Pacific Graphics 2018
H. Fu, A. Ghosh, and J. Kopf
(Guest Editors)

Volume 37 (2018), Number 7

ParamNet: Towards A Network of *Parameterization Prediction* for Shape Generation from a Single Image

ID: paper1147

Abstract

We propose an end-to-end deep learning framework that maps from unit sphere surface to target surface, given a single color image. Previous methods usually represent a 3D shape in volume or point cloud, and it is non-trivial to convert them to mesh models which are ready for wider application. Unlike the existing methods, our network represents 3D shape by a mapping from a fixed parameter domain. In our framework, the mapping is progressively carried out by the parameterization network, given parameters that semantic network predicted from a single color image. Our framework can directly generate mesh as output and it also allows us to integrate mesh based operation (e.g. Laplacian smooth) and mesh related losses into the network. Experiments show that our method not only qualitatively produces mesh model more visually appealing, but also achieves comparable 3D shape estimation error and even lower compared to the state-of-the-art. (see http://www.acm.org/about/class/class/2012)

CCS Concepts

•***Computing methodologies*** → ***Shape representations; Neural networks; Mesh models;***

1. Introduction

Inferring 3D shape from a single perspective is a traditional problem for computer vision. In computer graphics, 3D modeling with a given image has also been studied. Though it have been studied for tens of years, the problem remains challenging in both fields. The challenge arises from the fact that 3D-to-2D projection is not invertible and large portions of the 3D shape features are excluded in the 2D image. Recently, great success has been achieved for 3d shape generation from a single color image using deep learning techniques [CXG*16, FSG16]. By the using convolutional layers on regular grids or multi-layer perception on unordered 3D coordinates, the estimated 3D shape is represented as either a volume occupancy [CXG*16] or point cloud [FSG16] in neural networks. However, both representations lose important surface details, and is non-trivial to recover continue surface from.

As a matter of fact, mesh is a more desirable form of 3D shape representation for many applications in computer graphics, since it is capable of modeling shape details, easy to deform for animation, ready to render with various surface material...

Effort have been made by works like [DSK17], which integrate the technique of 3D morphable model from computer graphics into neural network to make an end-to-end trainable network that can output 3D mesh. However, such network can only ouput mesh for a specific class of object (the specific class is human face in [DSK17]).

In order to develop an end-to-end trainable network that can output mesh for multiple class of objects, we propose a brand new framework of neural network in this paper. Our framework is composed of two neural networks, the parameterization network and the semantic network. The semantic network predict parameters for the parameterization network which maps the unit sphere surface to the target surface. Such network structure can enable the representation of shape as spatial distribution. This perspective is inspired by VAE [KW13] in the sense that a large portion of distribution can be approximated by standard normal distribution plus a learned complicate mapping. In our framework, we actually represent the output shape using a uniform distribution defined on sphere surface plus a predicted complicate mapping (carried out by the parameterization network). These two combined can represent complicate distribution of surface. Therefore, such representation can enable the network to generate mesh as output and enable the mesh related operation and losses for the network.

In summary, our contributions are

- Introduction of parameterization network that enable mesh generation by representing the 3D shape as a spherical uniform distribution plus a learned/predicted mapping.
- Exploring the idea that use semantic network to predict parameters for parameterization network. Such structure relate input image to parameterization network.
- Though bijective is not ensured for the predicted mapping now, with the ability to integrate mesh based operation and losses, our framework push the neural network towards an actual *parameterization prediction* network that would allow more mesh operation former studied in computer graphics to be integrated into neural networks.

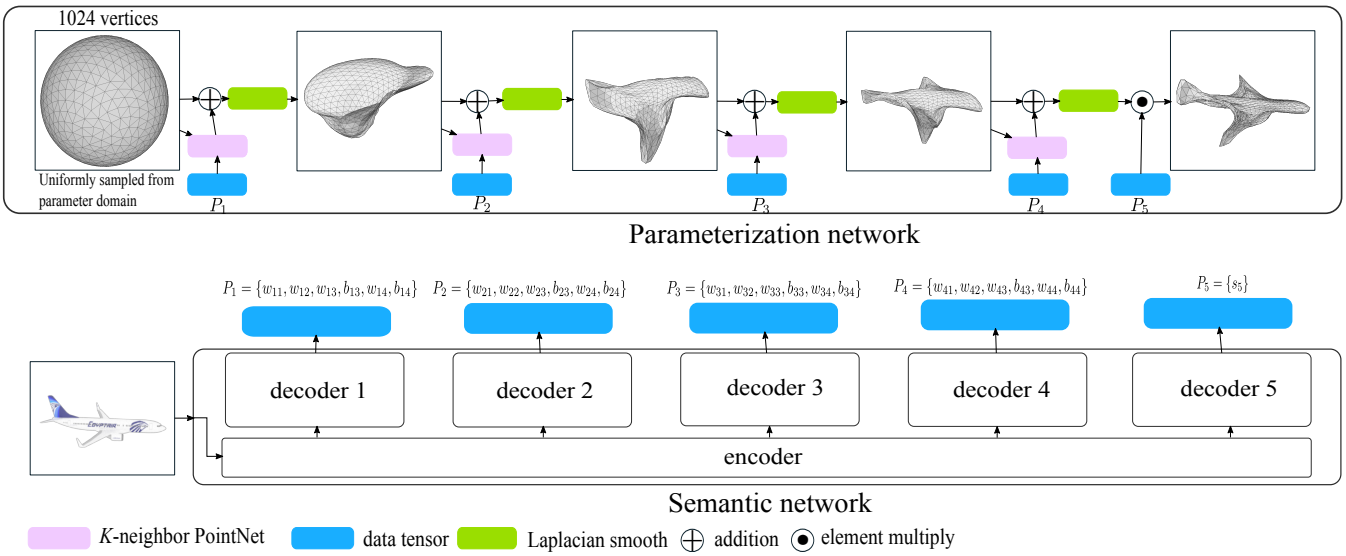

Figure 1: *The overview of the network: The proposed framework consists of two networks. One is the parameterization network that maps points sampled from parameter domain to target shape. The other is the semantic network that takes image as input and predict parameters for the parameterization network.*

2. Related work

3D reconstruction/modeling from single image have been well studied as the problem of *shape from monocular cues* (shading [ZTCS99], defocus [FSBO08,FS05], texture [Alo88]...). These methods usually only recover 2.5D surface from 2D image. Learning based approaches, especially deep learning methods can acquire more complicate prior by learning from dataset and recover much more complete 3D shape from a single image.

2.1. General learning approaches

As far as we known, early works of learning approaches can be traced back to [HEH07] and [SSN07]. More recent works like [SHM*14] and [HWK15] break down the problem to two stages, one is to retrieve shape components from a large dataset, the other is to assemble the components and deform the assembled shape to fit the observed image. However, shape retrieval from images itself is an ill-posed problem due to the loss of information in 3D-to-2D projection. To avoid such problem, [KTCM15] propose a solution that start reconstruction from a learned 3D deformable shape. In [KTCM15], images specify the shape variations and drive the shape deformation to target.

These learning approaches are relatively early. For their systems to work, complicate pre-process on database are usually needed. A more ideal solution would be to directly learn 3D shapes from single images under an end-to-end framework.

2.2. 3D neural networks

Most recently, researchers have developed techniques to represent 3D shapes inside deep learning framework and developed a serious of 3D neural networks to predict 3D shapes from images. Unlike images, 3D shapes are not canonical functions. This leads to multiple exploration for their representation. Therefore, we group these works according to their 3D representation.

Volume occupancy An intuitive way to extend the success of convolution network to 3D is to use volume occupancy of regular 3D grid to represent 3D shapes. Such 3D convolution was introduced by [WSK*15] and widely used in many works [CXG*16,GFRG16] for 3D shape generation.

The key disadvantage of such representation was the large memory consumption due to the raising of dimension when extend 2D grid to 3D. The most recent work of [TDB18] use an octree representation for shape generation (similar representation is used in for 3D shape classification and segmentation by [WLG*17]), which allows to higher resolution outputs with limited memory.

Point cloud [FSG16] proposed neural networks that regress unordered 3D point set for 3D shape generation. The point cloud representation do not consider local connections of the points, and thus the point positions have a very large degree of freedom. Consequently, it is difficult to reconstruct continue surface from the generated point cloud.

Besides the problem of generation/reconstruction, techniques like [CSKG17, QYSG17, LBSC18] have been developed so that the network can take unordered 3D point set as input and extract geometric features from 3D point set.

Mesh Among all three representations, mesh is the most popular one in game and movie industries. In addtion to the vertex positions , the mesh representation also consider local structure of vertices. It is non-trivial to use mesh representation for 3D shape with end-to-end trainable neural networks. [PKS*17] and [DSK17] produce mesh by linear interpolating base meshes. Since it is only possible to choose/learn base meshes for specific class of object, these two

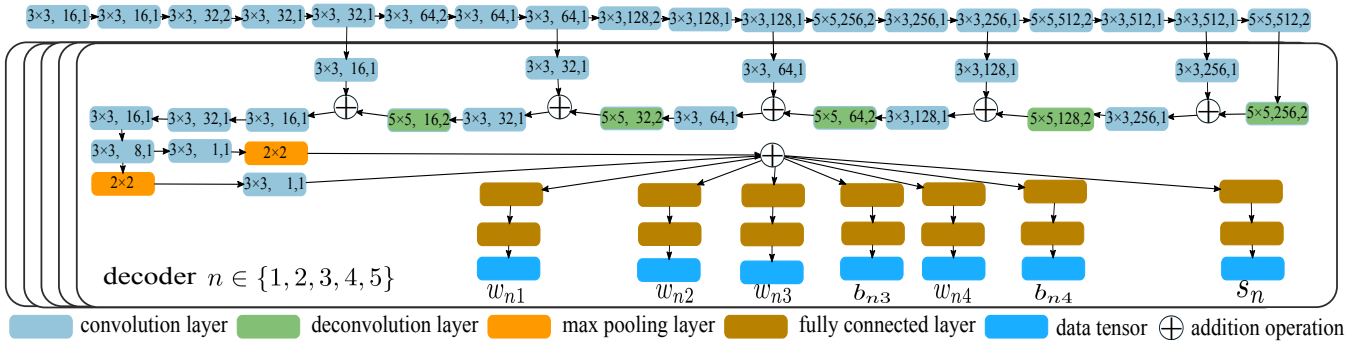

Figure 2: *Semantic network: The semantic network takes image as input and predict parameters for the K-neighbor PointNet and the global scale operation in the parameterization network. The semantic network have separate decoders for each K-neighbor PointNet and and the global scale operation.*

networks can only generate mesh for specific class of object (the latter one is only used for human face).

[KUH17] aims to make a differentiable approximation for 3D mesh rendering process, which would be much more significant if it can actually render gray image. In fact, the approximated renderer can only produce silhouette as output. When the it is used with mesh generation network, it can only take silhouette as supervision and hence does not perform well for complicated objects (car, airplane, etc.)

Our network produce mesh as output, but it draws a lot of experience from techniques designed for point cloud representation.

2.3. Parameterization in learning approaches

The idea that utilize parameterization in the representation of 3D shape have been explored by the works of [SUHR17, SBR16]. These methods involves a non-trainable procedure for the creation of geometry image. They also require manifold surface as training data so that the shapes can be parameterized using spherical parameterization and turned into geometry image. However, the public datasets like ShapeNet [CFG*15] contains meshes that are not manifold surfaces. Our idea is to represent the 3D surface by the parameterization. Our proposed network predict a mapping from the parameter domain to target surface. It do not require manifold surface as supervision. It can be trained with point cloud as supervision but produce mesh as output.

3. Network of *parameterization prediction*

In this section, we first explain the general framework of our proposed network in Sec 3.1, and then we elaborate the details of its structure part by part in the following sections.

3.1. Network overview

Our original idea about network of *parameterization prediction* was to use a semantic network that takes image as input to predict a mapping from the parameter domain to the target surface. In the proposed framework shown in Figure 1, the mapping is actually expressed by the parameterization network. Instead of directly predicting the mapping, the semantic network predicts parameters for the parameterization network, making the entire network end-to-end trainable.

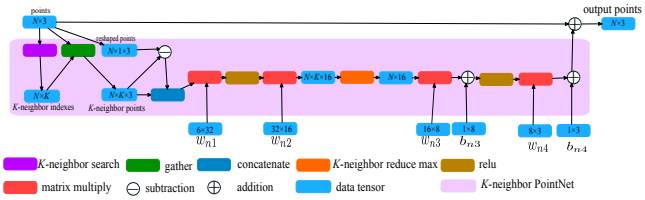

Figure 3: *The internal structure of K-neighbor PointNet*

The parameterization network is built by stacking several K-neighbor PointNet (explained in Sec 3.3). Each K-neighbor PointNet takes point set as input and predicts a offsets for each point and add them to the input points. In this way, the parameterization network can map/deform a randomly sampled point set to target shape. By using different randomly generated samples from the parameter domain in the training iterations, the parameterization network can learn to handle the *sampling variation* that is introduced by using specific point set as ground truth.

The semantic network is built on convolution, deconvolution and fully connected layers, it takes image as input to predict the parameters (i.e.$\{P_1, P_2, P_3, P_4, P_5\}$ in Figure 1) for the parameterization network. In this way, the semantic network relates the input image to the parameterization.

3.2. Parameter domain

In the proposed framework, we use unit sphere surface as parameter domain. As shown in Figure 1, at the beginning of the parameterization network $N = 1024$ points are uniformly sampled from the parameter domain. Triangulation is also applied on these sampled points. The edge connections built by triangulation are later used in the Laplacian smooth layer and the edge length regularization term. These connections are also used to connect output points to mesh.

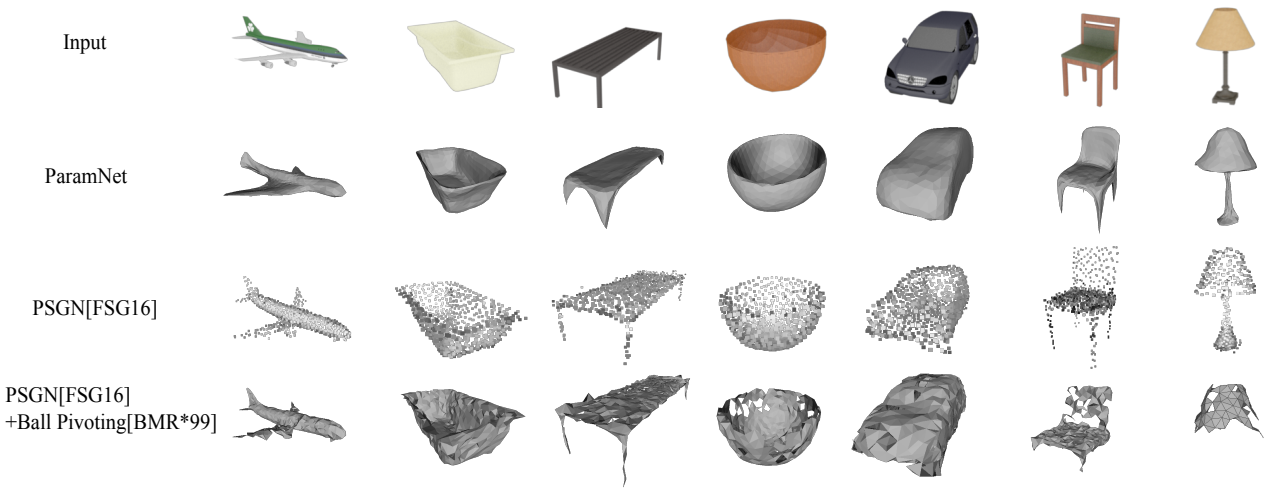

Figure 4: *The comparison of visual results*

3.3. *K*-neighbor PointNet for parameterization network

An essential building block for our parameterization network is K-neighbor PointNet. Figure 3 shows the structure of K-neighbor PointNet. K-neighbor PointNet is inspired by and named after PointNet [CSKG17] and its follow-up PointNet++ [QYSG17]. In order to take unordered point set as input, K-neighbor PointNet uses either point-wise operations on each point or symmetric functions on multiple points. As shown in Figure 3, the K-neighbor search operation finds the K neareast neighbor point for each point inside the N input points and output the indexes tensor. We employed the Bitonic Sort [Bat68] for its implementation on GPU. Then the *gather* operation forge the tensor of K-neighbor points based on these indexes and input points. This operation is tensorflow Subtracting the original input points from K-neighbor points we can get the local relative coordinates of K-neighbor points. By concatenating K-neighbor point coordinates and their local relative coordinates, we get the $N \times K \times 6$ tensor as geometric features. These features go through a series of operations as shown in Figure 3 to predict point-wise 3D offsets for each input point. Unlike the original PointNet [CSKG17], who extracts maximum value among all point as feature vector for input point set, our K-neighbor PointNet extract maximum value among every K-neighborhood as feature vector for each point. This is shown by the K-neighbor reduce max operation in Figure 3.

3.4. Laplacian smooth

Another essential building block for our parameterization network is the Laplacian smooth layer. Mesh based Laplacian smooth is commonly used in mesh processing. The Laplacian smooth layer applys the mesh based Laplacian smooth operation for each point in the point set as in (1). The mesh is constructed by triangulation on the sample points from parameter domain (i.e. the unit sphere surface). In (1), $\mathcal{N}(\mathbf{x})$ represents the one-ring neighborhood of $\mathbf{x}$. As described in (1), the Laplacian smooth is a local linear operation and therefore differentiable.

$$\mathbf{x}^* = \frac{1}{|\mathcal{N}(\mathbf{x})|} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \mathbf{y} \tag{1}$$

Our ablation study shows that the Laplacian smooth significantly improves the regularity of the generated surface and makes the output mesh visually much more appealing. It also prevent self-intersection of the surface locally

3.5. Semantic network

In our semantic network, we use convolution layers to extract semantic features from input image as shown in Figure 2. Our semantic network adopt the U-shape structure that is similar to UNet [RFB15]. To fit with our framework, we use separate decoder branches to predict different parameters parameterization network. These predicted parameters (i.e. $P_n = \{w_{n1}, w_{n2}, w_{n3}, b_{n3}, w_{n4}, b_{n4}, s_n\}$) are plugged into the parameterization network in the way shown in Figure 1

3.6. Losses

The whole loss we use is shown in Eq.(2), it is composed of the four terms as follows:

$$l = l_c + \alpha l_2 + \beta l_{reg} + l_n \tag{2}$$

Chamfer loss The Chamfer distance is directly borrowed from [FSG16]. This loss can drive the output point set to approach the target. As shown in Eq.(3). In Eq.(3), $\mathbf{x}$ are output points and $\mathbf{y}$ are the points from the groundtruth

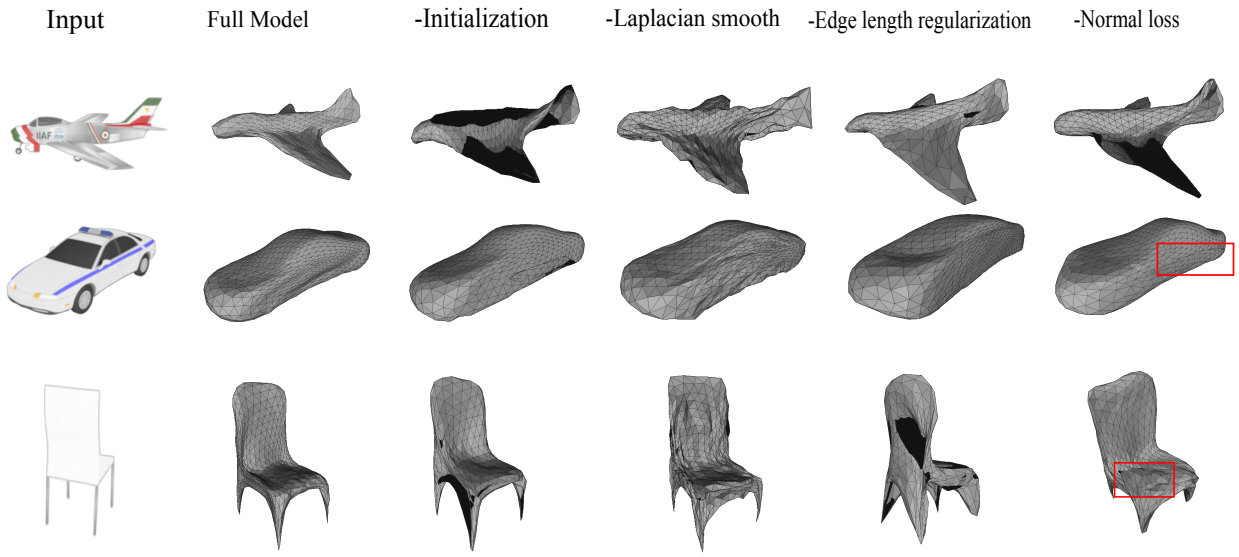

Figure 5: *Visual result for ablation study: In some cases, the view angle is adjusted for better exposure of imperfection. The black regions in the meshes are regions with flipped triangles*

Table 1: *Ablation study with respect to different components*

Models	Full Model	-Initialization	-Laplation smooth	-Edge length regularization	-Norm loss
Chamfer	0.297	0.424	0.394	0.405	0.390
EMD	0.834	1.524	1.369	4.195	1.415

$$l_c = \sum_{\mathbf{x}} \min ||\mathbf{x}-\mathbf{y}||_2^2 + \sum_{\mathbf{y}} \min ||\mathbf{x}-\mathbf{y}||_2^2 \tag{3}$$

L2 Regularization We use L2 regularization for the parameters in our semantic network.

Edge Length Regularization To discourage over stretched triangles in output mesh, we add a regularization term to minimize the variance of the edge length for the output mesh as shown in (4). In (4), $\mathcal{E}$ is the set of edge, recorded as a set of paired vertex indexes (e.g. (i, j)).

$$l_{reg} = \sum_{(i,j)\in\mathcal{E}} (||\mathbf{x}_i - \mathbf{x}_j||_2 - \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} ||\mathbf{x}_i - \mathbf{x}_j||_2)^2 \tag{4}$$

Normal loss To better guide surface to approach the groundtruth, instead of emphasizing only on the approaching of output vertices, we envolve normal into our loss. This loss is described in (5). In (5), the n_{y_i} and n_{y_j} are normals for vertices $\mathbf{y}_i$ and $\mathbf{y}_j$. $\mathbf{y}_i$ and $\mathbf{y}_j$ are respectively closest vertices to $\mathbf{x}_i$ and $\mathbf{x}_j$ in the groundtruth. They are found on calculating Chamfer loss (3). This Normal loss term encourages the edges to remain perpendicular to the corresponding normals of its two end points.

$$l_n = \sum_{(i,j)\in\mathcal{E}} || < \mathbf{x}_i - \mathbf{x}_j, \mathbf{n}_{\mathbf{y_i}} > ||^2 + || < \mathbf{x}_i - \mathbf{x}_j, \mathbf{n}_{\mathbf{y_j}} > ||^2. \tag{5}$$

3.7. Network initialization

When start training from scratch, the proposed loss and network still can not prevent the network to output surface with severe global self-intersection. To alleviate this problem, we adopt a network initialization step to reset the parameterization network to start at a state that output surface without self-intersection. At this step we train the network with the loss shown in Eq.(6). The $\mathbf{y}$ used in Eq.(6) is not the points from groundtruth, but the points sampled from parameter domain with a 0.5 scale down. In other words, this step will reset the network so that the parameterization network will map from a sphere to a smaller sphere, when taking any image as input. We do this training of initialization by one epoch on the training data.

$$l_{init} = \sum_{\mathbf{x}} ||\mathbf{x}-\mathbf{y}||_2^2 \tag{6}$$

4. Experiments

In this section, we evaluate our network through a series of experiments.

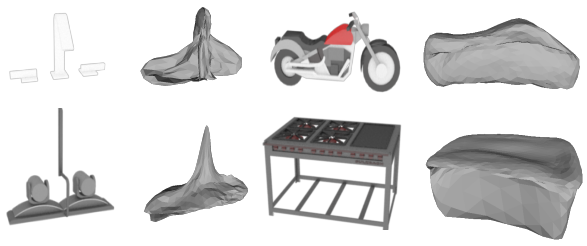

Figure 6: *Failure cases: In some cases, the view angle is adjusted for better exposure of imperfection*

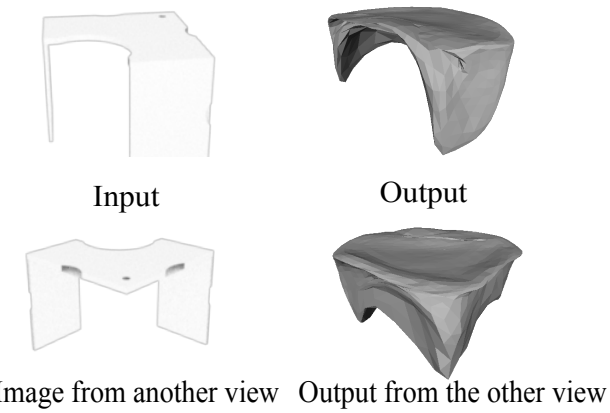

Figure 7: *Failure case: In this case, the network have output an extra leg that doesn't exist for the dresser*

4.1. Experimental setup

We train and evaluate our models on the ShapeNet [CFG*15]. Specifically, we use the ShapeNetCore55.v2 which contains over 50k of manually created and cleaned 3D CAD models in 55 category. The images for training and testing are rendered in random angles to provide synthetic training data for the model. In total, 51,856 shape models are covered. For the training/validation/testing split, we follow the CSV file provided by the ShapeNet website and resulted in 36,622/5,110/10,124 shapes. The 3D CAD objects are stored as meshes, we re-sample the meshes into point sets. In order to capture only the shape surface, we uses the code from [WLG*17] and execute "virtual scan" for the re-sampling.

Table 2: *Ablation study with respect to number of K-neighbor PointNet*

Number of K-n PointNet	2	3	4	5
Chamfer	0.425	0.411	0.297	0.313
EMD	1.304	1.360	0.834	1.067

4.2. Comparing to state-of-the-art

In this subsection, we compare our approach with the state-of-the-art point set generation network (PSGN) [FSG16]. In Table 3, we compare the our approach with PSGN [FSG16] in quantitative evaluation over all category of objects in testing data. The results shows that our approach achieves over all comparable performance with PSGN [FSG16] under Chamfer distance and present evidently better performance under Earth mover distance. In Figure 4, we compare the visual results of our framework (ParamNet) and PSGN [FSG16]. In Figure 4, we also show mesh generated by ball-pivoting [BMR*99] from the output point set of PSGN [FSG16]. These visual results shows that it is difficult to recover continuous surface from the point set generated by PSGN [FSG16] since it does not explicitly consider local structure of the generated surface, while our method can generate continuous surface for various objects.

4.3. Ablation study

In this subsection, we do ablation study to investigate the functionality of each components in our framework. The evaluation are shown in Table 1 and Table 2. Some visual result for the ablation study are shown in Figure 5. Now we discuss these results as follows:

Laplacian smooth The Laplacian smooth force the output surface to have more regular local structure. When Laplacian smooth is removed, the output surface become much more rough as shown in Figure 5 under the column of "-Laplacian smooth".

Initialization The initialization step reset the network to a state that output surface without self-intersection. As shown in Figure 5 under the column of "-Initialization", severe triangle flipping and self-intersection happens when the initialization training is skipped.

Edge length regularization The edge length regularization term in the loss is designed to discourage over stretched triangles by minimizing the variance of edge length. According to our observation, when edge length regularization term is removed from the loss, the network tend to produce mesh that has more large triangles as shown by the airplane example in Figure 5 under column "-Edge length regularization ". Without such suppression for triangle over-stretching, the output become more easily to have self-intersection and triangle flipping as shown by the chair example in Figure 5 under column "-Edge length regularization ".

Normal loss The normal loss is designed to guide the learning by using the normal from groundtruth. According to our observation, the normal information is evidently useful to prevent some local self-intersection as highlighted by red rectangles in Figure 5.

Number of *K*-neighbor PointNet The number of K-neighbor PointNet is an important super parameter for our framework of networks. Based on the experiment results shown in Table 2, we choose to use four K-neighbor PointNet to construct the parameterization network, since the test error no longer diminishes when we use more (i.e. five).

Figure 8: *More visual results randomly picked from testing data*

Table 3: *Comparison with point set generation network [FSG16]*

Category	Category id	Chamfer		EMD	
		ParamNet	PSGN [FSG16]	ParamNet	PSGN [FSG16]
aircraft	02691156	0.068	0.058	0.248	0.502
dustbin	02747177	0.162	0.166	0.425	1.947
bag	02773838	0.398	0.453	0.921	3.258
basket	02801938	0.861	0.708	1.545	2.186
bathtub	02808440	0.072	0.070	0.175	0.472
bench	02828884	0.063	0.063	0.239	0.641
bed	02818832	0.362	0.291	0.865	1.523
birdhouse	02843684	0.855	0.624	1.972	4.332
shelf	02871439	0.100	0.081	0.407	1.475
bottle	02876657	0.075	0.084	0.289	1.340
bowl	02880940	0.726	0.423	0.954	1.646
bus	02924116	0.035	0.036	0.306	0.602
dresser	02933112	0.082	0.078	0.168	0.799
camera	02942699	0.818	0.752	1.889	3.124
can	02946921	0.165	0.180	0.778	4.247
cap	02954340	1.466	2.738	2.795	3.559
car	02958343	0.051	0.047	0.176	0.399
cellphone	02992529,04401088	0.143,0.039	0.128,0.030	0.965,0.150	7.583,0.974
chair	03001627	0.042	0.038	0.117	0.605
clock	03046257	0.152	0.119	0.387	1.194
keyboard	03085013	0.450	0.456	1.470	2.626
dishwasher	03207941	0.204	0.236	0.501	2.975
monitor	03211117	0.079	0.067	0.224	0.811
headphone	03261776	0.738	0.590	3.906	4.499
hydrant	03325088	0.177	0.151	0.748	1.123
file cabinet	03337140	0.121	0.113	0.325	1.949
guitar	03467517	0.014	0.015	0.278	1.234
helmet	03513137	0.789	1.199	1.693	2.387
vase	03593526	0.088	0.082	0.305	1.244
knife	03624134	0.034	0.029	0.280	2.089
lamp	03636649	0.089	0.084	0.503	0.869
laptop	03642806	0.174	0.154	0.537	1.262
speaker	03691459	0.125	0.117	0.315	0.806
mailbox	03710193	0.258	0.252	1.245	4.337
mike	03759954	1.599	1.301	3.149	6.842
microwave	03761084	0.305	0.302	0.675	2.647
motorcycle	03790512	0.153	0.139	0.522	1.495
mug	03797390	0.269	0.188	0.481	1.568
piano	03928116	0.234	0.227	0.693	1.587
pillow	03938244	0.614	0.526	0.996	2.101
handgun	03948459,04090263	0.372,0.049	0.309,0.049	1.158,0.304	2.369,0.790
planter	03991062	0.124	0.139	0.301	0.839
printer	04004475	0.500	0.413	1.064	1.716
remote	04074963	0.106	0.105	0.846	5.940
missile	04099429	0.220	0.187	1.874	4.113
skateboard	04225987	0.369	0.295	1.696	2.232
sofa	04256520	0.051	0.052	0.172	0.321
stove	04330267	0.184	0.214	0.488	1.528
table	04379243	0.077	0.075	0.271	0.537
tower	04460130	0.620	0.735	1.714	3.812
train	04468005	0.129	0.122	0.638	1.140
ship	04530566	0.067	0.057	0.273	0.620
washer	04554684	0.205	0.203	0.482	1.957
mean	-	0.297	0.298	0.834	2.086

4.4. Limitations and future work

A major limitation for our framework is that it cannot handle the surface generation for objects that are not homotopy equivalent to sphere surface. This limitation arises from the design of our framework and is clearly shown in the cases in Figure 6. For this issue, we are planning to combine our framework with the methods like [TSG*17, NLX18]. In this way, we can generate objects with several separate parts and therefore represent objects with more complicate topology.

Another limitation is that our semantic network failed to correctly infer the parameter for shape from the input image. Figure 7 shows an example of such failure case. This limitation arises from the ambiguity in the 2D to 3D inference. For this issue, we are planning to try VAE method [KW13] or min-of-n loss proposed by [FSG16] to encode the ambiguity.

5. Conclusions

We propose an end-to-end trainable framework of networks that can learn to generate shape from single image without knowing the type of the shape in image. Our approach can generate better continuous surface than the state-of-art PSGN [FSG16].

References

[Alo88] ALOIMONOS J.: Shape from texture. *Biological Cybernetics 58*, 5 (Apr 1988), 345–360. URL: https://doi.org/10.1007/BF00363944, doi:10.1007/BF00363944. 2

[Bat68] BATCHER K.: Sorting networks and their applications. 307–314. 4

[BMR*99] BERNARDINI F., MITTLEMAN J., RUSHMEIER H., SILVA C., TAUBIN G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics 5*, 4 (Oct. 1999), 349–359. URL: http://dx.doi.org/10.1109/2945.817351, doi:10.1109/2945.817351. 6

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: *ShapeNet: An Information-Rich 3D Model Repository*. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 3, 6

[CSKG17] CHARLES R. Q., SU H., KAICHUN M., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 77–85. doi:10.1109/CVPR.2017.16. 2, 4

[CXG*16] CHOY C. B., XU D., GWAK J., CHEN K., SAVARESE S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. *CoRR abs/1604.00449* (2016). URL: http://arxiv.org/abs/1604.00449. 1, 2

[DSK17] DOU P., SHAH S. K., KAKADIARIS I. A.: End-to-end 3d face reconstruction with deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), vol. 00, pp. 1503–1512. URL: doi.ieeecomputersociety.org/10.1109/CVPR.2017.164, doi:10.1109/CVPR.2017.164. 1, 2

[FS05] FAVARO P., SOATTO S.: A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 3 (March 2005), 406–417. doi:10.1109/TPAMI.2005.43. 2

[FSBO08] FAVARO P., SOATTO S., BURGER M., OSHER S. J.: Shape from defocus via diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 3 (March 2008), 518–531. doi:10.1109/TPAMI.2007.1175. 2

[FSG16] FAN H., SU H., GUIBAS L.: A point set generation network for 3d object reconstruction from a single image. 1, 2, 4, 6, 8, 9

[GFRG16] GIRDHAR R., FOUHEY D. F., RODRIGUEZ M., GUPTA A.: Learning a predictable and generative vector representation for objects. In *Computer Vision – ECCV 2016* (Cham, 2016), Leibe B., Matas J., Sebe N., Welling M., (Eds.), Springer International Publishing, pp. 484–499. 2

[HEH07] HOIEM D., EFROS A. A., HEBERT M.: Recovering surface layout from an image. *International Journal of Computer Vision 75*, 1 (Oct 2007), 151–172. URL: https://doi.org/10.1007/s11263-006-0031-y, doi:10.1007/s11263-006-0031-y. 2

[HWK15] HUANG Q., WANG H., KOLTUN V.: Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph. 34*, 4 (July 2015), 87:1–87:10. URL: http://doi.acm.org/10.1145/2766890, doi:10.1145/2766890. 2

[KTCM15] KAR A., TULSIANI S., CARREIRA J., MALIK J.: Category-specific object reconstruction from a single image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), pp. 1966–1974. doi:10.1109/CVPR.2015.7298807. 2

[KUH17] KATO H., USHIKU Y., HARADA T.: Neural 3d mesh renderer. *CoRR abs/1711.07566* (2017). URL: http://arxiv.org/abs/1711.07566, arXiv:1711.07566. 3

[KW13] KINGMA D. P., WELLING M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013). 1, 9

[LBSC18] LI Y., BU R., SUN M., CHEN B.: Pointcnn. *CoRR abs/1801.07791* (2018). URL: http://arxiv.org/abs/1801.07791, arXiv:1801.07791. 2

[NLX18] NIU C., LI J., XU K.: Im2struct: Recovering 3d shape structure from a single RGB image. *CoRR abs/1804.05469* (2018). URL: http://arxiv.org/abs/1804.05469, arXiv:1804.05469. 9

[PKS*17] PONTES J. K., KONG C., SRIDHARAN S., LUCEY S., ERIKSSON A. P., FOOKES C.: Image2mesh: A learning framework for single image 3d reconstruction. *CoRR abs/1711.10669* (2017). URL: http://arxiv.org/abs/1711.10669, arXiv:1711.10669. 2

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems 30*, Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., (Eds.). Curran Associates, Inc., 2017, pp. 5105–5114. URL: http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature/learning-on-point-sets-in-a-metric-space.pdf. 2, 4

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (Cham, 2015), Navab N., Hornegger J., Wells W. M., Frangi A. F., (Eds.), Springer International Publishing, pp. 234–241. 4

[SBR16] SINHA A., BAI J., RAMANI K.: Deep learning 3d shape surfaces using geometry images. In *Computer Vision – ECCV 2016* (Cham, 2016), Leibe B., Matas J., Sebe N., Welling M., (Eds.), Springer International Publishing, pp. 223–240. 3

[SHM*14] SU H., HUANG Q., MITRA N. J., LI Y., GUIBAS L.: Estimating image depth using shape collections. *ACM Trans. Graph. 33*, 4 (July 2014), 37:1–37:11. URL: http://doi.acm.org/10.1145/2601097.2601159, doi:10.1145/2601097.2601159. 2

[SSN07] SAXENA A., SUN M., NG A. Y.: Learning 3-d scene structure

from a single still image. In *2007 IEEE 11th International Conference on Computer Vision* (Oct 2007), pp. 1–8. doi:10.1109/ICCV.2007.4408828. 2

[SUHR17] SINHA A., UNMESH A., HUANG Q., RAMANI K.: Surfnet: Generating 3d shape surfaces using deep residual networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), vol. 00, pp. 791–800. URL: doi.ieeecomputersociety.org/10.1109/CVPR.2017.91, doi:10.1109/CVPR.2017.91. 3

[TDB18] TATARCHENKO M., DOSOVITSKIY A., BROX T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct. 2018), vol. 00, pp. 2107–2115. URL: doi.ieeecomputersociety.org/10.1109/ICCV.2017.230, doi:10.1109/ICCV.2017.230. 2

[TSG*17] TULSIANI S., SU H., GUIBAS L. J., EFROS A. A., MALIK J.: Learning shape abstractions by assembling volumetric primitives. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 1466–1474. doi:10.1109/CVPR.2017.160. 9

[WLG*17] WANG P.-S., LIU Y., GUO Y.-X., SUN C.-Y., TONG X.: O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics (SIGGRAPH) 36*, 4 (2017). 2, 6

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), pp. 1912–1920. doi:10.1109/CVPR.2015.7298801. 2

[ZTCS99] ZHANG R., TSAI P.-S., CRYER J. E., SHAH M.: Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 8 (Aug 1999), 690–706. doi:10.1109/34.784284. 2