Jingyuan Sha

February 23, 2022

**Abstract**

# Chapter 1

# Related Work

In order to estimate normals of an object surface.

In 2012, Holzer et al. [2] 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 2019, Ben-Shabat et al. [1] presented a CNN based method.

In 2021, Zhou et al. [3]

# Chapter 2

# Approach

The valid points of point clouds are usually not constant. However, the input of a CNN has a fixed shape. Thus, in order to enable a CNN to predict the normals for irregular sized point clouds, a suitable shape alignment method need to be present. That is, find a function

$$f(P_i) = M, for\ i = 1, 2, ...$$

where $P_i$ is i-th point cloud with $N_i$ points number. The shape of $P_i$ is $N_i \times 3$. $M$ is CNN input with a constant shape $N_c \times 3$.

In order to align different point clouds to a standard shape, using dimension reduction method is a straightforward way. Like PCA and autoencoder.

Another approach, consider each pixel as the input of the CNN instead of the whole image, each pixel has a fixed number of neighbors, the neighbor matrix can always be the same size, then use this matrix as the input of CNN.

# Chapter 3

# Formular

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

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

### 3.0.1 Normal from k neighbors

Given a point $p$ locating on plane $\Pi$, calculate the normal $n$ of plane $\Pi$.

First, find the nearest $k$ neighbors $p_1, p_2, ..., p_k$ of point $p$ using KNN-algorithms. The plane $\Pi$ 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 $\Pi$, 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

$$\begin{aligned} \min \quad & \|P\mathbf{n}\|^2 \\ \text{s.t.} \quad & \|\mathbf{n}\|^2 = 1 \end{aligned} \tag{3.1}$$

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

### 3.0.2 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] 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.

[3] 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.