Pixel2Mesh for 3D Face Reconstruction

Yiwen Liu ge42lor@tum.de
Yunhe Nie ge95yaf@tum.de
Pengcheng Yu ge95hej@mytum.de
Keyue Zhang ge42zab@mytum.de

Abstract

Pixel2mesh is an end-to-end deep learning architecture that generates 3D triangular meshes from single-color images. The majority of previous works represent 3D shapes in volumes or point clouds, while the Pixel2Mesh network represents 3D shapes in meshes, which are essentially well graph suited for graph-based convolutional neural networks. Besides, this network has considerable potential to be applied in specific domains. Our group is committed to adapting pixel2mesh in the field of face generation and improving its performance through a series of optimizations. Codes can be found on the GitHub[1].

1. Introduction

3D reconstruction is an important research content in the field of computer vision, there are also plenty of applications in industry. Among those 3D reconstruction models, 3D face reconstruction has always been a heated direction. High-quality reconstruction of 3D faces is of great significance in the fields of face recognition, game entertainment, medical treatment, etc. Therefore, we hope to explore the face reconstruction model based on the traditional pixel2mesh network in this project. Our contribution is summarized in:

1) Focus on face reconstruction instead of general object reconstruction.

2) Improve the baseline model.

3) Ablation study within our model and compare related criteria with original P2M.

2. Related Works

2.1. Learning-Based 3D Shape Reconstruction

After years of development, there are many mature models for 3D reconstruction task, among which Scan2mesh [4] converts unstructured and possibly incomplete range scans into structured 3D mesh representations. Another classic baseline for the 3D reconstruction task is Pixel2mesh [14], it extracts features from a single 2D RGB image and generates the corresponding 3D mesh.

2.2. 3D Face Reconstruction

Blanz and Vetter [1] came up with the first significant contribution, 3D Morphable Face, which marked the evolution of facial reconstruction. Recent methods based on neural networks have shown their effectiveness in producing reconstructions rich in detail [5]. We think improving P2M model to face field is also a feasible solution.

3. Methods

3.1. Network Architecture

Our network includes three main parts: a pre-processing network, a 2D face feature extracting network, and a 3D deformation network. The whole network structure can be found in Figure 1.

3.1.1 Pre-processing Network

Our main consideration is to help the 2D network better extract useful features from the not clean-background faces and reduce the useless computation complexity. Additionally, some researchers have done experiments showing the classical P2M fails to reconstruct the furniture from noisy-background images, which supports our pre-processing network to detect and crop the input faces [10].

We have considered two different schemes. We first consider base on the Mask Scoring R-CNN [7], which uses Resnet101 and FPN as the RPN network with RolAilgn to help us find the precise location of faces [6, 10]. Although it can extract features for the 3D network and combine the first two networks, our Pixel2Mesh baseline is complex enough that no more complexity can be added.

With the help of the face recognition area is a very active field, we can use some very mature face-detection networks. Finally, the pre-processing network contains an MTCNN face detection network [16] to quickly detect the face regions and uses the information from bounding boxes to crop the local face region. We can see an example in the upper left corner of Figure 1.

[1] https://github.com/ML3Dproject/Pixel2Mesh_original-pytorch

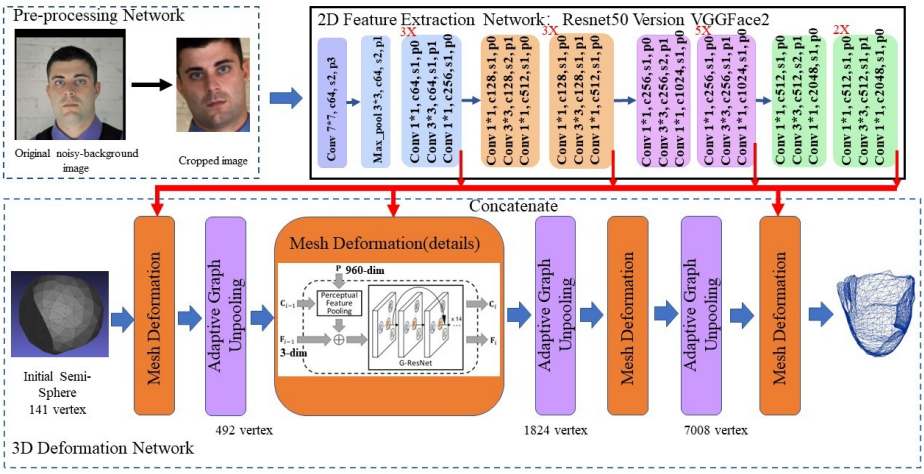

Figure 1. **Overall structure of the model:** The three main networks and some detailed information are marked separately. "c" means the number of output channels, "s" means the size of the stride, "p" means the size of the padding. [13, 14]

Note that we expand the size of the box by a factor of 1.1 so that it can better frame the entire face. Besides, for the faces that cannot be detected by MTCNN, we do average cropping on the original image using information from detected images. And for two faces in one image, we use the detection likelihood to crop the max likelihood face.

3.1.2 2D Face Feature Extracting Network

After the above analysis, we replace the original VGG-16 2D backbone with the VGGFace2 backbone [2]. It is a Resnet50 backbone that firstly pre-trained on MS1M and then fine-tuned on the VGGFace2 dataset. The original task for the VGGFace2 network is to classify 8631 faces and match the corresponding names. It will be more specialized to extract features from the crop faces images than the general VGG-16 backbone. Figure 1 clearly shows the 2D network and fusion ways.

3.1.3 3D Deformation Network

The overall structure is similar to the classical Pixel2Mesh backbone. We follow the basic mesh deformation and graph unpooling structure [2]. But we change the original ellipsoid into a semi-sphere as the initial deformation mesh, which has 141 vertices and is created using Blender [8]. Because the 3D point cloud ground truths only contain the front face surface and we can reduce the computation complexity by using fewer vertices in initial mesh [11].

In the mesh deformation block, we still use perceptual feature pooling to combine the 2D and 3D information [14]. But the new dataset uses parallel projection, not orthogonal projections. So we change the corresponding fusion method. After fusion and concatenation with the initial coordinate of 3 dimensions, the 963-dimension feature is fed into the G-ResNet Block and outputs the 3D coordinates of the new vertex. The detailed information can be found in Figure 1.

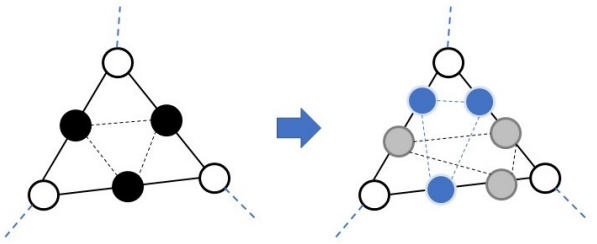

Figure 2. **Original Graph Unpooling(left), Adaptive Unpooling(right)**: The original way to add new vertices is at the center of each edge. Improved Adaptive Unpooling can let the model learn the position of interpolated vertices. The gray nodes and blue nodes are both two possible positions.

After each deformation block, there is a graph unpooling block to add more vertices in the initial mesh [14]. We let the locations of new points in edges be learnable parameters. Figure 2 compares the original unpooling method and our adaptive unpooling. Instead of increasing uniformly everywhere, we can have more points in the face details where more points are needed, and sparser points where more points are not needed.

Due to faces having more details and a more complex topology structure, we decide to add one more deformation block and unpooling block so that the mesh can be fully deformed to capture the detail of faces.

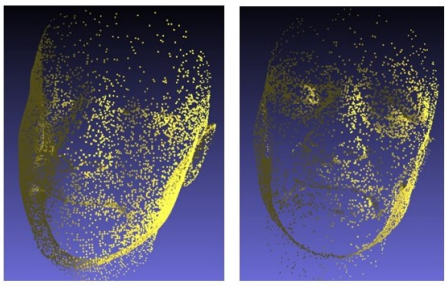

Figure 3. **Uniform Sampling(left), Weighted Sampling(right)**: We can see that more points are clustered in important locations. So these positions are more specific when computing the losses.

3.2. Loss Functions

We follow the four main loss functions used in [14]. Using chamfer loss and normal loss to control the location of vertices and quality of the mesh. And the Laplacian regularization and edge length regularization are responsible for controlling the deformation process [14]. The overall loss is a weighted sum of losses:

$$l_{all} = l_C + \lambda_1 l_n + \lambda_2 l_{lap} + \lambda_3 l_{mov} + \lambda_4 l_{loc}$$

with $\lambda_1 = 1.6e-4$, $\lambda_2 = 0.5$, $\lambda_3 = 0.033$, $\lambda_4 = 0.1$. And for chamfer loss, we give [0.5,1.,1.,1.] for four times deformations.

To help the model capture the detailed feature around eyes, noses, and mouths. We design two ways: The first one is that we can let the points around these areas have larger weight when computing loss [15]. But it is difficult for us to handle which points are near these areas when computing the loss and the chamfer loss is the external library, we cannot easily modify the source code.

As a result, we choose the second way to do weighted sampling in the 3D point cloud ground truth as a preprocessing step. Using the 68 pt3d landmarks from the dataset [17], we first query different distances around the landmarks and give each distance a weight to show its importance. In the data loader, we randomly sample different numbers of points in different regions. Also notes the query distances and sample numbers are also hyperparameters. We will discuss these in detail in Chapter 4.4.2. During computing the loss, we can push our vertex to deform toward these sampled points and control the vertex to be denser in key regions.

4. Experiments

In this section, some experimental results will be introduced and discussed. Firstly we trained our model on 6×RTX 2080 Ti GPUs multiple times for different configurations to figure out the effects of different model components or parameters. Secondly we compared our proposed model with the baseline model.

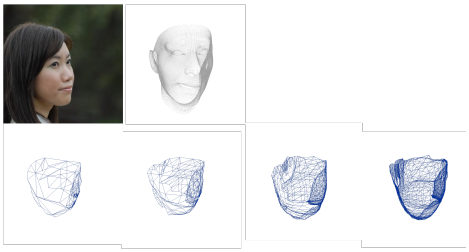

Figure 4. **Top:** input image and corresponding point cloud; **bottom:** four-time mesh deformation, which shows a coarse to fine process

4.1. Dataset

AFLW2000-3D dataset [17] is a very famous face dataset and is used for all our training. We also have prepared a much larger dataset 300W-LP [17], which contains more than 40,000 images, but due to time and device limits, we cannot do experiments on the larger one. The original AFLW2000-3D dataset contains 2000 3×450×450 RGB images of human faces in different angles, backgrounds, and lighting conditions. Some images contain multiple faces and the quality of the images are poor. We use BFM(Basel Face Model) [12] to generate the 3D point cloud ground truth, including vertices coordinates and normal vectors.

4.2. Evaluation

Chamfer Distance and F-score are used to measure the quality of the 3D shape reconstruction [10]. Lower is better for CD, and higher is better for F-score. Normalized Mean Error(NME) [9] is a very common criterion in face reconstruction field, while the normalized factor in NME is not unified for the different datasets, so the value of NME and other papers are not comparable. We finally abandon this criterion.

4.3. Results of Our Method

We use the standard split for AFLW2000 dataset [17], where 1600 instances are for training, 200 instances for validation, and 200 for testing. We use our improved model and set the batch size to 1 and the epoch to 80. We load part of the checkpoint of the original P2M to the 3D deformation network. The learning rate is initialized as 1e-4 and decays with factor 0.3 at epoch (25, 40, 60).

During training, our network takes the 2D images as input and corresponding point clouds as ground truth. During inference, our network takes the 2D images as input and predicts corresponding 3D meshes.

Figure 4 shows an example of the "coarse to fine" deformation process. We present the qualitative results

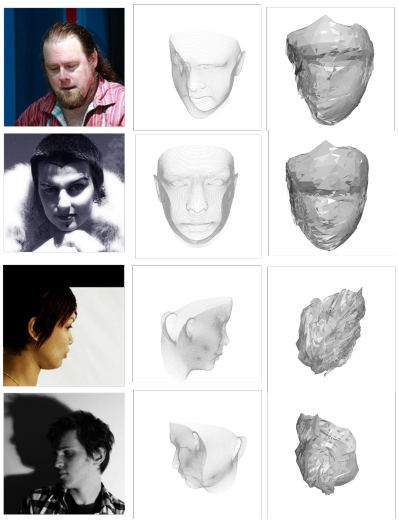

Figure 5. **Left:**input images; **middle:** corresponding point clouds; **right:** predicted meshes.

of our method in Figure 5, which show plausible mesh predictions of our methods, taking human face images from different angles as input.

Table 1 shows quantitative results on AFLW2000-3D, compared with the claimed performance of the original P2M model on the ShapeNet [3]. The F1 score of our method achieves comparable results compared with the original Pixel2Mesh. Note that in the AFLW2000 dataset, the number of points for each ground truth point cloud is around 60,000 and our network finally predicts 7008 vertices for each image input, while the original Pixel2Mesh paper predicts 2,468 points with around 10000 points as ground truth. For our model, when we compute the chamfer distance between predicted vertice coordinates and ground truth point clouds, for each point in both settings, it is more likely to find a better matching point with a lower distance, that's why the chamfer distance results show difference while the F1 scores are comparable.

Methods	Dataset	cd	$F1^{\tau}$	$F1^{2\tau}$
original	ShapeNet	0.482	65.22	78.80
ours	AFLW2000	0.039	57.28	77.54

Table 1. Different number of points in three regions.

4.4. Ablation Study

4.4.1 Experiments on Adaptive Unpooling

We add the Adaptive Unpooling(AU), introduced in section 3.1.3, to the baseline model to overfit a small part of the data (10 samples). The result is shown in Table 2, and the epoch by 10%, which proves that AU speeds up the model convergence. One possible explanation is that AU learns to move the vertices to correct places before mesh deformation takes effect. In larger datasets or more complex models, AU may have a greater impact.

Method	Without Adaptive Unpooling	Adaptive Unpooling
Epochs	5000	4500

Table 2. Results of epochs to achieve same evaluation criterion.

4.4.2 Experiments on weighted sampling

As mentioned in section 3.2, the query distances and weights are shown in the follows: (The distance unit is mm.)

$$Sample = \begin{cases} weight = 2 & x < 10 \\ weight = 1 & 10 < x <= 17 \\ weight = 0 & x > 17 \end{cases}$$

To verify the impact of the weighted sampling on training, we adopt three point-sampling methods in different weighted regions shown in Table 3, while keeping other model structures unchanged. Comparing the first two configurations, the quality of the reconstructed mesh and the evaluation performances are almost the same, which means the total sampling points have little impact on model training. For configurations 2 and 3, the reconstructed mesh has relatively much denser points around key regions, for example, the nose and eyes. However, too many vertices around the key regions will also lead to bad reconstruction performance. After experiments, the number of points in the above 3 regions is [11000,4000,3000].

Configuration	Face Boundaries (weight 0)	Sub-priority Area (weight 1)	Key Region (weight 2)	Total Number
1	11000	4000	3000	18000
2	5500	2000	1500	9000
3	2000	3000	4000	9000

Table 3. Different number of points in three regions.

4.4.3 Experiments on 2D pretrained model

We refined our model based on different 2D feature extractors: Resnet50 [13] and VGGFace2 pre-trained models. The information about VGGFace2 pre-trained models can be found in Chapter 3.1.2. The experiment shows that the model using VGGFace2 is slightly better than the model using ResNet50, and there is no essential difference overall.

5. Conclusion

In conclusion, our project ports the Pixel2Mesh to 3D face reconstruction field. We improve the baseline model structure and our methods can predict plausible human face meshes and achieve comparable results to the original Pixel2Mesh method used on ShapeNet. It shows that the modified Pixel2Mesh model is competent for more specialized object reconstruction.

Our method shares the same limitation as many of other baseline models: does not produce fine-scale details. In future work we can train the model in a larger dataset 300W-LP [17] to get a better generalization ability and improve the information extraction ability of 2D network using some 2D reconstruction methods.

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. pages 187–194, 1999. 1

[2] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018. 2

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4

[4] Angela Dai and Matthias Nießner. Scan2mesh: From unstructured range scans to 3d meshes. pages 5574–5583, 2019. 1

[5] Qixin Deng, Luming Ma, Aobo Jin, Huikun Bi, Binh Huy Le, and Zhigang Deng. Plausible 3d face wrinkle generation using variational autoencoders. *IEEE Transactions on Visualization and Computer Graphics*, 28(9):3113–3125, 2021. 1

[6] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795, 2019. 1

[7] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6409–6418, 2019. 1

[8] Brian R Kent. *3D scientific visualization with Blender®*. Morgan & Claypool Publishers, 2015. 2

[9] Shenqi Lai, Zhenhua Chai, Shengxi Li, Huanhuan Meng, Mengzhao Yang, and Xiaoming Wei. Enhanced normalized mean error loss for robust facial landmark detection. In *BMVC*, page 111, 2019. 3

[10] Xi Li, Kuang Ping, Xiaofeng Gu, and Mingyun He. 3d shape reconstruction of furniture object from a single real indoor image. pages 101–104, 2020. 1, 3

[11] Mehdi Malah, Mounir Hemam, and Fayçal Abbas. 3d face reconstruction from single image with generative adversarial networks. *Journal of King Saud University-Computer and Information Sciences*, 35(1):250–256, 2023. 2

[12] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee, 2009. 3

[13] Dhananjay Theckedath and RR Sedamkar. Detecting affect states using vgg16, resnet50 and se-resnet50 networks. *SN Computer Science*, 1:1–7, 2020. 2, 4

[14] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, pages 52–67, 2018. 1, 2, 3

[15] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Hang Yu, Wei Liu, Xiangyang Xue, and Yu-Gang Jiang. Pixel2mesh: 3d mesh model generation via image guided deformation. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3600–3613, 2020. 3

[16] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016. 1

[17] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z Li. Face alignment in full pose range: A 3d total solution. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):78–92, 2017. 3, 5