obtained from random appropriate 3DMM parameters or parameters calculated by minimizing the L_2 difference between 2D image landmarks and the projected 3DMM landmarks. Textures also come from two parts. One comes from 3DMM's appearance model using random parameters mixed with light color through three point light sources which are random in locations and colors. The other comes from real face images textures extracted by the estimated shapes, but only the images with near frontal face are used, e.g. the image from Chicago face database (CFD) [34] and Multi-pie [35].

We train our NaNet on the data mentioned above, where only the face shape is trained by supervised loss and the other outputs R, N and γ are learned automatically by the self-supervised loss. Hence, our training strategy falls into the semi-supervised categories.

IV. Evaluation and Results

This section demonstrates the results of our work. For qualitative analyses, we choose BFM2017 as topological structure and test it on synthetic images which are created the same as it in Sec. III-F. For quantitative analyses, we choose BFMclip as topological structure and test it on real images. This is because 300W-LP training set is based on real images. All the testing images never occur in training set.

A. Appearance Learning

Our NaNet automatically learns appearance representation, which is differently from the work [18] that merges the appearances and illumination, and the work [11] that learns appearance parameters based on the low space 3DMM color model. Fig. 6 shows the results of estimated textures and appearances.

As shown in Fig. 6, our NaNet learns an approximate uniform appearance representation. The first three rows are the results from our NaNet trained directly, their appearances tend to be whity. Although NaNet can separate illumination and appearance, there are many ways of combination to stand for the two items. The last two rows show another way to stand for appearances which tend to be yellowy. This is achieved by pre-training with a loss to minimize the difference between the appearance and the image texture.

B. Post-processing and Final Reconstruction Results

PRN [21] shows that face shapes estimated from position maps have great performance in reconstruction quantitative evaluation. But the vertexes of face surface are grid-like placed and the meshes generated from the shapes are not smooth, see Fig. 7 (PRN tailors the vertexes near the edges). These problems damage the face shape presentation and may not be appropriate for animation applications.

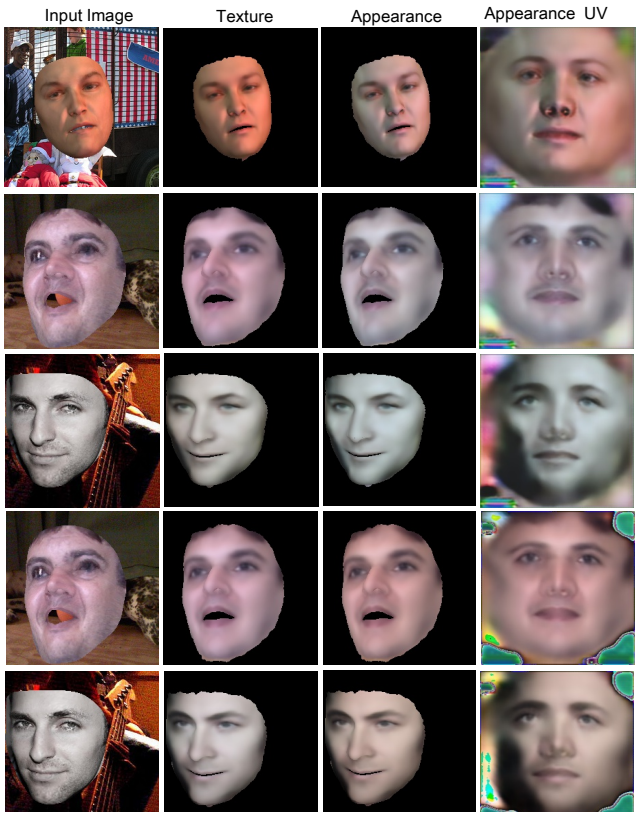

Fig. 6. Appearance automatic learning. Faces in the last two rows are trained with a pre-training strategy.

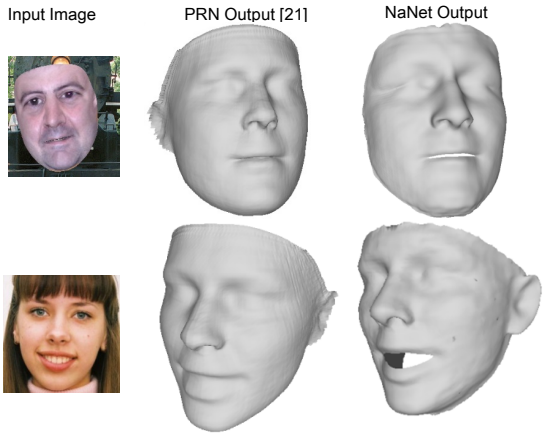

Fig. 7. Facial shapes generate from position maps are usually grid-like and not smooth.

To solve the above problems, we use our post-processing method to obtain smooth shapes and add the information from the EFN at the same time. Fig. 8 shows the results of our two-step post-processing. In Fig. 8, the third column is the final shape after deformation. We can see that due to our post-processing, face shapes become smoother and have few crashes on the mesh boundary. The forth column indicates the effects of our EFN. The heat maps are generated as follows. We firstly calculate the normals from the estimated shape and use them in our post-processing to obtain another final

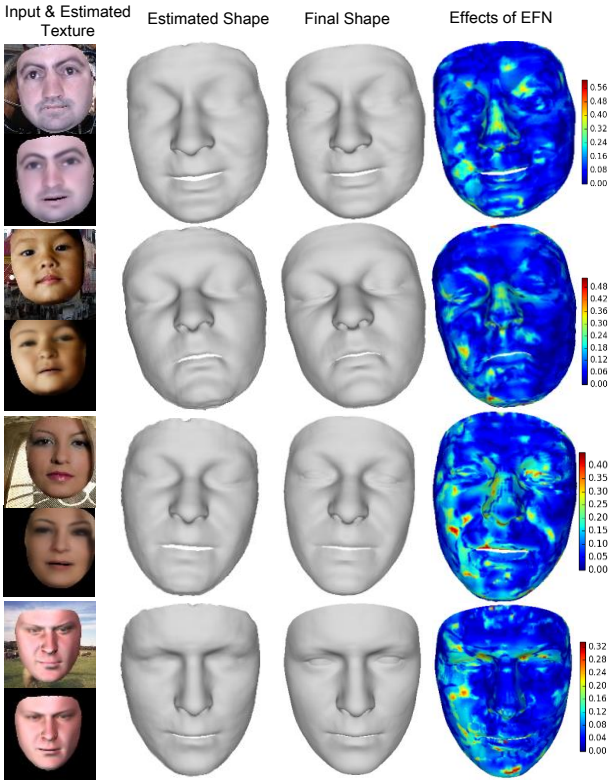

Fig. 8. Post-processing results on shape and the effects of EFN. The last column is the Euler distances visualized as heat maps between final shapes using EFN and CFN. The distances is measured in pixel space.

shape denoted as CFN-S. We denote as EFN-S the finally generated shape by the EFN. Then, points distances between CFN-S and EFN-S are calculated and converted to heat map colors. Finally, we use the heat map colors as vertical colors of CFN-S and then create the heat map in Fig. 8. The heat maps demonstrate that the EFN catches details information, especially on the corners, edges and regions with large texture differences between input images and estimated textures. In our experiment, $\alpha_{p2point} = 3, \alpha_{p2plane} = 10, \alpha_{smooth} = 0.01$.

C. UV Map Comparison

In Sec. III-C we have made qualitative analyses and shown the superiorities of ARAP-P. In Fig. 3 (or see the 3D shape results in [18], [22]), it shows that the big problem for cylindrical unwarp is that it can't handle overlapping geometries. The same problem should also occur when using Tutte Embedding which is used in PRN [21]. However, PRN has made additional artificial pretreatments on nose vertexes. Now, we focus on quantitative analyses of them in this section.

Because we only evaluate the shape performance of different UV maps, we simplify our NaNet to make it like PRN. We only preserve NaNet's facial shape estimation part and denote it as NaNet-shape. The architecture and parameters of NaNet-shape are also like PRN, except that most of our convolution kernel sizes are 3 while PRN's sizes are 4. NaNet-shape is trained supervised just like PRN. The evaluation criteria used in our experiments is defined as:

$$\text{NME} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|v_i^* - v_i\|_2}{d}, \tag{9}$$

where N is the number of vertexes used for estimating the L_2 distances, d is the 3D outer interocular distance of ground-truth mesh, x_i and x_i^* are the estimated and ground-truth vertex locations. The vertexes used for comparison do not include the facial margin and the nostril region in view of the overlapping problem of cylindrical unwarp. Fig. 9 shows the shape vertexes used for comparison. $N = 27378$ in this experiment (total number of vertexes in BFMclip is 46990).

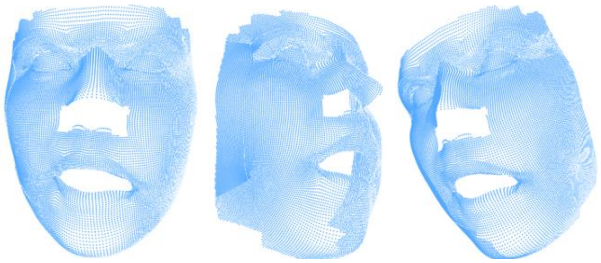

Fig. 9. An example of shape vertexes used for comparisons. These vertexes correspond to parts of BFMclip vertexes.

TABLE I
PERFORMANCE COMPARISONS. THE NMES (%) ARE REPORTED AND THE LOWER IS THE BETTER.

		PRN [21]	Cylindrical	ARPA-P
DB-A	No ICP	-	9.52	**9.43**
	ICP once	2.75	**2.72**	2.82
DB-B	No ICP	-	2.42	**2.41**
	ICP once	4.25	4.14	**4.13**

Our test datasets include AFLW2000-3D [33] which has 2000 samples denote as DB-A, and parts of IBUG and LFPW test images of 300W-LP which has 5452 samples denote as DB-B. Note that the other parts of IBUG and LFPW test images (also 5424 samples) are as valid set and the left images in 300W-LP are as training set. Results are presented in Tab. I. In Tab. I, ICP means Iterative Closest Point algorithm which is usually used for alignment and finding corresponding points before calculating NME. "ICP once" means we apply ICP only once based on manually selected six positions which is similar to [36] and "No ICP" means we directly calculate the dense alignment error considering the the training set is consistent with the test set. As Tab. I shows, UV map from ARPA-P has better performance on 3D face reconstruction than UV map from cylindrical unwarp under "No ICP" criteria even we do not include the nostril region. Although, under "ICP once" criteria, ARPA-P not always has the best performances, the criteria may not robust. ICP calculating rigid transform based on manually marked a few points may be inaccurate, which may be the reason causing the performance drop on DB-B. Thus, according to the qualitative and quantitative analysis, ARAP-P is a more suitable way to generate face geometry UV map than cylindrical unwarp.

D. Limitations

There are some limitations in our work. First of all, ARAP-P usually generates rectangular UV maps after mapping 3D face meshes. Our NaNet outputs are square, thus we need to resize the UV maps to suit the output size, which may damage the ARAP feature and performances. Then, the estimated textures are blurry (see the second column in Fig. 6), which appear commonly in [12], [18], [22]. It seems hard for our NaNet to capture fine appearances and geometric details. At last, the effects of estimated normals are weak in Fig. 8. In the future work, we may adopt better approaches to merge the EFN, e.g., the Laplacian-based surface editing technique [7].

V. CONCLUSION

In this paper, we propose a normal estimation assisted neural network, NaNet, for single image face reconstruction. The predicted vertical features are on the UV map to preserve spatial adjacency relations among vertexes. We adopt ARAP-P to project face geometry in UV space, which unfold 3D surface harmoniously and evenly to avoid the crowding and overlapping problems of cylindrical unwarp. The most novel trait of NaNet is EFN. Because EFN is not the actual shape normal, it can catch detail information according to textures. Our NaNet is trained in a semi-supervised style where only the shape is supervised training. Besides, we propose a post process method to add geometric details by the EFN and to generate a more realistic face shape. Qualitative results demonstrate our method can separate appearance and lighting, learn the uniform appearance representation and make more realistic facial shape by our post-processing method. Qualitative and quantitative results both demonstrate the presented ARAP-P UV map is superior to cylindrical unwarp. In the future, we will do more efforts on the real face reconstruction and fine detailed shape reconstruction.

ACKNOWLEDGEMENTS

This work is supported by the National Key R&D Plan of China (No. 2017YFB1002804), the National Natural Science Foundation of China (No. 61471359) and the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDB32070000).

REFERENCES

[1] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt, "State of the art on monocular 3d face reconstruction, tracking, and applications," in *Computer Graphics Forum*, vol. 37, no. 2, 2018, pp. 523–550.

[2] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Li, "High-fidelity pose and expression normalization for face recognition in the wild," in *CVPR*, 2015, pp. 787–796.

[3] A. E. Ichim, S. Bouaziz, and M. Pauly, "Dynamic 3d avatar creation from hand-held video input," *ACM ToG*, vol. 34, no. 4, p. 45, 2015.

[4] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *CVPR*, 2016, pp. 2387–2395.

[5] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time high-fidelity facial performance capture," *ACM ToG*, vol. 34, no. 4, p. 46, 2015.

[6] D. Sibbing and L. Kobbelt, "Building a large database of facial movements for deformation model-based 3d face tracking," in *CGF*, vol. 36, no. 8, 2017, pp. 285–301.

[7] J. Roth, Y. Tong, and X. Liu, "Adaptive 3d face reconstruction from unconstrained photo collections," in *CVPR*, 2016, pp. 4197–4206.

[8] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt, "Reconstruction of personalized 3d face rigs from monocular video," *ACM ToG*, vol. 35, no. 3, p. 28, 2016.

[9] S. Yamaguchi, S. Saito, K. Nagano, Y. Zhao, W. Chen, K. Olszewski, S. Morishima, and H. Li, "High-fidelity facial reflectance and geometry inference from an unconstrained image," *ACM ToG*, vol. 37, no. 4, p. 162, 2018.

[10] L. Jiang, J. Zhang, B. Deng, H. Li, and L. Liu, "3d face reconstruction with geometry details from a single image," *ToIP*, vol. 27, no. 10, pp. 4756–4770, 2018.

[11] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez, and C. Theobalt, "Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction," in *ICCV*, vol. 2, no. 3, 2017, p. 5.

[12] A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez, and C. Theobalt, "Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz," *arXiv:1712.02859*, vol. 2, 2017.

[13] N. Chinaev, A. Chigorin, and I. Laptev, "Mobileface: 3d face reconstruction with efficient cnn regression," *arXiv:1809.08809*, 2018.

[14] E. Richardson, M. Sela, and R. Kimmel, "3d face reconstruction by learning from synthetic data," in *3DV*, 2016, pp. 460–469.

[15] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *SIGGRAPH*, 1999, pp. 187–194.

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

[17] T. Gerig, A. Morel-Forster, C. Blumer, B. Egger, M. Luthi, S. Schönborn, and T. Vetter, "Morphable face models-an open framework," in *FG*, 2018, pp. 75–82.

[18] L. Tran and X. Liu, "Nonlinear 3d face morphable model," *arXiv:1804.03786*, 2018.

[19] M. Sela, E. Richardson, and R. Kimmel, "Unrestricted facial geometry reconstruction using image-to-image translation," in *ICCV*, 2017, pp. 1585–1594.

[20] A. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, "Large pose 3d face reconstruction from a single image via direct volumetric cnn regression," in *ICCV*, 2017, pp. 1031–1039.

[21] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3d face reconstruction and dense alignment with position map regression network," *arXiv:1803.07835*, 2018.

[22] L. Tran and X. Liu, "On learning 3d face morphable model from in-the-wild images," *arXiv:1808.09560*, 2018.

[23] E. Richardson, M. Sela, R. Or-El, and R. Kimmel, "Learning detailed face reconstruction from a single image," in *CVPR*, 2017, pp. 5553–5562.

[24] A. Tran, T. Hassner, I. Masi, E. Paz, Y. Nirkin, and G. Medioni, "Extreme 3d face reconstruction: Seeing through occlusions," in *CVPR*, 2018.

[25] B. K. P. Horn, *Shape from Shading: A Method for Obtaining the Shape of A Smooth Opaque Object from One View*. Massachusetts Institute of Technology, 1970.

[26] G. Trigeorgis, P. Snape, I. Kokkinos, and S. Zafeiriou, "Face normals "in-the-wild" using fully convolutional networks," in *CVPR*, 2017, pp. 340–349.

[27] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz, "Total moving face reconstruction," in *ECCV*, 2014, pp. 796–812.

[28] I. Kemelmacher-Shlizerman and R. Basri, "3d face reconstruction from a single image using a single reference face shape." *ToPAMI*, vol. 33, no. 2, pp. 394–405, 2010.

[29] F. Shi, H. Wu, X. Tong, and J. Chai, "Automatic acquisition of high-fidelity facial performances using monocular videos," *ACM ToG*, vol. 33, no. 6, p. 222, 2014.

[30] D. Frolova, D. Simakov, and R. Basri, "Accuracy of spherical harmonic approximations for images of lambertian objects under far and near lighting," in *ECCV*, 2004, pp. 574–587.

[31] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. Gortler, "A local/global approach to mesh parameterization," in *Computer Graphics Forum*, vol. 27, no. 5, 2008, pp. 1495–1504.

[32] A. Jacobson, D. Panozzo *et al.*, "libigl: A simple C++ geometry processing library," 2018, http://libigl.github.io/libigl/.

[33] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Li, "Face alignment across large poses: A 3d solution," in *CVPR*, 2016, pp. 146–155.

[34] D. Ma, J. Correll, and B. Wittenbrink, "The chicago face database: A free stimulus set of faces and norming data," *Behavior research methods*, vol. 47, no. 4, pp. 1122–1135, 2015.

[35] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

[36] Z. Feng, P. Huber, J. Kittler, P. Hancock, X. Wu, Q. Zhao, P. Koppen, and M. Rätsch, "Evaluation of dense 3d reconstruction from 2d face images in the wild," in *FG*, 2018, pp. 780–786.