3D Point Clouds Data Super Resolution-Aided LiDAR Odometry for Vehicular Positioning in Urban Canyons

Jiang Yue, Weisong Wen, Jin Han, and Li-Ta Hsu

***Abstract*—LiDAR odometry algorithms are extensively studied for vehicular positioning. However, achieving high-precision positioning using low-cost 16-channel LiDAR in urban canyons remains a challenging problem due to the limited point cloud density from low-cost LiDAR and excessive dynamic surrounding objects. To fill this gap, this paper proposes enriching sparse 3D point clouds to denser clouds via a novel deep learning-based superresolution (SR) algorithm before its utilization in 3D LiDAR odometry. We validate the effectiveness of the proposed method using the KITTI dataset and a challenging dataset collected in an urban canyon (with complex environmental structures and dynamic objects) of Hong Kong. We conclude that significantly denser point clouds are achieved with considerable accuracy. In addition, significantly improved performance of 3D LiDAR odometry is obtained in the evaluated dataset collected in an urban canyon of Hong Kong.**

***Index Terms*—LiDAR, sparse point clouds, superresolution, convolutional neural network, NDT, LiDAR odometry, vehicular positioning.**

I. Introduction

ACCURATE positioning is a fundamental part of the realization of safety-critical autonomous driving vehicles (ADVs) [1]. Light detection and ranging (LiDAR) is a popular sensor for providing positioning at a high frequency by LiDAR odometry [2] or map matching [3], [4]. Satisfactory accuracy can be achieved using high-grade 3D LiDAR with 64 channels (cost approximately $75000), which provides dense point clouds of surroundings. Unfortunately, the lasting high price is one of the major barriers that prevent its commercialization for ADV. The cost-effective 16-channel LiDAR sensor (cost approximately $4000) can provide positioning accuracy similar to that of the 64-channel sensor in constrained areas with sufficient environmental features [5]. Unfortunately, the performance is significantly degraded in challenging areas [6], as only limited and sparse point clouds are supplied by the low-cost LiDAR. More importantly, excessive dynamic objects can significantly distort point cloud registration [7], leading to numerous local minimums.

The major principle of 3D LiDAR odometry is to accumulate the estimation of the relative motion between consecutive frames of point clouds. Therefore, the performance of LiDAR odometry relies heavily on the accuracy of point cloud registration. In recent decades, several point cloud registration methods have been proposed, such as the generalized ICP [8], normal distribution transform (NDT) [9], and LiDAR odometry and mapping (LOAM) algorithms [10]. According to the extensive comparison in [11], the NDT is more robust in the evaluated scenarios due to the cell-based detailed modeling of point clouds using the normal distribution. The LOAM algorithm proposes extracting the edge and planar features for data association. Outperforming performance can be obtained in scenarios with abundant environmental features. However, the performance cannot be guaranteed in sparse areas with limited environmental features, especially when low-cost LiDAR is employed. Numerous studies [12]–[14] have been conducted to improve the performance of LiDAR odometry using low-cost 3D LiDAR in urban canyons. The straightforward method to increase both the accuracy and robustness of LiDAR odometry is to integrate additional sensors. The work in [15] proposes to make use of the inertial measurement unit (IMU) to provide pose prediction and motion distortion compensation for point cloud registration. However, the bias corrections of IMU still rely on the performance of point cloud registration. In other words, the core of the system relies heavily on the accuracy of point cloud registration. The work in [14] goes one step further, where the 16-channel LiDAR and IMU are tightly coupled via factor graph optimization. Significantly improved performance is obtained compared with the work in [15]. Unfortunately, only very limited point clouds are supplied using the low-cost 16-channel LiDAR. Moreover, the performance of tight integration relies heavily on the accuracy of the IMU, which is determined by its price. In short, the limited density of 3D point clouds is one of the major challenges for improving the performance of LiDAR odometry using low-cost 3D LiDAR in urban canyons.

Manuscript received May 9, 2020; revised August 27, 2020 and February 1, 2021; accepted March 14, 2021. Date of publication March 29, 2021; date of current version June 9, 2021. This work was supported in part by the Hong Kong PolyU internal Grant on the project ZVKZ, "Navigation for Autonomous Driving Vehicle using Sensor Integration" and in part by the National Natural Science Foundation of China under Grant 61601225. The review of this article was coordinated by Prof. Z. Ma. *(Corresponding author: Weisong Wen.)*

Jiang Yue is with the Nanjing University of Science and Technology, Nanjing, Jiangsu 210014, China, and also with the Intelligent Positioning and Navigation Lab, The Hong Kong Polytechnic University 999077, Hong Kong (e-mail: jiang.yue@polyu.edu.hk).

Weisong Wen and Li-Ta Hsu are with the Intelligent Positioning and Navigation Lab, The Hong Kong Polytechnic University, Hong Kong (e-mail: weisong.wen@connect.polyu.hk; lt.hsu@polyu.edu.hk).

Jin Han is with the Nanjing University of Science and Technology, Nanjing, Jiangsu, China (e-mail: njusthanjng@163.com).

Digital Object Identifier 10.1109/TVT.2021.3069212

Instead of improving the performance of LiDAR odometry via integration with additional sensors, the work in [16] proposes a deep neural network (DNN)-based superresolution algorithm to enrich sparse point clouds before its utilization in LiDAR odometry. This is the first work that employs the SR of sparse point clouds to improve the performance of 3D LiDAR odometry. However, only simulated data is validated. How the enriched point clouds can work in real and dynamic urban canyon remains an open question. Moreover, the performance of the applied DNN for superresolution (SR) fails to explore the discontinuity of the surrounding objects in the scene. As a result, its performance can be significantly challenged in dynamic urban scenarios. To fully make use of the discontinuity and improve the performance of point cloud SR, many neural networks have been studied and mainly validated using the KITTI dataset [17]. Point cloud data completion is well known as superresolution in the field of computer vision [18]. Normally, there are two types of sets: LiDAR input only [19] and LiDAR and image [20]. This superresolution started with filter research. It draws more attention when deep learning-based methods are proposed [21], [22]. A sparse convolution network that explicitly considers the location of missing data is proposed to realize superresolution at sparse depths, achieving a mean absolute error (MAE) of approximately 0.54 meters [19]. The result is the baseline of depth completion on the KITTI [17] depth dataset, and the input is the depth only. The surface normal used as a new depth local representation is proposed to predict the neighborhood pixel depth, and it reduces the MAE of the result by 0.226 meters [23]. Additionally, similar to the guided image filter, a pixel is a weighted average of nearby pixels. The weights are inferred from the image by the neuronal network and applied to sparse depths for a highly dense depth map [24]. The results show that it reduced the MAE by approximately 0.218 meters, which currently ranks first in the KITTI leaderboard (by Dec 2019). However, according to our findings in [25], the contribution from the image for 3D point cloud SR is limited compared with point cloud input only. Moreover, the SR based on the fusion of image and point clouds relies heavily on accurate spatial and temporary calibration. Moreover, panorama cameras or multicamera combinations are needed when both the camera and 3D LiDAR are used in SR, leading to additional costs.

Conventionally, depth map enrichment has poor results on the edges of objects [20], [26] due to the discontinuity between objects. A similar phenomenon can also be found in [16]. Moreover, the discontinuity can be even more severe in urban canyons with numerous dynamic objects. In short, using a DNN to enrich the depth map is a promising solution to obtain a dense depth map. However, the discontinuity of the objects inside the point clouds is one of the major challenges for point cloud SR in urban canyons.

Inspired by the work in [16] and our finding in [25], we go one step further by proposing a novel DNN-based and LiDAR-only SR algorithm, which fully explores the discontinuity between objects inside 3D point clouds, to enrich sparse 3D point clouds before their utilization in LiDAR odometry. Moreover, we bring the SR of 3D point clouds to real applications and explore its potential in a challenging urban canyon of Hong Kong. We first analyzed the sparsity of the depth map and the performance of edge detection with depth input, showing the feasibility of LiDAR standalone-based point cloud SR. The depth maps were acquired from the raw sparse point clouds based on the calibration between the camera and LiDAR [17]. An efficient residual factorized net (ERFNet) [27] was then employed to segment the depth map, which explores the discontinuity of the depth map. Moreover, the sparsity invariant CNN (SCNN) [19] was employed to predict the dense depth map based on a sparse depth map, which was projected from the raw sparse point clouds. Then, the predicted dense depth was fused with the segmentation outputs from ERFNet using a novel multilayer convolutional neural network (MCNN) to refine the predicted depth map. The dense point clouds were recovered from the predicted depth map. Finally, the enriched point clouds were employed to perform LiDAR odometry based on the normal distribution transform (NDT). We believe that the proposed method can have a positive impact on both the academic and industrial fields of 3D LiDAR odometry using low-cost sensors.

The remainder of this paper is structured as follows. The feasibility analysis of a single LiDAR-based SR is presented in Section II. An overview of the proposed method is given in Section III. Section IV presents the proposed methodology before the experimental evaluation is presented in Section V. Finally, conclusions and future work are drawn in Section VI.

II. Feasibility Study of Single LiDAR-Based Super Resolution

In this section, we analyze the sparsity of the depth map and the performance of edge detection with depth input to show the feasibility of LiDAR standalone-based depth map SR. According to [25], the sparsity property is important for depth data SR. If the data are sparse, there are many redundant data in the depth map. As the 3D point clouds from LiDAR are sparse, it is theoretically feasible to realize data enrichment with LiDAR input only without image information. In contrast, if the data are not sparse, then extra input should be needed to maintain the reliability of the enrichment results. The following section makes use of depth data compression as an example to show the feasibility of depth map standalone-based SR.

In contrast to compressive sensing [28], we consider the compression of the local information within the depth map. Inspired by the image compression of JPEG (Joint Photographic Experts Group) [29], the discrete cosine transform (DCT) [30] algorithm is employed to compress the depth map under different compression rates. The DCT transformation [31] is widely employed in image, video, and speech coding because of its good decorrelation and energy compaction properties. Fig. 1 shows the flowchart of the implemented compression of depth data using DCT. The input is the depth map, which includes dense depth data. The depth map is first split into multiple blocks. Then, DCT is performed for each block. The quantization process is conducted before compression. Note that the depth map becomes sparse after compression. To validate the feasibility of SR of sparse point clouds, reverse DCT is performed to recover the dense depth map, and the process is named SR.

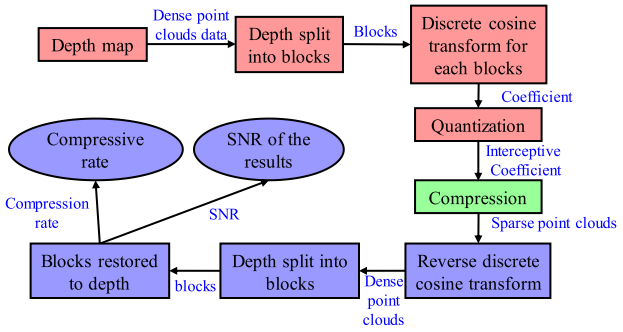


Fig. 1. The baseline sequential flowchart to analyze the compression of the depth using DCT.

More specifically, DCT in the two-dimensional image (8×8 blocks) can be written as follows:

$$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \times \cos\left[\frac{\pi}{N} u \left(x + \frac{1}{2}\right)\right] \cos\left[\frac{\pi}{N} v \left(y + \frac{1}{2}\right)\right] \quad (1)$$

where the function $f(x,y)$ is the intensity of the given image, x and y are pixel coordinates of the image, and $x, y = 01, 2 \ldots N-1$. The variable N is the scale of the block, $F(u,v)$ is the coefficient, and $u, v = 01, 2 \ldots N-1$ are indices. Since we have the image in the form of cosine functions, it can be compressed by reducing the accuracy of the coefficient. Subsequently, the small coefficients will exactly be zeros.

To evaluate the compressed results, the reverse DCT transformation is employed to restore the compressed data, which is written as follows:

$$f(x,y) = \frac{1}{2N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \times \cos\left[\frac{\pi}{N} u \left(x + \frac{1}{2}\right)\right] \cos\left[\frac{\pi}{N} v \left(y + \frac{1}{2}\right)\right] \quad (2)$$

The quality of the restored result will be evaluated by the peak signal-to-noise ratio (PSNR) and compressed rate (CR). The CR is calculated by the following equation

$$CR = \frac{\| \tilde{f}(x,y) \|_{l0}}{\| f(x,y) \|_{l0}} \quad (3)$$

where $\tilde{f}(x,y)$ is the restored result, $f(x,y)$ is the ground truth of the image, and $\| * \|_{l_0}$ is the l_0 norm operator.

The DCT is utilized to predict the superresolution of the depth map. We presented the well-known superresolution algorithm SRCNN [32] on two standard benchmark datasets, Set5 [33] and Set14 [34], and there are 14 images in total. Additionally, the SSIMs of those images compressed by the DCT are presented in Fig. 2. As can be seen in Fig. 2, the DCT results have a significant correlation with the superresolution results.

The depth image from the Middlebury [35] stereo dataset is employed to benchmark the sparsity of the depth image. One of

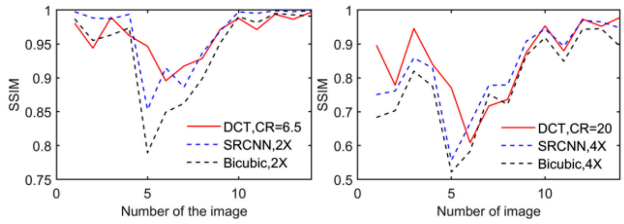


Fig. 2. The correlation of DCT compression and superresolution.

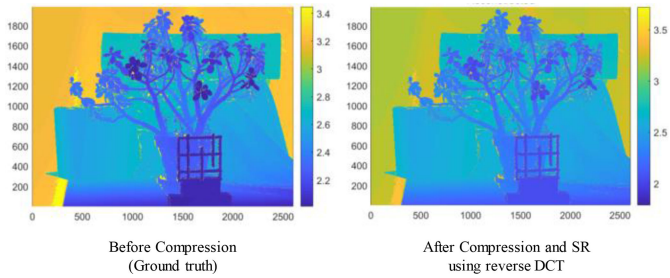


Fig. 3. The SR result using reverse DCT on the Middlebury dataset with $CR = 0.05$ and $MAE = 0.074$ meters.

TABLE I
THE COMPRESSED RATE AND RECONSTRUCTED QUALITY

Filename [35]	CR	1/CR	PSNR (dB)	MAE (m)	Range (m)
Jadeplant	0.1	10	68.1	2.4e-04	3.44
Jadeplant	0.05	20	32.1	0.049	3.44
Playtable	0.1	10	68.9	2.4e-04	4.19
Playtable	0.05	20	37.3	0.005	4.19
Playroom	0.1	10	67.1	3.0e-04	4.93
Playroom	0.05	20	29.7	0.009	4.93
Motorcycle	0.1	10	63.9	4.2e-04	5.88
Motorcycle	0.05	20	27.0	0.013	5.88
Classroom2	0.1	10	61.7	5.1e-04	12.13
Classroom2	0.05	20	19.3	0.030	12.13

*Compressed rate CR = 1 means there is no compress.

the depth maps is shown in Fig. 3. The left-side figure shows the original depth map, where the horizontal and vertical directions denote the pixel positions. The color denotes the depth of each pixel. After applying the DCT and reverse DCT algorithms (shown in Fig. 1) the result is shown on the right-hand side of Fig. 3 with a high compression rate (CR) of 0.05. Interestingly, the accuracy of SR of the depth map still reaches a high accuracy with an MAE of 0.074 meters.

To show the feasibility of the depth map SR, we evaluated the performance of the compression and reverse DCT on several different range depth images via different compression rates in the Middlebury dataset. The results are shown in Table I.

In Table I, the variable $1/CR$ is used for the reverse DCT, corresponding to the CR for DCT. The MAE denotes the accuracy of reverse DCT. The "range" represents the maximum range of the depth map collected using stereo cameras. We can see that the depth image could be well compressed without significant information loss. Therefore, we argue that sparse depth maps have a high potential for recovering dense depth maps using

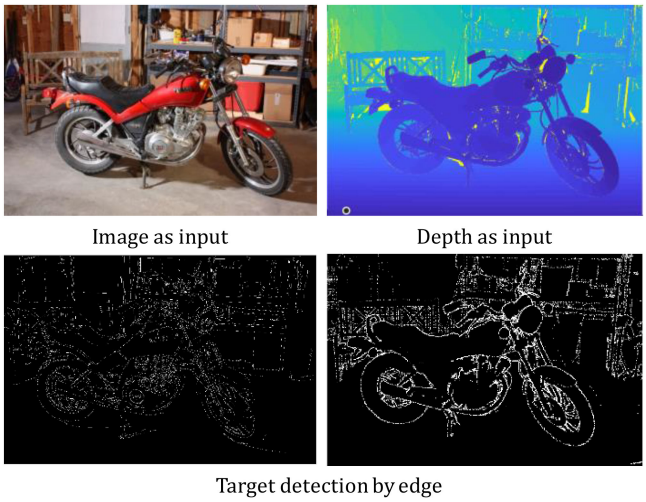


Fig. 4. Edge of image (left image) and edge of depth image (right image). The edges extracted from the image are very ambiguous, and it is difficult to provide a useful target before LiDAR enrichment.

SR algorithms (e.g., reverse DCT) without relying on additional information such as images. Therefore, we go one step further in this paper. Instead of using the conventional reverse DCT, we make use of the state-of-the-art SCNN [19] to recover the dense 3D point clouds from the sparse point clouds. Unfortunately, the SCNN cannot cope with the discontinuity of the scene, which limits the accuracy of the SR using the SCNN.

A general understanding of depth completion is that scenes with semantically similar appearances should have similar depth distributions [36]. The discontinuity of the scene usually occurs between the edges of different objects. Therefore, detecting the edge and producing a decent segmentation of objects can be a promising solution to improve the SCNN standalone-based SR. Segmentation of a scene with texture only (e.g., an image) is an ill-posed problem. As a result, the exact edges cannot be effectively detected. However, the depth map opens a new window for edge detection in a scene. The edges of objects are obtained after scene segmentation is effectively conducted; however, they rely on extra priors to estimate accurate results. Conversely, the depth map has a natural advantage for scene segmentation compared with image-based segmentation.

To show this advantage, we extract the edges from the same scene using both the colored image and depth map. The left side of Fig. 4 shows the edge extraction of a colored image using the Canny detector [37]. As can be seen, the extracted edges from the image are ambiguous due to the color of the objects. Edge extraction methods normally treat the surface as a continuous problem. Therefore, it is difficult to effectively extract the edges between different objects simply based on the colored image. Fortunately, the right side of Fig. 4 shows significantly clearer edge extraction results compared with the image-based method. Instead of using the conventional edge detection method, this paper makes use of ERFNet to segment the depth map to obtain the discontinuity of the scene. Then, both the predicted dense depth map from SCNN and the segmentation results from ERFNet are fused via a proposed MCNN. To the best of the author's knowledge, this is the first work that effectively explores the discontinuity of scenes for depth map SR by integrating both the SCNN and ERFNet.

III. Overview of the Proposed Method

The overview of the proposed method is shown in Fig. 5. The input of the system is the sparse point clouds from low-cost 3D LiDAR. The depth maps are acquired from the raw sparse point clouds. The SCNN is employed to predict the dense depth map from a sparse map. To address the discontinuity of the input depth map, ERFNet [27] is employed to perform the segmentation of the depth map to extract edges of different objects. Subsequently, a four convolutional layer branch is proposed to smooth and integrate both the segmentation and the predicated dense depth map result. The dense point clouds are recovered from the predicted depth map. Finally, the enriched 3D point clouds are utilized in 3D LiDAR odometry. The outputs of the system include dense point clouds and pose estimation from 3D LiDAR odometry.

In this paper, two open-source codes are employed to implement our algorithm. The PyTorch implementation of ENet is employed as the framework of our algorithm implementation [38]. Another open-source code employed in this paper is the ERFNet model, which is reused here [39]. Furthermore, according to [19], we implement the SCNN model in our network architecture.

To better understand the proposed architecture, the pseudocode of the training is presented in algorithm 1. Each module function of the implementation, including input and output, is presented.

The major contributions of this paper are listed as follows:

1) This paper first analyses the sparsity and the performance of edge detection of the depth data. Based on the analysis, a novel SR framework is proposed to enrich the sparse point clouds before its utilization in 3D LiDAR odometry. By exploring the discontinuity of objects, we relax the drawback of [16]. Moreover, compared with our previous work in [25], this paper eliminates the reliance on panorama image information, which can lead to high cost. With the help of the enriched point clouds, improved accuracy of 3D LiDAR odometry is obtained.
2) This paper extensively evaluates the effectiveness of the proposed SR method using the KITTI dataset. Moreover, it explores the potential of the proposed point cloud SR in 3D LiDAR odometry using data collected in a dynamic and complex urban canyon of Hong Kong.

IV. Methodology

A. Observation Matrix of SCNN

Regarding the employed SCNN, an observation mask is added to render the filter output, which is invariant to the actual number of observed inputs. The following figure shows the observation matrix, called sparse convolution.

Due to the sparsity of the depth input, the feature input involves many zero values. The traditional convolution layer

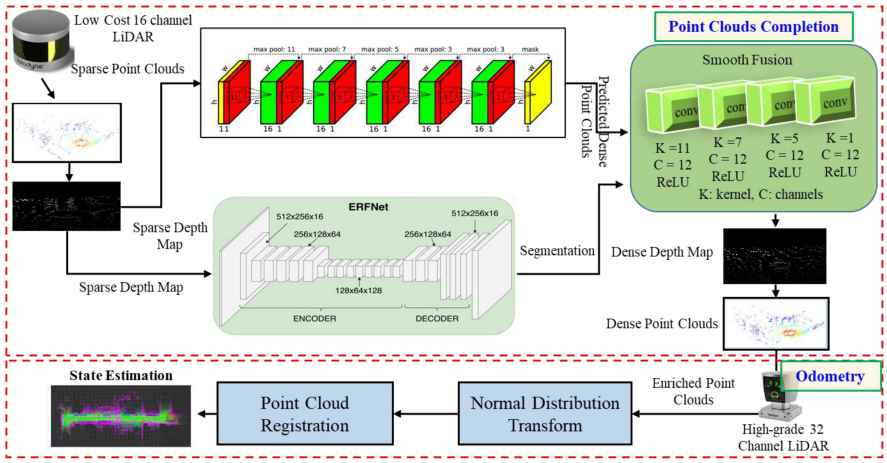


Fig. 5. The framework of the proposed method. The inputs are the 3D point clouds from a low-cost LiDAR. The output is the dense 3D point clouds and the pose estimation using LiDAR odometry.

Algorithm 1: Training and Testing of the Point Cloud Superresolution Model.

Input:

Training data: ground truth $\mathbf{Y}_{TR}$(high-resolution point cloud), input $\mathbf{X}_{TR}$(low-resolution point cloud);

test data: ground truth $\mathbf{Y}_{TE}$(high-resolution point cloud), input $\mathbf{X}_{TE}$(low-resolution point cloud)

Output:

The trained LiDAR superresolution model M; the predicated high-resolution results $\widetilde{\mathbf{Y}}_{TE}$ from input test data $\mathbf{X}_{TE}$

Initialization:

- Initial the training model of point cloud superresolution
- TR ← split the training data ($\mathbf{X}_{TR}$, $\mathbf{Y}_{TE}$) into equal parts of *K*
- Set the parameters of the deep network, *N* is the number of epoch

Steps:

1: **for** each epoch $i = 1, 2, \ldots, N$ do

2: each sample $\mathbf{L}(i)$ in the $\mathbf{X}_{TR}$ fitted in the sparsity invariant CNN (SCNN) and the ERFNet at the same time

3: Output of the SCNN, $\widetilde{\mathbf{H}}_{\mathbf{L}(i)}$, predicated high-resolution result of $\mathbf{L}_i$

3: Output of the ERFNet, $\widetilde{\mathbf{S}}_{\mathbf{L}(i)}$, predicated segmentation of $\mathbf{L}_i$; each pixel of the $\mathbf{L}_i$ will be labeled one class in $\widetilde{\mathbf{S}}_{\mathbf{L}(i)}$

4: Apply Hadamard product between $\widetilde{\mathbf{S}}_{\mathbf{L}(i)}$ and $\widetilde{\mathbf{H}}_{\mathbf{L}(i)}$, having $\widetilde{\mathbf{HS}}_{\mathbf{L}(i)}$. These three matrixes have the same scale

5: Feed the $\widetilde{\mathbf{HS}}_{\mathbf{L}(i)}$ into smooth fusion branch, having the final predicated high-resolution result $\widetilde{\mathbf{Y}}_{\mathbf{L}(i)}$

→ Calculate the loss of $\widetilde{\mathbf{Y}}_{\mathbf{L}(i)}$

→ Back Propagation

→ Update the weight

6: Fit the training model $\mathrm{M_i}$

7: Model $\mathrm{M_i}$ evaluation of the test dataset $\mathbf{X}_{TE}$ and $\mathbf{Y}_{TE}$

8: If $\mathrm{M_i}$ is the best model, $\mathrm{M} = \mathrm{M_i}$

9: **end for**

10: Output evaluation results $\widetilde{\mathbf{Y}}_{TE}$ of model M on the input $\mathbf{X}_{TE}$

cannot effectively cope with these unexpected zero values. In contrast to the traditional convolution kernel, the applied sparse

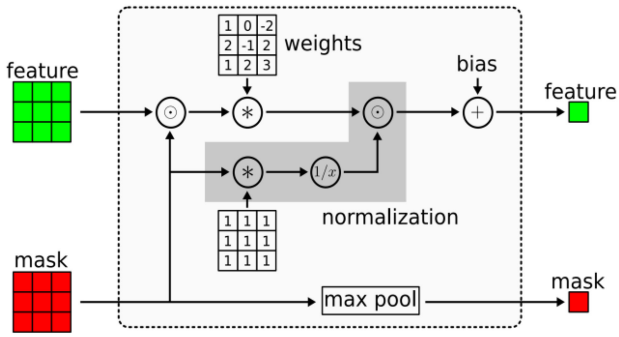


Fig. 6. The observation matrix [19] of the input sparse depth data, which keeps each convolutional layer weight as the input point only. The feature means the depth input. Mask consists of '0' and '1', and it would be 1 if the depth is valid; otherwise, it is '0'. $\odot$ denotes elementwise multiplication, and $*$ denotes convolution.

convolution operation is as follows:

$$f_{u,v}(\boldsymbol{x}, \boldsymbol{o}) = \frac{\sum_{i,j=-k}^{k} o_{u+i,v+j} x_{u+i,v+j} w_{i,j}}{\sum_{i,j=-k}^{k} o_{u+i,v+j} + \epsilon} + b \qquad (4)$$

where $o_{u+i,v+j}$ is the pixel in the observed matrix, and the kernel size is $2k+1$. $x_{u+i,v+j}$ is the pixel in the depth map, and $w_{i,j}$ is the weight to be trained.

As mentioned earlier, the segmentation of the scene is important for depth map superresolution, especially for improving the quality of the results around the edges of the targets. In this paper, we employed a 2D convolution network, ERFNet [27], to segment the targets in the scene. Taking advantage of ERFNet, the sparse depth map could lead to effective scene segmentation, which is utilized to improve the superresolution of the point cloud. To effectively fuse the information from both ERFNet and SCNN, a smooth branch is proposed (as shown in Fig. 5) to integrate the scene segmentation results. The network architecture is mainly constructed by four conventional kernels, which are inspired by the VGG16 [40]. Rectified linear units (ReLUs) are used as activation functions. This branch attempts to extract the local feature to upsample the sparse depth map.

TABLE II
THE NETWORK OF SMOOTH FUSION LAYER

Layer	Kernel	Filters in
Conv/Relu	11x11	12
Conv/Relu	7x7	12
Conv/Relu	5x5	12
Conv/Relu	1x1	12

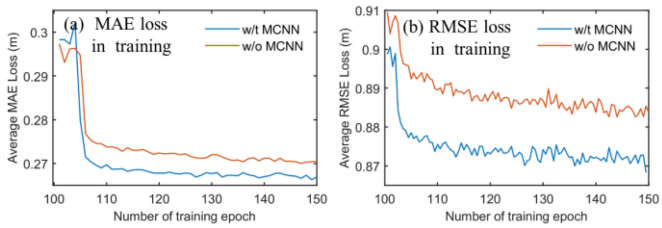


Fig. 7. The training loss of MAE and RMSE in the KITTI dataset.

B. Loss Function for the Proposed SR Network

The L2-norm (L2) is employed as the loss function to maintain the smoothness of the result. However, the L1-norm (L1) is robust at disparity discontinuities and has low sensitivity to outliers or noise [41]. Therefore, we make use of both the L1 and L2 loss functions, which is shown as follows:

$$L = \sum_{i \in \boldsymbol{P}} \left(\| \boldsymbol{D}_i^{\mathrm{gt}} - \boldsymbol{D}_i \|^2 + \lambda \| \boldsymbol{D}_i^{\mathrm{gt}} - \boldsymbol{D}_i \|^1 \right) \tag{5}$$

where λ is the regularization parameter, which is given as 0.004 in this paper.

As we know, the KITTI dataset has transferred the point cloud data into 16-bit Int to save as an image. The proposed network architecture processes the input of the point cloud as an image, including the SCNN and ERFNet branches. For evaluation, in the implementation of our algorithm, the data are transferred back to distance (unit, meter) by dividing 256. This would significantly improve the imbalance between the L2-norm and L1-norm in the loss function. In this paper, we try to compensate for the unbalanced loss by multiplying the efficiency $(1/256)$ by the regularization parameter.

C. The Smooth Fusion Using an MCNN

The image segmentation algorithms, such as ERFNet, assign pixels from one class with the same label number. The different label numbers would help the predicted depth of the SCNN distinguish different objects in the scene. It would improve the predicted depth results sensing ability on the discontinuity of the point cloud. However, it would cause an unexpected basis. The smooth fusion branch is proposed to correct the unexpected basis. The smooth fusion consists of four convolution layers, specifically, as shown in Table II.

ERFNet follows an encoder-decoder architecture similar to most segmentation learning methods. The encoder in ERFNet consists of residual blocks and downsampling blocks. The downsampling reduces the spatial resolution and pixel precision [27]. It works well for image segmentation, but it will cause unexpected error based on depth superresolution.

Compared with the MCNN included in our implementation, the algorithm without the MCNN will have less accuracy in both MAE and RMSE loss. The smooth fusion between depth from SCNN and segmentation predicted by ERFNet is necessary. This conclusion can be inferred from the test results, which can be found in Table IV.

D. LiDAR Odometry Based on Normal Distribution Transform (NDT)

The principle of LiDAR odometry [10] is to track the motion differences between two successive frames of 3D point clouds by matching the two frames (called a reference and an input point cloud in this paper). The matching process is also called point cloud registration. The objective of point cloud registration is to obtain the optimal transformation matrix to match or align the reference and input point clouds.

Typically, the objective function of the iterative closest point (ICP) [42], which is a well-known solution for point cloud registration, can be expressed as follows [42]:

$$C\left(\hat{\mathbf{R}},\ \hat{\mathbf{T}}\right) = \arg\min \sum_{i=1}^{N} ||(\mathbf{R}\mathbf{p}_i + \mathbf{T}) - \mathbf{q}_i||^2 \tag{6}$$

where N indicates the number of points in one scan $\mathbf{p}$, and a $\mathbf{R}$ and $\mathbf{T}$ indicate the rotation and translation matrix, respectively, to transform the input point cloud ($\mathbf{p}$) into the reference point cloud ($\mathbf{q}$). The objective function $\mathrm{C}(\hat{\mathbf{R}},\ \hat{\mathbf{T}})$ indicates the error of the transformation. One of the main drawbacks of this method is that ICP can easily enter the local minimum problem. The normal distribution transform [43] (NDT) is a state-of-art method to align two consecutive scans with the modeling of points based on a Gaussian distribution. The NDT innovatively divides the point cloud's space into cells. Each cell is continuously modeled by a Gaussian distribution. In this case, the discrete point clouds are transformed into successive continuous functions. In this paper, NDT is employed as the point cloud registration method for LiDAR odometry. Assume that the transformation between two consecutive frames of point clouds can be expressed as $\mathbf{T} = [\,t_x\ t_y\ t_z\ \phi_x\ \phi_y\ \phi_z\,]^T$. t_i indicates the translation in the x-, y-, and z-axes, respectively. ϕ_x represents the orientation angle of the roll, pitch, and yaw angles, respectively. The steps of estimating the relative pose between the reference and the input point clouds are as follows:

1) Fetch all the points $\boldsymbol{x}_{i\,=\,1\ldots n}$ contained in a 3D cell [44]. Calculate the geometry mean $\mathbf{q} = \frac{1}{n} \sum_i \boldsymbol{x}_i$. Calculate the covariance matrix

$$\boldsymbol{\Sigma} = \frac{1}{n} \sum_i (\boldsymbol{x}_i - \mathbf{q})(\boldsymbol{x}_i - \mathbf{q})^T \tag{7}$$

2) The matching score is modeled as:

$$f(\mathbf{p}) = \sum_i \exp\left(-\frac{(\boldsymbol{x}_i{}' - \boldsymbol{q}_i)^T \boldsymbol{\Sigma}_i{}^{-1} (\boldsymbol{x}_i{}' - \boldsymbol{q}_i)}{2} \right) \tag{8}$$

where $\boldsymbol{x}_i$ indicates the points in the current frame of scan $\mathbf{p}$. $\boldsymbol{x}_i{}'$ denotes the point in the previous scan mapped from the current frame using $\mathbf{T}$. $\boldsymbol{q}_i$ and $\boldsymbol{\Sigma}_i$ indicate the mean and

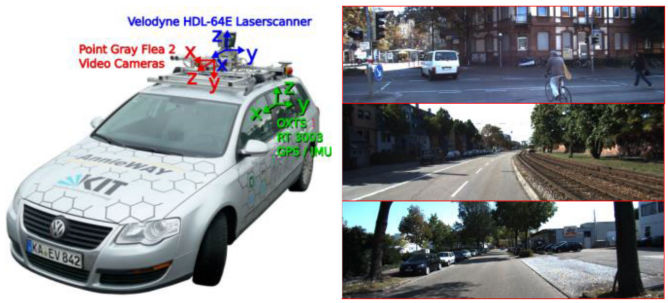


Fig. 8. Evaluated scene of point cloud SR from the KITTI dataset [17].

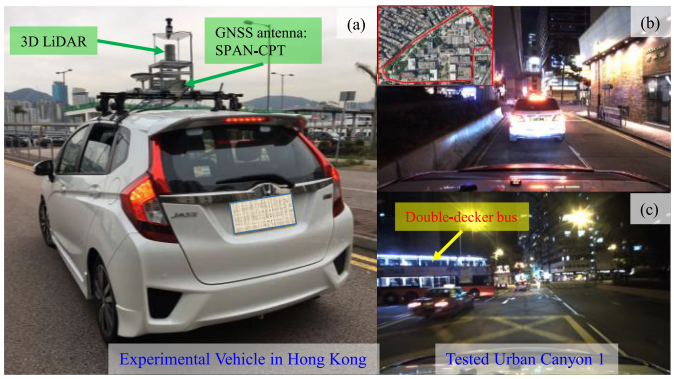


Fig. 9. Experimental vehicle and the evaluated scene of the dataset collected in Hong Kong.

the covariance of the corresponding normal distribution to point $\boldsymbol{x}_i^{'}$ in the NDT of the previous scan.

3) Update the pose using the quasi-Newton method based on the objective function to minimize the score, $f(\mathbf{p})$.

With all the points in one frame of point clouds being modeled as cells, the objective of the optimization for NDT is to match current cells into the previous cells with the highest probability. The optimization function $f(\mathbf{p})$ can be found in [43]. Therefore, $\mathbf{T}$ can be estimated by optimization. Finally, the LiDAR odometry is derived by accumulating the motion ($\mathbf{T}$) between frames.

V. Experimental Evaluation

A. Experimental Setup

To verify the effectiveness of the proposed method, both the open-sourced KITTI dataset [17] and real data collected in dynamic urban canyons of Hong Kong are evaluated. KITTI provides an edited dataset that is specifically provided for depth map SR algorithm evaluation with limited dynamic objects. Moreover, the road structures are similar, which is more friendly for SR algorithms. The environment scene is shown in Fig. 8, which mainly involves static environmental structures.

The evaluated scene from an urban canyon of Hong Kong is shown in Fig. 9, which involves complex road structures (narrow and broad streets, which can be seen in Fig. 9(b)) and numerous dynamic objects (car, high-rising double-decker bus, which can be seen in Fig. 9(c)), leading to considerable challenges to the proposed SR algorithms. Moreover, dynamic objects can also challenge the performance of 3D LiDAR odometry. It can be of interest to determine how the proposed SR algorithm works in this kind of challenging scene. During the experiments, a 3D LiDAR sensor (Velodyne HDL 32E) was employed to collect raw 3D point clouds at a frequency of 10 Hz. In addition, the NovAtel SPAN-CPT, a dual-frequency GNSS (GPS, GLONASS, and Beidou) RTK/INS (fiber-optic gyroscopes, FOG) integrated in navigation system, was used to provide the ground truth of 3D LiDAR odometry-based positioning. The gyro bias in-run stability of the FOG is 1 degree per hour, and its random walk is 0.067 degrees per hour. The baseline between the rover and GNSS base stations is less than 7 km. All the data were collected and synchronized using the robot operation system (ROS) [45]. The coordinate systems between all the sensors were calibrated before the experiment.

Regarding the performance evaluation of 3D point cloud enrichment, this paper follows KITTI, where both mean absolute error (MAE) and root mean squared error (RMSE) metrics are employed. Regarding the evaluation of 3D LiDAR odometry, we make use of the popular EVO [46] toolkit to calculate the relative pose error (RPE). Of note, both translation and rotation are evaluated via EVO. As the dataset from KITTI for SR is not continuous, 3D LiDAR odometry cannot be implemented accordingly. Therefore, we only evaluated the performance of 3D LiDAR odometry using the dataset collected in Hong Kong.

B. Dataset for Training and Testing

Two 2080Ti GPUs were used for training and testing based on PyTorch [47]. We evaluate our framework by computing the loss on all pixels corresponding to the ground truth.

Regarding the dataset from KITTI, 85898 frames of point clouds were selected for training, which is from 64 channels of LiDAR data. One thousand frames of point clouds are selected as the testing dataset. The ground truth of the SR is the dense point clouds, which accumulate 10 times 64 channels of the LiDAR data. Thus, we try to obtain denser point clouds based on 64 channel clouds using our proposed SR algorithm.

Regarding the dataset collected in Hong Kong, 4000 frames of point clouds were selected for training, which were downsampled from 32 channels of LiDAR data to simulate the 16 channels of LiDAR. Based on the manual book Velodyne HDL-32E [49], each laser channel is fixed at a particular elevation angle relative to the horizontal plane of the sensor. In this paper, we calculate the HEA of each laser channel with 32 different HEAs. Every other HEA is selected, forming 16 simulated channels of LiDAR. Approximately 600 frames of point clouds are selected as the testing dataset. Thus, we try to obtain 32 channel point clouds based on 16 channel point clouds using our proposed SR algorithm.

The applied parameters during the training are shown in Table III. The same learning rate decay of 0.01 is selected in both the KITTI and Hong Kong datasets.

C. Evaluation Using the KITTI Dataset

1) Evaluation of Data Enrichment: Table IV Shows the Results of the Proposed 3D Point Cloud SR Based on the Evaluated KITTI dataset. An MAE of 0.68 Meters Is Obtained Based on

TABLE III
THE APPLIED PARAMETERS IN THIS PAPER

Parameters	KITTI Dataset	Hong Kong Dataset
Batch Size	48	128
Epochs	100	100
Learning Rate	1e-3	4e-3
Learning Rate Decay	0.01	0.01

TABLE IV
PERFORMANCE COMPARISON OF THE PROPOSED SR ALGORITHM AND EXISTING ALGORITHMS USING THE KITTI DATASET

Method	MAE (m)	RMSE (m)
SCNN [19]	0.68	2.01
L^u [50]	0.356	1.326
Self-Supervised [26]	0.358	1.384
Local Net [51]	0.268	0.995
Proposed SR with range (0~max range) without MCNN	0.299	1.053
Proposed SR with range (0~max range)	0.283	1.041
Proposed SR with range (<10 m)	0.095	0.291
Proposed SR with range (<15 m)	0.125	0.393
Proposed SR with range (<20 m)	0.149	0.485

SCNN (see Second Row of Table IV). With the Help of the Proposed Method, Which Explores the Discontinuity of Objects, Both MAE and RMSE Decrease By Almost Two-Fold (see Sixth Row of Table IV). The Significantly Improved Accuracy Shows the Effectiveness of the Proposed Point Cloud SR method. The Proposed Method Outperforms Both the L^u [49] and the Self-Supervised [26] Methods Based on the Validated KITTI dataset. Moreover, We Obtain Similar Performance Compared with the Recently Proposed *Local Net* [50]. In Short, the Proposed SR Method Achieves Considerable Performance, Even Compared with the Existing State-Of-The-Art Methods.

To present the performance of the proposed SR algorithm in detail, we compare the accuracies of point cloud SR between different distance ranges. The distance is between the point cloud and the center of the 3D LiDAR. An MAE of 0.095 meters is obtained regarding the point clouds between 0∼10 meters relative to the center of 3D LiDAR, with an RMSE of 0.291 meters. Both MAE and RMSE slightly increase to 0.149 and 0.485 meters for the point clouds between 0∼20 meters, respectively. Interestingly, we can find that decent performance is obtained regarding point clouds between 0∼20 meters, which are suitable for indoor perception or SLAM applications. Therefore, we believe that the application of the proposed SR method in indoor mobile robotic applications is also an area of interest to explore.

Fig. 10 shows a selected frame of the depth map from the validated KITTI dataset. Fig. 10(a) shows the input 64 channel depth map. Note that the point clouds are already projected into the depth map in the KITTI dataset. With the help of the proposed SR algorithm, the density is significantly enhanced, which can be seen in Fig. 10(b) (MAE: 0.310 meters). Fig. 10(c) shows that the depth map only includes the point clouds with ranges between 0∼20 meters (MAE: 0.260 meters). Moreover, the number of points is increased by two-fold from 19622 (see Fig. 10(a)) to 62010 (see Fig. 10(b)). To show the details of the SR, Fig. 11 shows the amplified area corresponding to the selected areas of Fig. 10. We can see that the proposed SR (Fig. 11(b)) obtains similar details compared with the ground truth depth map (Fig. 11(d)). The significantly enhanced detail again shows the effectiveness of the proposed algorithm.

2) Error Distribution of the SR Point Clouds: Regarding the original point clouds provided by 3D LiDAR (e.g., HDL 32E), the accuracy is bounded at -0.2 to $+0.2$ meters [48], and the error distribution can be modeled using Gaussian noise. However, as Table IV shows, the RMSE can still reach 1.073 meters even when using the proposed SR method. This inspired us to determine how the point cloud errors are distributed, which is significant for the sensor integration of LiDAR with other sensors, such as inertial measurement units (IMUs). Fig. 12 shows the histogram of the errors corresponding to Fig. 10(b). The left bound of the error reaches -33.6 meters, which is significantly larger than the original accuracy (-0.2 to $+0.2$ meters). Moreover, the right bound reaches 8.07 meters, which is caused by outliers. As a result, the unexpected outliers lead to the long-tail phenomenon, which can be seen in Fig. 12. Therefore, it is difficult to use a single Gaussian noise to model it. Interestingly, recent work in [51] proposed using the Gaussian mixture model (GMM) to describe the non-Gaussian distribution. Inspired by their work, we employ the GMM, denoted by the blue curve in Fig. 12, to describe the error of the point clouds. We find that 3 components can effectively fit the histogram. The table in Fig. 12 shows the parameters (mean, standard deviation, and weightings). We can see that the first Gaussian component with a mean of approximately zero only makes up 77.74% of the GMM. The rest of the GMM is contributed by the other two Gaussian components that describe the long-tail phenomenon. In short, we conclude that it is difficult to use a conventional Gaussian distribution to describe the error distribution of point clouds after applying SR. The GMM is a promising solution to effectively model point clouds after SR, especially in sensor fusion.

D. Evaluation in the Dynamic Urban Canyon of Hong Kong

Instead of relying on the KITTI dataset, this section tests a challenging dataset (the whole length is approximately 2.01 km) collected from an urban canyon of Hong Kong to validate the proposed SR method. In addition, as the collected dataset is continuous, which is different from the evaluated KITTI dataset in Section C, we also present the performance analysis of 3D LiDAR odometry before and after applying the depth map SR.

1) Evaluation of Data Enrichment: Since the performance evaluation of SR on our Hong Kong dataset is not available based on previously mentioned methods [26], [49], [50], we only evaluate the performance of our proposed SR method in this paper. Moreover, we open-source our evaluated Hong Kong dataset by linking it (https://www.polyu-ipn-lab.com/download) to the benchmark with other researchers.

Table V shows the results of the point cloud SR. An MAE of 0.472 meters is obtained using the proposed method, which is slightly larger than the MAE (0.306 meters) in the evaluated

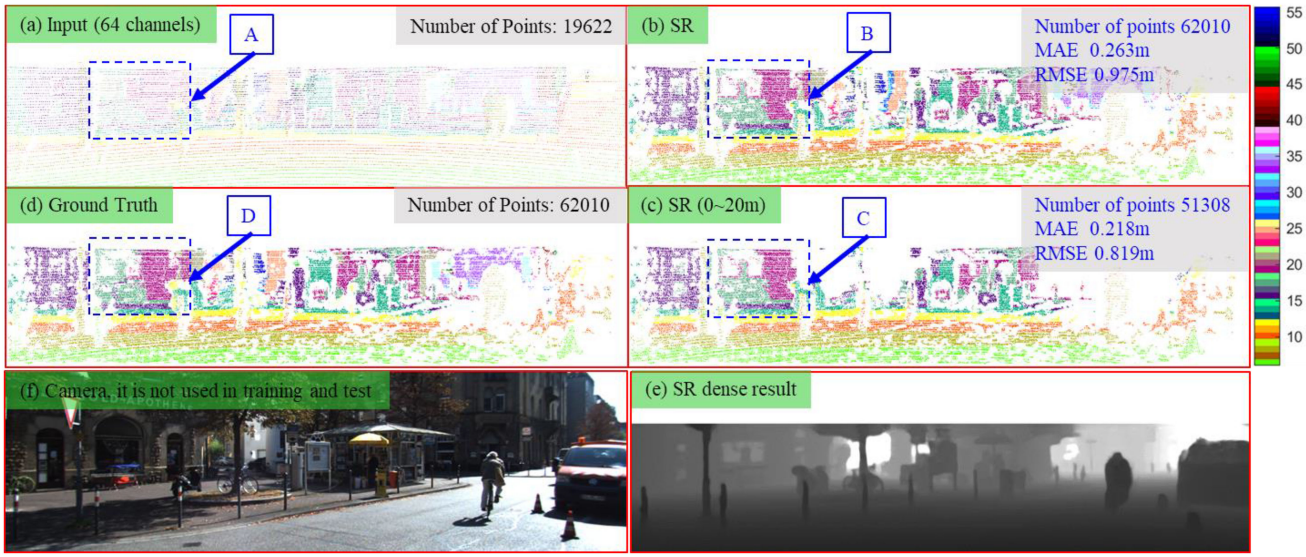


Fig. 10. (a) The 64 channels input from the KITTI dataset; (b) SR result; (c) SR result with point clouds limited between 0∼20 meters; (d) the ground truth of dense point clouds from the KITTI dataset; (e) the SR dense result, from which the results of (b) and (c) are acquired; (e) the related camera image, it is not used in the training or test, as a reference of the dense result only. The point is colored according to the depth, and the color scale is shown on the right of the figure.

Fig. 11. The details of the selected areas corresponding to the blue dashed rectangles in Fig. 10.

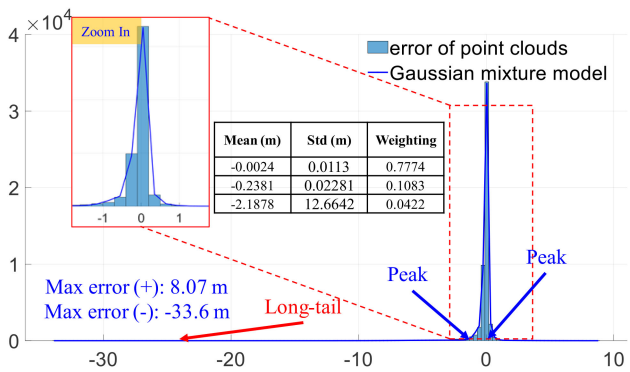


Fig. 12. The error distribution of the point clouds after applying the proposed SR method in the KITTI dataset. The x-axis denotes the error range. The y-axis denotes the number of counts.

TABLE V
PERFORMANCE OF DATA ENRICHMENT USING THE HONG KONG DATASET

Method	MAE (m)	RMSE (m)
SR with range (0~max range)	0.472	2.213
SR with range (<5 m)	0.117	0.362
SR with range (<10 m)	0.137	0.624
SR with range (<15 m)	0.196	0.926
SR with range (<20 m)	0.255	1.147

KITTI dataset. This phenomenon is due to the more complex environmental structures shown in Fig. 9. Regarding the point clouds within 0∼5 meters, the MAE decreases to only 0.117 meters with an RMSE of 0.362 meters. Both the MAE and RMSE increase gradually with increasing distance. Interestingly, an MAE of 0.255 meters is obtained regarding the point clouds within 0∼20 meters with an RMSE of 1.147 meters. We believe that this kind of accuracy is still a promising solution for indoor applications. In short, the proposed depth map SR method obtains good accuracy even in the evaluated challenging dataset collected in urban canyons.

Fig. 13 shows a few selected frames of the point clouds from the Hong Kong dataset. Fig. 13(a) shows the input 16 channel point clouds. Note that the 16 channel point clouds are strictly and homogeneously extracted from the 32 channel point clouds based on the geometric distribution of the rings. Thus, one ring is extracted for 16 channel point clouds from every two rings from 32 channel point clouds. With the help of the proposed SR algorithm, the density is significantly enhanced, which can be seen in Fig. 13(b) (MAE: 0.325 meters). Fig. 13(c) shows the point clouds that only include the point clouds with ranges between 0∼20 meters (MAE: 0.178 meters). Moreover, the number of points is increased by almost two-fold from 31014 (see Fig. 13(a)) to 60285 (see Fig. 13(b)).

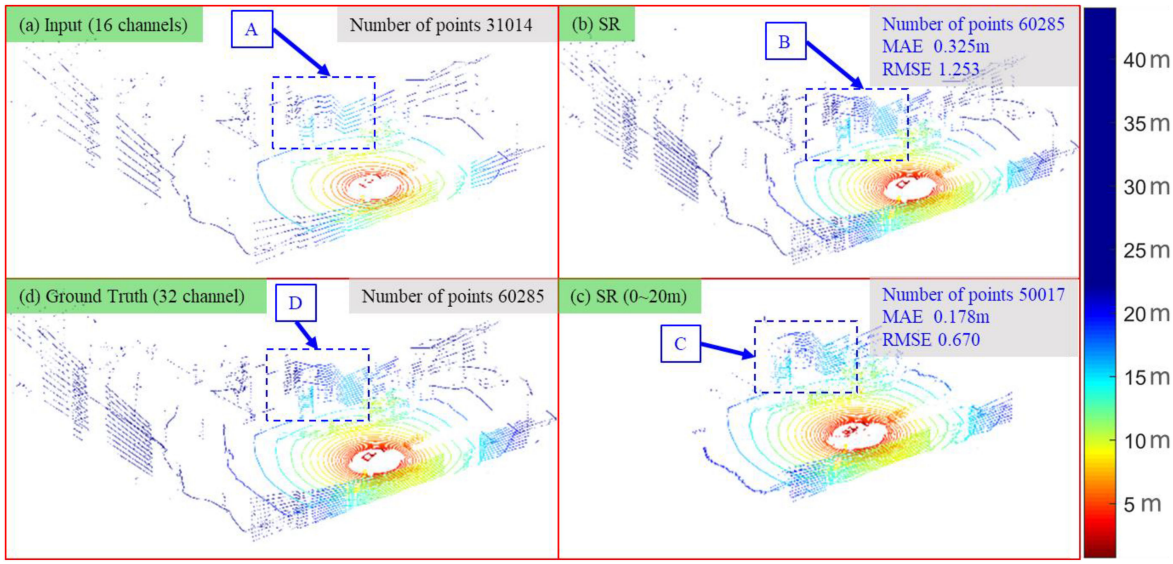


Fig. 13. (a) The 16 channels input from the Hong Kong dataset; (b) SR result; (c) SR result with point clouds limited between 0∼20 meters; (d) the ground truth of dense point clouds from the 32-channel point clouds. The point is colored by its depth from the sensor, and the color scale is shown on the right of the figure. A more detailed video can be found at link (https://www.youtube.com/watch?v=a-hUjiu4Byw).

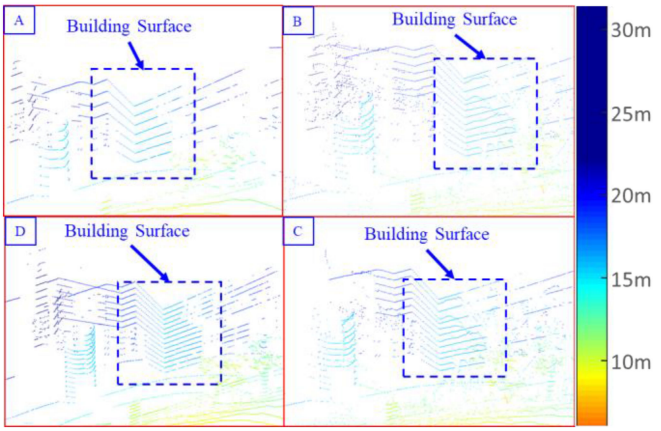


Fig. 14. The details of the selected areas corresponding to the blue dashed rectangles in Fig. 13.

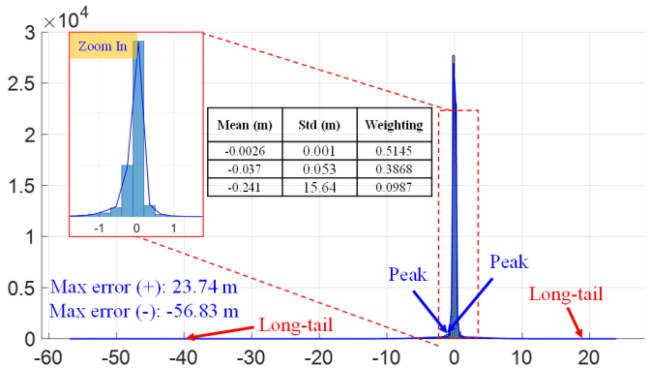


Fig. 15. The error distribution of the point clouds after applying the proposed SR method in the Hong Kong dataset. The x-axis denotes the error range. The y-axis denotes the number of counts.

To show the details of the SR based on the Hong Kong dataset, Fig. 14 shows the amplified area corresponding to the selected areas of Fig. 13. The selected area denotes the part of a building surface. We can see that the proposed SR (Fig. 14(a)) obtains similar details compared with the ground truth depth map (Fig. 14(d)). The significantly enhanced detail again shows the effectiveness of the proposed method. The video of the SR of the evaluated Hong Kong dataset can be found at the link.

2) Error Distribution of the SR Point Clouds: Similarly, the error distribution of the point clouds in Fig. 13(b) is also presented in Fig. 15. The left bound of the error reaches -56.83 meters, which is significantly larger than the original accuracy (−0.2 to +0.2 meters). Moreover, the right bound reaches 23.74 meters, which is caused by outliers. As a result, the unexpected outliers lead to a similar long-tail phenomenon, which can be seen in Fig. 15. An interesting finding is that both the KITTI and Hong Kong datasets introduce larger left bounds. Thus, the proposed SR method tends to underestimate the distance of point clouds. However, the reason behind this phenomenon is still an open question that will be explored in our future work.

The table in Fig. 12 shows the parameters (mean, standard deviation, and weightings). We can see that the first Gaussian component with a mean of approximately zero only makes up 51.45% of the GMM, which is significantly smaller than the one (77.74%) in the evaluated KITTI dataset. Thus, it is even harder to describe the error distribution of the Hong Kong dataset using only one Gaussian component. The rest of the GMM is contributed by the other two Gaussian components that describe the long-tail phenomenon. In short, we conclude that the error of point clouds from the Hong Kong dataset after applying SR leads to a severe long-tail phenomenon. The GMM is a promising solution to effectively model the potential error.

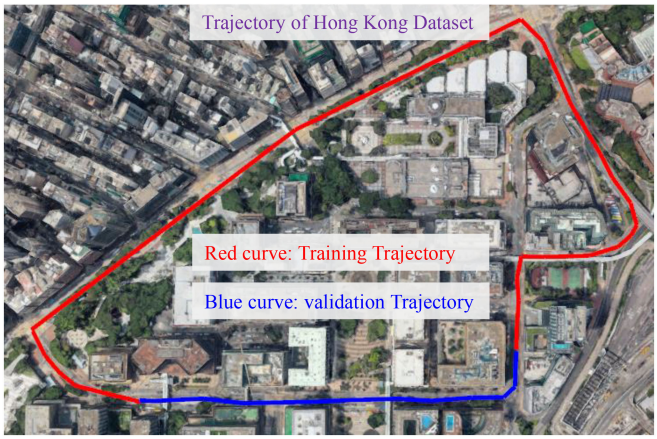


Fig. 16. The trajectory of the evaluated Hong Kong dataset. The red part is employed to train the SR method in Fig. 5. The blue part is utilized for both SR validation and LiDAR odometry evaluation.

TABLE VI
PERFORMANCE COMPARISON OF ODOMETRY USING 3D POINT CLOUDS WITH DIFFERENT DENSITIES

All data	16 Channels (m)	SR-aided (m)	32 channels (m)
Max error	8.12	0.93	0.92
Mean error	2.09	0.35	0.32
Min error	0.06	0.05	0.06
RMSE	3.51	0.42	0.39
STD	2.82	0.23	0.20

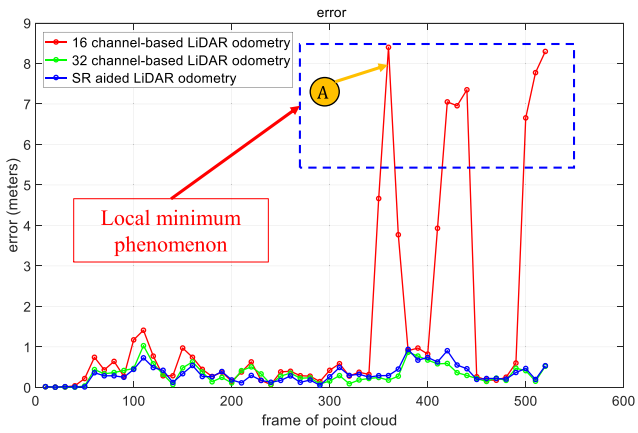


Fig. 17. The positioning errors of the tested methods in the Hong Kong dataset. The red and green curves denote the positioning errors of 16- and 32-channel point cloud-based odometry, respectively. The blue curve represents the odometry result using the enriched 3D point clouds.

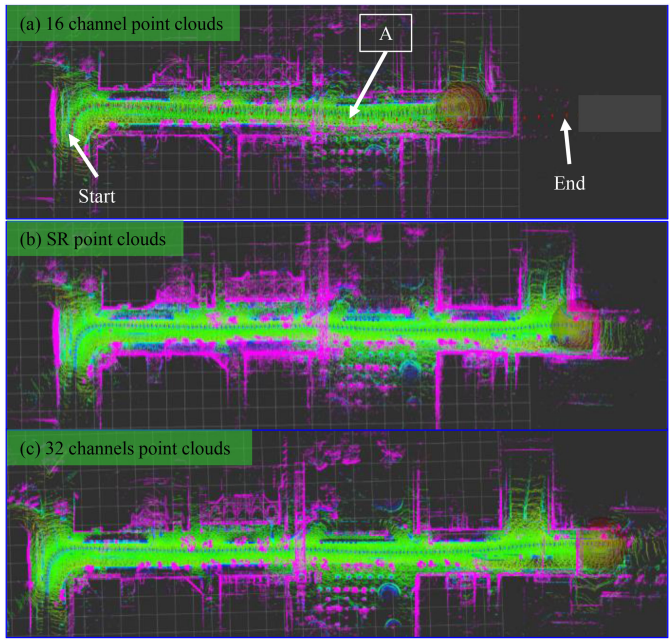


Fig. 18. The generated point cloud map based on (a) 16-channel point clouds, (b) SR-aided point clouds, and (c) 32-channel point clouds. The color is determined by the height value of the point cloud.

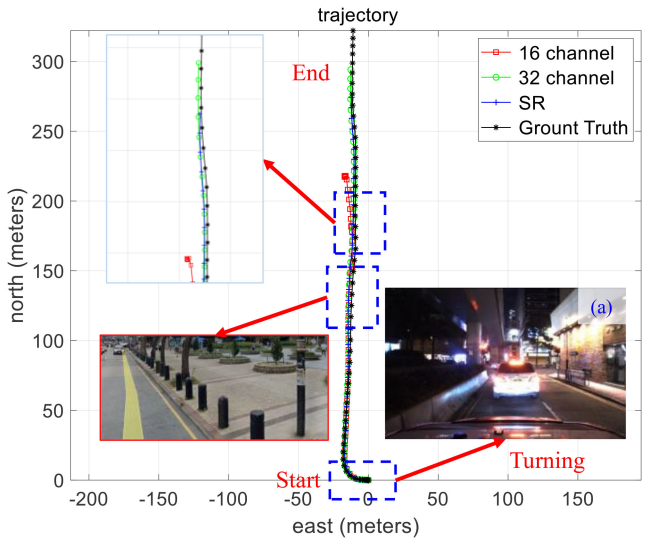


Fig. 19. The trajectories of the tested methods in the Hong Kong dataset. The red and green curves denote the trajectories of 16- and 32-channel point cloud-based odometry, respectively. The blue curve represents the LiDAR odometry result using the enriched 3D point clouds. The black curve represents the ground truth trajectory.

3) Evaluation of Odometry Based on Enriched 3D Point Clouds: In this section, we evaluate the potential of the enriched point clouds in 3D LiDAR odometry. Fig. 16 shows the trajectory for training (red curve) of the SR algorithm and validation (blue) purpose. Three different 3D LiDAR-based odometry methods are evaluated: (1) 16 channel point clouds, (2) enriched point clouds using the proposed SR method, and (3) 32 channel point clouds.

Table VI Shows the Performance of the Odometry evaluation. A Mean Error of 2.09 Meters Is Obtained Using the 16-channel Point Clouds with a Maximum Error of 8.12 meters. This Phenomenon Is Mainly Caused By the Limited Density of Point Clouds, Leading to the Local Minimum Problem in NDT-based LiDAR odometry. Fig. 17 Shows the Positioning Error Throughout the test. The X-Axis Denotes the Frame Index of the Point clouds. The Y-Axis Represents the Positioning error.

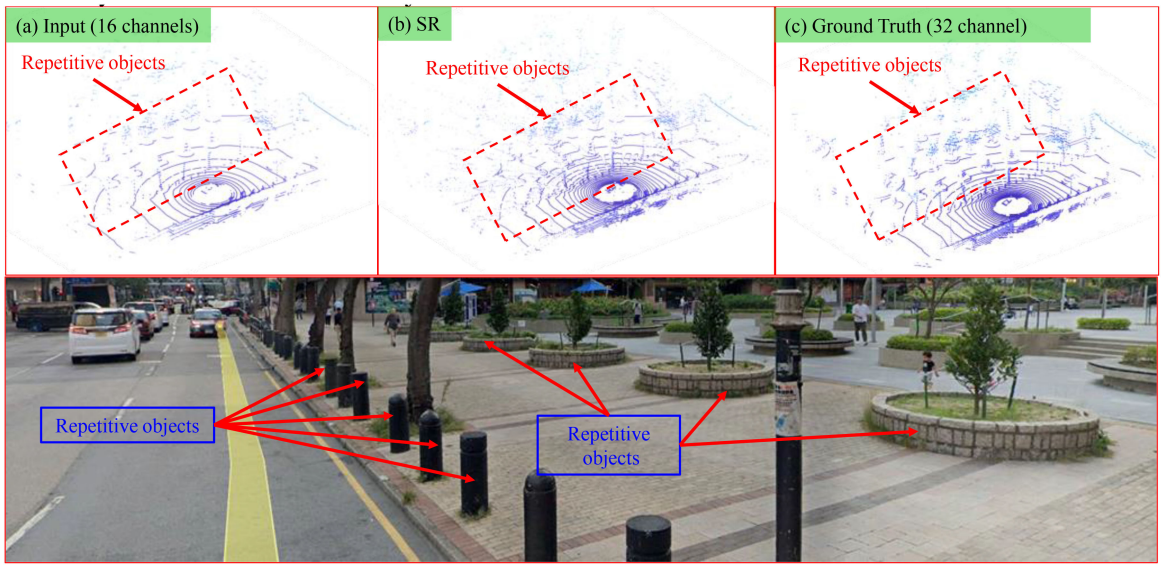


Fig. 20. The top panel shows the point clouds at epoch C annotated in Fig. 16 from the (a) input 16 channel, (b) SR-aided, and (c) 32 channel. The bottom panel shows the scene captured from Google Earth.

We Can See That the Error Reaches More Than 5 Meters After Frame 330. With the Help of the Proposed SR Method, the Mean Error Significantly Decreases to 0.33 Meters with a Maximum Error of 0.93 meters. Moreover, the STD Is Only 0.23 meters.

The fourth column of Table VI shows the accuracy of the 3D LiDAR based on the original 32-channel point clouds. We can see that we obtain similar accuracy compared with the original 32-channel point cloud-based 3D LiDAR odometry.

Fig. 18 shows the generated point cloud map based on the three listed LiDAR odometry methods. The map generated by 16 channels is significantly smaller due to the erroneous pose estimation caused by the local minimum. The trajectories of the three methods are shown in Fig. 19. To show the details of the reasons leading to a local minimum of NDT-based point cloud registration, we present the point clouds at epoch A as in Fig. 19. We can see from the bottom panel that there are numerous repetitive objects, such as small flower nurseries and trees. The repetitive scene is one of the major factors leading to the unexpected local minimum phenomenon. Thus, it is difficult for the point cloud registration method to find the global optimum when there are numerous repetitive scenes. Moreover, we can see from the top panel of Fig. 20(a) that the point cloud from 16 channels is very sparse, leading to the more severe local minimum phenomenon. With the help of the proposed SR method, the density of point clouds is significantly enhanced (see Fig. 20(b)). As a result, more details of objects are recovered in the generated point cloud map accordingly. In short, we conclude the following:

1) Although the accuracy of the point clouds after applying SR reaches 0.472 meters (see Table V), we still achieve similar accuracy in 3D LiDAR odometry in the evaluated Hong Kong dataset compared with the 32-channel dataset. Moreover, the denser point cloud map is generated with the help of the proposed SR method, which can be seen in Fig. 18.

TABLE VII
PERFORMANCE COMPARISON OF ODOMETRY BASED ON ICP [54] USING 3D POINT CLOUDS WITH DIFFERENT DENSITIES

All data	16 Channels (m)	SR-aided (m)	32 channels (m)
Max error	0.53	0.47	0.42
Mean error	0.35	0.27	0.25
Min error	0.04	0.04	0.04
RMSE	0.43	0.31	0.27
STD	0.25	0.19	0.13

2) We do not argue that our method can achieve similar performance compared with the 32-channel method in all scenarios. However, the proposed method can significantly improve the performance of 16-channel LiDAR-based odometry in sparse scenarios (e.g., Fig. 19) with limited or repetitive environmental structures.

4) Discussions: To show the generalization capabilities of the proposed method, we present the performance of the LiDAR odometry the other two typical point cloud registration methods, the generalized iterative closest point (G-ICP) [8], [10], and the iterative closest point (ICP) method [5], [52]. The trajectories of the LiDAR odometry using the G-ICP method are shown in Fig. 21, which is similar to Fig. 19. The positioning error of the G-ICP is shown in Table VII. We can see that a similar improvement is also obtained for G-ICP-based LiDAR odometry, with the mean error decreasing from 0.35 m (16-channel 3D point clouds) to 0.27 meters (SR-aided 3D point clouds). This improvement can also be seen from the trajectories shown in Fig. 21. Moreover, we obtain similar accuracy after applying the proposed enriched point clouds compared with the 32 channel point cloud-based LiDAR odometry.

Fig. 22 shows the trajectories of the LiDAR odometry using ICP, which is similar to Fig. 19. Interestingly, all three

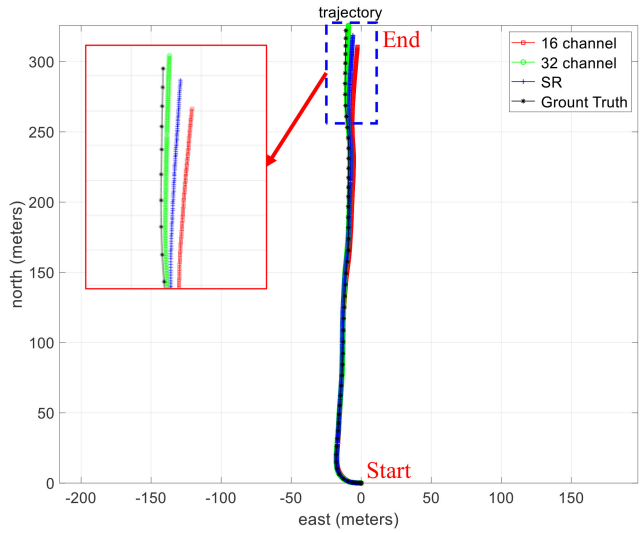


Fig. 21. The trajectories of LiDAR odometry based on G-ICP using the Hong Kong dataset. The red and green curves denote the trajectories of 16- and 32-channel point cloud-based odometry, respectively. The blue curve represents the LiDAR odometry result using the enriched 3D point clouds. The black curve represents the ground truth trajectory.

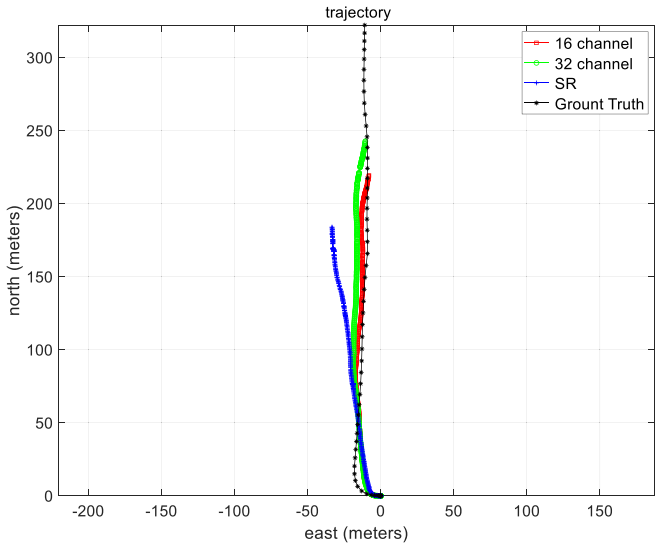


Fig. 22. The trajectories of LiDAR odometry based on ICP using the Hong Kong dataset. The red and green curves denote the trajectories of 16- and 32-channel point cloud-based odometry, respectively. The blue curve represents the LiDAR odometry result using the enriched 3D point clouds. The black curve represents the ground truth trajectory.

trajectories are significantly shorter than the ground truth trajectory (black curve), leading to a large mean error. Interestingly, although the mean error is large, we find that the LiDAR odometry based on the proposed enriched point clouds deviates dramatically from the ground truth trajectory, which is even worse than the other two methods (green and red curves). This phenomenon occurs because ICP directly estimates the transformation between two frames of point clouds using the point distance, which relies heavily on the initial guess of the transformation. Otherwise, it can easily get into a local minimum. After enriching the point clouds using the proposed SR method, the local minimum issues can be even more severe. A similar phenomenon was also witnessed in [54]. Instead of simply estimating the transformation between two frames of point clouds using the point distance, the G-ICP and NDT methods make use of the Gaussian model to describe the geometric distribution of the point clouds. Moreover, the G-ICP and NDT are less sensitive to the initial guess due to the point description using the Gaussian model compared with the ICP. With the help of the proposed superresolution method, more details (see Fig. 13) are recovered, leading to a better geometric distribution. As a result, improved accuracy is obtained in G-ICP- and NDT-based LiDAR odometry tests.

VI. Conclusion and Future Work

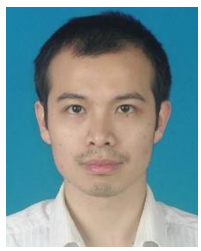
Achieving accurate positioning in urban canyons using a low-cost 16-channel 3D LiDAR sensor is still a challenging problem due to the sparsity of supplied 3D point clouds. This article opens a new window for improving the performance of 3D LiDAR odometry in urban canyons by proposing to enrich sparse 3D point clouds to a denser one via a novel deep learning-based superresolution (SR) algorithm before its utilization in 3D LiDAR odometry. In contrast to SCNN, this paper relaxes the drawbacks of the work in [16], which fails to explicitly explore the discontinuity of 3D point clouds. Moreover, this paper eliminates the reliance on the camera compared with our previous work [25]. Regarding the performance of point cloud SR, we deliver considerable accuracy, even compared with the state-of-the-art methods [26], [49], [50]. The results show that exploring the discontinuity of 3D point clouds can effectively improve the accuracy of SR. Moreover, the performance of 3D LiDAR odometry is significantly improved in the evaluated Hong Kong dataset with the help of the enriched 3D point clouds. The improved results show the effectiveness of the proposed method.

With the significantly enhanced point cloud density, the capability of environmental description can be improved accordingly. We will explore the potential of the enriched 3D point clouds in detecting [55], [56] or correcting [57] global navigation satellite system (GNSS) outlier measurements in the future. We will explore the potential of the integration of the enriched 3D point clouds with other sensors, such as IMU and GNSS. Moreover, the study of the proposed SR method in indoor applications will also be performed, which we believe can be a promising research direction.

References

[1] P. A. Ioannou and C.-C. Chien, "Autonomous intelligent cruise control," *IEEE Trans. Veh. Technol.*, vol. 42, no. 4, pp. 657–672, Nov. 1993.
[2] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual SLAM with building structure lines," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1364–1375, Apr. 2015.
[3] W. Wen, X. Bai, W. Zhan, M. Tomizuka, and L.-T. Hsu, "Uncertainty estimation of LiDAR matching aided by dynamic vehicle detection and high definition map," *Electron. Lett.*, vol. 55, no. 6, pp. 348–349, 2019.
[4] S. S. Saab, "A map matching approach for train positioning. I. Development and analysis," *IEEE Trans. Veh. Technol.*, vol. 49, no. 2, pp. 467–475, Mar. 2000.

[5] T. Shan and B. Englot, "LeGo-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
[6] W. Wen *et al.*, "UrbanLoco: A full sensor suite dataset for mapping and localization in urban scenes," 2019, *arXiv:1912.09513*.
[7] W. Wen, L.-T. Hsu, and G. Zhang, "Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of hong kong," *Sensors*, vol. 18, no. 11, 2018, Art. no. 3928.
[8] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," *Robot.: Sci. Syst.*, vol. 2, no. 4, p. 435, 2009.
[9] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 2743–2748.
[10] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," *Robot.: Sci. Syst.*, vol. 2, no. 9, 2014, pp. 1–9.
[11] S. Pang, D. Kent, X. Cai, H. Al-Qassab, D. Morris, and H. Radha, "3D Scan registration based localization for autonomous vehicles-A comparison of NDT and ICP under realistic conditions," in *Proc. IEEE 88th Veh. Technol. Conf.*, 2018, pp. 1–5.
[12] L. Zheng, Y. Zhu, B. Xue, M. Liu, and R. Fan, "Low-cost GPS-aided lidar state estimation and map building," 2019, *arXiv:1910.12731*.
[13] S. Zhao, Z. Fang, H. Li, and S. Scherer, "A robust laser-inertial odometry and mapping method for large-scale highway environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1285-1292.
[14] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D lidar inertial odometry and mapping," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 3144–3150.
[15] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Proc. Int. Conf. Robot. Automat.*, 2015, pp. 2174–2181.
[16] T. Shan, "Minimalistic and learning-enabled navigation algorithms," Ph.D. dissertation, Stevens Inst. Technol., Hoboken, NJ, USA, 2019.
[17] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
[18] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[19] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 11–20.
[20] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy liDAR completion with RGB guidance and uncertainty," 2019, *arXiv:05356*.
[21] T.-W. Hui, C. C. Loy, and X. Tang, "Depth map super-resolution by deep multi-scale guidance," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 353–369.
[22] W. Dong, G. Shi, X. Li, K. Peng, J. Wu, and Z. Guo, "Color-guided depth recovery via joint local structural and nonlocal low-rank regularization," *IEEE Trans. Multimedia*, vol. 19, no. 2, pp. 293–301, Feb. 2017.
[23] J. Qiu *et al.*, "DeepLiDAR: deep surface normal guided depth prediction for outdoor scene from sparse LiDAR data and single color image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3308–3317.
[24] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," 2019, *arXiv:1908.01238*.
[25] J. Yue, W. Wen, J. Han, and L.-T. Hsu, "LiDAR data enrichment using deep learning based on high-resolution image: An approach to achieve high-performance LiDAR SLAM using Low-cost LiDAR," 2020, *arXiv:2008.03694*.
[26] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 3288–3295.
[27] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.
[28] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.
[29] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.
[30] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.
[31] Z. Xiong, K. Ramchandran, M. T. Orchard, and Y.-Q. Zhang, "A comparative study of DCT-and wavelet-based image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 5, pp. 692–695, Aug. 1999.
[32] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.
[33] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–10.
[34] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surf.*, 2010, pp. 711–730.
[35] D. Scharstein *et al.*, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. German Conf. Pattern Recognit.*, 2014, pp. 31–42.
[36] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.
[37] L. Ding and A. Goshtasby, "On the canny edge detector," *Pattern Recognit.*, vol. 34, no. 3, pp. 721–725, 2001.
[38] D. Silva, "PyTorch-ENet," [Online]. Available: https://github.com/davidtvs/PyTorch-ENet
[39] E. Romera, "erfnet_pytorch," [Online]. Available: https://github.com/Eromera/erfnet_pytorch
[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[41] F. Zhang, V. Prisacariu, R. Yang, and P. H. S. Torr, "GA-Net: Guided aggregation net for end-to-end stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 185-194.
[42] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm," *Image Vis. Comput.*, vol. 23, no. 3, pp. 299–309, 2005.
[43] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007.
[44] M. Magnusson, "The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro Univ., Örebro, Sweden, 2009.
[45] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. Int. Conf. Robot. Automat., Open-Source Softw. Workshop*, 2009, p. 5.
[46] M. Grupp, "evo: Python package for the evaluation of odometry and slam," in *Note: https://github.com/MichaelGrupp/evo Cited by: Table*, vol. 7, 2017.
[47] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.
[48] V. LiDAR, *HDL-32E User Manual, Velodyne LiDAR*. Velodyne LiDAR, Inc., 2018.
[49] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (DDP) from single image and sparse range," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3348–3357.
[50] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy LiDAR completion with RGB guidance and uncertainty," in *Proc. 16th Int. Conf. Mach. Vis. Appl.*, 2019, pp. 1–6.
[51] T. Pfeifer and P. Protzel, "Expectation-maximization for adaptive mixture models in graph optimization," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 3151–3157.
[52] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, 1992.
[53] K. Koide, J. Miura, and E. Menegatti, "A portable 3D LIDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, 2018, Art. no. 1729881419841532.
[54] S. Pang, D. Kent, X. Cai, H. Al-Qassab, D. Morris, and H. Radha, "3D scan registration based localization for autonomous vehicles-A comparison of NDT and ICP under realistic conditions," in *Proc. IEEE 88th Veh. Technol. Conf.*, 2018, pp. 1–5.
[55] W. Wen, G. Zhang, and L.-T. Hsu, "GNSS NLOS exclusion based on dynamic object detection using LiDAR point cloud," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 853–862, Feb. 2021.
[56] W. Wen, X. Bai, Y.-C. Kan, and L.-T. Hsu, "Tightly coupled GNSS/INS integration via factor graph and aided by fish-eye camera," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10651–10662, Nov. 2019.
[57] W. Wen, G. Zhang, and L. T. Hsu, "Correcting NLOS by 3D LiDAR and building height to improve GNSS single point positioning," *Navigation*, vol. 66, no. 4, pp. 705–718, 2019.



Jiang Yue received the B.S. and Ph.D. degrees in electronic science and technology from the Nanjing University of Science and Technology (NUST), China, in 2008 and 2014, respectively. He has been employed as a Lecturer with the State Key Laboratory of Transient Physics, Nanjing University of Science and Technology (NUST) since 2014. He is currently a Postdoctoral Fellow with the Interdisciplinary Division of Aeronautical and Aviation Engineering (AAE), The Hong Kong Polytechnic University. His research interests include signal processing, LiDAR & image fusion, high-dimensional data denoising, and superresolution.

Weisong Wen was born in Ganzhou, Jiangxi, China. He is working toward the Ph.D. degree in mechanical engineering with Hong Kong Polytechnic University. His research interests include multisensor integrated localization for autonomous vehicles, SLAM, and GNSS positioning in urban canyons. He was a Visiting Student Researcher with the University of California, Berkeley (UCB), in 2018.

Jin Han received B.S. and Ph.D. degrees from the Nanjing University of Science and Technology, China, in 2009 and 2015, respectively. She is currently an Associate Professor with the Nanjing University of Science and Technology. Her research interests include computer vision, computing imaging and deep learning in scattering imaging.

Li-Ta Hsu received the B.S. and Ph.D. degrees in aeronautics and astronautics from National Cheng Kung University, Taiwan, in 2007 and 2013, respectively. He is currently an Assistant Professor with the Interdisciplinary Division of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University. Previously, he was a Postdoctoral Researcher with the Institute of Industrial Science, the University of Tokyo, Japan. In 2012, he was a Visiting Scholar with University College London, the U.K. His research interests include GNSS positioning in challenging environments and localization for pedestrians, autonomous driving vehicles, and unmanned aerial vehicles.