Single-image Mesh Reconstruction and Pose Estimation via Generative Normal Map

Nan Xiang
Bournemouth University
nxiang@bournemouth.ac.uk

Li Wang
Bournemouth University
lwang@bournemouth.ac.uk

Tao Jiang
Bournemouth University
tjiang@bournemouth.ac.uk

Yanran Li
Bournemouth University
liy@bournemouth.ac.uk

Xiaosong Yang*
Bournemouth University
xyang@bournemouth.ac.uk

Jianjun Zhang
Bournemouth University
jzhang@bournemouth.ac.uk

ABSTRACT

We present a unified learning framework for recovering both 3D mesh and camera pose of the object from a single image. Our approach learns to recover outer shape and surface geometric details of the mesh without relying on 3D supervision. We adopt multi-view normal maps as the 2D supervision so that the silhouette and geometric details information can be transferred to neural network. A normal mismatch based objective function is introduced to train the network, and the camera pose is parameterized into the objective, it integrates pose estimation with the mesh reconstruction in a same optimization procedure. We demonstrate the abilities of the proposed approach in generating 3D mesh and estimating camera pose with qualitative and quantitative experiments.

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction**; *Mesh models*; *Neural networks*.

KEYWORDS

mesh reconstruction, pose estimation, deep learning

ACM Reference Format:
Nan Xiang, Li Wang, Tao Jiang, Yanran Li, Xiaosong Yang, and Jianjun Zhang. 2019. Single-image Mesh Reconstruction and Pose Estimation via Generative Normal Map. In *Computer Animation and Social Agents (CASA '19), July 1–3, 2019, PARIS, France*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3328756.3328766

1 INTRODUCTION

How do humans understand 3D world from a single 2D image? Researches in visual perception indicate that humans recognize objects from a single image by building a description of shapes and spatial positions according to the empirical knowledge[11, 16, 20]. Therefore, recovering 3D shapes and poses from images is a worthwhile study that has became an active research area in computer graphics and computer vision. While single-image shape and pose recovering is still an ill-posed problem due to the multitude of ambiguities in a single image. In this paper, we consider the issue of single-image 3D shape reconstruction and pose estimation under learning based methods.

Recent progress in deep neural networks has sparked a growing research interest in using deep learning methods to recover 3D shape from a single-image. Many works trained convolutional neural networks (CNNs) to predict 3D voxel models [2, 15, 19, 23] or 3D point clouds [3]. However, voxels and point clouds do not convey surface information of the object, cannot be used to geometry editing, texture mapping and light rendering. While the polygon mesh which consists of vertices and triangular faces is a considerable 3D representation type for further editing.

To take advantages of polygon mesh, we train an encoder-decoder architecture CNN to generate 3D mesh. Specifically, the network is trained to deform a underlying mesh into a new shape. Several studies have adopted this approach to recover 3D mesh from a single image. Ellipsoid [22] or sphere [5] was per-defined as the underlying mesh, however, they require a large number of 3D meshes as supervision during training phase. Our method based on multi-view 2D observation, the network can be trained without relying on 3D supervision.

Current Non-3D-supervision learning frameworks [8, 23] used 2D silhouette supervision, the network was trained by optimizing an Intersection over Union (IoU) based objective function, which is a simple silhouette constraint, does not perform well for recovering surface geometric structures. We adopt normal map instead of silhouette image as the supervision. On this basis, a normal vector based objective function is introduced, which constrains both silhouette and surface geometric structures of the mesh, such that the reconstructed mesh keeps correct outer contour and surface geometric structure.

Shape reconstruction and pose estimation from a single image are two fundamental issues in 3D understanding area. Many recent studies specialized in using deep learning methods to learn pose estimation from the 2D image [9, 10, 14, 17]. Some frameworks attempted to predict shape and pose together by training several networks[21] or learning from annotated images[7]. In our work, the pose estimation can be straightforward. We calculate the normal vector in 3D camera space, such that the pose parameters are

*Corresponding author.

CASA '19, July 1–3, 2019, PARIS, France

ACM ISBN 978-1-4503-7159-9/19/07…$15.00
https://doi.org/10.1145/3328756.3328766

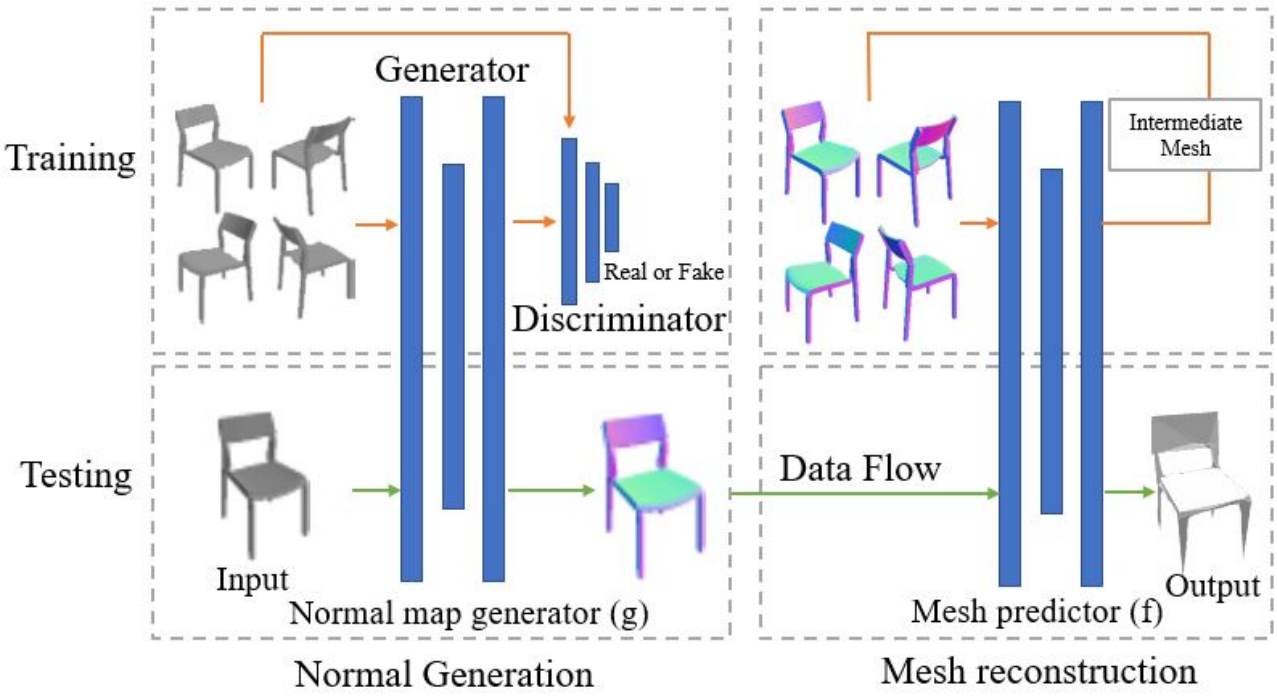

Figure 1: The overview of proposed framework. The left column is a CGAN for normal map generation; the right is an encoder-decoder CNN for mesh reconstruction. The first row shows the training phases, the multi-view 2D observations (lighting images and normal images) are rendered from 3D models. The second row shows the workflow of the trained normal map generator and the mesh predictor, it takes a single RGB image as input, the output is the reconstructed 3D mesh and its corresponding pose.

easily integrated in the final objective function, then the pose estimation can be done along with 3D mesh reconstruction in the same optimization process.

In this paper, we propose an end-to-end learning framework for recovering detailed 3D mesh and camera pose from a single image. Figure 1 illustrates the overview of the framework. It takes a single image as input, a Conditional Generative Adversarial Network (CGAN) [12] architecture is adopted to train a generator for normal map generation. The normal map is then transferred to an encoder-decoder CNN architecture for processing mesh reconstruction and pose estimation. As far as we know, 2D normal maps have never been directly fed to neural networks for learning single-image 3D reconstruction and pose estimation.

Our main contributions can be summarized as:

- We present a unified learning framework to recover both 3D mesh and camera pose from a single RGB image. The two different processes are integrated in the same optimization procedure.
- We propose the method for recovering outer contour and surface geometric details of mesh without learning from 3D supervision. This is made by introducing a normal mismatch based objective function.
- We demonstrate the advantages of our approach both qualitatively and quantitatively in mesh reconstruction and pose estimation.

2 RELATED WORK

Our work is based on the recent progress in deep learning methods to tackle the problem of single-image mesh reconstruction and pose estimation.

2.1 Single-image 3D reconstruction

Deep learning methods based single-image 3D reconstruction has became an active research topic. In previous studies, The voxel [2, 15, 19, 23] is the most widely adopted 3D representation type, on account of the voxel is considered as the 3D extension of the 2D pixel, that can be processed by convolutional neural networks (CNNs). Another representation is point cloud, [3] introduced the PointOutNet to generate point set, which was used in 3D point cloud reconstruction from a single image. However, both of them lose important surface details of geometric structure, making it difficult to further geometry editing, texture sampling and light rendering. Due to the clear surface geometric structure of polygon mesh, our framework generates the mesh instead of voxels or point clouds.

Integrating mesh data into the neural networks is a big challenge due to its graph structure. [22] represented the vertices of the mesh as nodes, which were fed to a graph convolutional network (GCN), the new mesh was obtained by learning to deform an ellipsoid mesh to the target shape. [5] parameterized a pre-defined mesh in a

bidirectional 2D space, mesh generating amounts to learning parametric transformation in the 2D space. Both works adopted a 3D chamfer distance based loss function to train the networks, which requires 3D vertex sets as supervision. Our learning framework allows the network to be trained without 3D supervision.

Previous studies have proved that single-image 3D reconstruction can be realized without explicit 3D supervision. [23] introduced a silhouette loss function for learning 3D voxel reconstruction under 2D silhouette observation, [8] employed this approach in single-image 3D mesh reconstruction. However, the silhouette supervision cannot constrain the geometric structure of mesh surface, hence does not perform well for recovering surface details. Considering that the normal map conveys both silhouette information and geometric details, we use normal maps instead of silhouette images as the supervision.

2.2 Pose estimation

3D pose estimation from images is an essential task in 3D understanding area such as 3D recognition and human-machine interaction. Deep learning methods have shown to be effective for the single-image pose estimation. [10] introduced the PoseNet, a pose estimation learning framework based on the GoogLeNet, it was trained to regress the 6-DOF camera pose from a single image. [9] proposed geometric loss functions that improved the PoseNet's performance. [17] adopted images that rendered from 3D model datasets to train CNN for learning pose estimation on real images. [14] trained a CNN to predict semantic key-points of the object from a single RGB image, then combined the key-points with a predefined deformable model to calculate the pose of the object. [21] attempted to predict both shape and pose from a single RGB image by respectively training two CNNs. Our framework integrates the pose estimation with the mesh reconstruction, and both outperform state of the art.

2.3 Normal map generation via image translation

Isola *et al.*[6] explored image-to-image translation problem in using CGAN [12], they used the image as the extra information that fed to the network along with a noise vector to train an image "translator". On this basis, [18] trained a normal map generator that converts sketch images to normal maps, in the meanwhile, they presented an interactive interface to enhance user-specific refining. Inspired by these works, we employ the CGAN to train the normal map generator without user-specific enhancing.

3 METHOD

We aim to learn a mesh predictor f and a normal map generator g. The f can infer the 3D structure for the underlying mesh from a single normal map n, while the g generates the normal map n from a single RGB image x, $g : x \rightarrow n$. The final prediction $(f \circ g)(x)$ is comprised of the 3D shape with the specific camera pose corresponding to the input image. We detail the methods in following subsections.

3.1 Mesh reconstruction

Instead of relying on 3D supervision, we train the predictor function f from multiple-view observations of objects, which is similar to the previous works in [8, 23]. But they used silhouette images as 2D supervision, while we use normal maps $\mathcal{N}$, the viewport is denoted as $\mathcal{V}$.

In every iteration of the training process, $n_i \in \mathcal{N}$ denotes the i-th normal map, the mesh prediction is $f(n_i)$. Then we calculate its surface normals, and map the normal values into RGB range $[0, 1]$, which are finally rendered in the given viewport $v_i \in \mathcal{V}$. The process can be expressed as $\hat{n}_i = R_{normal}(f(n_i), v_i)$, where $R_{normal}(\cdot)$ denotes the rendering process of the normal map. To allow the backward propagation of normal errors, we use a differentiable mesh renderer [8] to implement the rendering process. The predictor function f is trained by optimizing a normal mismatch based objective function.

In consideration of the normal maps contain both 2D silhouette information and 3D mesh surface details, we directly calculate the distance between n_i and $\hat{n}_i$ as a global normal loss. In this work, L1 outperform L2 distance in the prediction, especially in the outline preservation (see Figure 2). Thus, we define the global normal loss as:

$$\mathcal{L}_{normal} = ||\hat{n}_i - n_i||_1, \tag{1}$$

The normal value is a direction vector, therefore, we introduce a loss function based on angular distance to regulate the local face orientations of the mesh:

$$\mathcal{L}_{rotation} = ||(1 - \hat{d}_i \odot d_i) \odot M||_2, \tag{2}$$

where

$$d_i = 2n_i - 1, \hat{d}_i = 2\hat{n}_i - 1$$

The symbol $\odot$ presents an element-wise product. The normal vectors d_i and $\hat{d}_i$ can be obtained by mapping the RGB value of each normal map pixel back to the range of $[-1, 1]$ without normalization, due to the normals have been normalized when stored as the normal map. To eliminate the effect to the outline of the predicted mesh, we apply a mask M that indicates the intersection between n_i and $\hat{n}_i$. While M can be easily obtained by applying an element-wise product between n_i and $\hat{n}_i$ based on their alpha channels.

In addition, we add two geometric constraints to regularize the angles between neighboring faces, and the length of edges. The smooth loss $\mathcal{L}_{smooth}$ is used to insure the consistency of the surface. $\mathcal{L}_{smooth}$ acts on the predicted mesh $f(n_i)$ directly:

$$\mathcal{L}_{smooth} = \sum_{(a_j, b_j) \in F_i} (1 + cos < a_j, b_j >)^2, \tag{3}$$

Here a_j, b_j are two adjacent faces of the mesh $f(n_i)$, and F_i denotes the set of all adjacent face pairs.

The edge loss is added to prevent the 3D mesh deforming too much in local areas:

$$\mathcal{L}_{edge} = \sqrt{\frac{1}{N} \sum_{e_j \in E_i} (e_j - \bar{e})} \tag{4}$$

We calculate the standard deviation of the edge length set, e_j presents one edge (length) in the edge set E_i, where E_i contains all edges of the mesh, N denotes the number of edges, $\bar{e}$ is the mean length of the edges.

The final objective function for the shape prediction is the weighted sum of above loss functions:

$$\mathcal{L} = \lambda_n \mathcal{L}_{normal} + \lambda_r \mathcal{L}_{rotation} + \lambda_s \mathcal{L}_{smooth} + \lambda_e \mathcal{L}_{edge} \tag{5}$$

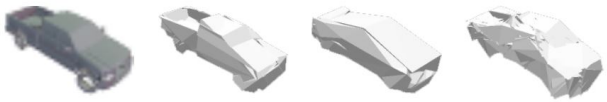

Figure 2: The results with different loss terms. First is the input image, second is the result with full loss terms, third is the result with L2 normal distance, fourth is the result without geometric loss.

3.2 Pose parameterization

We assume that the camera intrinsics are known, then the pose can be parameterized as the camera extrinsic matrix $T_c = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times4}$, which is comprised of a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a translation vector $t \in \mathbb{R}^3$.

We presented the normal map rendering function $\hat{n} = R(f(n), v)$, here $n = g(x)$ is the normal map generated from the given single sketch image x, and v is the viewport that denotes the camera pose T_c.The normal vectors of the mesh are calculated in the camera space, for ease of explanation, we omit the symbol of the rendering, the normal map rendering function can be rewritten as:

$$\hat{n} = T_c(f \circ g)(x) \tag{6}$$

Then taking it into the formula (5), and λ_s, λ_e can be set to zero due to the geometric constraints do not affect the result of pose estimation.

3.3 Normal map generation

We treat the normal map generation for a single RGB image as an image-to-image translation problem. Inspired by previous studies [6, 18], we use conditional adversarial networks (CGANs) [12] mix a global L1 distance and an area sampling regularizer to train the normal map generator.

GANs consist of a generative model G and a discriminative model D, G is generator that maps a random random vector z to an image y, $G : z \rightarrow y$ [4]. CGANs are a variant model that conditioned on extra information x [12], in this work, x is the sketch image. We define the objective function based on CGANs that is same to [6, 18] :

$$\mathcal{L}_{CGANs}(G, D) = \mathbb{E}_{x,y}[log\ D(x, y)] + \mathbb{E}_{x,y}[log\ (1 - D(x, G(x, z)))], \tag{7}$$

where G and D presents the Generator and Discriminator, x is the input image and y is the normal map, z is the random vector. By simulating the gaming between G and D to train a qualified Generator, $\hat{G} = \arg\min_G \max_D \mathcal{L}_{CGANs}(G, D)$.

Previous methods have indicated that it is beneficial to add L1 or L2 distance loss to GANs objective [6, 13]. Compare to the L1 distance, L2 distance encourages image blurring [13], therefore we calculate the L1 distance between the generated normal map and the ground truth to regulate the global image distribution:

$$\mathcal{L}_{global}(G) = \mathbb{E}_{x,y,z}[||G(x, z) - y||_1], \tag{8}$$

Additionally, in each iteration of training, we sample a few pixels from strong geometric structure areas to enhance the constraint of local features in normal map. The local area regularizer can be expressed as:

$$\mathcal{L}_{local}(G) = \mathbb{E}_{\hat{\varphi}\subset G(x,z),\varphi\subset y}[||\hat{\varphi} - \varphi||_1], \tag{9}$$

The normal map generator g is obtained by optimizing the objective:

$$g = \arg\min_G \max_D \mathcal{L}_{CGANs}(G, D) + \lambda_g \mathcal{L}_{global}(G) + \lambda_l \mathcal{L}_{local}(G). \tag{10}$$

4 EXPERIMENTS

4.1 Experimental setup

Dataset. To compare our approach with the state of art Non-3D-supervision learning frameworks, we use the ShapeNet [1] dataset that they used. For each model, we render its lighting images and normal images from 24 azimuth angles with 30 degree elevation angle. The lighting images are rendered under same light intensity with Phong shading. The normal maps are rendered by mapping vertex normal vectors to RGB range, the normal values are calculate in the camera view space. The resolution is 256×256. For category, we random select 1000 models for rendering. Thus, we obtain $1000\times$ 24 lighting images and 1000×24 normal maps with the resolution of 256×256 for each training category.

Network architecture. The overview of the framework is shown in Figure 1. Specifically, for fair comparison, the encoder of f is nearly same to that of [8, 23], which consists of 3 convolution layers with 64, 128 and 256 channels respectively, and the fixed filter size of 5×5; after the convolution layers are 3 fully-connected layers, the first two layers have 1024 neurons, the last layer has 512 neurons. The decoder decodes the latent unit that get from the encoder by 3 other fully-connected layers with 1024, 2048 and 642 neurons respectively. The network for training normal map generator g is a CGAN architecture, it is composed of a Generator and a Discriminator. The CGAN is a common choice for image-to-image translation [6, 12, 18], we follow the settings in [18].

Evaluation metric. We evaluate the quality of shape prediction by calculating the mean IoU ratio between the ground-truth and predicted shapes, which is a standard metric for single-image reconstruction. In addition, we calculate the pixel-wise L1 distance of normal map between the ground-truth and predicted shapes for surface geometric evaluation.

4.2 Training

Normal map generation. In every iteration, we fed 256×256 pixels lighting image to the Generator, the rendered normal map and the generated intermediate normal map were transferred to the Discriminator. We use the Sobel filter to calculate the gradient of the normal map to obtain a wait list of pixels for local sampling, due to the fact that the normal map conveys the geometric features

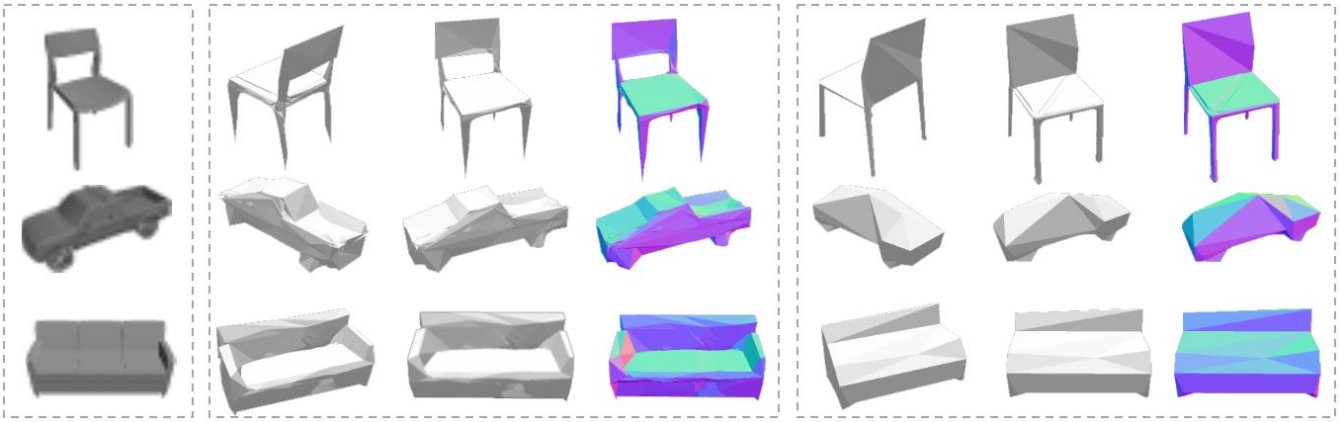

Figure 3: Comparison experiments. First column: input image. Second to fourth columns: result of our approach. Fifth to seventh columns: result of [8].

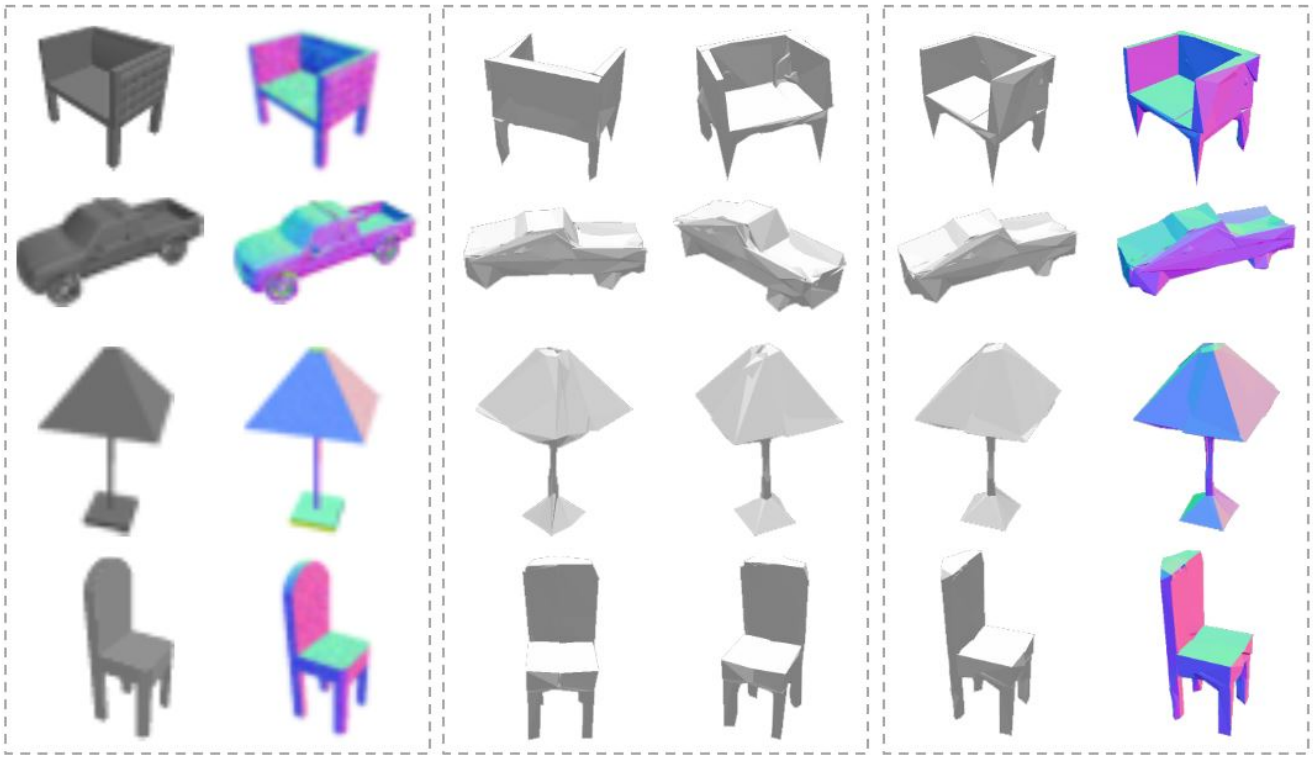

Figure 4: 3D mesh reconstruction and pose estimation from a single image. First column is the input image. Second is the normal map that generated by the normal map generator g. Third to fourth columns are predicted mesh that rendered from two random viewports. Fifth and sixth column show the lighting image and normal map that rendered under the estimated camera pose.

Table 1: L1 Normal distance (The lower is better)

	chair	car	bench	lamp	airplane
Silhouette supervision [8]	0.6493	0.4214	0.6010	0.4905	0.4172
Normal map supervision (ours)	**0.5045**	**0.2526**	**0.4880**	**0.3372**	**0.3351**

Table 2: IoU ratio (The higher is better)

	chair	car	bench	lamp	airplane
Silhouette supervision [8]	0.5091	0.7251	**0.4728**	0.4126	**0.6085**
Normal map supervision (ours)	**0.5711**	**0.8132**	0.4356	**0.6157**	0.5314

of model surface. The RMSProp optimizer with 5e-5 learning rate was used.

Mesh reconstruction. For fair comparison, we down-sampled the normal images to 64×64, then fed the resized images to the network. A pre-defined isotropic sphere with 642 vertices was used as the underlying mesh, which is identical to the settings in [8]. In all cases we set the weights with $\lambda_n = 0.001, \lambda_r = 1e5, \lambda_s = 0.01, \lambda_v = 1.0$. The Adam optimizer with $\alpha = 0.0001, \beta_1 = 0.9$ and $\beta_2 = 0.999$ is used. For each category, we set the batch size to 64, iteration to 20000, then the mean training time is 125.47 minutes under the condition of a single GTX1080 GPU.

4.3 Result evaluation

We trained several models from each category, and compared our results with the state of art Non-3D-supervision mesh reconstruction method [8]. For each category, we random tested 10 groups and calculated the mean value of the IoU ratio and L1 normal distance. The results of [8] were obtained based on their pre-trained models. Figure 3, Table 1 and Table 2 present a part of the results. Our approaches have achieved encouraging results in recovering mesh surface details and camera poses.

5 CONCLUSION

In this paper, We presented a learning framework for recovering 3D detailed mesh and camera pose of the object from a single RGB image. It shows that the multi-view normal maps can be used as the supervision for learning 3D detailed mesh reconstruction. The normal map conveys both 2D silhouette and 3D geometric details information that is beneficial to train the network for recovering surface geometric structure of the mesh. We also proved that the 3D pose can be parameterized into the proposed objective function of mesh reconstruction. The encouraging results indicate our approaches outperform state of art in learning single-image detailed mesh reconstruction and pose estimation without relying on 3D supervision. The generated normal map has a significant influence on final result of mesh prediction, it would be interesting to apply similar ideas for learning mesh surface editing or 3D style transfer in the future study.

ACKNOWLEDGMENTS

The authors would like to appreciate the open dataset *ShapeNet* and open deep learning framework *Pytorch*. Yanran Li has received research grants from the South West Creative Technology Network.

REFERENCES

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

[2] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*. Springer, 628–644.

[3] Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 605–613.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[5] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 2018. A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 216–224.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

[7] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. 2018. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 371–386.

[8] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3907–3916.

[9] Alex Kendall and Roberto Cipolla. 2017. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5974–5983.

[10] Alex Kendall, Matthew Grimes, and Roberto Cipolla. 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*. 2938–2946.

[11] David Marr. 1982. Vision: A computational investigation into the human representation and processing of visual information, henry holt and co. *Inc., New York, NY* 2, 4.2 (1982).

[12] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[13] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2536–2544.

[14] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 2017. 6-dof object pose from semantic keypoints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011–2018.

[15] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. 2017. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3577–3586.

[16] Peggy Seriès and Aaron Seitz. 2013. Learning what to expect (in visual perception). *Frontiers in human neuroscience* 7 (2013), 668.

[17] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. 2015. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*. 2686–2694.

[18] Wanchao Su, Dong Du, Xin Yang, Shizhe Zhou, and Hongbo Fu. 2018. Interactive sketch-based normal map generation with deep neural networks. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 1 (2018), 22.

[19] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. 2017. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*. 2088–2096.

[20] James T Todd. 2004. The visual perception of 3D shape. *Trends in cognitive sciences* 8, 3 (2004), 115–121.

[21] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. 2018. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2897–2905.

[22] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. 2018. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 52–67.

[23] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. 2016. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*. 1696–1704.