Lidar Based Point Cloud Denoising with Manhattan World Assumption

Tao Liu
2022231023
liutao2022@
shanghaitech.edu.cn

Zijia Dai
2022233158
daizj2022@
shanghaitech.edu.cn

Xinyue Zhang
2019233140
zhangxy11@
shanghaitech.edu.cn

Shaoxun Wu
2019533211
wushx@
shanghaitech.edu.cn

Abstract

Point clouds acquired from LiDAR are often perturbed by noise, which affects downstream tasks such as ICP registration and analysis. To denoise a noisy point cloud, we propose a neural network architecture to estimate the uncertainty of point cloud based on the parameters of LiDAR sensors. We add the uncertainty to the Score-based denoising network to re-extract the features of the point cloud to optimize the network scoring results. We derive loss functions for training the network and develop a denoising algorithm leveraging on the estimated scores and uncertainty. Experiments demonstrate that the proposed model outperforms state-of-the-art methods under a variety of noise models, and shows the potential to be applied in other tasks such as point cloud registration.

1. Introduction

Point cloud, as with meshes and RGB-D images, is one of the most popular representations for 3D objects and environments. Recently, LiDAR-based Simultaneous Localization and Mapping (SLAM) has reached a significant level of maturity in many applications such as 3D reconstruction , autonomous driving, robotics, and augmented reality. Point clouds acquired with the LiDAR sensor, however, inevitably suffer from different levels of noise and outliers caused by measurement errors [2]. A plethora of noise sources can affect point clouds, such as the acquisition device, limitations of sensors, and the lighting or reflective nature of the surface. The noise not only degrades the quality of point clouds, but also hinders downstream geometry processing applications.

Most of the point cloud denoising methods are based on the 3D information of the point cloud – a set of $N * 3$ datas.

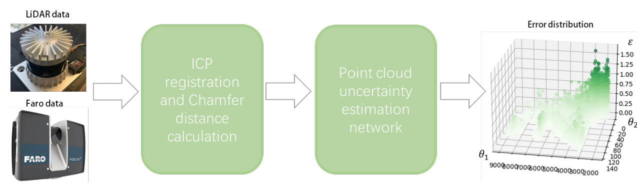

Figure 1. Process of point cloud uncertainty estimation

They do not take advantage of the prior knowledge of the sensor itself, such as the depth, the surface reflectivity, signal photons, ambiant photons and channels etc. And this knowledge can obtain the estimation of the uncertainty of each points after training by our MPL network, that is, the variance μ and the average α of different points under the uniform parameter region. This powerful prior knowledge can be used to assist many downstream point cloud processing tasks, such as point cloud denoising tasks, ICP registration tasks and map reconstruction.

Early point cloud denoising is done using optimization-based methods, which rely heavily on geometric priors and sometimes struggle to strike a balance between detail preservation and denoising effectiveness. With the emergence of neural network architectures designed for point clouds [28, 29, 35], point cloud denoising methods based on deep learning have emerged and achieved good results. Including Luo et al. [22] proposed to explicitly learn the underlying manifold of a noisy point cloud for denoising via an autoencoder-like network. The encoder, which builds on the Dynamic Graph CNN (DGCNN) [35], learns both local and non-local feature representations of each point and then samples points with low noise that tend to be closer to the underlying surfaces via an adaptive differentiable pooling.

We add the strong prior knowledge, that is, the uncertainty of point cloud predicted by the network based on the parameters of Lidar sensors, into the noise reduction network, and re-extract the features of point cloud to optimize the network scoring results. And the average error a in the uncertainty of point cloud is used to predict the displacement of noise points in the plane, so as to optimize the noise reduction results and reduce the number of network iterations.

To summarize, the contributions of this work include:

- Based on LiDAR sensors, we design and train a network to learn the error distribution in the parameter space, so as to obtain a point cloud uncertainty estimation model, which can be used to predict the mean and variance of the error for any point in the LiDAR point cloud, providing strong prior knowledge.

- We add the uncertainty of the point cloud to the Score-based denoising network, and re-extract the features of the point cloud to optimize the network scoring results, and use it to predict the displacement of the noise points in the plane, and optimize the results and efficiency.

- We use faro and Ouster OS-0 128-line Lidar to collect a large number of point cloud data, and make a data set to train the corresponding model of LiDAR point cloud uncertainty estimation model.

2. Related Work

The techniques of point cloud denoising have been growing with the development of machine learning during the past decades. From the classical filter-based methods to the optimization-based methods and now to the advanced deep learning-based approaches, various ways are proposed to denoise point clouds and have reached promising performance so far. However real-world noise is much more complex and varies with sensors and scenarios, the focus has recently moved from the geometric structure to the noisy distribution. In the following, we briefly review the relevant approaches to point cloud denoising.

2.1. Geometric Structure-based denoising

Early point cloud denoising methods inherited from image processing, which smoothing the local region by a filter. Various filters were designed to operate on point positions or point normals. For instance, bilateral filters consider the spatial and normal closeness but imprecise near the sharp features [3, 6, 11, 15, 39, 42], guided filters try to preserve sharp features by guidance information but highly depend on the guidance [13, 14, 34, 36, 43, 44] and graph filters take advantage of underlying geometric structure from the graph but suffer from the instability [8, 9, 17, 18, 40]. Later on, as traditional machine learning developed, optimization-based denoising methods emerged. This kind of method is formulated to find a denoised point cloud that best fits the noisy input with a set of constraints designed from the geometric structure and noise assumption. One of the most representative methods is sparsity-based [7, 19, 24]. Inspired by dictionary learning, the sparse dictionary is built on the assumption that the surface is smooth almost everywhere except at some sparse features such as sharp edges and corners. Therefore points can be updated by solving a global minimization problem [24]. In practice, these methods may lead to over-smoothing or over-sharping arising from poor normal estimation. Afterwards, the technique of low-rank matrix approximation is also used in point cloud denoising [4, 20, 45]. To sum up, geometric structure-based methods developed in the manner of traditional optimizations. Its nature highly depends on the geometric interpretation which limits its performance. In other words, these approaches may suffer from artefacts and performance degradation when the input point cloud lacks in similar patches. Besides, the computational complexity of these methods is usually high.

2.2. Displacement-based denoising

The success of deep learning is also extended to the field of point cloud denoising. Instead of recovering the geometric structure of point clouds, deep learning-based methods provide a new view that is learning a mapping from the noisy inputs to the noise-free counterparts [5, 16, 21, 25–27, 30, 33, 37, 41], which is typically referred as displacement-based denoising. One of the typical approaches is PointNet-based, which is a kind of supervised method based on the spirit that extracts the local feature of a single point. Taking PointNet as the backbone network, the denoising network of PointCleanNet is designed to estimate the displacement of each point [30]. Notice that the PointNet-based methods neglect the information about the neighbourhood, convolution-based methods are proposed to learn hierarchical features. For instance, PointProNet projects the noisy point cloud into 2D heightmap [33] and GPDNet represents it as graphs and makes use of graph convolution operations [26, 27] the latter is more robust to the high level of noise. Supervised methods have achieved impressive results on point cloud denoising, but they heavily rely on the training data which is commonly generated by adding man-made noise to synthetic clean data. They may suffer from poor performance when it comes to complex noisy scenarios that have different noise setting from training data. Soon afterwards, an unsupervised denoising method TotalDenoising is proposed [16]. It is based on the assumption that points with denser surroundings are closer to the underlying surface. However, this method cannot preserve sharp features due to a lack of sharp feature information during the training stage.

2.3. Noise Distribution-based denoising

As mentioned above, in supervised denoising the assumption of specific noise is too ideal to be true for real-world noisy point clouds, where the noise is much more complex and varies with different scenes and sensors. But the unsupervised displacement-based method lost sharp details. Instead, [21] proposed to explicitly learn the underlying manifold of a noisy point cloud for denoising via an autoencoder-like network, but its downsampling may also discard some informative details, leading to over-smoothing. Notably, score-based Denoising is proposed to learn the distribution of the noisy point cloud by formulating the convolution between the sample distribution and some noise model [1, 20]. Compared with displacement-based methods which only consider the position of each input point, score-based methods model 3D continuous distribution supported by a 2D manifold. Compared with the downsample-upsample method [21], they preserve more informative details at low noise levels.

Although there have been a few unsupervised methods developed for real-world noisy point cloud denoising. These algorithms aim to directly learn latent representations for denoising from the noisy point cloud in an unsupervised manner. However, their overall performance on real-world noise is still limited. Therefore, it is desirable to further investigate the problem of real-world noisy point cloud denoising.

3. Methods

We first provide an overview of the proposed method. Then, we elaborate on the LiDAR point cloud uncertainty estimation model, propose the training objective for the network, and develop a score estimation network with error distribution.

3.1. LiDAR point cloud uncertainty estimation model

The normal way to denoise the point cloud is directly input the raw point cloud data. This method is easy to implement but for LiDAR point cloud, it means we abandoned the abundant parameter features like range(depth), reflectivity, signal photons and so on. If we can leverage this information to enrich the preprocessing procedure, it will make the denoising downstream work more solid. So we provide the uncertainty estimation model to estimate the displacement from the noise point to the ground truth point in the parameter dimension, which will give the prior knowledge for the score estimation network.

Every point cloud has its own parameters. So we first establish a coordinate system centred on parameters. For example, we can build a coordinate with the x-axis representing range(depth) and the y-axis representing reflectivity. Then the z-axis is the displacement from the noise point to the ground truth point which also can be called an error. The Fig. 2 shows the 3d representation.

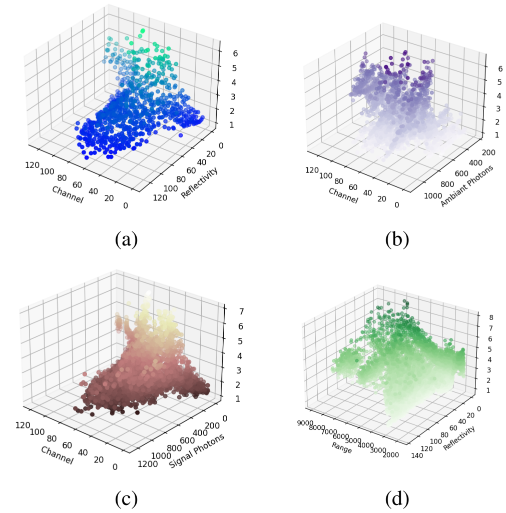

Figure 2. Error distribution in parameter coordinates

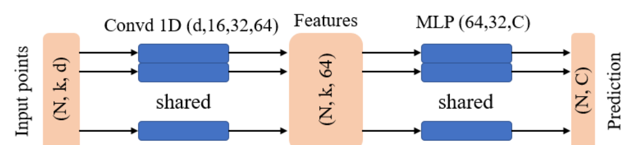

Figure 3. Architecture of uncertainty estimation model

For better visualizing, we only take two parameters in Fig. 2. But in the real situations, the dimension of parameters will be much larger. So traditional surface fitting method will be much more complicated. We choose to use a neural network to predict error distribution. The architecture of the uncertainty estimation model is shown in Fig. 3.

3.2. Score Estimation Network with Error Distribution

Given a noisy point cloud $X = \{x_i\}_{i=1}^{N}$ as input, we model the underlying clean point cloud as a set of samples from a 3D distribution p supported by a 2D manifold and assume that the noise follows a distribution n. Then the distribution of the noisy point cloud can be modelled as a convolution between p and n, denoted as $p{*}n$. Use the score estimation network to predict $\nabla_x \log[(p * N)(x_i)]$ for each point in X - the gradient of the log-probability function, only from X. We estimate the score for each point x_i on a local region, which means the network aims to estimate the score function in the neighbourhood space around x_i, denoted as $S_i(r)$.

$$S_i(x) = Score(x - x_i, h_i) \tag{1}$$

$S_i(r)$ is estimated by a neural network consisting of a feature extraction unit and a score estimation unit. The feature extraction unit produces features that encode the local and non-local geometry of each point. The extracted features are subsequently fed as parameters to the score estimation unit to construct the score functions. The figure 4 is showing the details of features network.

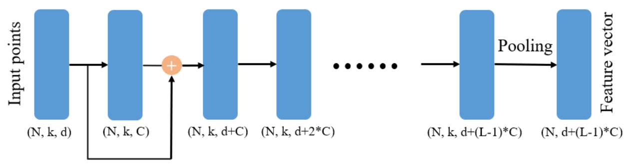

Figure 4. The network of feature extraction

The score estimation unit is parameterized by the feature h_i of point x_i. It takes some 3D coordinate $x \in R^3$ near x_i as input and outputs the score $S_i(x)$. $Score(\cdot)$ is a multi-layer perceptron (MLP), we also add our parametric network to the it. Its structure is shown in the figure5.

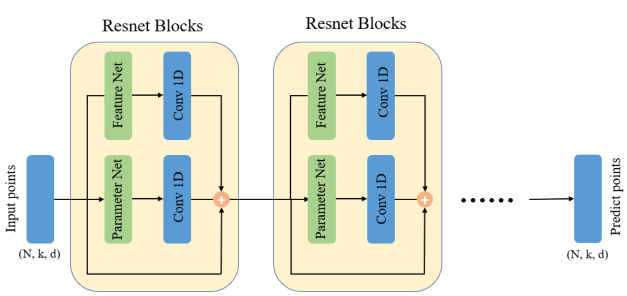

Figure 5. The network of feature extraction

Next we introduce the loss function of training, we denote the ground truth clean point cloud as $Y = \{y_i\}_{i=1}^N$. Using the ground truth Y, we define the score for some point $x \in R^3$ as follows, where $NN(x, Y)$ returns the point nearest to x in Y.

$$s(x) = NN(x, Y) - x, x \in R^3 \tag{2}$$

The loss function aligns the network prediction score with the ground truth score defined above, where $N(x_i)$ is the distribution concentrated around x_i in R^3 space:

$$L^{(i)} = E_{x \sim N(x_i)}[||s(x) - S_i(x)||_2^2] \tag{3}$$

The final loss function is simply an aggregation of the objective for each local score function:

$$L = \frac{1}{N} \sum_{i=1}^{N} L^{(i)}. \tag{4}$$

3.3. The Error Distribution Based Denoising Algorithm

For the final denoising step, we need the noisy points close to the plane with a certain stride to determine the next position of the noisy point cloud. In order to enhance the robustness and reduce the estimation bias, we use the ensemble score function, which fully considers the influence of neighbor:

$$\varepsilon_i(x) = \frac{1}{K} \sum_{x_j \in kNN(x_i)} S_j(x), x \in R^3 \tag{5}$$

where $kNN(x_i)$ is x_i's k-nearest neighborhood.

Finally, denoising a point cloud amounts to updating each point's position via gradient ascent:

$$\begin{aligned} x_i^{(t)} &= x_i^{(t-1)} + \alpha_t \varepsilon_i(x_i^{t-1}), t = 1, \cdots, T, \\ x_i^{(0)} &= x_i, x_i \in X \end{aligned} \tag{6}$$

where α_t is the step size at the t-th step. We regularize the average error $\alpha_i, i = 1, \cdots, N$ previously obtained from the point cloud uncertainty estimation network to the interval 0 and 1 to be used as the initial value of α_t for each noisy points, and then decrease it to 0 to ensure convergence. In this way, it can converge as soon as possible for the points with large noise, while avoiding excessive denoising for the points with small noise.

4. Analysis

In this section, we elaborate on the distribution model for noisy point clouds.

4.1. Point-point Distance to Point-plane Distance

In previous paper, we only consider the nearest ground truth point to the noisy point cloud. But with the Manhattan World Assumption, most planes are one of three mutually orthogonal planes. And interior scene usually have a floor plane and a ceiling plane, which parallel to each other. So it is much more suitable to switch the distance calculation from point-point to point-plane. Once we have this, we can also use the information about the local planes as an additional input to our noise prediction model, because just as it depends on intensity or on the depth of the point, the expected noise may also depend on the inclination of the plane, which can not be achieved by point-point distance method. Fig. 7 illustrate this baseline.

4.2. Iteratively Denoising

We want to follow up on this idea of reweighting each arrow by the expected uncertainty and then realign our scans to the ground through scans and redo our model. So we really want to do that iteratively because like this, we will

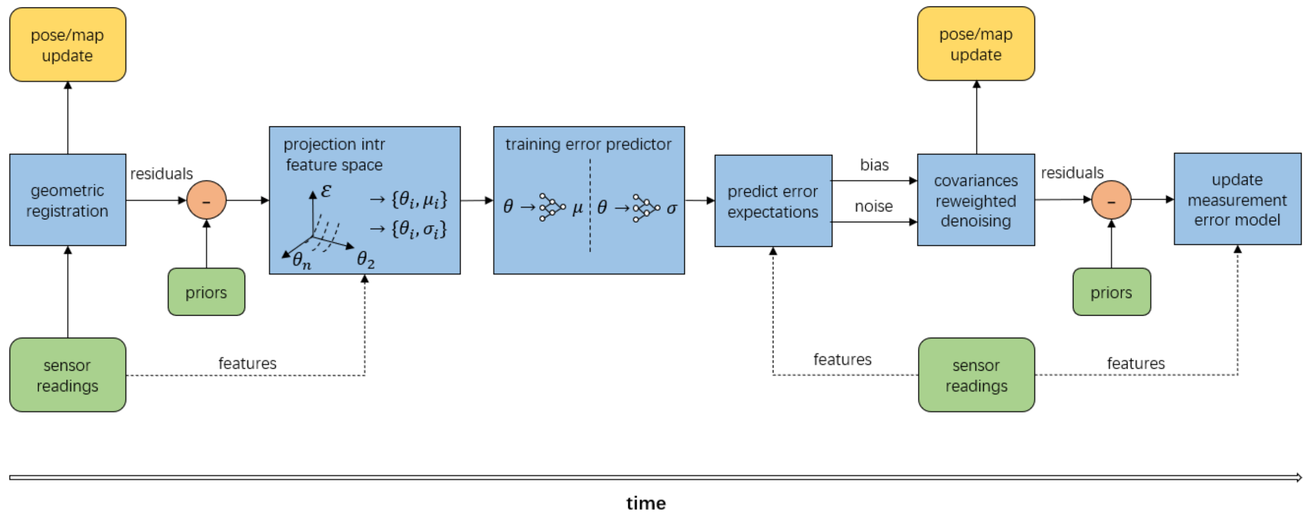

Figure 6. Iteratively denoising pipeline.

find the argument that can use the gotten noise model in the beginning, but then try to fit another noise model in the end. So this is not consistent. By doing the iterations over these two parts, we will get rid of this issue. And it's important to do this experiment so that we can figure out if the whole manifold is actually converging. The whole pipeline is showed in Fig. 6.

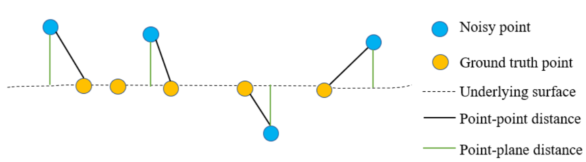

Figure 7. Point-point distance and point-plane distance comparison. Dotted black line denotes the underlying scanned surface, orange points denote original points, and blue points denote the noisy points. Black line represents point-point distance and green line represents point-plane distance. The point-plane distance is more accurate to point-point distance with the Manhattan World Assumption.

4.3. Weak Supervised Method

What have been done now is totally based on supervised learning method. But due to the large amounts of data and disorder property of point cloud, it is expensive to do the human labeling and get the high accurate ground truth point cloud. So it is necessary to move away from using FARO scan as ground truth and change towards the weak supervision method. Everything has been done now is very valuable, because it is already a reference for us to validate whether or not the weekly supervised method actually works and can give the same result afterwords.

5. Experiments

We evaluate a classroom dataset for point cloud denoising to qualitatively and quantitatively compare our method against the state-of-the-art method [23]. We only employ real-world dataset since the proposed method takes reflectivity besides coordinates information as input.

5.1. Experiment Setting

Datasets We use our own datasets captured in a typical classroom with regular clutters. Data is captured by a tripod placed statically on the ground and carrying an Ouster OS-0 128-line Lidar. For ground truth production, We follow the approach proposed by Ramezani et al. [31] and use the Iterative Closest Point (ICP) method to match the motion-compensated LiDAR scans with a prescanned dense point cloud of the environment captured by the survey-grade FARO laser scanner.

Train Setting We use Adam optimizer and set the batch size to 32, learning rate to 0.001, weight decay to 1e-4. We take multilayer perception (MLP [12]) as the main block thus we reduce the learning rate adaptation method. For efficiency, we collect meshes from PU-Net [38] as training data.

Metrics We employ Chamfer distance (CD) [10], which is widely used by point cloud reconstruction and aims to show the difference between two point clouds, to perform a quantitative evaluation of our model. The point-to-mesh distance

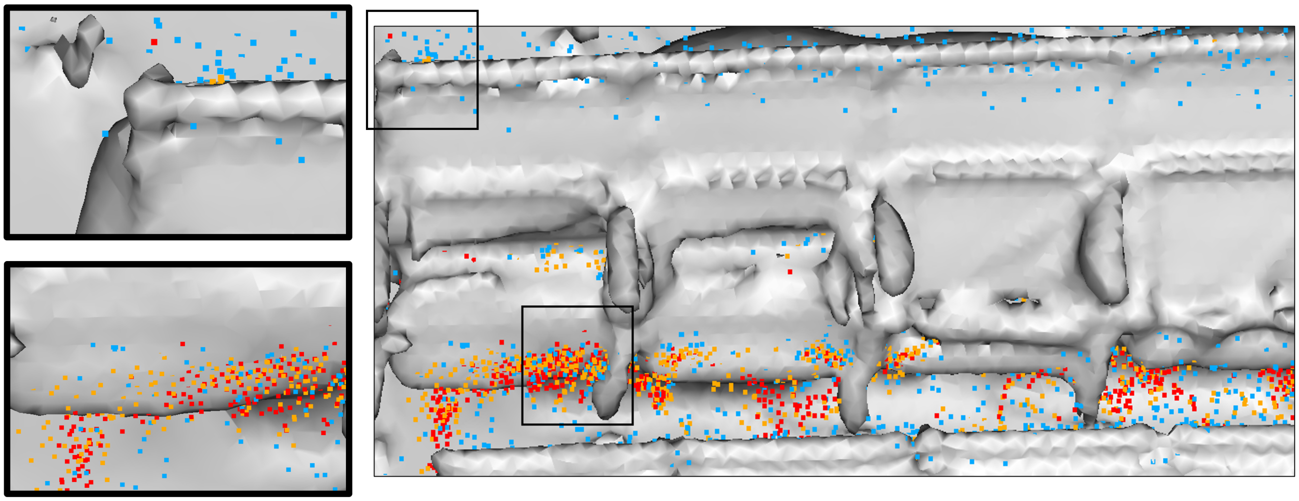

Figure 8. The comparison results. The Grey object is the reconstructed mesh of a row of chairs in a typical classroom. The blue, orange, and red points respectively are noisy point clouds, the ScoreNet results, and our results. The Lidar scan does not contain the front side of the chairs due to the scan angle.

	Noisy	ScoreNet	ours
CD [cm]	20.7224	22.0314	17.4576
P2M [cm]	4.8627	5.3834	4.8812

Table 1. The quantitative results on the captured dataset. Here P2M is computed with mesh reconstructed from a high-quality point cloud captured by a FARO scanner. ScoreNet is the state-of-the-art method [23].

(P2M) [32] is also set for reference with the reconstructed mesh from a dense high-quality point cloud recorded by a FARO scanner. Since the size of point clouds varies, we normalize the denoised results into the unit sphere before computing the metrics.

5.2. Quantitative Results

We evaluate our model on our own dataset. As presented in Tab. 1, our model significantly outperforms the state-of-the-art method [23], which indicates that our method has the potential with processing implicit objects in the space, see chapter Sec. 5.4 for the details.

5.3. Qualitative Results

As shown in Fig. 9, our results are notably cleaner and workable. For a point cloud captured from an indoor scan, the number and distribution of points on the outer side of the wall as shown in Fig. 9 can be set as the quality of the point cloud. It can be seen in the upper part of the illustration that the SOTA method does not remove the outliers of

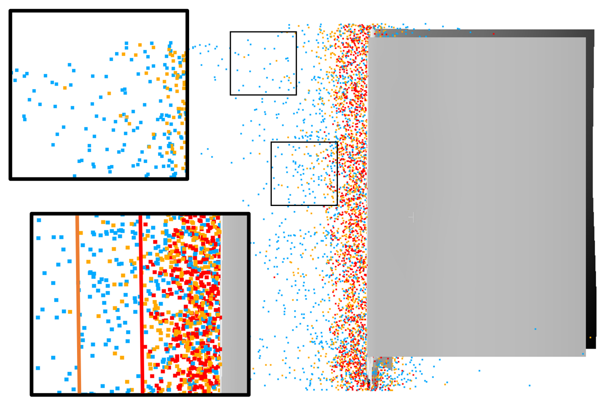

Figure 9. The comparison results. The Grey object is the reconstructed mesh of one wall. The blue, orange, and red points respectively are noisy point clouds, the ScoreNet results, and our results.

the original point cloud, while our results do not show similar outliers. For the lower part of the illustration, the noisy point cloud, the results of the SOTA method, and our results constitute a clear boundary, which strongly illustrates the superiority of our method.

5.4. Error Mode Analysis

In this section, we will analyze the error model of the model and provide an explanation for the reason why the SOTA results in Tab. 1 are instead worse than the original point cloud.

In Fig. 8, the original point cloud contains a large number of points on the chair back, and all these points are cut off after denoising. However, the original points near the ground are retained after optimization. We believe that since our training data from PU-Net [38] does not contain cases of objects inside the point cloud, the model inclines to the point cloud should have convexity inside, which is not consistent with our test data. In this case, the model will optimize some of the hanging points, such as the points scanned on the chair back. However, the points close to the ground are retained, which is also consistent with the experimental results.

Since the SOTA method only employs coordinates information, it cannot deal with this kind of situation well, so the metrics rise after optimization, and our method employs more information, so the final result is better.

6. Conclusion

Based on LiDAR sensors, we design and train a network to learn the error distribution in the parameter space, so as to obtain a point cloud uncertainty estimation model, which can be used to predict the mean and variance of the error for any point in the LiDAR point cloud, providing strong prior knowledge. We add the uncertainty of the point cloud to the Score-based denoising network, and re-extract the features of the point cloud to optimize the network scoring results, and use it to predict the displacement of the noise points in the plane, and optimize the results and efficiency. We use faro and Ouster OS-0 128-line Lidar to collect a large number of point cloud data, and make a data set to train the corresponding model of LiDAR point cloud uncertainty estimation model. Experimental results validate the superiority of our model and further show the potential to be applied to other tasks such as point cloud registration.

References

[1] Haolan Chen, Shitong Luo, Wei Hu, et al. Deep point set resampling via gradient fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[2] H. Chen and J. Shen. Denoising of point cloud data for computer-aided design, engineering, and manufacturing. *Engineering with Computers*, 2017. 1

[3] Hao Chen and Jie Shen. Denoising of point cloud data for computer-aided design, engineering, and manufacturing. *Engineering with Computers*, 34(3):523–541, 2018. 2

[4] Honghua Chen, Mingqiang Wei, Yangxing Sun, Xingyu Xie, and Jun Wang. Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint. *IEEE transactions on visualization and computer graphics*, 26(11):3255–3270, 2019. 2

[5] Siheng Chen, Chaojing Duan, Yaoqing Yang, Duanshun Li, Chen Feng, and Dong Tian. Deep unsupervised learning of 3d point clouds via graph topology inference and filtering. *IEEE transactions on image processing*, 29:3183–3198, 2019. 2

[6] Julie Digne and Carlo De Franchis. The bilateral filter for point clouds. *Image Processing On Line*, 7:278–287, 2017. 2

[7] Julie Digne, Sébastien Valette, and Raphaëlle Chaine. Sparse geometric representation through local shape probing. *IEEE transactions on visualization and computer graphics*, 24(7):2238–2250, 2017. 2

[8] Chinthaka Dinesh, Gene Cheung, and Ivan V Bajić. Point cloud denoising via feature graph laplacian regularization. *IEEE Transactions on Image Processing*, 29:4143–4158, 2020. 2

[9] Chaojing Duan, Siheng Chen, and Jelena Kovacevic. Weighted multi-projection: 3d point cloud denoising with tangent planes. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 725–729. IEEE, 2018. 2

[10] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *computer vision and pattern recognition*, 2016. 5

[11] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. In *ACM SIGGRAPH 2003 Papers*, pages 950–953. 2003. 2

[12] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998. 5

[13] Xian-Feng Han, Jesse S Jin, Ming-Jie Wang, and Wei Jiang. Guided 3d point cloud filtering. *Multimedia Tools and Applications*, 77(13):17397–17411, 2018. 2

[14] Xian-Feng Han, Jesse S Jin, Ming-Jie Wang, and Wei Jiang. Iterative guidance normal filter for point cloud. *Multimedia Tools and Applications*, 77(13):16887–16902, 2018. 2

[15] Xian-Feng Han, Jesse S Jin, Ming-Jie Wang, Wei Jiang, Lei Gao, and Liping Xiao. A review of algorithms for filtering the 3d point cloud. *Signal Processing: Image Communication*, 57:103–112, 2017. 2

[16] Pedro Hermosilla, Tobias Ritschel, and Timo Ropinski. Total denoising: Unsupervised learning of 3d point cloud cleaning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 52–60, 2019. 2

[17] Wei Hu, Xiang Gao, Gene Cheung, and Zongming Guo. Feature graph learning for 3d point cloud denoising. *IEEE Transactions on Signal Processing*, 68:2841–2856, 2020. 2

[18] Muhammad Abeer Irfan and Enrico Magli. Exploiting color for graph-based 3d point cloud denoising. *Journal of Visual Communication and Image Representation*, 75:103027, 2021. 2

[19] Esmeide Leal, German Sanchez-Torres, and John W Branch. Sparse regularization-based approach for point cloud denoising and sharp features enhancement. *Sensors*, 20(11):3206, 2020. 2

[20] Xuequan Lu, Scott Schaefer, Jun Luo, Lizhuang Ma, and Ying He. Low rank matrix approximation for 3d geometry filtering. *IEEE Transactions on Visualization and Computer Graphics*, 2020. 2, 3

[21] Shitong Luo and Wei Hu. Differentiable manifold reconstruction for point cloud denoising. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1330–1338, 2020. 2, 3

[22] S. Luo and W. Hu. Score-based point cloud denoising, 2021. 1

[23] Shitong Luo and Wei Hu. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4583–4592, 2021. 5, 6

[24] Enrico Mattei and Alexey Castrodad. Point cloud denoising via moving rpca. In *Computer Graphics Forum*, volume 36, pages 123–137. Wiley Online Library, 2017. 2

[25] Hailun Nan, Dejian Lu, et al. A potential marker dyf371 for differentiating han population from non-han population in chinese. *Journal of Forensic Science and Medicine*, 6(1):1, 2020. 2

[26] Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli. Learning graph-convolutional representations for point cloud denoising. In *European conference on computer vision*, pages 103–118. Springer, 2020. 2

[27] Francesca Pistilli, Giulia Fracastoro, Diego Valsesia, and Enrico Magli. Learning robust graph-convolutional representations for point cloud denoising. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):402–414, 2020. 2

[28] C. R. Qi, Y. Li, S. Hao, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017. 1

[29] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[30] Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J Mitra, and Maks Ovsjanikov. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In *Computer Graphics Forum*, volume 39, pages 185–203. Wiley Online Library, 2020. 2

[31] Milad Ramezani, Yiduo Wang, Marco Camurri, David Wisth, Matias Mattamala, and Maurice Fallon. The newer college dataset: Handheld lidar, inertial and vision with ground truth. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4353–4360. IEEE, 2020. 5

[32] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. 2023. 6

[33] Riccardo Roveri, A Cengiz Öztireli, Ioana Pandele, and Markus Gross. Pointpronets: Consolidation of point clouds with convolutional neural networks. In *Computer Graphics Forum*, volume 37, pages 87–99. Wiley Online Library, 2018. 2

[34] Yangxing Sun, Honghua Chen, Jing Qin, Hongwei Li, Mingqiang Wei, and Hua Zong. Reliable rolling-guided point normal filtering for surface texture removal. In *Computer graphics forum*, volume 38, pages 721–732. Wiley Online Library, 2019. 2

[35] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds, 2018. 1

[36] Sunil Kumar Yadav, Ulrich Reitebuch, Martin Skrodzki, Eric Zimmermann, and Konrad Polthier. Constraint-based point set denoising using normal voting tensor and restricted quadratic error metrics. *Computers & Graphics*, 74:234–243, 2018. 2

[37] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Ec-net: an edge-aware point set consolidation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 386–402, 2018. 2

[38] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. *computer vision and pattern recognition*, 2018. 5, 7

[39] Faisal Zaman, Ya Ping Wong, and Boon Yian Ng. Density-based denoising of point cloud. In *9th International Conference on Robotic, Vision, Signal Processing and Power Applications*, pages 287–295. Springer, 2017. 2

[40] Jin Zeng, Gene Cheung, Michael Ng, Jiahao Pang, and Cheng Yang. 3d point cloud denoising using graph laplacian regularization of a low dimensional manifold model. *IEEE Transactions on Image Processing*, 29:3474–3489, 2019. 2

[41] Dongbo Zhang, Xuequan Lu, Hong Qin, and Ying He. Pointfilter: Point cloud filtering via encoder-decoder modeling. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2015–2027, 2020. 2

[42] Feng Zhang, Chao Zhang, Huamin Yang, and Lin Zhao. Point cloud denoising with principal component analysis and a novel bilateral filter point cloud denoising with principal component analysis and a novel bilateral filter. 2

[43] Yinglong Zheng, Guiqing Li, Shihao Wu, Yuxin Liu, and Yuefang Gao. Guided point cloud denoising via sharp feature skeletons. *The Visual Computer*, 33(6):857–867, 2017. 2

[44] Yinglong Zheng, Guiqing Li, Xuemiao Xu, Shihao Wu, and Yongwei Nie. Rolling normal filtering for point clouds. *Computer Aided Geometric Design*, 62:16–28, 2018. 2

[45] Yiyao Zhou, Rui Chen, Yiqiang Zhao, Xiding Ai, and Guoqing Zhou. Point cloud denoising using non-local collaborative projections. *Pattern Recognition*, 120:108128, 2021. 2