

Learning Free-Form Deformation for 3D Face Reconstruction from In-The-Wild Images

Harim Jung¹, Myeong-Seok Oh², and Seong-Wan Lee¹

Abstract—The 3D Morphable Model (3DMM), which is a Principal Component Analysis (PCA) based statistical model that represents a 3D face using linear basis functions, has shown promising results for reconstructing 3D faces from single-view in-the-wild images. However, 3DMM has restricted representation power due to the limited number of 3D scans and global linear basis. To address the limitations of 3DMM, we propose a straightforward learning-based method that reconstructs a 3D face mesh through Free-Form Deformation (FFD) for the first time. FFD is a geometric modeling method that embeds a reference mesh within a parallelepiped grid and deforms the mesh by moving the sparse control points of the grid. As FFD is based on mathematically defined basis functions, it has no limitation in representation power. Thus, we can recover accurate 3D face meshes by estimating the appropriate deviation of control points as deformation parameters. Although both 3DMM and FFD are parametric models, deformation parameters of FFD are easier to interpret in terms of their effect on the final shape. This practical advantage of FFD allows the resulting mesh and control points to serve as a good starting point for 3D face modeling, in that ordinary users can fine-tune the mesh by using widely available 3D software tools. Experiments on multiple datasets demonstrate how our method successfully estimates the 3D face geometry and facial expressions from 2D face images, achieving comparable performance to the state-of-the-art methods.

Index Terms—3D face reconstruction, Free-form deformation, 3D morphable model

I. INTRODUCTION

The task of inferring the 3D face geometry and appearance from 2D images has been extensively explored in computer vision and graphics research, as it is essential for many face-related tasks and applications such as face recognition, anti-spoofing, tracking, virtual and augmented reality, animation, and gaming. Recovering a 3D face mesh from a single unconstrained in-the-wild image is especially challenging, due to large head pose variations, extreme expressions, occlusions, lighting conditions, and complex backgrounds.

Research in computer vision has quickly advanced and has been widely used in various applications [1]–[5]. Recently, considerable improvement has been made in 3D face reconstruction, with the help of Convolutional Neural Networks (CNNs). Previous research on 3D face reconstruction can be

mainly categorized into model-based and model-free methods. The statistical PCA-based face model, so-called the 3D Morphable Model (3DMM), has established the foundations of model-based methods. 3DMM is a globally linear model, where the face shape is represented as a linear combination of basis meshes obtained from a set of collected 3D face scans. Lately, many face reconstruction methods began to employ CNNs to regress 3DMM parameters [5]–[9].

On the other hand, model-free methods do not rely on a predefined face model but directly regress 3D vertices using volumetric representations [10] or UV position maps [11] for example. The idea of nonlinear 3DMM [12], [13] was also introduced, where the nonlinear decoder of a deep neural network maps the shape and texture parameters to the 3D shape and texture. Since the decoder forms the final mesh through direct vertices regression rather than the parameters, it can be considered as model-free. While 3DMM has a model space restricted to the distribution of a specific set of 3D face scans, model-free methods do not have limited representation power. Nevertheless, they tend to be computationally inefficient due to the high degrees of freedom.

In order to address the limitations of model-based and model-free methods, we propose a learning-based 3D face reconstruction method that uses Free-Form Deformation (FFD) [14], for the first time to the best of our knowledge. FFD is a well-known geometric modeling technique. It embeds a reference mesh within a parallelepiped grid and deforms the mesh by shifting the control points of the grid. Since our FFD-based method does not have limited representation power nor excessively high degrees of freedom, it can be seen to fall into the category between model-based and model-free methods. It is “model-free” in the sense that it does not reflect actual face scans as in 3DMM. Our method tends to have relatively more parameters to learn than 3DMM-based methods for this reason. On the other hand, it is “model-based” in the sense that it does not directly use vertices to represent a mesh but lower dimensional control points and mathematically defined basis functions. Thus, it requires less parameters than direct vertices regression. Our goal is to train the network to find the proper deviation of control points, which deforms the reference mesh to be similar to the target face. We summarize our main contributions as follows:

- We explore how FFD can be applied to 3D face meshes in the context of deep learning. Our method attempts to discover the appropriate number of control points, their distribution, and their range of influence over vertices.
- While 3DMM-based methods tend to be restricted in the

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program (Korea University)).

¹H. Jung and S.-W. Lee are with the Department of Artificial Intelligence, Korea University, Seoul 02841, South Korea. hr.jung@korea.ac.kr and sw.lee@korea.ac.kr

²M.-S. Oh is with the Department of Computer and Radio Communications Engineering, Korea University, Seoul 02841, South Korea. ms.oh@korea.ac.kr

spanned model space of linear bases, our method has no limit in representation power and generalizes well for unseen faces, as FFD is based on mathematically defined basis functions rather than PCA basis functions.

- Our method can be readily utilized for practical purposes, since the reconstructed mesh and control points can provide a solid starting point for 3D face modeling and can be easily modified for detailed adjustments by using widely available 3D software tools.
- Our method either outperforms or achieves comparable results to existing 3D face reconstruction methods, both in quantitative and qualitative experiments.

II. RELATED WORK

A. 3D Face Reconstruction from a Single Image

Since Blanz *et al.* [15] proposed the 3DMM, it has played a dominant role in 3D face modeling and reconstruction. Later on, more advanced variants of 3DMM were designed utilizing larger 3D scan databases and higher dimensional basis [16]–[19]. Most recent methods use CNNs to regress the 3DMM parameters in a supervised manner, which are used to reconstruct 3D faces [5]–[7]. Moreover, 3DMM parameters may be found through unsupervised methods [8], [9] without the help of training data. Sanyal *et al.* [8] proposed a novel shape consistency loss that induces the face shape to be similar for images of the same person and dissimilar for different people.

Model-based methods, however, have limited representation power and model-free methods [10], [11], [20] were proposed to overcome this limitation. Jackson *et al.* [10] proposed to map image pixels to a volumetric representation, while Feng *et al.* [11] developed a UV position map to represent the 3D shape. In a similar fashion, Deng *et al.* [20] directly regressed 3D vertex coordinates in the image space. Tran *et al.* [13] first introduced the concept of nonlinear 3DMM, where the decoders act as nonlinear models that map the parameters to the actual 3D shape and texture.

B. 3D Shape Reconstruction using Free-Form Deformation

There have been previous works that attempted to learn FFD for 3D shape reconstruction for rigid objects. Kuryenkov *et al.* [21] first searched for the nearest shape template from a database and applied FFD to manipulate the template to match the target image. Jack *et al.* [22] proposed a similar approach where they learned to deform points sampled from high-quality meshes.

To the best of our knowledge, FFD has never been applied to learning-based 3D face reconstruction, which is fairly different from general object reconstruction. The object reconstruction methods [21], [22] incorporated FFD using Bernstein basis functions, which is appropriate for rigid shapes, as its control points impose global influence on vertices. This method is not necessarily suitable for deformable shapes such as the human face, so we further experiment with FFD using B-spline basis functions. Moreover, they did not consider the pose of objects by using images obtained from the same viewpoint. However, considering the head

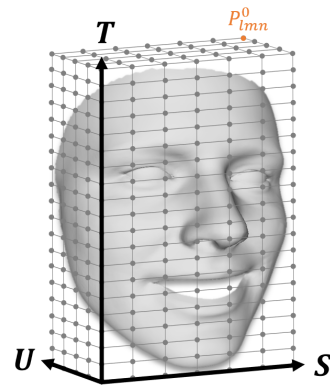


Fig. 1. Reference face mesh with frontal pose embedded in the 3D parallelepiped grid of 700 control points ($l = 6, m = 19, n = 4$). Each point on the intersections represents a control point and P_{lmn}^0 refers to the last control point, where l, m, n are the number of divisions along the S, T, U axes respectively. The control points are displaced from their original positions to deform the shape of the embedded reference mesh to match the target face.

orientation is important in the face domain, since human faces tend to have more obvious changes in pose. For this reason, we train our model to not only regress deformation parameters but also camera projection parameters.

III. METHOD

A. Deforming 3D Face Mesh with Free-Form Deformation

The goal of our method is to reconstruct the dense 3D face geometry from a single in-the-wild image. Since there are numerous vertices in a mesh, using all vertices as free variables to represent a mesh would be computationally inefficient. The more fundamental problem with this approach, however, is that these variables are not all free because the vertices should be constrained by each other to construct a mesh. Thus, we need to define new variables whose degrees of freedom are just enough to represent a mesh. For this purpose, we use FFD to represent and manipulate 3D meshes.

FFD is a shape modification method that has been widely used for geometric modeling in computer graphics and is supported by almost all 3D softwares. It embeds a reference mesh in a parallelepiped grid and deforms it by moving the control points of the grid. Each vertex of a deformed mesh is computed by a linear combination of control points by means of coefficient basis functions [14]. Originally, FFD was implemented by using Bernstein polynomials as basis functions [23], which we refer to as Bernstein FFD. In this approach, each vertex of the mesh is influenced by all control points of the grid. Therefore, we further experiment with FFD using B-spline basis functions [14], in which each vertex is influenced by only a small number of neighbor control points. We refer to this method as B-spline FFD.

An arbitrary volume $v(s, t, u)$ can be represented by taking a linear combination of control points using B-spline polynomials as coefficients, defined as follows:

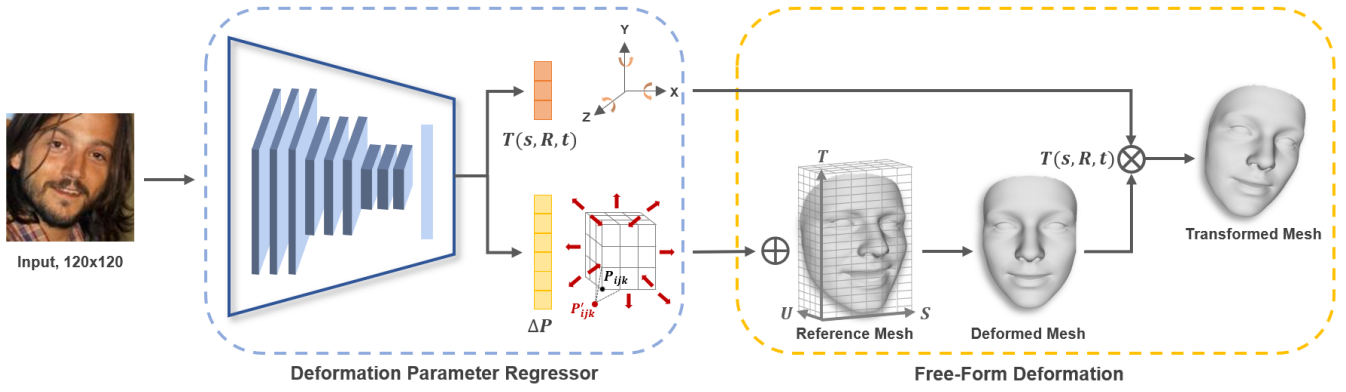


Fig. 2. Overview of our method. The Deformation Parameter Regressor includes the ResNet-50 backbone, which regresses the pose parameters $T(s, R, t)$ and the deformation parameters ΔP of 700 control points. The grid on the right of ΔP illustrates the possible deviation of control points in all directions of the red arrows and as an example, the left-bottom control point P_{ijk} could be moved to the position of P'_{ijk} , which would effect the reference mesh accordingly. The Free-Form Deformation part takes in the predicted ΔP , which is added to the original control points P^0 of the reference mesh, and the displaced control points are multiplied to B^0 , to obtain the deformed mesh in the world coordinate system, as in Eq. (2). $T(s, R, t)$ is a 3D scaled orthographic projection (3×4 affine transformation), which is multiplied to the deformed mesh to output a transformed mesh in the camera coordinate system. R, t are the rotation and translation parameters and s is the scale factor applied to all x, y, z directions.

$$\begin{aligned}
 v(s, t, u) &= \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n B_{ijk}(s, t, u) P_{ijk}, \\
 B_{i,j,k}(s, t, u) &= \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n B_{i,p}(s) B_{j,p}(t) B_{k,p}(u), \\
 \text{where } 0 \leq s, t, u &\leq 1.
 \end{aligned} \tag{1}$$

Here $B_{i,p}(s)$ is a B-spline basis function of degree p defined over the knot spans dividing the range of s . Basis functions $B_{j,p}(t)$ and $B_{k,p}(u)$ are similarly defined. In this work, we use B-spline functions of degree 3, that is, $p = 3$. The number of control points are $(l + 1)$, $(m + 1)$, and $(n + 1)$ in each direction. The control points P_{ijk} are distributed in a lattice structure where the space between the control points is nonuniform in general, but we particularly use a uniform grid with different dimensions on each axis.

We choose one of the mesh data from 300W-LP [5] as the reference mesh, as shown in Fig. 1. Note that the reference mesh can be any face mesh in the world coordinate system, since the goal is to find the appropriate deformation to reach the target mesh. To represent the reference mesh in the context of FFD, we need to define a parametric representation of the mesh by obtaining the values of the parameters (s_q, t_q, u_q) corresponding to each vertex (x_q, y_q, z_q) . Considering the relationship between each vertex and the parameters expressed in the following Eq. (2), we can obtain these parameters by solving the nonlinear equations:

$$\begin{aligned}
 (x_q, y_q, z_q) &= v(s_q, t_q, u_q) \\
 &= \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n B_{ijk}^0(s_q, t_q, u_q) P_{ijk}^0, \\
 \text{for each } (x_q, y_q, z_q) &\in \text{RefMesh},
 \end{aligned} \tag{2}$$

where RefMesh refers to the reference mesh and P^0 refers to the control points of the undeformed, initial grid embedding

the reference mesh. $B_{ijk}^0(s_q, t_q, u_q)$ is the B-spline coefficient function computed with respect to the reference mesh, which has the property that given a vertex (x_q, y_q, z_q) , it has a large value if the control point P_{ijk} is close to the vertex and a small value if it is far from the vertex.

By representing Eq. (2) in a matrix multiplication form, the reference mesh V^0 is represented in terms of the coefficient matrix B^0 and the control points P^0 as follows:

$$V^0 = B^0 P^0, \tag{3}$$

that is,

$$\begin{bmatrix} v^0(s_1, t_1, u_1) \\ \vdots \\ v^0(s_N, t_N, u_N) \end{bmatrix} = \begin{bmatrix} B_{000}^0(s_1, t_1, u_1) \dots B_{lmn}^0(s_1, t_1, u_1) \\ \vdots \\ B_{000}^0(s_N, t_N, u_N) \dots B_{lmn}^0(s_N, t_N, u_N) \end{bmatrix} \begin{bmatrix} P_{000}^0 \\ \vdots \\ P_{lmn}^0 \end{bmatrix}. \tag{4}$$

In Eq. (3), V^0 represents the reference mesh with N vertices and $B^0 \in \mathbb{R}^{N \times M}$ is the B-spline coefficient matrix which expresses the influence of each control point on each vertex of the mesh. $P^0 \in \mathbb{R}^{M \times 3}$ expresses the 3D coordinates of M control points. Once the coefficient matrix B^0 is computed using the reference mesh V^0 , any mesh V can be represented by deforming the control points as follows:

$$V(\Delta P(I^i)) = B^0(P^0 + \Delta P(I^i)), \tag{5}$$

where I^i indicates an arbitrary input image. Given a list of training data pairs of image I^i and its corresponding ground truth mesh V^i , we train the neural network to extract image features and predict the appropriate deformation $\Delta P(I^i)$ of the control points from those features, so that the deformed mesh V matches the ground truth mesh. All deformation of the mesh happens so that the vertex indices and triangle connectivity remain the same. The overview of our proposed method is shown in Fig. 2.

B. Applying Camera Projection to 3D Face Mesh

Our model regresses pose parameters, as well as deformation parameters. It is a natural approach to consider the shape deformation in the world coordinate system and the camera projection separately. $T(s, R, t)$ indicates a 3D scaled orthographic projection (3×4 affine transformation). The scale factor s is applied to x, y, z directions. This projection is called 3D in that it preserves the depths (z -values) of vertices, scaling them in the same factor as in the x and y directions. The transformation matrix $T(s, R, t)$ is multiplied to the deformed mesh to output the final transformed deformed mesh in the camera coordinate system.

C. Loss Functions

1) Vertex Loss

Since the ground truth mesh and the predicted mesh share the same topology, it is possible to calculate the distance between the vertices that share the same indices. The Mean Squared Error (MSE) of the vertex loss is defined as:

$$L_{Vertex} = \frac{1}{N} \sum_{i=0}^N (T(V(\Delta P(I^i))) - T_g(V^i))^2, \quad (6)$$

where N refers to the number of data, V^i refers to the ground truth mesh of the training image I^i , and V refers to Eq. (5) which outputs the deformed mesh from the predicted ΔP . T is a 3×4 transformation matrix and T_g is the ground truth pose.

2) Landmark Region Loss

There exists a set of predefined vertex indices corresponding to 68 facial landmarks. If we simply use MSE, each facial region would be weighted depending on the number of landmarks in each region. In order to control the importance of each region, we divide the 68 landmarks into 9 different regions and form a weighted loss. The regions are divided into the left and right eyebrow, left and right eye, upper and lower nose, upper and lower lip, and contour. The loss of each landmark region, abbreviated as $LMRegion$, can be computed by sampling the mesh with landmark region indices, abbreviated as $LMIndex$ in the following equation.

$$L_{LMRegion} = \frac{1}{N} \sum_{i=0}^N \frac{1}{M} \sum_{j \in LMIndex} (T(V_j(\Delta P(I^i))) - T_g(V_j^i))^2, \quad (7)$$

where N refers to the number of data, M refers to the number of landmarks of a specific region, and V_j refers to the sampled vertices using $LMIndex$. The total loss can be computed as the weighted average of L_{Vertex} defined in Eq. (6) and all $L_{LMRegion}$ defined in Eq. (7). In our implementation, the weight of the vertex loss is set to 0.46, and the weight of each landmark region loss is set to 0.06.

IV. EXPERIMENTS

A. Datasets and Protocols

1) 300W-LP

300W-LP [5] is composed of synthesized large-pose face images. The dataset consists of pairs of face images and

3DMM parameters, which were found by fitting a 3DMM built from a combination of the Basel Face Model [17] and FaceWarehouse [16]. We used an augmented version of 300W-LP [24] for training, which has a more variety of extreme poses. Although the ground truth face meshes are based on 3DMM, our method has the capacity to generate deformed meshes beyond the linearly spanned space of 3DMM because the reconstructed mesh is not represented by 3DMM parameters, but by the deformation parameters.

2) AFLW2000-3D

AFLW2000-3D [5] contains the first 2,000 images of AFLW [25], with ground truth 3DMM parameters defined in a consistent manner with 300W-LP [5] and 68 3D landmarks. The dataset consists of images with large pose variations of yaw angles ranging from -90° to 90° and with variations of illumination conditions and facial expressions. Thus, it is often used to evaluate the 3D face reconstruction performance on challenging in-the-wild images.

Quantitative evaluation of 3D face reconstruction for in-the-wild images is challenging due to the lack of pairs of 2D images and 3D models collected in an unconstrained setting. As an alternative, evaluation of facial landmarks can be indicative of the overall 3D face reconstruction accuracy, especially when considering both visible and invisible landmarks. We evaluated our method using facial landmark data of AFLW2000-3D. The protocol followed [5], [6], [11] to calculate the Normalized Mean Error (NME) normalized by the bounding box size. The result is reported by the range of yaw angles, which is divided into $[0^\circ, 30^\circ]$, $[30^\circ, 60^\circ]$ and $[60^\circ, 90^\circ]$. Following the works of [5], [11], we randomly sampled 696 faces to balance the distribution so that each pose range has the same number of data.

3) CelebA

CelebA [26] is a large-scale face dataset with over 200K in-the-wild images of celebrities. The images have large variations in pose and background clutter. We use the provided aligned images resized to 218×178 as our test data.

We use CelebA to conduct a qualitative evaluation of our model. As shown in Fig. 3, reconstructed 3D face mesh is rendered on top of the input image using scaled orthographic projection. The rendered result shows the resemblance between the original face of the 2D image and the deformed face mesh.

B. Implementation Details

The training and experiments were implemented based on Pytorch [27]. We employed a ResNet-50 architecture [2] as the backbone of our network. At the end, we implemented a fully connected layer with 2,112 nodes to regress the deformation of 700 control points in 3D coordinates and 12 pose parameters for the transformation matrix. We divided the grid uniformly but with different dimensions on each axis ($l = 6, m = 19, n = 4$).

For training, we used an augmented version of the 300W-LP dataset [24] with 687,854 images, which were obtained through applying random perturbations to pitch, yaw, and roll angles. Then, the images were cropped around the facial

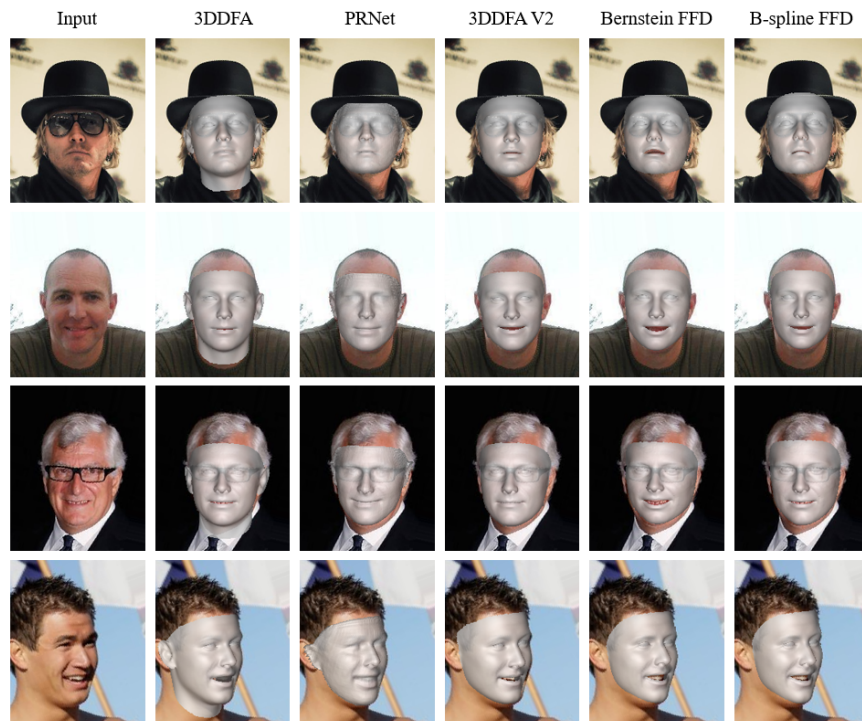


Fig. 3. Qualitative results of our method on CelebA compared to 3DDFA [5], PRNet [11] and 3DDFA V2 [6]. Our results of two separate cases were reported, each using Bernstein FFD and B-spline FFD. B-spline FFD showed better results on the mouth area, since only considering neighbor control points and imposing local influence on vertices allow for finer control. B-spline FFD also outperformed other existing 3D face reconstruction methods.

TABLE I

COMPARISON BETWEEN BERNSTEIN FFD AND B-SPLINE FFD ON AFLW2000-3D. THE NME (%) OF FACIAL LANDMARKS WITH DIFFERENT YAW ANGLES ARE REPORTED.

Method	0° to 30°	30° to 60°	60° to 90°	Mean
Bernstein FFD	2.97	3.70	4.90	3.86
B-spline FFD	2.60	3.44	4.50	3.51

region and resized to 120×120 . Finally, the images were normalized by the RGB mean and standard deviation. We constructed the mesh data with 35,709 vertices covering the face area, excluding the ears and neck, using the provided 3DMM parameters. Our model was trained with the Adam optimizer using a learning rate of 0.001 and weight decay of 0.0005, for 50 epochs with a mini-batch size of 128.

C. Comparison between Bernstein FFD and B-spline FFD

We have experimented with FFD using both Bernstein [23] and B-spline basis functions [14], where the control points of Bernstein FFD have global influence on vertices while those of B-spline FFD have local influence. Empirically, we observed that one of the most challenging issues is properly controlling the mouth area, mostly due to the opened space between the lips. In fact, the qualitative results shown in Fig. 3 demonstrate that Bernstein FFD could hardly control the lips. We observed B-spline FFD yielded better results, since the human face is a deformable shape and requires local control. Furthermore, Table I shows that B-spline FFD

TABLE II

PERFORMANCE EVALUATION OF 68 LANDMARKS ON AFLW2000-3D. THE NME (%) WITH DIFFERENT YAW ANGLES ARE REPORTED. THE BEST RESULTS ARE HIGHLIGHTED.

Method	0° to 30°	30° to 60°	60° to 90°	Mean
SDM [28]	3.67	4.94	9.76	6.12
3DDFA [5]	3.78	4.54	7.93	5.42
3DDFA + SDM [5]	3.43	4.24	7.17	4.94
DeFA [7]	-	-	-	4.50
Yu <i>et al.</i> [29]	3.62	6.06	9.56	6.41
3DSTN [30]	3.15	4.33	5.98	4.49
3D-FAN [31]	3.15	3.53	4.60	3.76
PRNet [11]	2.75	3.51	4.61	3.62
CMD [12]	-	-	-	3.98
3DDFA V2 [6]	2.63	3.42	4.48	3.51
B-spline FFD	2.60	3.44	4.50	3.51

achieved higher accuracy on predicting facial landmarks in general. For fair comparison, both methods were tested with the same number of control points and grid dimensions.

D. Quantitative Results

Table II reports the NME of facial landmarks compared to existing methods including SDM [28], 3DDFA [5], DeFA [7], Yu *et al.* [29], 3DSTN [30], 3D-FAN [31], PRNet [11], CMD [12], and 3DDFA V2 [6]. Our method, B-spline FFD, showed better accuracy than most of the compared methods and achieved comparable performance to 3DDFA V2 [6]. It is worth noting that the landmarks are inherently a subset

of 3DMM fitted meshes, so 3DMM-based methods tend to show better results.

E. Qualitative Results

Our reconstruction results of Bernstein FFD and B-spline FFD on the images of the CelebA dataset can be viewed at Fig. 3, in comparison to 3DDFA [5], PRNet [11], and 3DDFA V2 [6]. The qualitative evaluation demonstrates that our model is robust to large poses, not only frontal poses, as illustrated in the last row. In addition, it can reconstruct 3D meshes even when part of the face is occluded by glasses, as shown in the first and third row. Overall, this experiment validated that our model works well for in-the-wild images with various lighting, expressions, poses, and occlusions.

V. CONCLUSION

In this paper, we proposed a new approach to learning-based 3D face shape reconstruction through B-spline FFD. FFD can be considered as both model-based and model-free; the mesh is represented by low dimensional control points and polynomial basis functions, but still has no limit in expressiveness or model space. Moreover, the regressed deformation parameters are intuitive and self-explanatory. This further allows one to readily utilize the estimated mesh and control points for detailed adjustment and further deformation. Lastly, our model is meaningful in that it achieved comparable results to state-of-the-art methods of 3D face reconstruction from a single in-the-wild image.

REFERENCES

- [1] H.-D. Yang and S.-W. Lee, "Reconstruction of 3D human body pose from stereo image sequences based on top-down learning," *Pattern Recognition*, vol. 40, no. 11, pp. 3120–3131, 2007.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [3] S.-W. Lee, J. H. Kim, and F. C. Groen, "Translation-, rotation- and scale-invariant recognition of hand-drawn symbols in schematic diagrams," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 4, no. 1, pp. 1–25, 1990.
- [4] Y.-K. Lim, S.-H. Choi, and S.-W. Lee, "Text extraction in MPEG compressed video for content-based indexing," in *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 4, 2000, pp. 409–412.
- [5] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3D solution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 146–155.
- [6] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li, "Towards fast, accurate and stable 3D dense face alignment," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 152–168.
- [7] Y. Liu, A. Jourabloo, W. Ren, and X. Liu, "Dense face alignment," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1619–1628.
- [8] S. Sanyal, T. Bolkart, H. Feng, and M. J. Black, "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, pp. 7763–7772.
- [9] A. Tewari, M. Zollhofer, H. Kim, P. Garrido, F. Bernard, P. Perez, and C. Theobalt, "MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1274–1283.
- [10] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, "Large pose 3D face reconstruction from a single image via direct volumetric cnn regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1031–1039.
- [11] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3D face reconstruction and dense alignment with position map regression network," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 534–551.
- [12] Y. Zhou, J. Deng, I. Kotsia, and S. Zafeiriou, "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, pp. 1097–1106.
- [13] L. Tran and X. Liu, "Nonlinear 3D face morphable model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7346–7355.
- [14] W. M. Hsu, J. F. Hughes, and H. Kaufman, "Direct manipulation of free-form deformations," *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 177–184, 1992.
- [15] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 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," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 413–425, 2013.
- [17] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proceedings of the IEEE International Conference on Advanced Video and Signal-based Surveillance*, 2009, pp. 296–301.
- [18] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniah, and D. Dunaway, "A 3D morphable model learnt from 10,000 faces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5543–5552.
- [19] A. Shin, S.-W. Lee, H. Bülthoff, and C. Wallraven, "A morphable 3D-model of Korean faces," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2012, pp. 2283–2288.
- [20] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5203–5212.
- [21] A. Kuryankov, J. Ji, A. Garg, V. Mehta, J. Gwak, C. Choy, and S. Savarese, "Deformnet: Free-form deformation network for 3D shape reconstruction from a single image," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2018, pp. 858–866.
- [22] D. Jack, J. K. Pontes, S. Sridharan, C. Fookes, S. Shirazi, F. Maire, and A. Eriksson, "Learning free-form deformations for 3D object reconstruction," in *Proceedings of the Asian Conference on Computer Vision*, 2018, pp. 317–333.
- [23] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, 1986, pp. 151–160.
- [24] J. Guo, X. Zhu, and Z. Lei, "3DDFA," <https://github.com/cleardusk/3DDFA>, 2018.
- [25] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2011, pp. 2144–2151.
- [26] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3730–3738.
- [27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [28] X. Xiong and F. De la Torre, "Global supervised descent method," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2664–2673.
- [29] R. Yu, S. Saito, H. Li, D. Ceylan, and H. Li, "Learning dense facial correspondences in unconstrained images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4723–4732.
- [30] C. Bhagavatula, C. Zhu, K. Luu, and M. Savvides, "Faster than real-time facial alignment: A 3D spatial transformer network approach in unconstrained poses," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3980–3989.
- [31] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks)," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1021–1030.