CIVILEN 6451 Photogrammetry

Final Project

Bing Zha

zha.44@osu.edu

Surface Normal Estimation from RGB Image using Deep Learning

1. Introduction

In this final project of photogrammetry class, my topic is to estimate surface normal from monocular RGB image using deep learning technologies. Given a RGB image, there are much information in this image we can extract, such as semantic information and geometric information. For semantic information, we can use segmentation or labeling method to assign every pixel a specific object class. For geometric information, we still can extract some useful information, including depth or surface normal. Therefore, estimating the orientation of surfaces in an image is an important in reconstructing a 3D model of the scene. Recently, the surface normal estimation from monocular RGB images, has been an active area of research in computer vision.

Overall, the problem we address in this project is to estimate pixel-wise surface normal from monocular RGB images of the indoor scenes.

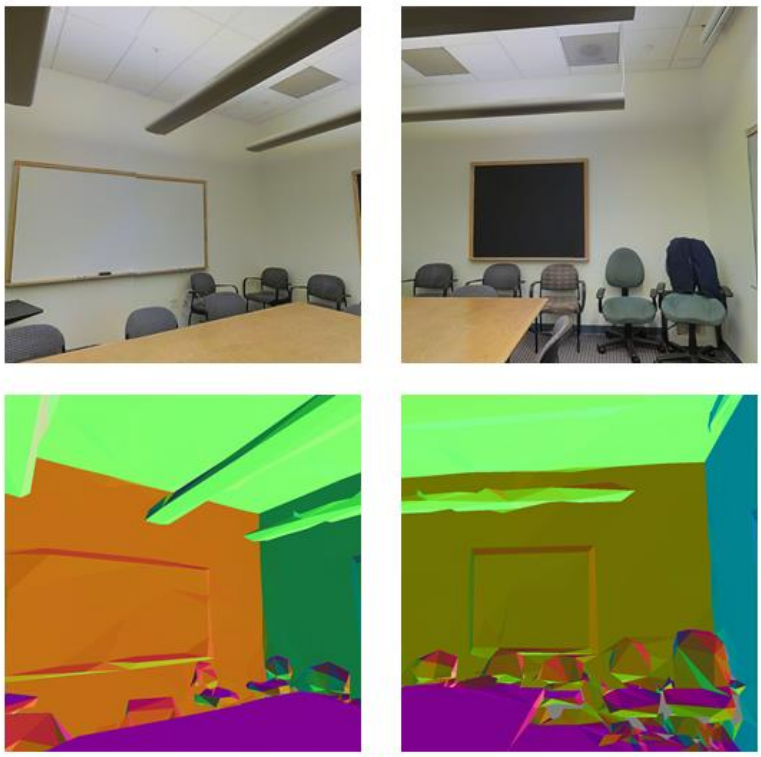


Figure 1: Top: RGB images of indoor scenes, Bottom: corresponding surface normal maps

2. Related Work

Before discussing the related work on surface normal estimation from RGB images, it is best to mention a similar problem: estimating the depth map from RGB images. By having the depth map of scene, it is possible to calculate the pixel-wise surface normal by simply fitting a least-square plane for the neighboring sets of points in the 3D point cloud. However, depth map estimation is also a challenging task. In fact, some of the works discussed in this section (Dharmasiri, et al., 2017; Qi, et al., 2018), attempt to jointly estimate depth and surface normal

maps. Other methods for depth map estimation have been proposed, too; method such as Markov Random Fields, Conditional Random Fields, focus information and higher order statistics, semantic labels, deep convolutional neural fields, and multi-scale deep neural networks.

In recent years, there have been several attempts to tackle the problem of surface normal estimation from RGB images. Some of the better performing works are all based on convolutional neural networks. The main difference between them lies in their network architecture. Therefore, the main focus of this section in discussing these methods, is the architecture of underlying CNN models.

3. Our Methods

Motivated by pixel-wise semantic labeling work using SegNet (Badrinarayanan, et al., 2016), we first normalize and transform the normal surface components (x, y, z) into unit spherical coordinates $(r,\ \theta,\ \varphi)$. The value of r is always equal to 1. The range of θ is in [0, 360] and φ is in [0, 180]. Then, we divide the unit sphere and assign normal surface a specific class bin of unit sphere.

3.1 Network Architecture

We adopt the SegNet as our network architecture, which is an encoder-decoder type network design. The first 13 layers in the VGG16 network comprise the

encoder network in SegNet. Each layer is 3x3 convolutions stack on each other. The encoder received three channel image input to generate a low dimensional representation which is passed onto the decode that classifies pixels within the image.

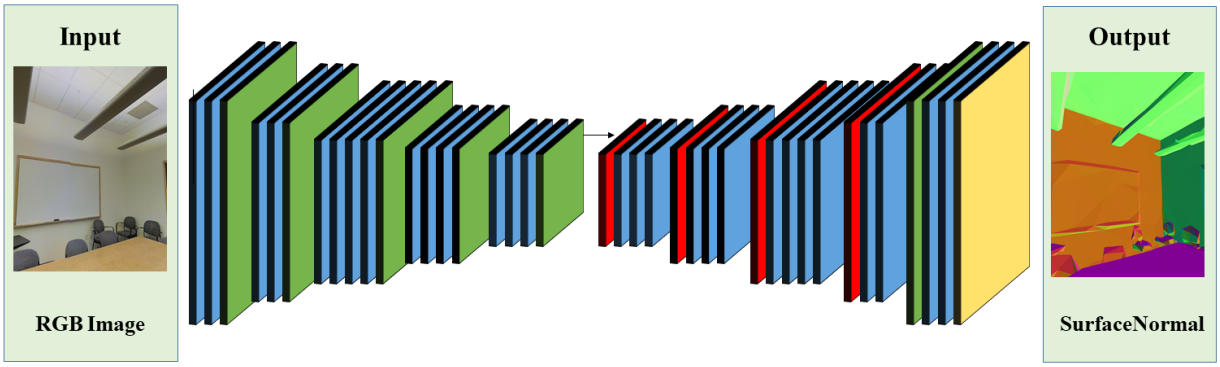


Figure 2: SegNet based encoder-decoder architecture for normal surface estimation using RGB images

The decoder is the mirror of the encoder, such that for each layer in the encoder network there is a corresponding layer in the encoder network. Class labels are generated in the last layer of the decoder using a softmax classifer, which predicts pixel labels.

The breakdown of the *encoder* is as follows:

1. Generate feature maps from various filters
2. Batch normalize the generated feature maps
3. Apply ReLU
4. Perform max-pooling with 2x2 window + stride of size 2
5. Downsize the output by a factor of 2

The breakdown of the *decoder* is as follows:

1. Generate sparse feature maps using max-pooling indices from the corresponding encoder layer
2. Generate dense feature maps by convolving sparse feature maps with trainable filters
3. Batch normalize generated feature maps
4. Predicts pixel labels through a multi-class softmax function

3.2 Discretization of Surface Normal

The original surface normal coordinates is Cartesian coordinates (x, y, z). We used the following equation to transform and then get the spherical coordinates.

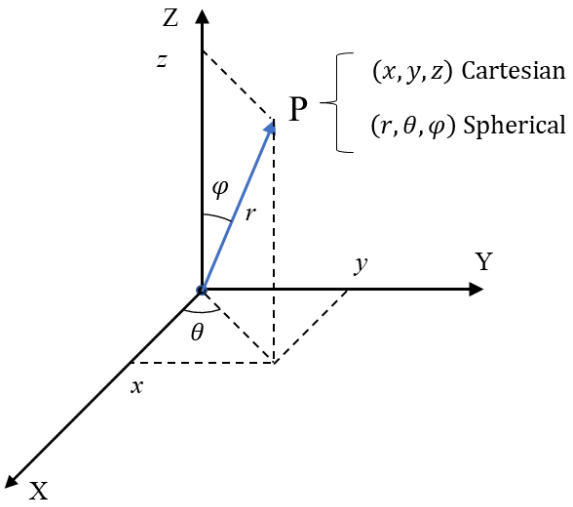


Figure 3: Transformation between Cartesian coordinates and Spherical coordinates

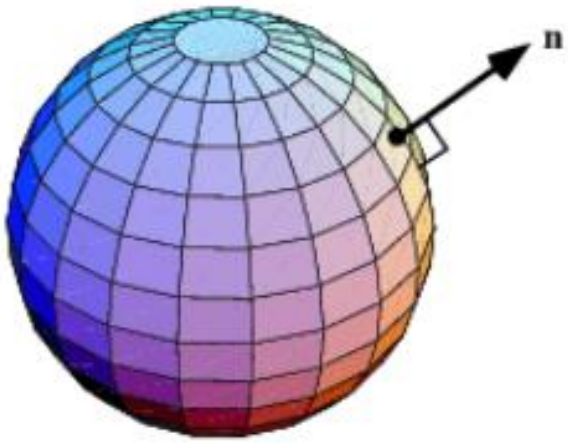


Figure 4: Divide the unit Sphere and quantize the surface normal

$$\begin{cases} x = \dfrac{x}{\sqrt{x^2+y^2+z^2}} \\ y = \dfrac{y}{\sqrt{x^2+y^2+z^2}} \\ z = \dfrac{z}{\sqrt{x^2+y^2+z^2}} \end{cases}$$

$$\begin{cases} r = \sqrt{x^2+y^2+z^2} \\ \theta = \tan^{-1}\dfrac{y}{x} \\ \varphi = \cos^{-1} z \end{cases}$$

$$\mathrm{r} = 1, \theta \in [0, 360^\circ\ ], \varphi \in [0, 180^\circ\ ]$$

θ \ φ	[0-60°]	[60-120°]	[120-180°]
[0-60°]	1	2	3
[60-120°]	4	5	6
[120-180°]	7	8	9
[180-240°]	10	11	12
[240-300°]	13	14	15
[300-360°]	16	17	18

Table 1: Assign a single number to quantized surface normal

4. Experiments and Results

4.1 Datasets

To test the proposed approach, we used Stanford 2D-3D Semantics Dataset (2D-3D-S) (Armeni et al., 2017), which contains RGB images as well as the corresponding surface normal images for 11 types of indoor scenes. Due to educational and office use, most of the dataset are office rooms and hallways and just only a small part of other rooms such as lobby and auditorium. The data is collected using the Matterport Camera, which combines 3 structured-light sensors to capture RGB and 360-degree depth images.

The surface normal are computed from a normal pass in Blender and are saved as 24-bit RGB PNGs. The surface normal in 3D corresponding to each pixel are computed from the 3D mesh instead of directly from the depth image. The normal

vector is saved in the RGB color value where Red is the horizontal value (more red to the right), Green is vertical (more green downwards), and Blue is towards the camera. Each channel is 127.5-centered, so both values to the left and right (of the axis) are possible. For example, a surface normal pointing straight at the camera would be colored (128, 128, 255) since pixels must be integer-valued.

Missing values take (128, 128, 128) which is convenient in practice as it is not a unit normal and is clearly visually distinguishable from the surrounding values. The convention is that surface normals cannot point away from the camera.

4.2 Preprocessing

0: 0.5239
1: 0.0001
2: 1.9022
3: 8.9514
4: 0.0002
5: 13.0027
6: 4.2505
7: 0.0003
8: 2.5191
9: 11.1274
10: 0.0014
11: 4.9667
12: 15.8426
13: 0.0009

14: 13.8093

15: 5.3435

16: 0.0009

17: 5.1841

18: 12.5728

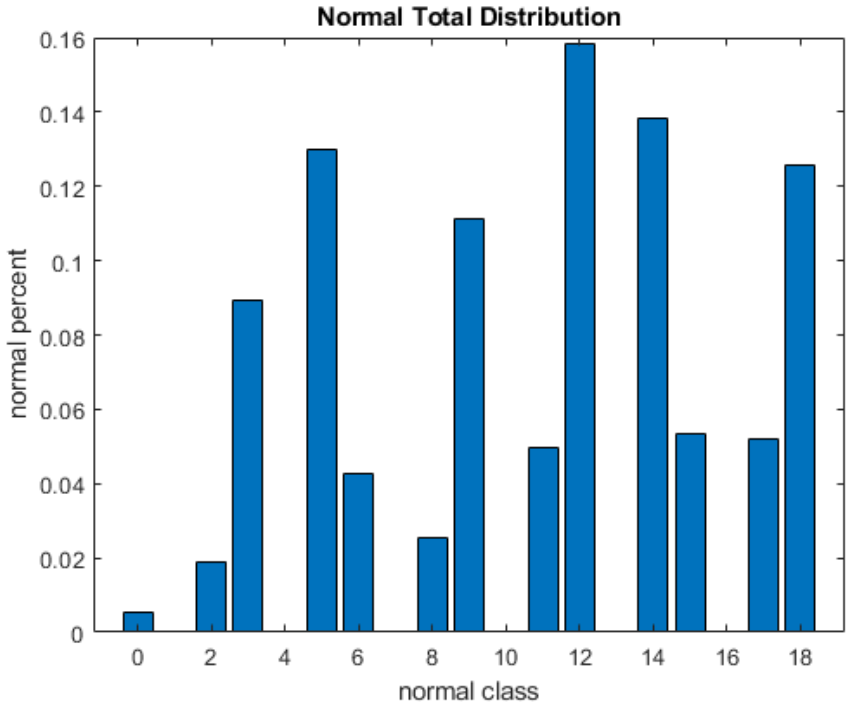
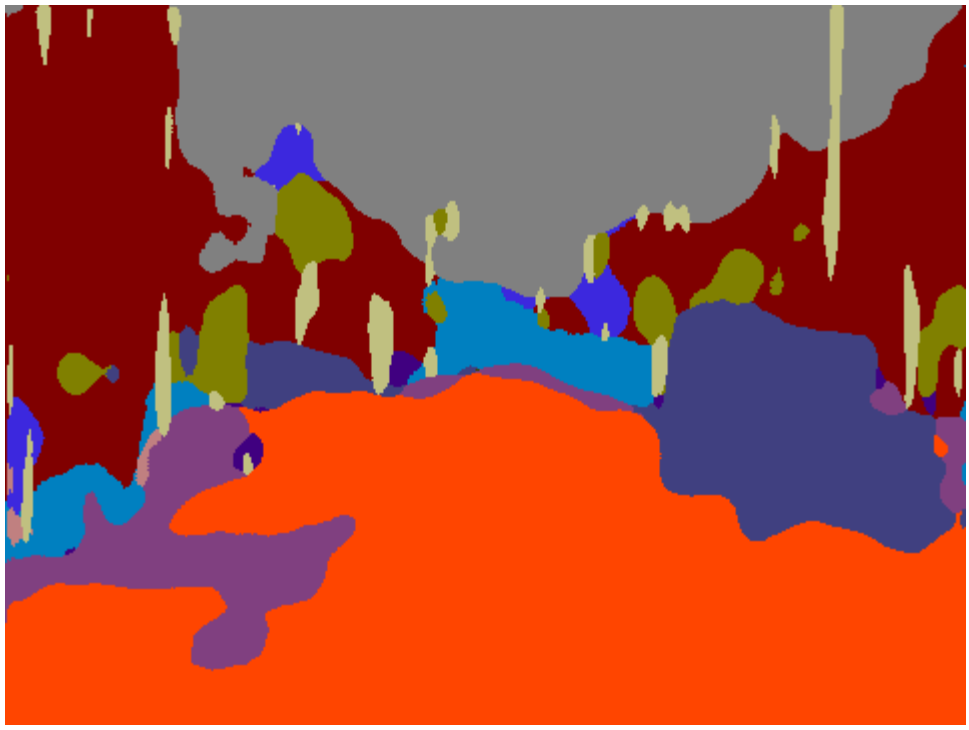


Figure 5: Quantized surface normal class distribution

4.3 Experiments and Results

This is just semantic labeling: test acc: 0.7585696083690987 mean IU: 0.4246364025620113



5. Discussion and Conclusion

In this project, I did pixel-wise surface normal estimation from monocular RGB image. For now, I am still doing the training and testing part.

References

Dharmasiri, T., Spek, A., & Drummond, T. (2017). Joint Prediction of Depths, Normals and Surface Curvature from RGB Images using CNNs. arXiv preprint arXiv:1706.07593.

Qi, X., Liao, R., Liu, Z., Urtasun, R., & Jia, J. (2018). GeoNet: Geometric Neural Network for Joint Depth and Surface Normal Estimation. CVPR 2018

Wang, X., Fouhey, D., & Gupta, A. (2015). Designing deep networks for surface Normal estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 539-547).

Eigen, D., & Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2650-2658).

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), 2481-2495.

Armeni, I., Sax, S., Zamir, A. R., & Savarese, S. (2017). Joint 2D-3D-Semantic Data for Indoor Scene Understanding. arXiv preprint arXiv:1702.01105.