A Comparative Ablation Study Between Pixel2Mesh and Depth-Aware Pixel2Mesh

Philipp Hermüller, Philipp Steininger, Valérie Redl, Timur Krüger
Technical University of Munich
Munich, Germany
philipp.hermueller@tum.de, ph.steininger@tum.de,
valerie.redl@tum.de, timur.krueger@tum.de

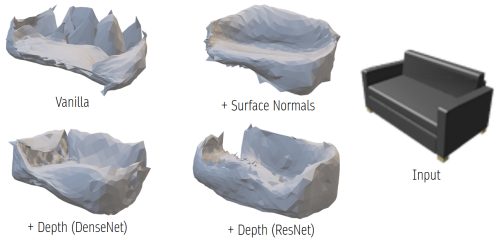


Figure 1. Shows the predicted meshes of the modified models trained for 10 epochs on a training set of 3352 images of one class.

Abstract

By generating depth for RGB images before using them in training, Quevedo et al. [12] were able to improve the performance of Pixel2Mesh [14]. They stated "[...] when adding the depth channel to the input images, the model was able to converge slightly faster [and has] lower overall loss on the unseen data as well" [12, p. 5]. However, an ablation study comparing different depth-generation models to create this additional information was not part of Quedevo et al.'s paper. Therefore, we compare the performance of Pixel2Mesh and the adjusted Pixel2Mesh implementations that use depth and surface normal models to generate additional input channels. Especially the usage of surface normals resulted in more robust and qualitative better results, as we will show throughout this report.

1. Introduction

Monocular 3D object reconstruction is the reconstruction of three-dimensional objects from RGB images. Multiple models already exist that tackle this task in different representations, e.g., via occupation grid [4], point clouds [6], deforming an initial mesh [14], and refining an occupation grid to a mesh [7]. However, monocular 3D object reconstruction remains challenging, especially for highly complex objects.

Out of all these approaches, Pixel2Mesh has many variations and extensions. These extensions improve upon the network itself or explore additional input while remaining monocular, for example, Quevedo et al. [12], Dongsheng et al. [5], and Wen et al. [15], which utilize depth, attention, and multi-view respectively. The work of Quevedo et al. extends the initial RGB image into the third dimension by generating depth as an additional input dimension for Pixel2Mesh. To achieve this, Manvi et al. apply MiDaS [13], a monocular depth estimation, to the image. Therefore the overall reconstruction remains monocular.

Even though they stated faster convergence as an improvement of their model, this is not clearly visible in their paper's provided loss curves [12, p. 5]. Additionally, no ablation study was made exploring the specific consequences and detailed effects of including depth as another input channel.

Our work explores the effects of an additional input dimension stemming from another pre-trained model in the form of an ablation study. We inspect the repercussions of employing different depth models for the additional input dimension. Further, we show that providing the Pixel2Mesh model with surface-normal estimations as another input dimension improves the training more than providing depth estimation. In addition, we show the improvement/decline of the predictions regarding robustness towards brightness, blur, noise, and saturation.

The resulting predictions for an exemplary input image using the modified models used throughout our study can be seen in Figure 1.

Overall, our contributions include:

- A comparison between Pixel2Mesh and depth-aware Pixel2Mesh regarding robustness, loss convergence, and performance on different loss scores (e.g. loss normal, loss edge) and evaluation metrics.
- A more robust and accurate variation of Pixel2Mesh using surface normal estimations as additional input by passing the initial RGB input through a pre-trained surface normal estimation network.

2. Methods

To provide Pixel2Mesh with a diversified input, we use depth and surface normal estimation models to generate additional input channels. This way, the initial RGB image gets transformed to RGB-D and RGB-XYZ input tensors. Upon those tensors, segmentation is performed to discard empty space around the displayed object.

2.1. Monocular Depth Models

A Dense Convolutional Network (DenseNet) [9] connects the layers in a feed-forward fashion. DenseNet has several compelling advantages, according to Huang et al., such as strong gradient flow, parameter and computations efficiency, and maintaining low complexity features [9]. This architecture can be used to achieve detailed high-resolution depth maps [1].

A Residual Neural Network (ResNet) [8] consists of hundreds of layers, much deeper than the previously mentioned neural network. Due to insufficient labeled samples, a small amount of labeled training data can cause problems such as overfitting in deeper networks. Many researchers have employed residual networks to solve this problem. For example, Mou et al. [11] uses ResNet to deal with image classification.

Both monocular depth estimation models that we employ are Feature-Pyramid Networks [10] with different backbones. While the model proposed by Alhashim et al. [1] uses DenseNet, the Depth-Estimation-PyTorch model [1] utilizes ResNet. Due to their similar performance in our predictions, we select the DenseNet-based implementation of Alhashim et al. to represent depth-aware Pixel2Mesh implementations.

[1] https://github.com/wolverinn/Depth-Estimation-PyTorch

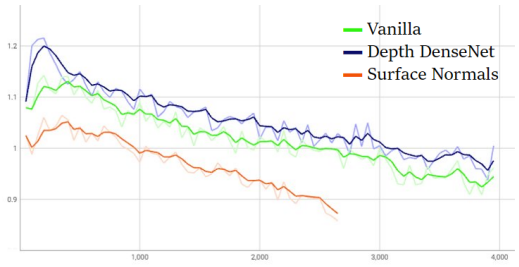
2.2. Surface Normal Models

Surface normals indicate a surface's orientation and allow the possibility to draw better conclusions about its original 3D space captured by an image. We use the state-of-the-art surface normal estimation method proposed by Bae et al. [2] to incorporate surface normals into Pixel2Mesh. This method achieves its results by sampling the pixels based on the estimated aleatoric uncertainty in the surface normals.

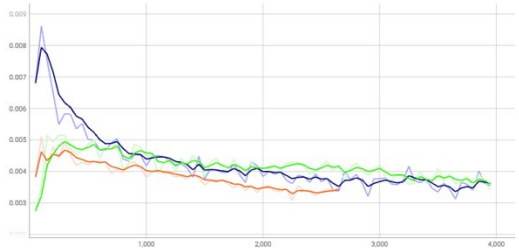
2.3. Training

As a baseline for our research serves the PyTorch implementation of Pixel2Mesh[2]. The pre-trained model to generate DenseNet[3] and ResNet [4] based depth, as well as the surface normals[5], are applied in the data loader. We utilize a ShapeNet [3] subset consisting of the sofa class with a data split of 3352 images for training and 589 images for validation and testing. We apply the already-defined optimization parameters. Because of limited resources, we constrain our training to ten epochs. The vanilla and depth-aware Pixel2Mesh takes around eight hours of training on an Nvidia GTX 1070 with a batch size of 32, and the surface normal-aware Pixel2Mesh takes about ten hours with a batch size of 48 on an Nvidia RTX 3070.

[2] https://github.com/noahcao/Pixel2Mesh
[3] https://github.com/alinstein/Depth_estimation
[4] https://github.com/wolverinn/Depth-Estimation-PyTorch
[5] https://github.com/baegwangbin/surface_normal_uncertainty



(a) Normal Loss



(b) Edge Loss

Figure 2. The above charts depict two different loss scores of the used models.

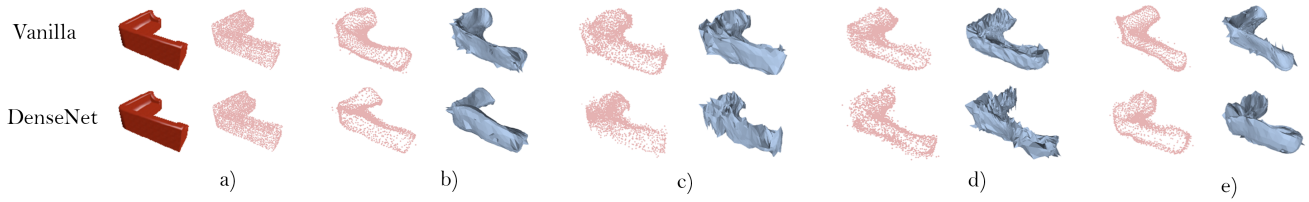


Figure 3. Point cloud and reformed mesh comparison between Pixel2Mesh (Vanilla) and depth-aware Pixel2Mesh (DenseNet) to depict the impact of different loss scores throughout the training. a) shows the input image and ground truth; b) utilizes all loss scores, disabling the c) edge loss, d) Laplacian loss, and e) normal loss.

3. Evaluation and Generalization

To evaluate our models, we investigate their performance regarding input robustness and the impact of utilized loss scores.

3.1. Loss Functions

In the original Pixel2Mesh paper, Wang et al. define multiple losses to generate qualitatively appealing meshes. These include a Chamfer Distance loss, a surface normal loss, Laplacian regularization loss, and Edge length loss [14, p. 5]. While the different network's loss graphs mostly behave similarly during the training (e.g., Figure 2b), the surface normal loss, designed to favor smooth surfaces [14, p. 3], improves when utilizing surface normals as additional input, as seen in Figure 2a. Comparing the resulting meshes in Figure 1 one can see the comparatively smoother surface compared to the monocular depth-based models and the vanilla model.

Figure 3 depicts the impact of disabling edge loss, normal loss, and Laplacian loss for the vanilla Pixel2Mesh and the DenseNet Pixel2Mesh model after training for ten epochs. Disabling the Laplacian loss, which aims to prevent mesh face intersection, and the edge loss, designed to encourage the uniform distribution of mesh vertices [14, p. 3], increases noise in the respective point clouds. The effect appears stronger on the depth-aware Pixel2Mesh implementation. Removing the normal loss negatively impacts surface smoothness in both models.

3.2. Robustness

Real-world images often vary in e.g., brightness, saturation, and shadows depending on the location and camera settings. For a model to be used with real-world images, it should be invariant, or at least robust, to input variations.

Shadows In contrast to the Computer-Aided Design (CAD) based images from our training and test set, real-world images contain shadows. The monocular depth estimation models we use to estimate depth predict a closer distance to some shadows than to the object itself. This prediction often leads the models to create a surface for shadows even though it is not part of the object. Neither adding depth nor surface normal estimations as additional channels help against this problem in early epochs due to the inaccuracy of the predictions of our used methods.

Blur Sometimes cameras focus on the wrong objects, resulting in blurry images. To simulate this, we augmented the input images with a Gaussian blur of kernel size of 5. The predicted meshes of the vanilla Pixel2Mesh model become thicker, for example, a thicker armrest on a sofa. At the same time, blur augmentation has less influence on the depth and surface normal Pixel2Mesh implementations.

Background Color and Brightness The Pixel2Mesh implementation uses segmented images of objects as input with white as the consistent background color. This makes predictions of white objects on a white background more complicated due to parts of the object being considered as background. Real-life image color distribution depends on the brightness and can be very dark or very bright. To counter the conflict of background and object color we calculate the depth and surface normal estimation based on a black background while the RGB input for Pixel2Mesh's background stays white. This image manipulation, based on the image's alpha channel, enables distinction between object and background for at least one channel.

As seen in Table 1, the surface normal extension of Pixel2Mesh is more robust to extensive changes in brightness. We assume that while depth-aware Pixel2Mesh trains on only one channel containing the black background color, surface normal-aware Pixel2Mesh has an additional three. Therefore, in the prior case, the model relies more on the RGB part than on the depth part; in the latter, the model relies on both the RGB part and the surface normal estimations equally.

Model	Metric	(Standard)	160% Brightness	Difference
Depth [DenseNet]	CD	0.000897	0.004472	+0.003575
	F1-τ	32.9113%	25.9896%	-6.9217%
	F1-2τ	52.0284%	39.9660%	-12.0624%
Vanilla	CD	0.000865	0.003896	+0.003031
	F1-τ	33.6147%	24.6768%	-8.9379%
	F1-2τ	52.7496%	37.7286%	-15.021%
Surface Normals	CD	0.000875	0.000945	**+0.00007**
	F1-τ	34.4806%	33.9051%	**-0.5755%**
	F1-2τ	53.3432%	52.2931%	**-1.0501%**

Table 1. Evaluation metrics on the different Pixel2Mesh extensions when increasing the brightness by 60%.

3.3. Point Clouds and Meshes

The Chamfer distance and F1-score evaluation metrics are computed with point clouds consisting of the mesh's vertices and not the meshes themselves. Even though these evaluation metrics result in similar scores after ten epochs, a difference in the point clouds can be observed. As seen in Figure 4, the depth-aware model as well as the surface normal-aware model contain fewer points on flat surfaces and more in curvy or detailed areas early on in the training.

4. Future Work

Another possible variation of the Pixel2Mesh approach we have not yet explored would include:

- Surface Normal Uncertainty estimations as an additional input channel. These estimations contain higher values for smaller details like the legs of a chair, therefore, we would expect this additional input to help the model with these finer details.
- A comparison between the surface normal-aware Pixel2Mesh's loss function and the impact of loss scores for Pixel2Mesh and depth-aware Pixel2Mesh.
- A combination of depth and surface normal estimation as input.

5. Conclusion

The comparison of Pixel2Mesh, depth-aware Pixel2Mesh, and surface normal-aware Pixel2Mesh depicts a higher quality performance of our adjusted Pixel2Mesh implementations. While the evaluation scores of the different models behave similarly, the qualitative results improve when combining the input with additional channels. The addition of depth and surface normal information boosts the smoothness of the generated 3D model's surface as well as the robustness regarding the manipulation of the input images. During training, a lower normal loss can be observed when using the surface normal-aware Pixel2Mesh.

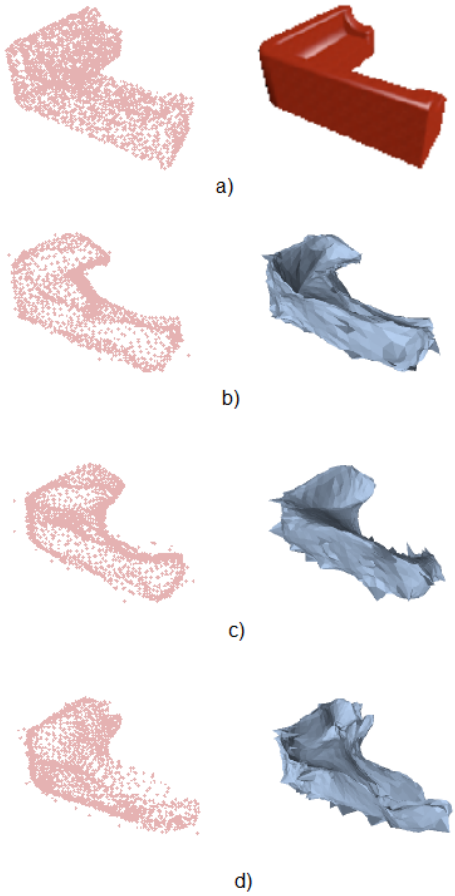


Figure 4. Point cloud and reformed mesh comparison between a) the ground truth, b) Depth-aware Pixel2Mesh (DenseNet), c) Pixel2Mesh with surface normals, and d) vanilla Pixel2Mesh. The point clouds in b-d) are the vertices from the predicted reformed mesh.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018. 2

[2] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13137–13146, 2021. 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. 2

[4] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 628–644. Springer, 2016. 1

[5] Yang Dongsheng, Kuang Ping, and Xiaofeng Gu. 3d reconstruction based on gat from a single image. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 122–125. IEEE, 2020. 1

[6] Haoqiang Fan, Hao Su, and Leonidas J Guibas. 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*, pages 605–613, 2017. 1

[7] 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

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 2

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[11] Lichao Mou, Pedram Ghamisi, and Xiao Xiang Zhu. Unsupervised spectral–spatial feature learning via deep residual conv–deconv network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(1):391–406, 2017. 2

[12] Rohin Manvi Kabir Jolly Julian Quevedo. Depth-aware pixel2mesh. 1

[13] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, 2019. 1

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

[15] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1042–1051, 2019. 1