Paper Reading No.1

Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images

Sheng Lian

June 2019

1 Brief Paper Intro

- ***Paper ref:*** ECCV 2018, http://arxiv.org/abs/1804.01654
- ***Authors:*** See Figure 1
- ***Paper summary:*** generating 3D mesh from single RGB images using GCN-based model in a cascaded manner.
- ***Reading motivation:*** I want to introduce Graph Convolutional Networks (GCN) into medical image segmentation task. I can refer to this paper that refine segmentation mask in a cascaded manner from ellipse using GCN.

Nanyang Wang[1]*, Yinda Zhang[2]*, Zhuwen Li[3]*,
Yanwei Fu[4], Wei Liu[5], Yu-Gang Jiang[1]†

[1]Shanghai Key Lab of Intelligent Information Processing,
School of Computer Science, Fudan University
[2]Princeton University [3]Intel Labs [4]School of Data Science, Fudan University [5]Tencent AI Lab
nywang16@fudan.edu.cn yindaz@cs.princeton.edu lzhuwen@gmail.com
yanweifu@fudan.edu.cn wl2223@columbia.edu ygj@fudan.edu.cn

Figure 1: authors' brief intro.

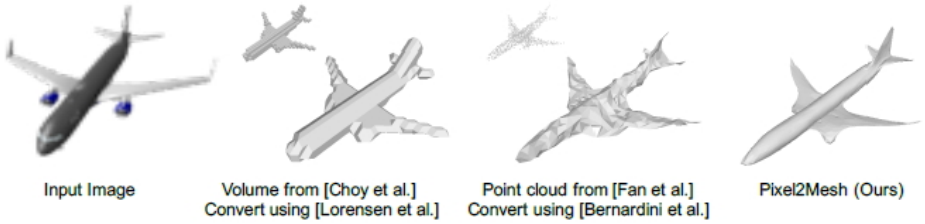

Figure 2: Comparison between 3 kinds of 3D representations: volume, point cloud, and mesh

2 Backgrounds

Inferring 3D shape from a single perspective is an extremely challenging task for computer vision community. The three most common 3D representations are **volume, point cloud**, and **mesh**. See Figure 2.

Brief comparison: Volume is limited by resolution, and it lacks a lot of details. For point cloud, there is no connection between points, and it lacks surface information of objects. In comparison, mesh has the characteristics of light weight and rich detail, and more importantly, easy to deform for animation.

3 Contributions/highlights

- An end-to-end neural network is proposed to generate 3D mesh from a single RGB image.
- This paper uses GCN to represent 3D mesh information. The features extracted from input images are adopted for ellipsoid mesh's deformation towards the correct geometry.
- This paper use a coarse-to-fine manner for stable deforming.
- Several loss functions have been proposed to make the model work better.

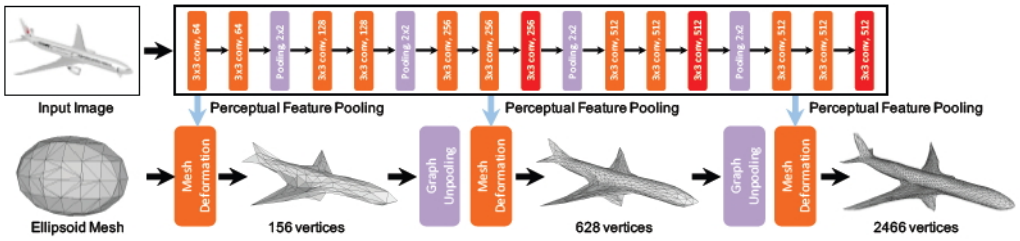

Figure 3: The framework of cascaded mesh deformation network.

4 Methods

4.1 Framework overview

The overview of the proposed framework is illustrated in Fig 3. The whole network consists a 2D CNN feature network and a cascaded GCN-based mesh deformation network. The main steps for this model are summarized as follows.

1) Given an input image

2) Initialize an ellipsoid mesh for the input image for 156 vertices using Meshlab [1]

3) The upper half of the model (which has similar structure as VGG16) is used for extracting 2D image features, which is leveraged by the mesh deformation network to progressively deform an ellipsoid mesh into the desired 3D model.

4) The second half is GCN-based cascaded mesh deformation network, which contains three deformation blocks intersected by two graph unpooling layers.

5) The graph unpooling layers increase the number of vertices to increase the capacity of handling details, while still maintain the triangular mesh topology.

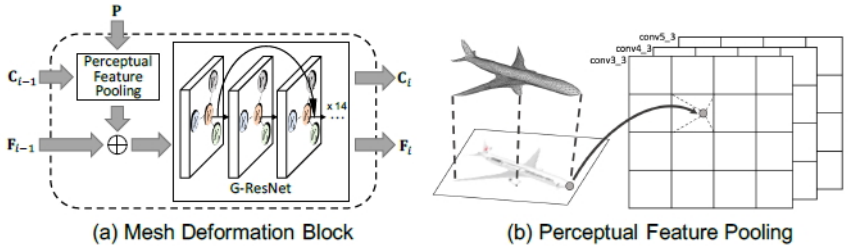

Fig. 3. (a) The vertex locations $\mathbf{C}_i$ are used to extract image features, which are then combined with vertex features $\mathbf{F}_i$ and fed into G-ResNet. $\bigoplus$ means a concatenation of the features. (b) The 3D vertices are projected to the image plane using camera intrinsics, and perceptual feature is pooled from the 2D-CNN layers using bilinear interpolation.

Figure 4:

4.2 Fusion of 2D & 3D info

In this paper, authors use perceptual feature pooling and mesh deformation block to leveraging 2D image information to help reconstructing 3D mesh. See Fig 4. C represents 3D vertex coordinates, P represents image features, and F represents 3D vertex features. The perceptual feature pooling layer is responsible for extracting corresponding information from the image feature P according to the 3D vertex coordinates C(i-1). The vertices extracted above are merged with the vertex feature F(i-1) at the previous moment as the input of following G-ResNet.

4.3 Losses

This paper define 4 kinds of losses for constraining the property of the output shape. Here, we use p for a vertex in the predicted mesh, q for a vertex in the ground truth mesh, $\mathcal{N}$(p) for the neighboring pixel of p, till the end of this section.

- **Chamfer loss** is used for measuring the distance of each vertex.

$$l_c = \sum_p \min_q \|p - q\|_2^2 + \sum_p \min_q \|p - q\|_2^2 \tag{1}$$

- **Normal loss** force the normal of a locally fitted tangent plane to be

consistent with the observation (smooth the surface)

$$l_n = \sum_p \sum_{q=\arg\min_p\left(\|p-q\|_2^2\right)} \|\langle p-k, \mathbf{n}_q\rangle\|_2^2 \tag{2}$$

- **Laplacian regularization** is used for preventing the vertices from moving too freely when deforming.

$$l_{lap} = \sum_p \left\|\delta'_p - \delta_p\right\|_2^2 \tag{3}$$

- **Edge lenth regularization** is adopted for penalizing flying vertices, which ususally cause long edge.

$$l_{loc} = \sum_p \sum_{k\in\mathcal{N}(p)} \|p-k\|_2^2 \tag{4}$$

The overall loss is a weighted sum of all four losses.

$$l_{all} = l_c + \lambda_1 l_n + \lambda_2 l_{lap} + \lambda_3 l_{loc} \tag{5}$$

5 Experiment

The paper choose ShapeNet dataset for experiment, and choose standard 3D reconstruction metric (F-score, Chamfer Distance (CD) and Earth Mover's Distance (EMD)) as evaluation metric. And the comparison results are listed in tables.

Authors also show the ablation study results in Fig 7.This figure truly reflects the contribution of each components, including losses, unpooling scheme, etc.

Furthermore, Fig 8 shows the effectiveness of the coarse-to-fine deformation scheme.

Finally, Fig 9 illustrated the quantitative results of the proposed model and other methods.

Threshold	τ				2τ			
Category	3D-R2N2	PSG	N3MR	Ours	3D-R2N2	PSG	N3MR	Ours
plane	41.46	68.20	62.10	**71.12**	63.23	81.22	77.15	**81.38**
bench	34.09	49.29	35.84	**57.57**	48.89	69.17	49.58	**71.86**
cabinet	49.88	39.93	21.04	**60.39**	64.83	67.03	35.16	**77.19**
car	37.80	50.70	36.66	**67.86**	54.84	77.79	53.93	**84.15**
chair	40.22	41.60	30.25	**54.38**	55.20	63.70	44.59	**70.42**
monitor	34.38	40.53	28.77	**51.39**	48.23	63.64	42.76	**67.01**
lamp	32.35	41.40	27.97	**48.15**	44.37	58.84	39.41	**61.50**
speaker	45.30	32.61	19.46	**48.84**	57.86	56.79	32.20	**65.61**
firearm	28.34	69.96	52.22	**73.20**	46.87	82.65	63.28	**83.47**
couch	40.01	36.59	25.04	**51.90**	53.42	62.95	39.90	**69.83**
table	43.79	53.44	28.40	**66.30**	59.49	73.10	41.73	**79.20**
cellphone	42.31	55.95	27.96	**70.24**	60.88	79.63	41.83	**82.86**
watercraft	37.10	51.28	43.71	**55.12**	52.19	**70.63**	58.85	69.99
mean	39.01	48.58	33.80	**59.72**	54.62	69.78	47.72	**74.19**

Figure 5: F-score on the ShapeNet test set at different thresholds

References

[1] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. Meshlab: an open-source mesh processing tool. In *Eurographics Italian chapter conference*, volume 2008, pages 129–136, 2008.

Category	CD				EMD			
	3D-R2N2	PSG	N3MR	Ours	3D-R2N2	PSG	N3MR	Ours
plane	0.895	**0.430**	0.450	0.477	0.606	**0.396**	7.498	0.579
bench	1.891	0.629	2.268	**0.624**	1.136	1.113	11.766	**0.965**
cabinet	0.735	0.439	2.555	**0.381**	**2.520**	2.986	17.062	2.563
car	0.845	0.333	2.298	**0.268**	1.670	1.747	11.641	**1.297**
chair	1.432	0.645	2.084	**0.610**	1.466	1.946	11.809	**1.399**
monitor	1.707	**0.722**	3.111	0.755	1.667	1.891	14.097	**1.536**
lamp	4.009	**1.193**	3.013	1.295	1.424	1.222	14.741	**1.314**
speaker	1.507	0.756	3.343	**0.739**	**2.732**	3.490	16.720	2.951
firearm	0.993	**0.423**	2.641	0.453	0.688	**0.397**	11.889	0.667
couch	1.135	0.549	3.512	**0.490**	2.114	2.207	14.876	**1.642**
table	1.116	0.517	2.383	**0.498**	1.641	2.121	12.842	**1.480**
cellphone	1.137	0.438	4.366	**0.421**	0.912	1.019	17.649	**0.724**
watercraft	1.215	**0.633**	2.154	0.670	0.935	0.945	11.425	**0.814**
mean	1.445	0.593	2.629	**0.591**	1.501	1.653	13.386	**1.380**

Figure 6: CD and EMD on the ShapeNet test set. Smaller is better

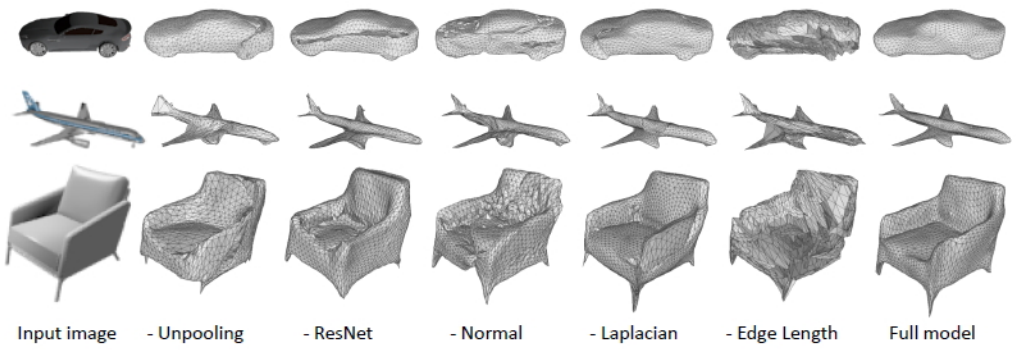

Figure 7: Qualitative results for ablation study.

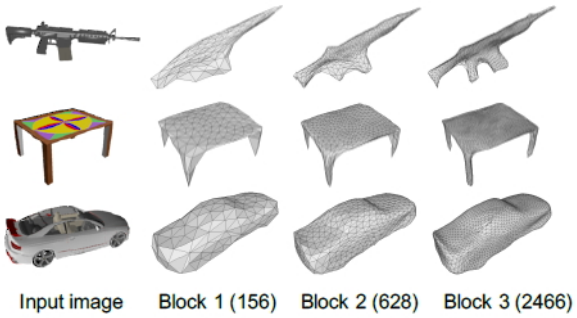

Figure 8: Sample examples showing the output after each block.

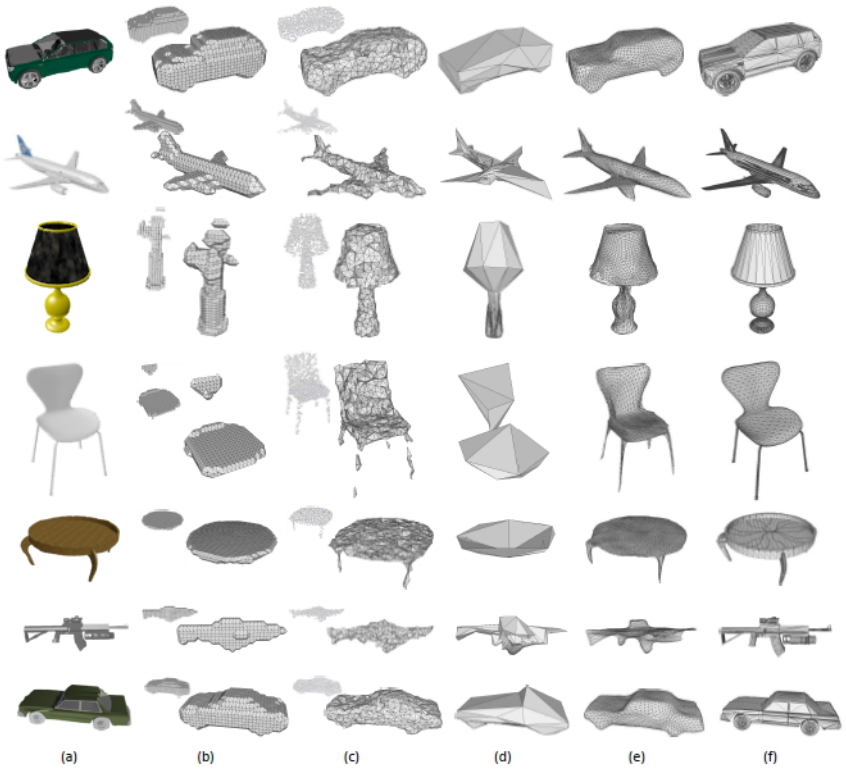

Fig. 8. Qualitative results. (a) Input image; (b) Volume from 3D-R2N2 [6], converted using Marching Cube [21]; (c) Point cloud from PSG [9], converted using ball pivoting [1]; (d) N3MR[17]; (e) Ours; (f) Ground truth.

Figure 9: Qualitative results.