Jingyuan Sha

May 16, 2022

Abstract

Chapter 1

Introduction

Standard methods compute normals from point cloud using neighboring information in image space or use Shape from Shading. However, the point clouds captured by sensors like Kinect or similar RGB-D, LiDAR sensors are only semi-dense. As shown in Figure 1.1. If map these normals to mesh, that is not good at all. Standard methods dependents on chosen neighbor size, if too small, it has larger noise sensitive, if too large, the output will too smooth and not crispy. Errors may occur in regions with inter-reflections, where we have errors in the 3D measurement.

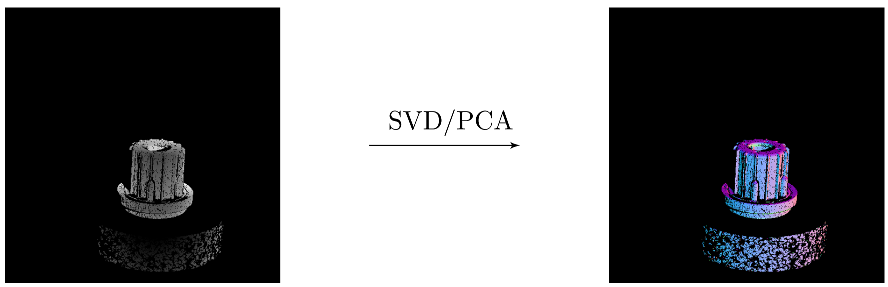


Figure 1.1: Left: Depth Map, Right: Semi-dense Normal Map

Although the depth image is incomplete, the depth sensor usually able to capture grayscale texture image, which are typically fully dense due to their passive nature. Furthermore, if the texture image is already illuminated by strong directional light of a video projector, whose position is known, then there should exist theoretical relations between light direction, normal direction, etc. Thus the normal can be inferenced better using the given image information and depth map.

Both depth and grayscale image are initially relevant to the machine learning algorithms. Based grayscale image and corresponding semi-dense depth map, a CNN model can be designed to inference the normal map, which gives more density and robust comparing to standard algorithm. However, the missing pixels in depth map can be distributed around the whole image, some of the region leaves complete empty pixels. This situation imposes further processing for the missing regions and some other challenges on the machine learning methods. In this thesis, we found a solution for the problems mentioned above.

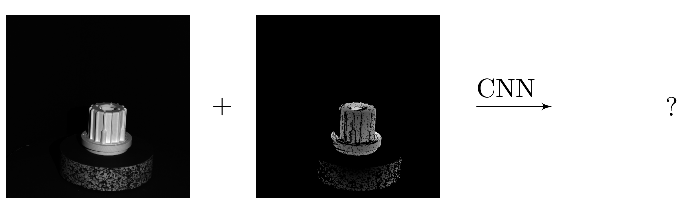


Figure 1.2: Left: Grayscale Image, Middle: Depth Map

Chapter 2

Related Work

2.1 Sparse Input processing

The depth map is supposed to be dense. Therefore, how to accept sparse depth map as input in CNNs is one of the most important problem. Some trivial solutions like median filters are good enough, if the missing pixels are sparse enough, however, for the case of huge missing holes in the depth map, it produces just a paltry result. Thus a reasonable guess is required for missing areas. Generally, it can be solved as image inpainting problems,[15],[11].

Some deep learning based method for image inpainting also achieved quite good performance for the hole mending task. Notably, in 2016, Oord et al. [10] proposed a gated activation unit for a CNN model,

$$\mathbf{y} = \tanh(W_{k,f} * \mathbf{x}) \odot \sigma(W_{k,g} * \mathbf{x})$$

to substitute the standard activation layer, where σ is the sigmoid function, which constricts the output value of second part between $[0, 1]$. The function is inspired by Long Short-Term Memory (LSTM) [6] and Rated Recurrent Unit (GRU).[2] It is originally used for learning complex interactions as LSTM gates does. In 2018, Yu et al. [16] employed same function for free-form image inpainting, which can be used to learn mask automatically from image it self.

Different to aforementioned approaches, Knutsson et al. in 2005 introduced normalized convolution [9] dealing with missing sample case for convolution operation, which aims to reconstruct the missing pixels from the sparse output sensed by cameras, which particularly considered the confidence of each interpolated pixels, since it provides the trustworthiness of the predicted value. The higher the reliability of the value inference, the better the model shape reconstruction.

In 2018, Eldesokey et al. [4] applied normalized convolution in CNN as normalized convolution layer that takes both sparse depth map and a binary confidence map as input to perform scene depth completion. In 2020, Eldesokey et al. [3] focus on modeling the uncertainty of depth data instead of assuming binary input confidence.

2.2 Normal Inference

In order to estimate normals of an object surface.

In 2012, Holzer et al. [7] presented a read-time method, which is able to run algorithm in a high frame speed. They smooth the depth data in order to handle the noise of depth image. The speed is accelerated via integral image. The drawbacks are, as mentioned in the paper, the normals error go up when point depths change severely.

In 2018, Yu et al. [16] presented a CNN based method with gated convolutional layer.

In 2018, Hua et al. [8] presented an approach integrating color image and certainty map into the network to enhance the performance of depth map density.

In 2019, Ben-Shabat et al. [1] presented a CNN based method.

In 2021, Zhou et al. [17]

The deep convolutional neural network is typically used for image classification, which achieved great success in last several years. [12], [14].

These kinds of network architecture takes a single image as input which usually employed for classification problems. The image is usually convoluted with convolutional layer and downsampling with pooling layers. The outputs of the network consists of a single value to represent the ID of corresponding class [14], or with set of values to represent the position of bounding boxes.[12].

However, in many other vision tasks, the output is demanded as an image, instead of predicting one or several classes of the input, but the classes of each pixel are predicted. In this case, the traditional network architecture is not suitable anymore.

Recently, Ronneberger et al proposed an architecture called UNet [13] for biomedical image segmentations. The architecture is shown in Figure 2.1.The first half network is a usual classification convolutional network, the second half replace the pooling layers to upsampling layers, thus in the end of the second half, the output is back to the input size. The proposed network can successfully assigned each pixel a class for segmentation tasks. Under this symmetric network, an input image is downsampled 3 times and upsampled 3 times. Output image has exactly the same size as input image. The downsampling and upsampling both have large number of feature channels, which guarantee the network propagates the information to higher resolution layers.

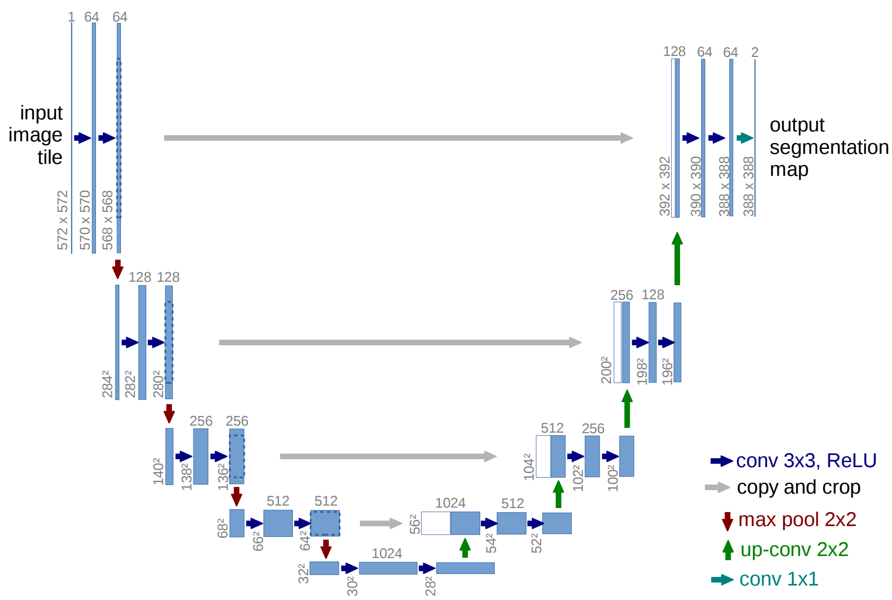


Figure 2.1: U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Chapter 3

Approach

Figure 3.1: Depth Map of an object captured by Kinect

3.1 Dataset

Since the images captured by Kinect usually have missing pixels, consequently, the ground truth of missing pixels are unknown. Therefore we generate synthetic 3D scene in game engine like Unity, in this case, all the information in the synthetic world can be measured, as well as normal map. Thus we captured the depth map and normal from Unity as input and ground truth. To construct the dataset, 22 3D models have been used and 1000 different scenes have been captured. Each scene has 3 images:

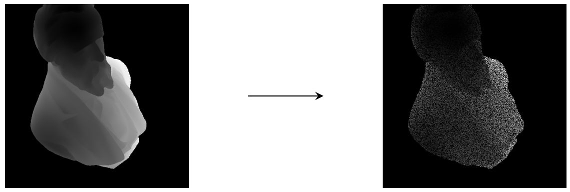
- Depth map
- Grayscale image
- Normal Map

3.1.1 Noise

The raw depth maps captured by Kinect usually have missing pixels. As shown in picture 3.1. Observing the depth map, the missing pixels can be divided into two groups.

- random missed single pixels
- missed dark areas

Since the input of evaluation is incomplete Kinect depth map, thus the input of training should have the similar pattern as well. That is, adding the similar noise to synthetic depth map. For the two kinds of depth noise, first can be simulated by adding uniformly distributed black pixels to the whole depth map, whereas the second can be dealed with a high-pass filter to filter out dark pixels and set them to 0.



Specifically, for each pixel with index (i, j) in a synthetic depth map D, if its value less than a threshold B, then set it to 0, i.e.

$$D(i,j) = \begin{cases} 0 & D(i,j) \leq B \\ D(i,j) & D(i,j) > B \end{cases}$$

for the rest of the pixels, the following two equal-opportunity situations will occur,

- pixel value remain same
- pixel value set to 0

3.2 Gated Convolution

Gated Convolution layer[16], the output of the layer with input size (N, C_{in}, H, W) and output $(N, C_{out}, H_{out}, W_{out})$ can be described as:

$$o(N_i, C_{o_j}) = \sigma(\sum_{k=0}^{C_{in}-1} w_g(C_{o_j}, k) \star i(N_i, k) + b_g(C_{o_j})) * \phi(\sum_{k=0}^{C_{in}-1} w_f(C_{o_j}, k) \star i(N_i, k) + b_f(C_{o_j})) \tag{3.1}$$

where ϕ is LeakyReLU function, σ is sigmoid function, thus the output values are in range $[0, 1]$, $\star$ is the valid 2D cross-correlation operator, N is batch size, C denotes a number of channels, H is a height of input planes in pixels, and W is width in pixels, $w(C_{o_j}, k)$ denotes the weight of j-th output channel corresponding k-th input channel, $i(N_i, k)$denotes the input of i-th batch corresponding k-th input channel, $b(C_{o_j})$ denotes the bias of j-th output channel.

3.3 Architecture

Based on the implementation mentioned above, we propose a Gated Convolutional Neural Network to perform guided normal inference. The architecture of trained network is shown in Figure 3.2. There are two stages in the downsampling and one stage in upsampling. First stage takes gray scale image as input then samples downward 3 times and extracts the features using 2D convolutional layers. The second stage takes 3D vertex as input then samples downward 3 times as well but extract the features using gated convolutional layers. Then concatenate two stages together and upsampling 3 times, two standard convolutional layers have been added at the end. The output normal map has the same size as the input 3d vertex map.

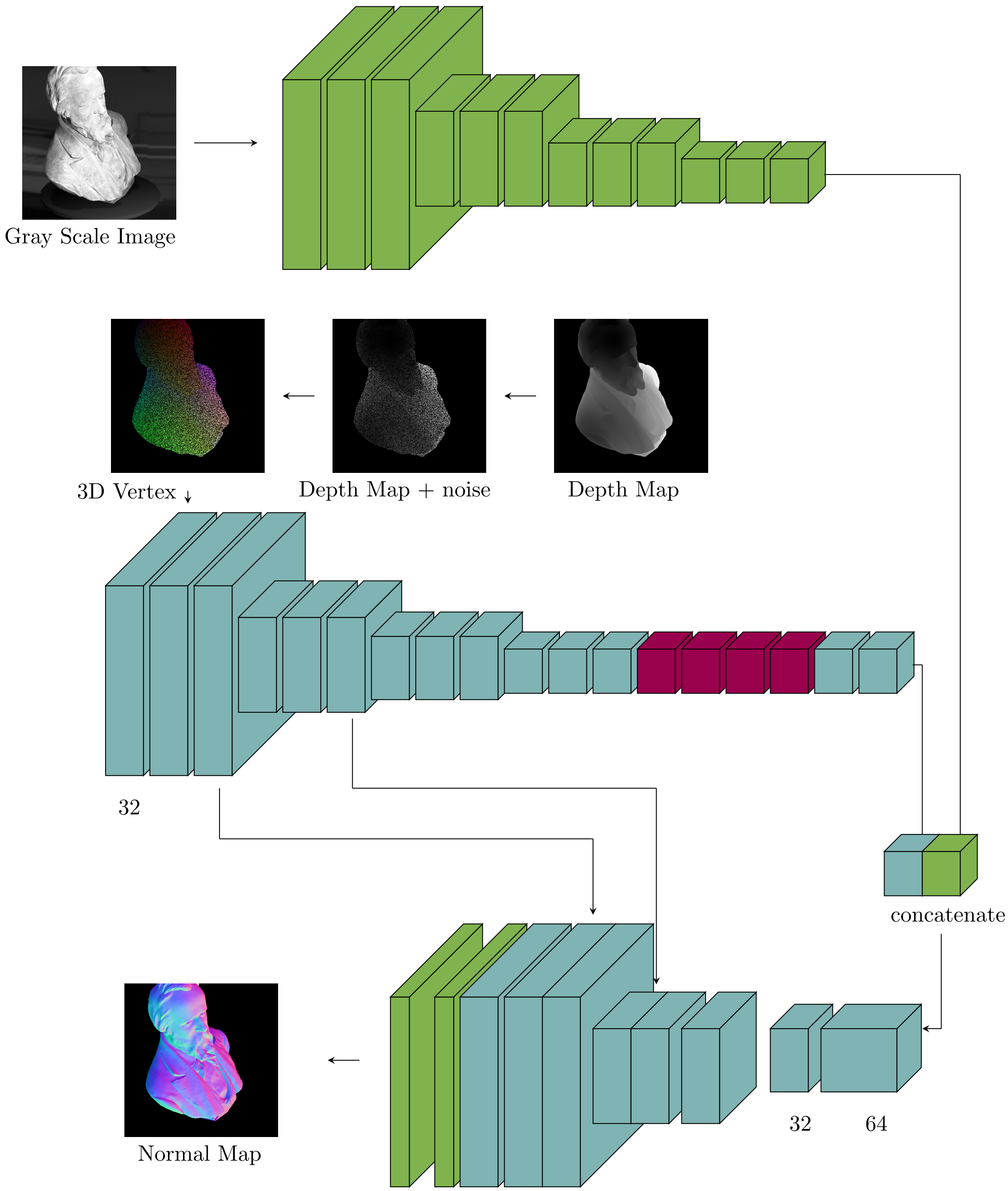


Figure 3.2: CNN Architecture

Chapter 4

Experiments

The model is trained with PyTorch 1.10.2, CUDA 10.2.89, GPU with single NVIDIA GEFORCE GTX 1080Ti.

Chapter 5

Formular

RGB image will be stored as gray value Scene using following equation:

$$gray : \frac{r + 2g + b}{4}$$

5.0.1 Normal from k neighbors

Given a point p locating on plane Π, calculate the normal n of plane Π.

First, find the nearest k neighbors $p_1, p_2, ..., p_k$ of point p using KNN-algorithms. The plane Π containing point p can be fitted using the neighbors of point p. Then the normal is available immediately.

Assume all the neighbors of point p are in plane $\Pi = ax + by + cz + d = 0$. Since we only need calculate the normal, thus with out loss of generation, we can set displacement $d = 0$. Then the normal $\mathbf{n} = (a, b, c)^T$.

Since all the neighbors of point p are located on plane Π, thus we have

$$P_{k\times 3} \cdot \mathbf{n}_{3\times 1} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

In order to avoid trivial solution, one more constraint should be added

$$\|\mathbf{n}_{3\times 1}\|_2^2 = 1$$

, which also let the normal to be a unit vector. In order to calculate a valid normal, 3 points are required at least. For the sake of robust, more points can be used to reduce the measuring error. In this case, the equation system is over-determined, which can be modeled as following optimization problem

$$\begin{aligned} \min \quad & \|P\mathbf{n}\|^2 \\ \text{s.t.} \quad & \|\mathbf{n}\|^2 = 1 \end{aligned} \tag{5.1}$$

Let the decomposition of $P = U\Sigma V^T$, The solution i.e. normal is the last column of V.

5.0.2 Normalized Convolution

Bibliography

[1] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. Nesti-net: Normal estimation for unstructured 3d point clouds using convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[2] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014.

[3] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Mikael Persson. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end, 2020.

[4] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through CNNs for guided sparse depth regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2423–2436, oct 2020.

[5] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through CNNs for guided sparse depth regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2423–2436, oct 2020.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[7] S. Holzer, R. B. Rusu, M. Dixon, S. Gedikli, and N. Navab. Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2684–2689, 2012.

[8] Jiashen Hua and Xiaojin Gong. A normalized convolutional neural network for guided sparse depth upsampling. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 2283–2290. AAAI Press, 2018.

[9] Hans Knutsson and Carl-Fredrik Westin. Normalized and differential convolution methods for interpolation and filtering of incomplete and uncertain data. 01 1993.

[10] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016.

[11] Fei Qi, Junyu Han, Pengjin Wang, Guangming Shi, and Fu Li. Structure guided fusion for depth map inpainting. *Pattern Recognition Letters*, 34(1):70–76, 2013. Extracting Semantics from Multi-Spectrum Video.

[12] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[14] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. 2019.

[15] Na-Eun Yang, Yong-Gon Kim, and Rae-Hong Park. Depth hole filling using the depth distribution of neighboring regions of depth holes in the kinect sensor. In *2012 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2012)*, pages 658–661, 2012.

[16] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Free-form image inpainting with gated convolution, 2018.

[17] Jun Zhou, Wei Jin, Mingjie Wang, Xiuping Liu, Zhiyang Li, and Zhaobin Liu. Fast and accurate normal estimation for point cloud via patch stitching, 2021.