A Coarse-to-Fine LiDAR-Based SLAM with Dynamic Object Removal in Dense Urban Areas

Feng Huang[1], Donghui Shen[1], Weisong Wen[1], Jiachen Zhang[1,2] and Li-Ta Hsu[1]

[1]Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong, China
[2]School of Precision Instrument and Opto-Electronics Engineering, Tianjin University, Tianjin 300072, China

BIOGRAPHY

Feng Huang received his bachelor's degree from Shenzhen University in Automation in 2014 and MSc in Electronic Engineering at Hong Kong University of Science and Technology in 2016. He is a Ph.D. student in the Department of Aeronautical and Aviation Engineering, Hong Kong Polytechnic University. His research interests including localization and sensor fusion for autonomous driving.

Shen Donghui received his bachelor's degree from Shandong University in Space Science and Technology in 2017 and MSc in Space Physics at Shandong University in 2020. His research interests including 3D perception and HD Map updating for autonomous driving.

Weisong Wen was born in Ganzhou, Jiangxi, China. He received a Ph.D. degree in mechanical engineering, the Hong Kong Polytechnic University, in 2020. He is currently a senior research fellow at the Hong Kong Polytechnic University. His research interests include multi-sensor integrated localization for autonomous vehicles, SLAM, and GNSS positioning in urban canyons. He was a visiting student researcher at the University of California, Berkeley (UCB) in 2018.

ZHANG Jiachen received her bachelor's degree from Tianjin University in Information Engineering in 2016 and is currently an enrolled, full-time graduate student at Tianjin University, majoring in Optical Engineering. She is working as a research assistant in the Intelligent Positioning and Navigation Laboratory. Her research interests including localization and sensor fusion for autonomous driving.

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 Division of Aeronautical and Aviation Engineering, Hong Kong Polytechnic University, before he served as post-doctoral researcher in Institute of Industrial Science at University of Tokyo, Japan. In 2012, he was a visiting scholar in University College London, U.K. He was a technical representative in ION in 2019-2021 and is an Associate Fellow of RIN. His research interests include GNSS positioning in challenging environments and localization for pedestrian, autonomous driving vehicle and unmanned aerial vehicle.

ABSTRACT

Robust and precise localization and mapping are essential for autonomous systems. Light detection and ranging (LiDAR) odometry is extensively studied in the past decades to achieve this goal. However, almost all the LiDAR-based approaches are built on top of the static world assumption. The performance of the LiDAR-based method is significantly degraded in urban canyons with enormous dynamic objects. To tackle this challenge, we propose a coarse-to-fine LiDAR-based solution with dynamic object removal. Both instant-level deep neural network (DNN) and point-wise discrepancy images are adopted to deal with the dynamic points. The evaluation results show that a 19.1% improvement of the LiDAR-based method in a highly urbanized area can be achieved by distinguishing dynamic objects from LiDAR scan while generating clean maps for real-world representation.

1. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is fundamental for most autonomous systems [1, 2]. The three-dimensions (3D) light detection and ranging (LiDAR), which provides dense 3D point clouds of the surroundings, had recently been widely studied to provide accurate and high-frequency LiDAR-Based positioning for autonomous systems [3]. However, most of the

LiDAR-based SLAM methods rely on the assumption that the environmental features are static then find a property transformation to minimize the distance between correspondences [4-6]. Therefore, the estimation accuracy could be highly degenerated [7, 8] in urban canyons due to excessive moving objects. Also, the existence of dynamic objects affects the static world representation [9] with long-tailed object generated, so that fail to meet the long-term mapping requirements for autonomous application.

To mitigate the impacts of the errors caused by dynamic objects, numerous literatures [10, 11] are presented to address the localization and mapping problems in changing environments. A random sampling consensus (RANSAC) based method [12] is proposed to eliminate the mismatching effects by excluding the moving objects as outliers. However, the performance will be significantly downgraded with an enormous number of dynamic points. The clustering method [13] is adopted to divide the point cloud into several organized groups, and the dynamic vehicles are efficiently extracted based on the classification modeling. But the parameter-based approaches are vulnerable to unknown classes or the threshold limit. In recent years, a large body of deep learning-based methods [14, 15] are presented to eliminate the effects of moving objects, which achieved satisfactory results in the widely evaluated KITTI dataset [16]. To tackle the dynamic objects in scenes, LO-Net [17] was proposed with mask-weighted geometric constraint loss which achieved similar results as LOAM. Recent work [18] proposes to improve the SLAM accuracy by predicting the point-wise semantic label based on the RangeNet++ [19] with a range image. For the generality of the learning-based method, an unsupervised dynamic awareness LO method [20] was proposed by the team from ETH. The dynamic objects are labeled automatically by the occupancy grid-based method. However, the performance of detection and motion estimation in a highly urbanized environment such as Hong Kong is still a challenging research topic.

Change detection [21] is another efficient way to detect objects. Yoon et al. [22] proposed a ray-tracing method with a false-positive filter. Another work in [23] proposes to build a static map based on voxel ray casting-based methods. But it is a time-consuming task as needs to traverse the voxel grid. To alleviate the computational load, a range image-based visibility check is proposed in [24] to remove the dynamic points on the map directly. But the detections are often containing static areas due to the limitation of the field of view (FOV).

The objective of the paper is to provide a complete pipeline to optimize the LiDAR-based SLAM performance via detecting and removing the dynamic objects. Our approach leverages the potential of deep neural network (DNN) and point-wise discrepancy comparison. First, a custom trained DNN [25] is employed to obtain precious feature representation and classification for the highly urbanized areas, an example of the vehicle detected in an intersection is shown in Fig. 1 (b). Secondly, an existing LiDAR SLAM [3] is utilized to yield coarse poses using clean scans. The poses generated can be used to construct a submap to further refine the dynamic objects in the LiDAR frame by comparing the range image-based discrepancy between scan and submap. The initial guess of odometry is provided in the coarse process and the refined scans are further processed by the LiDAR odometry to generate an accurate pose. Based on the data we have shown, the relative translation error is improved by 19.1% after filtering the dynamic objects. The results of Fig. 1 (c) and (d) show that the generated point cloud map is fewer non-static points compare to the original LiDAR SLAM method. The major contributions of the work are summarized as follows:

(1) The paper proposes a coarse-to-fine LiDAR-Based pipeline with dynamic object removal algorithms.
(2) Instant-level deep learning method and point-wise discrepancy calculation are adopted to deal with the dynamic points.
(3) Performance evaluation of the proposed pipeline using the challenging datasets collected in typical urban canyons of Hong Kong.

The rest of the paper is structured as follows: Section 2 presents our proposed method, before the performance evaluation being presented in Section 3. Finally, the conclusions are summarized, and future work is discussed in Section 4.

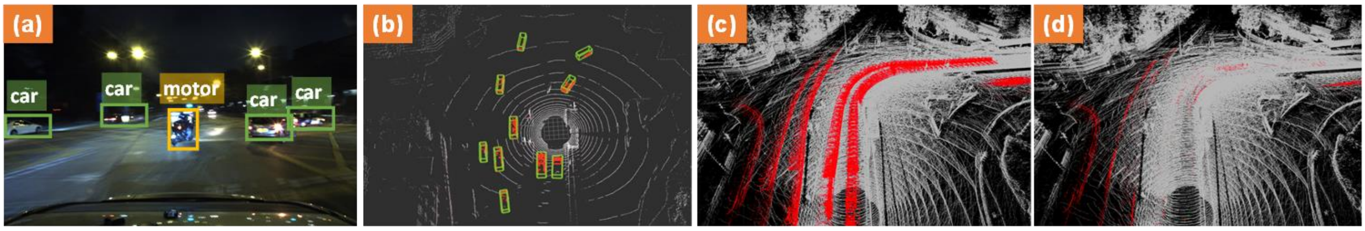


Fig. 1. (a) Numerous dynamic objects in a crossing road captured by the camera; (b) example of object detection using a custom DNN model; (c) The raw point cloud map generated by LiDAR SLAM. Static points are marked in greyscale, while dynamic points are labeled in red; (d) The refined point cloud map using the proposed method.

2. METHODOLOGY

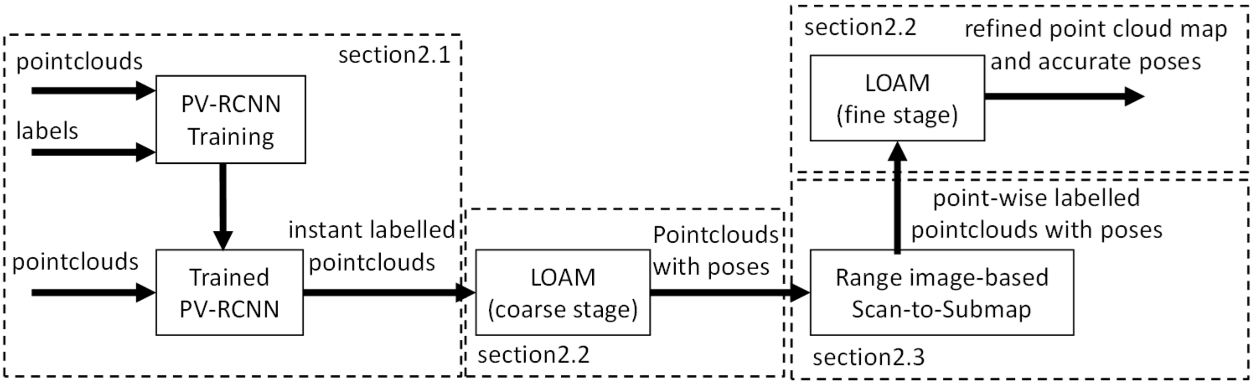


Fig. 2 Proposed pipeline for the proposed coarse-to-fine LiDAR-based SLAM with dynamic object removal

We present a coarse-to-fine LiDAR-Based SLAM with dynamic object removal. First, a 3D DNN network is trained offline to support the detection in urban canyons. Using the dynamic objects removed point clouds enables us to achieve more precise odometry. The range image-based scan-to-submap is conducted to further refine the point clouds. Finally, the refined point cloud map and accurate posed are yielded through LOAM in the fine stage based on the point-wise labeled point cloud and initial guess from the coarse stage. The full pipeline is shown in Fig. 2. We denote the points cloud received at timestamp k as $\mathbf{X}_k$. The i^{th} point $\mathbf{x}_i$, $\mathbf{x}_i \in \mathbf{X}_k$ is denoted as $\mathbf{x}_{(k,i)}$.

2.1 PV-RCNN Training and Detection

PV-RCNN was proposed by Shi et al. [25] to integrate both point-based set abstraction and 3D voxel-based convolutional neural network for point feature learning. The system architecture of the PC-RCVV is presented in Fig.3. Specifically, raw points data are first been divided into small voxels with the resolution of L × W × H. Features of each non-empty voxel are represented by the mean coordinates and intensities of all points inside it. A series of 3 × 3 × 3 3D sparse convolution is used to gradually extract feature volumes with 1x, 2x, 4x, 8x downsampled sizes to learn multi-scale semantic features. After encoded point clouds to 8x downsampled feature volumes, this feature volume is further stacked along the Z-axis to 2D bird's-eye-view (BEV) feature maps to generate 3D proposals following the approaches applied in [14] and [26].

In the voxel set abstraction branch, raw points are first sampled by the Furthest-Point-Sampling (FPS) algorithm to generate a smaller number of keypoints. The surrounding points of each keypoint are regular voxels with multi-scale semantic features encoded by 3D voxel CNN from different levels. These learned voxel-wise feature volumes at multiple neural layers are summarized into a small set of key points via the novel voxel set abstraction module. Finally, the keypoint features are aggregated to the RoI-grid points to learn proposal-specific features for fine-grained proposal refinement and confidence prediction. According to [25], this method gained higher performance than other methods in the KITTI 3D object detection challenge.

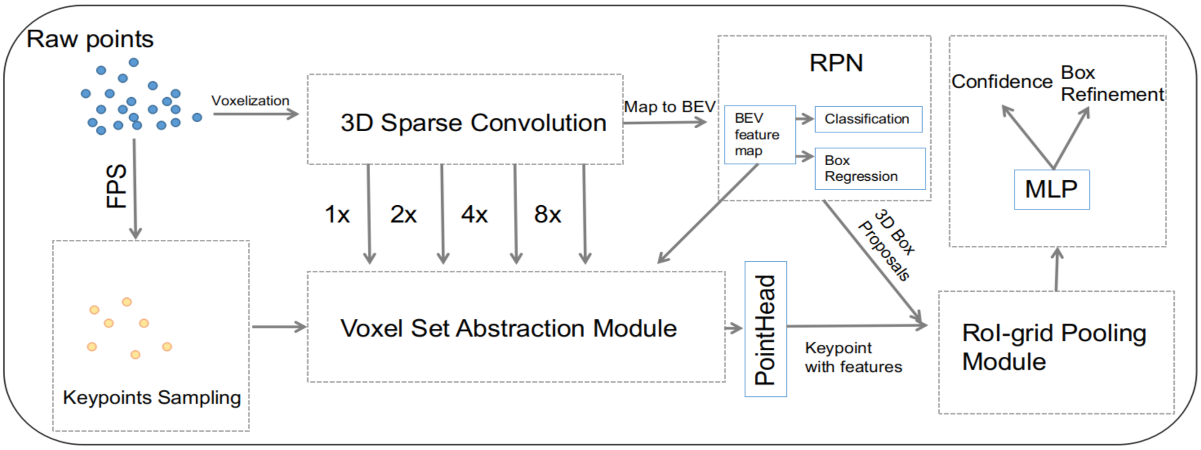


Fig. 3 PV-RCNN Framework [25]

Let $\mathbf{F}_k$ be a set of 3d bounding boxes detected at $\mathbf{X}_k$ with the custom trained PV-RCNN model. The object points $\mathbf{x}_{(k,i)}^{obj}$ are extracted within in the 3D boxes while reminded are classified as clean point $\mathbf{x}_{(k,i)}^{clean}$.

$$\mathbf{x}_{(k,i)} = \begin{cases} \mathbf{x}_{(k,i)}^{obj}, & \text{if } \mathbf{x}_{k,i} \text{is inside any of } \mathbf{F}_k \\ \mathbf{x}_{(k,i)}^{clean}, & \text{classified as clean if outside the boxes} \end{cases} \tag{1}$$

To implement PV-RCNN in customized datasets, we annotated 3D dynamic objects with SUSTechPoints [27], an example is provided in Fig.4 for the annotating process. The FOV is fine-tuned to support 360 degrees. After modifying the data type to suit the network, all data frames are classified into training, validation, and testing datasets to complete the training and testing task.

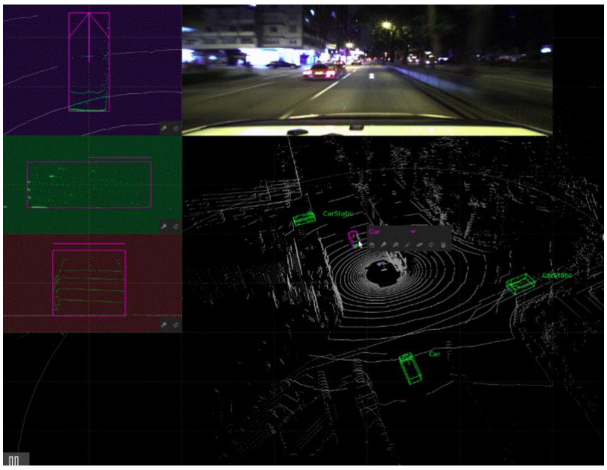


Fig. 4 Annotation using the open-sourced tool

2.2 LOAM

LOAM [3] was first introduced in 2014. It proposes to extract edge and planar features based on the smoothness of a small region near a given feature point. Let m indicates the ring number of point cloud $\mathbf{X}_k$. $\mathbf{S}_{(k,i)}^m$ be a set of continuous neighboring points of $\mathbf{x}_{(k,i)}$ in scan ring m. Normally the points in a ring are in clockwise or counterclockwise order according to the receiving time within a scan period (normally 0.1 seconds). The feature points are extracted according to the curvature c_i of point $\mathbf{x}_{(k,i)}$ and its successive points [3],

$$c_i = \frac{1}{N_{S^*}\left\|\mathbf{x}_{(k,i)}\right\|}\left\|\sum_{j\in\mathbf{S}_{(k,i)}^m, j\neq i}\left(\mathbf{x}_{(k,i)} - \mathbf{x}_{(k,j)}\right)\right\| \tag{2}$$

the $\mathbf{x}_{(k,j)}$ denotes the consecutive point of $\mathbf{x}_{(k,i)}$ within subset $\mathbf{S}_{(k,i)}^m$. N_s represents the number of points in $\mathbf{S}_{(k,i)}^m$, including $\mathbf{x}_{(k,i)}$ and consecutive points. The operator ||·|| indicates the L2 vector norm. An example of feature extraction results is shown in Fig. 5.

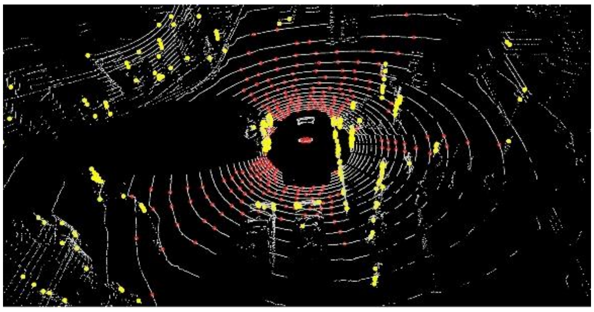


Fig. 5 Example of extracted edge (yellow) and planar points (red) concerning a frame of LiDAR point clouds (grey) in a crossing road using LOAM.

Then two steps of scan matching were performed for the state estimation by minimizing the distance between the edge-to-edge line and planar points-to-planar. The first step matching is to estimate the motion $\mathbf{T}^{L}_{(k,k+1)}$between two successive sweeps (scan-to-scan). The estimated $\mathbf{T}^{L}_{(k,k+1)}$ is used to correct the distortion of points in $\mathbf{X}_{k+1}$ and provide the initial guess to project $\mathbf{X}_{k}$ as $\widetilde{\mathbf{X}}_{k}$. During the next frame, the corresponding features can be found between $\widetilde{\mathbf{X}}_{k}$ and $\mathbf{X}_{k+1}$.

For each edge point $\mathbf{x}^{e}_{(k+1,i)}$, searching its nearest neighbors in $\widetilde{\mathbf{X}}_{k}$ to fit a line by, $\tilde{\mathbf{x}}^{e}_{(k,j)}, \tilde{\mathbf{x}}^{e}_{(k,l)} \in \widetilde{\mathbf{X}}_{k}$, as the corresponding edge. The distance $d^{e}_{(k+1,i)}$ between the edge point $\mathbf{x}^{e}_{(k+1,i)}$ and the fitted line represents residual of edge feature to be minimized, which can be described as,

$$d^{e}_{(k+1,i)} = \frac{\left|(\mathbf{x}^{e}_{(k+1,i)}-\tilde{\mathbf{x}}^{e}_{(k,j)})\times(\mathbf{x}^{e}_{(k+1,i)}-\tilde{\mathbf{x}}^{e}_{(k,\mathrm{l})})\right|}{\left|\tilde{\mathbf{x}}^{e}_{(k,j)}-\tilde{\mathbf{x}}^{e}_{(k,l)}\right|} \tag{3}$$

Similarly, for each plane point $\mathbf{x}^{p}_{(k+1,i)}$ in $\mathbf{X}_{k+1}$, the distance $d^{p}_{(k+1,i)}$ between the point and the fitted plane in $\widetilde{\mathbf{X}}_{k}$, is the residual of plane feature to be minimized, which can be represented as,

$$d^{p}_{(k+1,i)} = \frac{\left|\begin{matrix}\left(\mathrm{x}^{p}_{(k+1,i)}-\tilde{\mathbf{x}}^{p}_{(k,j)}\right)\cdot \\ (\tilde{\mathbf{x}}^{p}_{(k,j)}-\tilde{\mathbf{x}}^{p}_{(k,l)})\times(\tilde{\mathbf{x}}^{p}_{(k,j)}-\tilde{\mathbf{x}}^{p}_{(k,\mathrm{m})})\end{matrix}\right|}{\left|(\tilde{\mathbf{x}}^{p}_{(k,j)}-\tilde{\mathbf{x}}^{p}_{(k,l)})\times(\tilde{\mathbf{x}}^{p}_{(k,j)}-\tilde{\mathbf{x}}^{p}_{(k,m)})\right|} \tag{4}$$

where $\tilde{\mathbf{x}}^{p}_{(k,j)}, \tilde{\mathbf{x}}^{p}_{(k,l)}, \tilde{\mathbf{x}}^{p}_{(k,m)}$ are three nearest points of $\mathrm{x}^{p}_{(k+1,i)}$ among planar points in $\widetilde{\mathbf{X}}_{k}$ using k-d tree search. Then the optimization can be solved using the Levenberg-Marquardt [28] method by minimizing the distance of the features.

The second step matches the current scan and the point cloud map (scan-to-map) to mitigate the error estimation arising from scan-to-scan. According to our previous research [29], this computational load and accuracy of LOAM outperform other methods in highly urbanized areas.

2.3 Range Image-based Scan-to-Submap

Inspired by the work in [24], the dynamic objects are further refined by calculating the discrepancy between the range image of scan and submap within fifty meters. Current scan $\mathbf{X}_{k}$ and the surrounding map $\mathbf{X}^{M}_{k}$ are projected into fix-size range image $\mathbf{I}_{k}$ and $\mathbf{I}^{M}_{k}$, respectively. For a LiDAR sensor that has 40° vertical FOV and 360° horizon FOV, the size of the range image is 360*40 pixels as one pixel represents 1 degree for both horizontal and vertical FOV. An example of the current range image $\mathbf{I}_{k}$ and surrounding $\mathbf{I}^{M}_{k}$ is presented in Fig. 6.

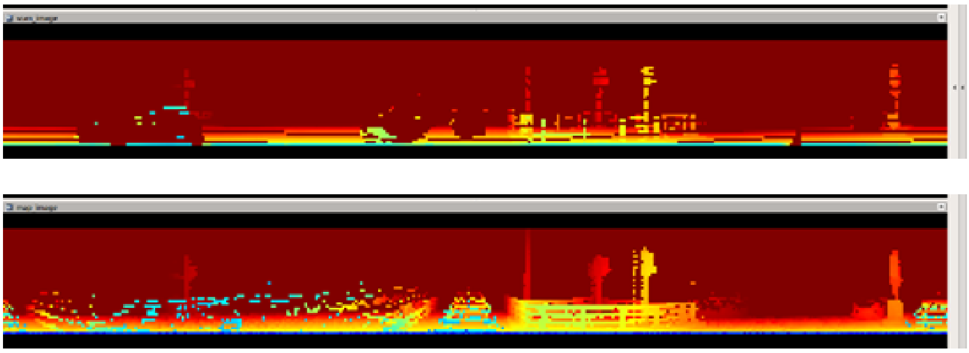


Fig. 6 The range image, the top and down are represents $\mathbf{I}_{k}$, $\mathbf{I}^{M}_{k}$, respectively.

Then the visibility check of the map points is calculated via pixel-wise subtraction between $\mathbf{I}_{k}$ and $\mathbf{I}^{M}_{k}$,

$$\mathbf{I}^{Diff}_{k} = \mathbf{I}_{k} - \mathbf{I}^{M}_{k} \tag{5}$$

Corresponding to Fig.7 (b) of which the point clouds in a crossing road, the range images $\mathbf{I}_k$, $\mathbf{I}_k^M$ and $\mathbf{I}_k^{Diff}$ are demonstrated in Fig.7 (a). The red pixel of $\mathbf{I}_k^{Diff}$ indicates the higher discrepancy between $\mathbf{I}_k$ and $\mathbf{I}_k^M$. Furthermore, the green boxes demonstrate the dynamic vehicles can be classified in red pixels of $\mathbf{I}_k^{Diff}$. We assign a point as dynamic on the map if its corresponding pixel (i, j) in $\mathbf{I}_k^{Diff}$ is larger than a certain threshold $\tau^{dynamic}$.

$$\mathbf{X}_k^{M,\, dynamic} = \{\mathbf{x}_{(k,ij)}^{dynamic} \mid \text{associated } \mathbf{I}_{k,ij}^{Diff} > \tau^{dynamic}\} \tag{6}$$

Finally, the scans within $\mathbf{X}_k^{M,\,\mathrm{dynamic}}$ can be further refined by the point-wise dynamic labels. But the method might contain several false positive points like trees and ground which cannot visible correctly by the current scan. Such that we apply a clustering and drivable area validation to filter the actual static points. The refined object points and the corresponding captured image are shown in Fig. 7 (b) and (c), respectively.

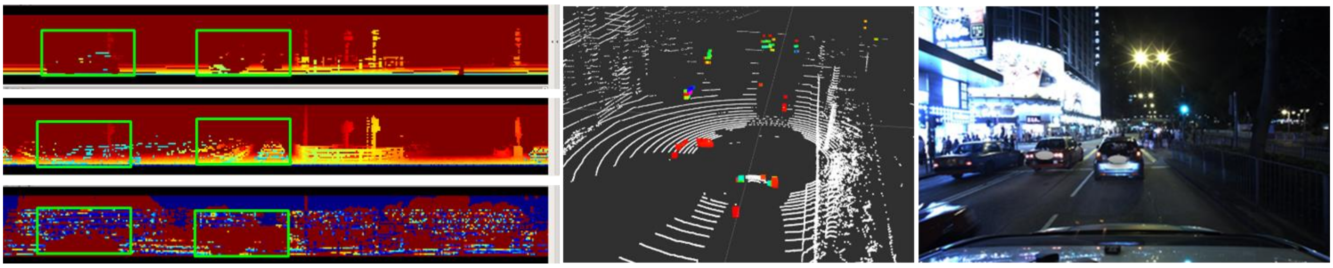

(a) (b) (c)

Fig. 7 (a) The range image, the row from top to down represents $\mathbf{I}_k$, $\mathbf{I}_k^M$, $\mathbf{I}_k^{Diff}$, respectively, the blue color indicates the pixel that is closer and red is far in the color map. Thus, the red pixels in the bottom range images $\mathbf{I}_k^{Diff}$ represent a high discrepancy between scan and submap thus classified as dynamic points; (b) the detected dynamic objects; (c) the image captured of the corresponding scenario.

3. PERFORMANCE ANALYZE RESULT

The performances of the proposed method are evaluated using our recently published UrbanNav dataset [30] which contains the data collected from various degrees of urban areas in Hong Kong and Tokyo. The dataset contains measurements from GNSS, IMU, camera, and LiDAR. Besides, the ground truth data is recorded by NovAtel SPANCPT, which integrates GNSS RTK with fiber optics gyroscope level of IMU.

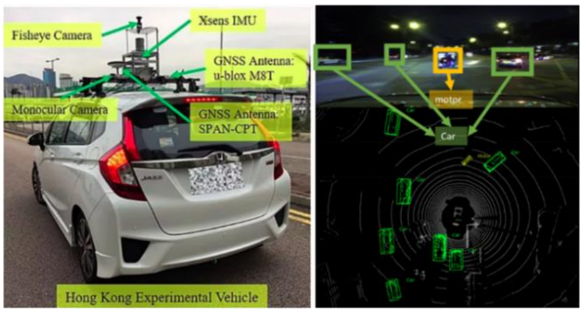


Fig. 8. Left: the sensors and vehicles for the data collection. Right: Variety of dynamic vehicles in HK-Data20190428.

3.1 PV-RCNN Object Training and Verification Results

The evaluation results are shown in Table 1. In this experiment, 486 annotated frames are separated into 292 frames are utilized as the training set, 97 frames are performed as the validation and testing sets, respectively. LiDAR scans and ground truth labels of the training set are taken as the input to the network for training. Bev is calculated only at the bird-eye-view, which losses the precision of the Z-axis. 3D is calculated at 3D space, which is a comprehensive index of evaluating the performance. Intersection over Union (IoU) means the overlapping level of prediction $\mathrm{A_{prediction}}$ and ground truth $\mathrm{A_{ground\ truth}}$ bounding boxes. The trained network is evaluated by the average precision (AP) [31] and recall at IoU threshold 0.5, namely AP_0.5 and recall_0.5, respectively.

$$\mathrm{IoU} = \frac{\mathrm{A_{prediction}} \cap \mathrm{A_{ground\ truth}}}{\mathrm{A_{prediction}} \cup \mathrm{A_{ground\ truth}}} \tag{7}$$

$$\mathrm{recall\_0.5} = \frac{\mathrm{Num_{prediction_{(IoU>0.5)}}}}{\mathrm{Num_{ground\ truth}}} \tag{8}$$

The fine-tuned PV-RCNN achieves 79.02% of AP_0.5 and 87.9% of recall_0.5 for the validation data in terms of dynamic object removal of dataset HK-Data20190428. Most ground truth objects can be successfully predicted with satisfactory accuracy from Table 1. Examples of the detection are shown in Fig. 9. We focus on the detection of the moving car in this paper and more types of dynamic objects will be included in future work, such as pedestrians, double-decker buses, and trucks.

Table 1. 3D object detection results on the validation set.

Type	AP_0.5
Bev	79.02%
3D	67.35%

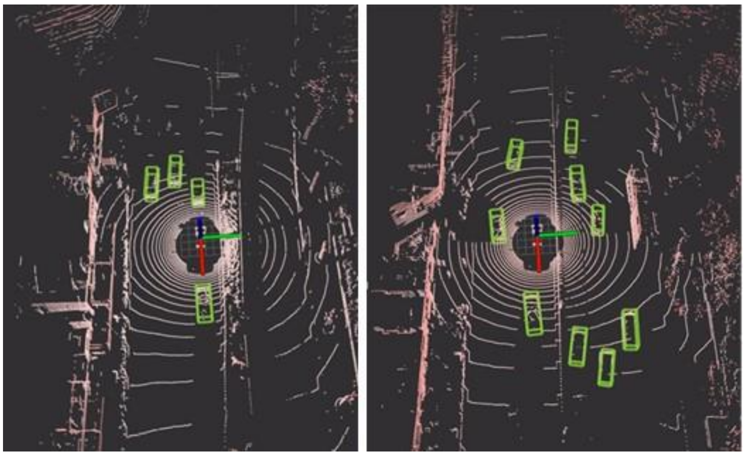


Fig. 9. Demonstration of the detection of the dynamic object using PV-RCNN.

3.2 LiDAR Odometry

To verify the performance of each component and the entire process, we separate the evaluation based on LOAM, LOAM-C, and LOAM-CF respectively.
LOAM: The original LOAM [3]
LOAM-C: The proposed LOAM with the coarse process, the DNN [25] is utilized to detect the dynamic objects.
LOAM-CF: The proposed LOAM with the coarse-to-fine process, the scan is further refined by the discrepancy of scan-to-submap.

We labeled the dynamic objects such as cars and buses in the datasets to explore how the dynamic objects affecting the position error. The density of dynamic objects factor [29] is defined as,

$$c = \left(\frac{N_{car}+N_{bus}}{N_{total}}\right) \times 100\% \tag{9}$$

which N_{car} and N_{bus} represent the number of LiDAR points of cars and buses separately in the current scan. The N_{total} indicates the total number of points in the current frame. The performance of the method listed was evaluated by relative pose error (RPE) via the popular EVO tool [32], a python package that is widely used for evaluating and comparing odometry or SLAM algorithms. The overall results are presented in Table 2 and Fig. 10. Compared to the standard LOAM, 19.1% improvement is achieved by evaluating the RMSE of translation error using the proposed pipeline LOAM-C. The mean error of LOAM-C decreases from 0.321m to 0.258m. Fig. 10 shows that LOAM-C can slightly mitigate the error that occurred by the dynamic points. To further evaluate the proposed method in highly dynamic scenarios of the urban canyon, an epoch-wise evaluation based on the scenarios

(A)(B)(C) of Fig. 10 are conducted to demonstrate the performance of LOAM, LOAM-C, and LOAM-CF, respectively.

Qualitative and quantitative epoch-wise results are presented in Fig. 11 and Table 3, respectively. The DNN network cannot fully detect the vehicle, therefore the discrepancy method based on range image is utilized to further refine the point cloud filtering by DNN labels. For scenarios (A)(C), the LOAM-C outperforms LOAM and LOAM-CF while for scenarios (B), the LOAM-CF achieved more accurate estimation than LOAM-C in terms of translation error and rotation error.

Table 2. Performance comparison between the LOAM and LOAM-C

Dataset	Trajectory Length	Method	Relative Translation Error (m)		Relative Rotation Error (deg)	
			RMSE	Mean	RMSE	Mean
HK-Data20190428	1.21 Km	LOAM LOAM-C	0.450 0.364	0.321 0.258	1.383 1.413	0.799 0.815

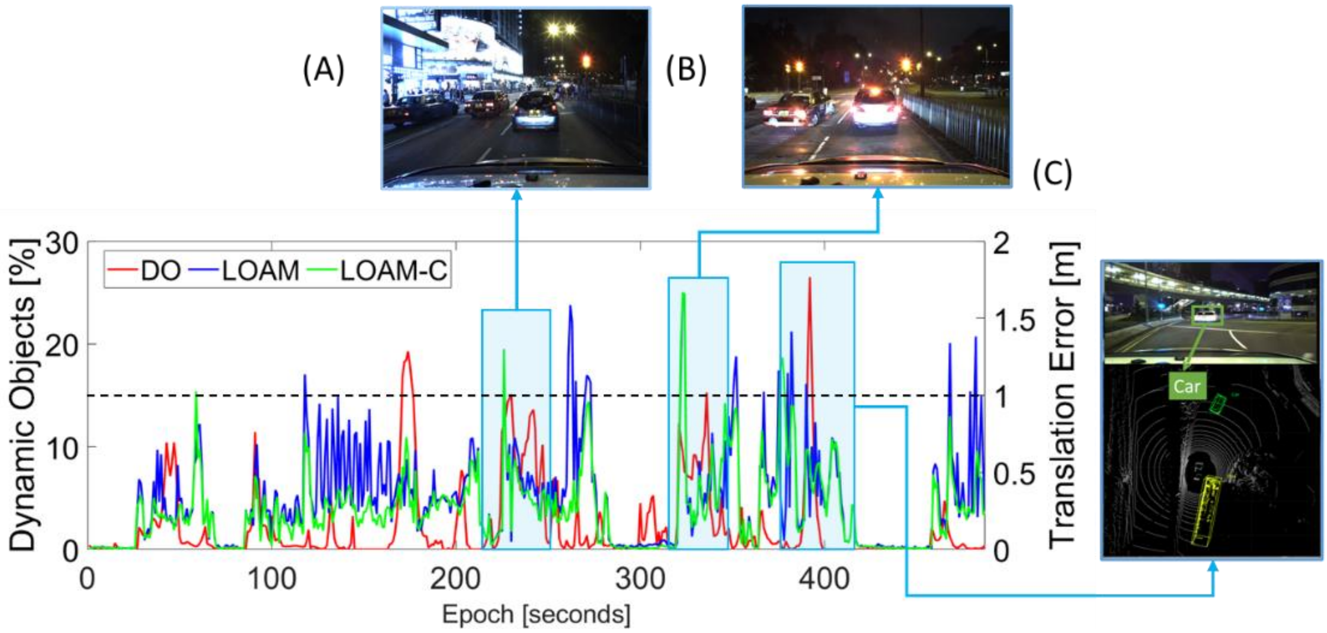
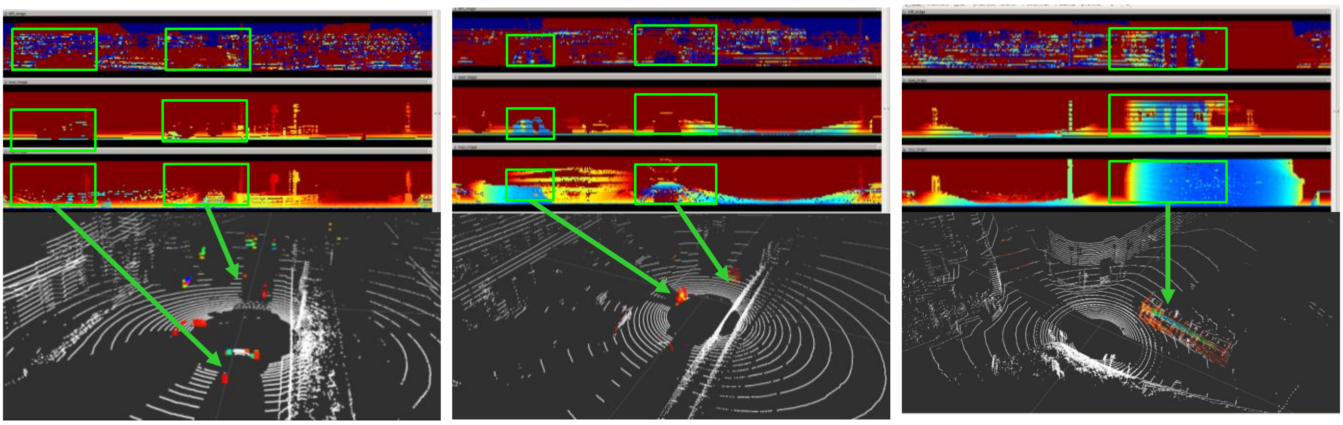


Fig. 10 The comparison of the dynamic objects (DO) versus translation error in LOAM and LOAM-C. (A)(B)(C) are the labels of the scenarios with numerous dynamic objects while the state estimation is degraded.

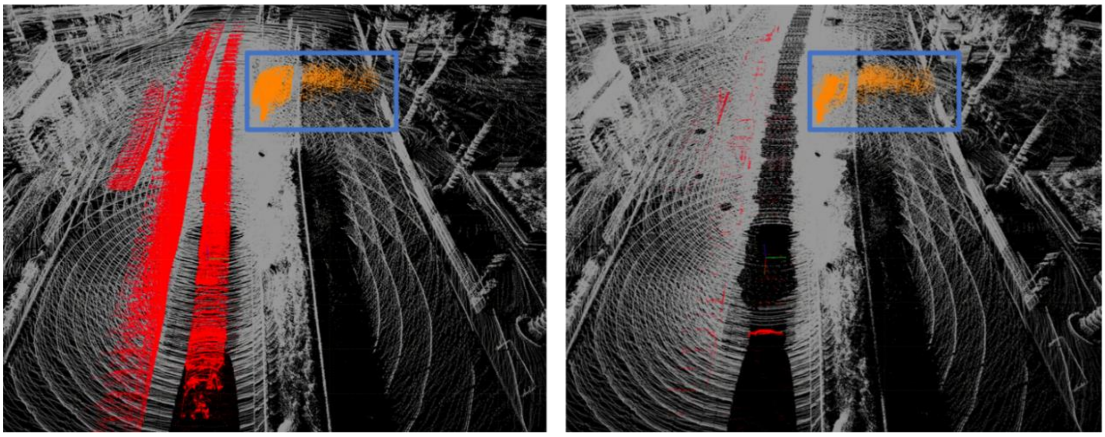


(a) **(b)** **(c)**

Fig. 11 Qualitative evaluation of the refined dynamic objects in scenarios (A)(B)(C) of Fig. 10 using the range image-based scan-to-submap (a) Scenario A: The ego-vehicle resumed as the traffic light turning back to green from red, numerous dynamic vehicles and pedestrians are nearby; (b) Scenario B: The car passes through a crossroad with dense traffic; (c) Scenario C: A moving double-decker bus covers a one-quarter FOV in the intersection.

Table 3. Epoch-wise performance comparison of LOAM, LOAM-C, and LOAM-CF under scenarios labeled in Fig. 10. The top performance of the method in different scenarios is highlighted in bold.

Scenario	Method	2D Relative Translation Error (m)		Relative Rotation Error (deg)	
		RMSE	Mean	RMSE	Mean
A	LOAM	0.415	0.354	0.464	0.374
	LOAM-C	**0.382**	0.337	**0.414**	0.333
	LOAM-CF	0.387	0.341	0.434	0.352
B	LOAM	0.53	0.406	2.597	1.767
	LOAM-C	0.476	0.357	2.39	1.65
	LOAM-CF	**0.475**	0.355	**2.376**	1.619
C	LOAM	0.5112	0.443	3.242	2.241
	LOAM-C	**0.5109**	0.443	3.245	2.233
	LOAM-CF	0.5169	0.446	**3.224**	2.225



(a) **(b)**

Fig. 12 Mapping results of Scenario A. (a)The raw point cloud map generated by LOAM. Static points are marked in greyscale, while dynamic points on the drivable lane are labeled in red and pedestrians are colored yellow; (b) The refined point cloud map using LOAM-CF.

3.3 Mapping Results

Mapping presented in Figs. 1 and 12 were yielding by the LOAM-CF. The refined map with satisfactory results compared to the original map. However, the dynamic pedestrian in Fig. 12 (b) is not removed because the pedestrian is not trained in our DNN model and filtered by the drivable lane in section 2.3. Generally speaking, removing dynamics using the proposed method in the urban canyons is a promising solution for construct a static map for long-term usage in autonomous driving.

4. CONCLUSIONS AND FUTURE WORKS

Dynamic object removal is significant for improving the performance of the LiDAR SLAM in urban canyons. In this paper, we presented a complete coarse-to-fine LiDAR-based pipeline with dynamic object removal and improves odometry by 19.1% in dense urban. In addition, we are able to construct a clearer point cloud map to represent the real world.

In the future, we will study to train a supervised/unsupervised DNN network to support more dynamic object types in urban canyons. Moreover, we will study to reweight the dynamic points for state estimation rather than simply remove the object that might lead to the artificial edge [20, 33] in the SLAM system. The lidar-inertial odometry [34] is another interesting topic to reduce positioning error. Last but not least, we will evaluate the performance under diverse urban areas.

1. Cadena, C., et al., *Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age.* IEEE Transactions on robotics, 2016. **32**(6): p. 1309-1332.
2. Lee, T.N. and A.J. Canciani, *MagSLAM: Aerial simultaneous localization and mapping using Earth's magnetic anomaly field.* NAVIGATION, 2020. **67**(1): p. 95-107.
3. Zhang, J. and S. Singh, *Low-drift and real-time lidar odometry and mapping.* Autonomous Robots, 2017. **41**(2): p. 401-416.
4. Besl, P.J. and N.D. McKay, *A method for registration of 3-D shapes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. **14**(2): p. 239-256.
5. Biber, P. and W. Straßer. *The normal distributions transform: A new approach to laser scan matching*. in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*. 2003. IEEE.
6. Segal, A., D. Haehnel, and S. Thrun. *Generalized-icp*. in *Robotics: science and systems*. 2009. Seattle, WA.
7. Wen, W., 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, 2018. **18**(11): p. 3928.
8. Kan, Y.C., L.T. Hsu, and E. Chung, *Performance Evaluation on Map-based NDT Scan Matching Localization using Simulated Occlusion Datasets.* IEEE Sensors Letters, 2021: p. 1-1.
9. Sun, L., et al., *Recurrent-OctoMap: Learning state-based map refinement for long-term semantic mapping with 3-D-Lidar data.* IEEE Robotics and Automation Letters, 2018. **3**(4): p. 3749-3756.
10. Wen, W., G. Zhang, and L.-T. Hsu. *Exclusion of GNSS NLOS receptions caused by dynamic objects in heavy traffic urban scenarios using real-time 3D point cloud: An approach without 3D maps*. in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 2018. IEEE.
11. Bescos, B., et al., *DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes.* IEEE Robotics and Automation Letters, 2018. **3**(4): p. 4076-4083.
12. Tan, W., et al. *Robust monocular SLAM in dynamic environments*. in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2013. IEEE.
13. Wen, W., G. Zhang, and L.-T. Hsu, *GNSS NLOS Exclusion Based on Dynamic Object Detection Using LiDAR Point Cloud.* IEEE Transactions on Intelligent Transportation Systems, 2019.
14. Lang, A.H., et al. *Pointpillars: Fast encoders for object detection from point clouds*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
15. Xu, C., et al. *Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation*. in *European Conference on Computer Vision*. 2020. Springer.
16. Geiger, A., et al., *Vision meets robotics: The kitti dataset.* The International Journal of Robotics Research, 2013. **32**(11): p. 1231-1237.
17. Li, Q., et al. *LO-Net: Deep Real-Time Lidar Odometry*. in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
18. Chen, X., et al. *Suma++: Efficient lidar-based semantic slam*. in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019. IEEE.
19. Milioto, A., et al. *Rangenet++: Fast and accurate lidar semantic segmentation*. in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019. IEEE.
20. Pfreundschuh, P., et al., *Dynamic Object Aware LiDAR SLAM based on Automatic Generation of Training Data*, in *IEEE International Conference on Robotics and Automation (ICRA)*. 2021.
21. Qin, R., J. Tian, and P. Reinartz, *3D change detection–approaches and applications.* ISPRS Journal of Photogrammetry and Remote Sensing, 2016. **122**: p. 41-56.
22. Yoon, D., T. Tang, and T. Barfoot. *Mapless online detection of dynamic objects in 3d lidar*. in *2019 16th Conference on Computer and Robot Vision (CRV)*. 2019. IEEE.
23. Schauer, J. and A. Nüchter, *The peopleremover—removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid.* IEEE robotics and automation letters, 2018. **3**(3): p. 1679-1686.
24. Kim, G. and A. Kim. *Remove, then Revert: Static Point cloud Map Construction using Multiresolution Range Images*. in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2020. IEEE/RSJ.
25. Shi, S., et al. *Pv-rcnn: Point-voxel feature set abstraction for 3d object detection*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
26. Yan, Y., Y. Mao, and B. Li, *Second: Sparsely embedded convolutional detection.* Sensors, 2018. **18**(10): p. 3337.
27. Li, E., et al. *SUSTech POINTS: A Portable 3D Point Cloud Interactive Annotation Platform System*. in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE.
28. Moré, J.J., *The Levenberg-Marquardt algorithm: implementation and theory*, in *Numerical analysis*. 1978, Springer. p. 105-116.
29. Huang, F., et al., *Point wise or Feature wise? Benchmark Comparison of Public Available LiDAR Odometry Algorithms in Urban Canyons.* IEEE Intelligent Transportation Systems Magazine (accepted), 2021.

30. Hsu, L.T., et al., *UrbanNav:An open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas (Accepted)*, in *In Proceedings of the ION GNSS+ 2021*. 2021: St. Louis, MO, USA.

31. Everingham, M., et al., *The pascal visual object classes (voc) challenge.* International journal of computer vision, 2010. **88**(2): p. 303-338.

32. Grupp, M., *evo: Python package for the evaluation of odometry and slam.* Note: https://github.com/MichaelGrupp/evo Cited by: Table, 2017. **7**.

33. Bai, X., et al. *Perception-aided Visual-Inertial Integrated Positioning in Dynamic Urban Areas*. in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 2020. IEEE.

34. Zhang, J., et al., *Coarse-to-Fine Loosely-Coupled LiDAR-Inertial Odometry for Urban Positioning and Mapping.* Remote Sensing, 2021. **13**(12): p. 2371.