ISPRS Journal of Photogrammetry and Remote Sensing 159 (2020) 90–100

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

A Frustum-based probabilistic framework for 3D object detection by fusion of LiDAR and camera data

Zheng Gong[a,b], Haojia Lin[a], Dedong Zhang[b], Zhipeng Luo[a], John Zelek[b], Yiping Chen[a], Abdul Nurunnabi[c], Cheng Wang[a], Jonathan Li[a,b,*]

[a] Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China
[b] Departments of Systems Design Engineering & Geography and Environmental Management, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
[c] SLG, Department of Statistics, University of Rajshahi, Rajshahi 6205, Bangladesh

ARTICLE INFO

Keywords:
3D object detection
CNN
Deep learning
LiDAR point clouds
MLS
SLAM

ABSTRACT

This paper presents a real-time 3D object detector based on LiDAR based Simultaneous Localization and Mapping (LiDAR-SLAM). The 3D point clouds acquired by mobile LiDAR systems, within the environment of buildings, are usually highly sparse, irregularly distributed, and often contain occlusion and structural ambiguity. Existing 3D object detection methods based on Convolutional Neural Networks (CNNs) rely heavily on both the stability of the 3D features and a large amount of labelling. A key challenge is efficient detection of 3D objects in point clouds of large-scale building environments without pre-training the 3D CNN model. To project image-based object detection results and LiDAR-SLAM results onto a 3D probability map, we combine visual and range information into a frustum-based probabilistic framework. As such, we solve the sparse and noise problem in LiDAR-SLAM data, in which any point cloud descriptor can hardly be applied. The 3D object detection results, obtained using both backpack LiDAR dataset and the well-known KITTI Vision Benchmark Suite, show that our method outperforms the state-of-the-art methods for object localization and bounding box estimation.

1. Introduction

Mobile Laser Scanning (MLS) such as Backpack Laser Scanning (BLS) systems are increasingly being used for Autonomous Vehicles (AVs) perception or indoor mapping (Broggi et al., 2013; Schreiber et al., 2013; Seo et al.,2015). MLS or BLS which usually contain multiple sensors, including Light Detection and Ranging (LiDAR) or laser scanners, optical cameras, an integrated GNSS (Global Navigation Satellite Systems), and IMU (Inertial Measurement Unit) positioning and orientation system (POS), rapidly collect almost accurate 3D point clouds and images over large areas. Effective processing and analyzing this huge amount of point clouds is a very challenging task and has been considered as an active research area. A large number of algorithms for point cloud processing have been proposed in recent years, such as for registration (Zai et al., 2017), segmentation (Trevor et al., 2013; Nurunnabi et al., 2016; Wang et al., 2018), recognition (Xie et al., 2018; Luo et al., 2019), object detection (Qi et al., 2017a) and extraction (Zai et al., 2018; Ma et al., 2018), and semantic information processing. Semantic information of 3D objects is not only necessary for generating building High-Definition (HD) maps of both outdoor and indoor environments, but also an important basis for navigation of AVs and Augmented and Virtual Reality (AR and VR) applications. Intelligently perceiving the surrounding environment and acquiring 3D object information (object's position, texture, pose and size) can effectively bring further strategic guidance to smart devices (AVs, personal smart assistants, intelligent robots, etc.). We propose a 3D object detection method through the fusion of multi-sensor (image and point cloud) data collected by MLS and BLS (see Fig. 2).

We identify three major challenges of 3D object detection that can be summarized as follows.

First, presence of occlusions: MLS/BLS collects point clouds that are usually incomplete due to occlusions generated by obstacles in complex built environment, see Fig. 1(a). Most of the existing 3D feature-based methods assume that point clouds acquired by MLS/BLS are consistent and complete, which is not always true and therefore do not perform well in the presence of occlusions.

Second, existence of structural ambiguity: 3D point clouds commonly contain structural ambiguity and are usually unorganized,

* Corresponding author at: Departments of Systems Design Engineering & Geography and Environmental Management, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

E-mail addresses: gongzheng@stu.xmu.edu.cn (Z. Gong), hlin@stu.xmu.edu.cn (H. Lin), jzelek@uwaterloo.ca (J. Zelek), chenyiping@xmu.edu.cn (Y. Chen), cwang@xmu.edu.cn (C. Wang), junli@xmu.edu.cn, junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.isprsjprs.2019.10.015
Received 30 March 2019; Received in revised form 24 August 2019; Accepted 28 October 2019
Available online 22 November 2019

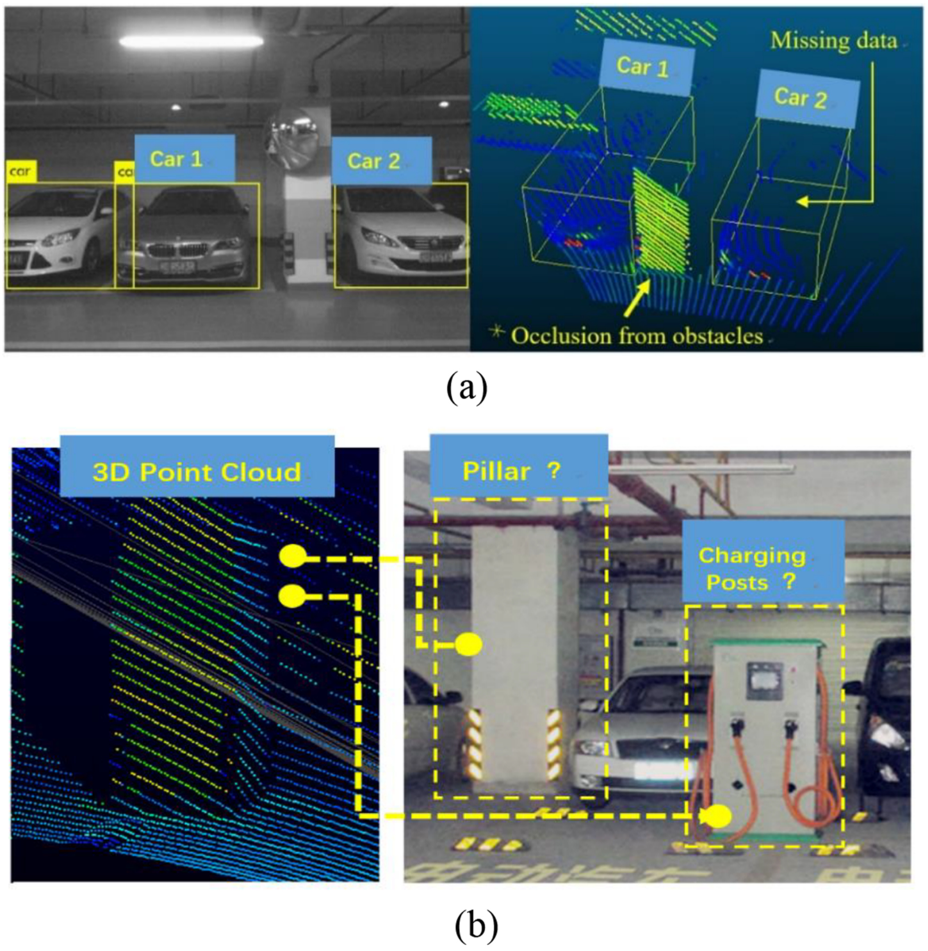

Fig. 1. Main challenges of 3D object detection in point clouds: (a) incomplete 3D objects, (b) structural ambiguity between point cloud and 3D objects.

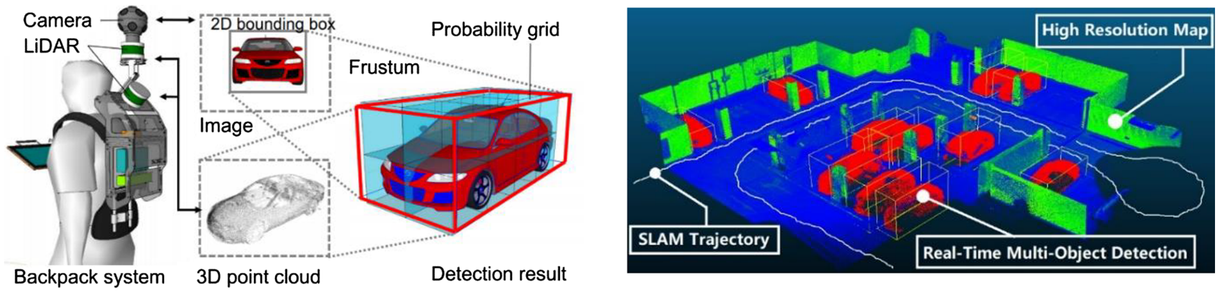

Fig. 2. Pipeline and 3D object detection results.

noisy, sparse, and inconsistent. Structural ambiguity refers to different objects having same structural features, such as pillar-like objects (e.g., pillars or charging posts) in an indoor environment, [see Fig. 1(b)]. This problem of structural ambiguity often makes the existing structure-feature-based methods ineffective.

Third, methods used in object detection: deep learning-based methods require a large amount of manual labeling. For specific 3D object detection, a huge training set must be prepared. In addition, to add an object type, the deep neural networks need retrained, which may result in decreased performance of the existing deep neural networks. In practical applications, the requirement for large training data sets is a bottleneck for deep learning-based methods.

Traditional methods for object detection are generally based on hand-crafted descriptors. Some pioneering works include: Spin Image (SI) (Johnson and Hebert, 1999), 3D shape context (3DSC) (Frome et al., 2004) and its improved versions (Tombari et al., 2010; Sukno et al., 2013; Dong et al., 2017), Fast Point Feature Histograms (FPFH) descriptor (Rusu et al., 2009), Signature of Histograms of Orientations (SHOT) (Tombari et al., 2010), and Rotational Projection Statistics (RoPS) (Guo et al., 2013). However, most of such hand-crafted descriptors suffer from poor descriptiveness and low accuracy, because they can catch only a partial geometric structure of a 3D object.

Very recently, machine learning has shown great progress in environmental perception tasks, such as 2/3D object detection, recognition and tracking (Ren et al., 2015; Redmon et al., 2016). With the development of LiDAR and laser scanning systems for the acquisition of high-density 3D point clouds, deep learning methods (e.g., CNN) have been applied to improve 3D object detection (Chen et al., 2015; Qi et al., 2017b; Qi et al., 2017a; Ma et al., 2019).

Most of the existing approaches for automated 3D object detection in indoor or outdoor scene are not satisfactory. The visual perception based online detector is a matured research topic in computer vision and has been commonly used for robust detection of 2D objects (Liu et al., 2016; Redmon et al., 2016; Redmon and Farhadi, 2017). But, simply relying on a 2D image to detect a 3D object and for obtaining high-precision depth information is considerably limited. The point clouds generated from 2D images by the visual-SLAM methods (e.g., Mur-Artal et al., 2015; Engel et al., 2017; Forster et al., 2014) are relatively sparse and have lack of necessary 3D features. The distribution of sparse point clouds on a small or distant object affects the accuracy of object detection. Moreover, LiDAR derived point clouds having lack of texture, which may cause structural ambiguity problems. Rather, LiDAR-based SLAM algorithms (e.g., Nuchter et al., 2007; Zhang and Singh, 2014; Hess et al., 2016) not only can accurately describe 3D structures, but also produce accurate maps and trajectories. In this study, we develop a multi-sensor (LiDAR and digital camera) 3D object detector for MLS/BLS datasets. Our approach, which falls under a probabilistic framework in a SLAM environment, takes advantage of LiDAR-based SLAM and 2D object detection to solve the problem of object detection in a 3D environment. Combining LiDAR-based SLAM with the characteristics of object detection in 2D images is advantageous for detecting small and distant objects. It reduces interference from noise and errors when detecting objects in 2D images, and narrows down the search space for specific 3D targets. Based on the above idea, we propose a Simultaneous Dynamic Triangulation Mapping (SDTM) framework. First, we use high-precision LiDAR point clouds in LiDAR-SLAM algorithm to restore the relative relationships between objects in 3D space and a global map. Second, we apply the SDTM probabilistic fusion model to estimate the location and point cloud distribution of 3D objects. Thereby, the box enclosing the specific target is estimated using the proposed PCA (principal component analysis) based method. Finally, in the fine-tuning step, a real-time Visual-LiDAR SLAM and object detection framework is implemented in wide dynamic and highly noisy SLAM environment.

The main contributions of this paper can be summarized as follows:

First, we develop a SDTM framework that combines visual and LiDAR-based SLAM data (e.g., point clouds, and trajectories) and uses contextual information. This framework can more effectively updates and dynamically estimates the location and shape of 3D objects in a highly noisy environment.

Second, we introduce a descriptor-free method (G-PO) for 3D bounding box estimation. This method combines a PCA-based rough oriented estimator and a generalized- Iterative Closest Point (g-ICP) registration method for region proposal and fine-tuning.

Third, we develop an experimental verification mechanism and 3D hybrid-SLAM-based (context sequence) object detection method applied to the point clouds acquired by our self-designed BLS system.

The rest of this paper is organized as follows: Section 2 provides a literature review of 3D object detection. Section 3 contains details of the proposed method. Section 4 presents experiments, results and discussion. Section 5 concludes the paper.

2. Related work

2D Object Detection: Traditionally, 2D object detection is based on manually designed descriptor features. Viola et al. (2001) used the Viola-Jones detector to detect sliding windows and used multi-scale Haar features to match similarities. Dalal and Triggs (2005) used the histogram of an oriented gradient as a detector feature, which plays a huge role in pedestrian detection. Felzenszwalb et al. (2008) proposed the Deformable Part-based Model (DPM), which uses hand-selected features and divide and conquer rule to detect targets, thereby greatly improved detection speed and accuracy. Although, in traditional works results are acceptable, descriptiveness in the existing methods is still not satisfactory. In deep learning, the R-CNN (Regions with Convolution Neural Networks) and Fast R-CNN (Fast Region-based Convolutional Networks) target detection framework proposed by Girshick et al. (2014). Ren et al. (2015) further improved the mean average precision (mAP), but the framework has low computational efficiency due to the net structure. To solve the above problems, Liu et al. (2016) proposed the Single Shot Detector (SSD), Redmon et al. (2016) proposed the You Only Look Once (YOLO) system, and Redmon and Farhadi (2017) proposed an integrated deep neural network, YOLO-v3, based on R-CNN, which improves the efficiency of the framework and accelerates the calculation. In our work, we used YOLO-v3 to detect 2D objects.

3D Object Detection in Point Clouds: For detecting 3D objects in point clouds, Munoz et al. (2009) proposed a functional gradient algorithm to learn an Associative Markov Network (AMN) model that was introduced by Taskar et al. (2004) for 3D point cloud semantic segmentation. Based on R-CNN, Gupta et al. (2014) detected and segmented 3D objects from different perspectives. Chen et al. (2015) proposed a new method, which provides energy equations for the different characteristics of 3D targets and optimizes the equations to obtain segmentation results. Song and Xiao (2016) proposed a Deep Sliding Shapes method based on a 3D CNN network and presented a multi-scale 3D Region Proposal Network (RPN) framework that uses multiple scales to detects small targets. The above two methods have a high detection rate in an ideal environment, but no one is good enough in the presence of noise and occlusion. Deng and Latecki (2017) discussed this problem and extracted the appropriate expression from the RGB-D data and, solved the problem to some extent based on the calculation of the 3D box proposals and Fast R-CNN regression. Qi et al. (2017a,b) presented the PointNet model, which defines the state-of-the-art for 3D shape analysis.

2D and 3D Joint Estimation: Because of the characterization of a point cloud after SLAM, the 3D feature descriptor hardly works well in sparse, noisy, and obscured data. Identifying targets from noisy 3D data easily leads to failure. Contrary to the methods used in 3D noisy point cloud data, 2D object detection methods based on CNN model perform better in 2D object detection. Because, the CNN model used in the 2D methods, not only well-designed and pre-trained with ImageNet, but also the image data used in these methods are denser and complete (Song and Xiao (2016)). Therefore, combing 2D images with 3D data has become popular to detect 3D objects. Chen et al. (2017) developed a Multi-View 3D networks (MV3D) based frame work to handle multi-view/sensor fusion and object detection. Ku et al. (2017) used LiDAR point clouds and RGB images to generate features. The authors combined different patterns of features to generate 3D proposals. Then these features were applied to a RPN and a second stage target detection network. Next, they predicted the range and orientation, and finally the objects were classified in 3D space. In another study, similar to our work, a frustum-based method (Qi et al., 2017a) was proposed. This method first uses a frustum to obtain a relevant single-frame point cloud in perspective field of view and then uses PointNet to segment and extract the point cloud in the frustum search box; as a result, effectively solves the occlusion problem, greatly reduces the large-scale search of the target point cloud and improves calculation efficiency.

3D object detection has evolved from traditional manual feature descriptor design methods to a deep learning-based process. As far as we know, few researchers effectively solved the problem of detecting 3D objects with sparse, noisy and obscured point cloud data, especially in a SLAM environment with high dynamics. Inspired by the frustum-based PointNet of Qi et al. (2017a,b), we design a 3D object position estimation and detection algorithm based on SLAM trajectory and mapping probability. The algorithm uses only one CNN to detect target in 2D images, and uses probability maps to detect objects in 3D point clouds. Also, the algorithm estimates target positions and correlations in 3D space, which significantly increases the stability of the algorithm and reduces computational complexity.

3. Proposed method

The new method is comprised of three main steps: (1) localize a 3D object, (2) estimate a 3D bounding box, and (3) fine-tune. The problem

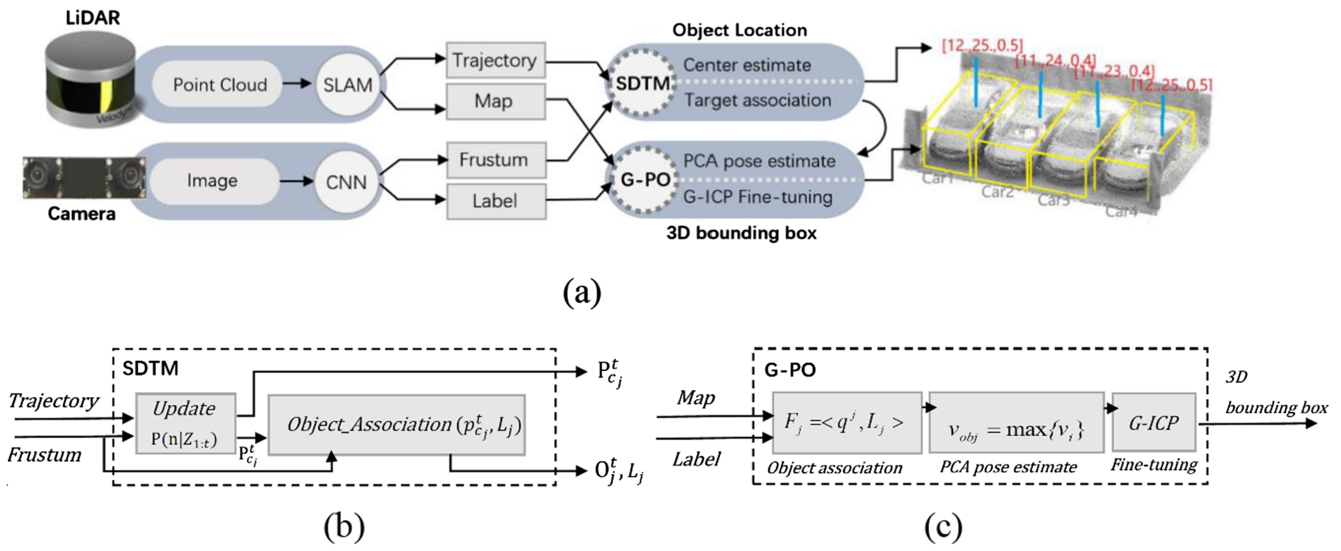

Fig. 3. (a) Diagram of the proposed framework. (b) SDTM module for object location and (c) G-PO module for 3D bounding box estimation and fine-tuning.

is defined as follows:

3.1. Problem Definition

Given LiDAR point cloud and image data as input, our goal is to localize and detect objects in 3D space. We obtain a 3D frustum from the 2D image region through the known projection matrix. We also estimate high-precision trajectories and establish 3D maps by using 3D SLAM, for example, Cartographer (Hess et al., 2016), and LOAM (LiDAR Odometry and Mapping) (Zhang and Singh, 2014). Each 3D object is expressed by a labeled point cloud, $obj(x_i, y_i, z_i, l)$ and an Amodal (Breckon and Fisher, 2005) 3D bounding box $Bb(cx, cy, cz, pl, pu)$, where x_i, y_i, z_i is the point cloud and l is the label of each detected object; cx, cy, cz is the center of the bounding box; pl and pu are the lower front and upper back corners of the box, respectively.

3.2. Main framework

An architectural diagram of our proposed method is given in Fig. 3(a). Given LiDAR and camera data, using a CNN, the framework first generates proposed detected 2D object regions in the image. In our work, we used YOLO-v3 to detect 2D objects. Each 2D region is then projected onto a 3D viewing frustum, resulting in a point cloud from LiDAR. Moreover, our method generates a 3D region probability by SDTM, which uses SLAM trajectory results and frustum point clouds to locate an object. From the above information, G-PO is performed for object pose estimation and pose/position fine-tuning, and finally a 3D bounding box is predicted.

3.3. Object localization in 3D space

The main issue for object detection is target location in wide 3D space. Qi et al. (2017a) proposed a method, which requires only the result of a frustum projection of a detected 2D target to carry out coarse positioning of the target in 3D space, uses PointNet (Qi et al., 2017b) to perform fine-grained segmentation of the point cloud inside the frustum to detect specific target. With a known camera projection matrix, a 2D bounding box can be lifted to a frustum that defines a 3D search space for the object. Two advantages of this method are: (i) reducing search space dramatically, and (ii) solving the occlusion problem well. On the downside, there are two main shortcomings for this method: (i) this method, based on merely a point cloud descriptor to detect an object, can lead to detection failure when processing point clouds with noise and ambiguity, and (ii) this method may become confused when multiple instances appear in the frustum region (Qi et al., 2017a). Considering all the above issues, we propose a novel object location module, SDTM, which retains the above two advantages, at the same time resolves the two disadvantages. That means, SDTM can robustly process point clouds in the presence of noise and ambiguity. More specifically, in a dynamic SLAM scene, SDTM projects the frustum from different views, and to achieve robust positioning of objects in 3D space it uses multi-frame context information and fusion of multi-sensor data.

There are three steps for implementing SDTM: (i) generating the proposed 2D object regions in images via a 2D-CNN, (ii) obtaining a 3D viewed frustum from the proposed 2D regions using frustum projection and transforming this frustum region into global coordinates, and (iii) selecting the candidate points belonging to target objects and calculating the target center based on the candidate points. In the second step, based on the 2D detection results (denoted as P_{2D}) of a 2D-CNN and the local pose (denoted as q_c) of the camera, we calculate the frustum vertex by function Γ:

The j^{th} frustum vertex $<v_c, v_{p1}, v_{p2}, v_{p3}, v_{p4}>$ of v_t^j in the local coordinate system for each frame F_t, $t \in (0{\sim}time_n)$[the depth of frustum D_j is estimate from prior target information and 2-D bounding box's height h_j and fitting parameters ρ, (Eq. (1))]. Then, v_t^j is transformed to V_t^j simultaneously to the global coordinate system through the LiDAR-Camera calibration matrix Θ and SLAM trajectory T_t (Eq. (2)). The above expressions can be defined mathematically as,

$$v_t^j < v_c, v_{p1}, v_{p2}, v_{p3}, v_{p4} > = \Gamma(q_c, p_{2d}, D_j)$$

$$\mathrm{D_j} = \gamma \cdot h_j + \rho \tag{1}$$

$$\mathrm{V_t^j} = \mathrm{T_t} * \Theta * v_t^j,\ t \in (0{\sim}n) \tag{2}$$

The third step is the core of SDTM. A key challenge in this step is to extract candidate object points from foreground or background noise, and then based on these points calculating the target center. We dynamically select or eliminate candidate object points following the probabilistic iteration method from different frustum views during the SLAM (see Fig. 4). Specifically, for each point cloud frame, $p_c (p_c \in \mathrm{V})$, we first transform p_c to an occupied cell structure, i.e., the octomap [in order to maximize the target positioning accuracy and the accuracy of the detection results, we use the maximum octree resolution of the 3D object, which is the same as the SLAM octree map (0.03 m)]. Then for the l^{th} target object, given the sensor measurements $z_{1:T}$, we apply the following formula to calculate the updated probability $\mathrm{P}(\mathrm{n}|z_{1:T})$, of each cell, n, to be occupied:

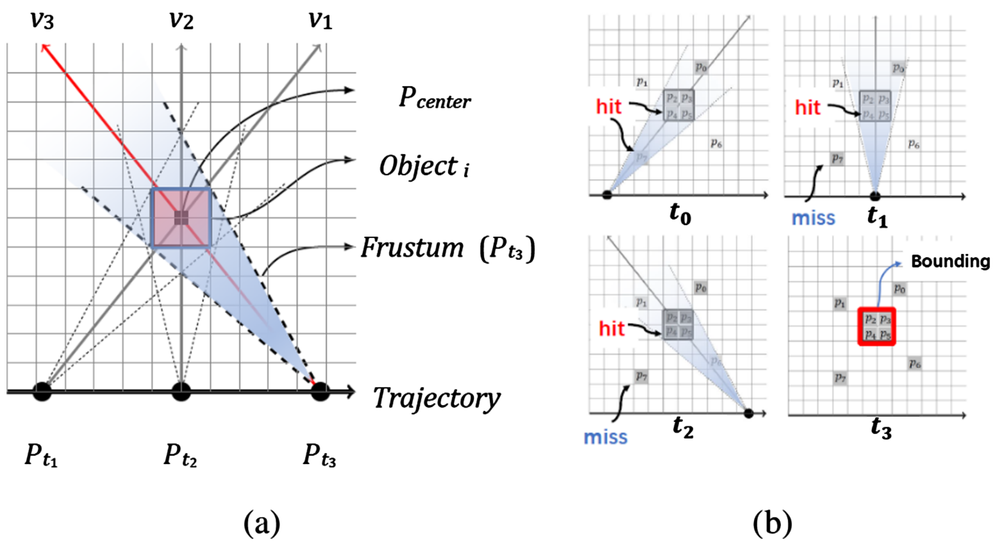

Fig. 4. Illustration of SDTM, (a) $P_{t1} \sim P_{t3}$ are observation points at different views on the SLAM trajectory point at different times, (b) 3D object probability model by updating (hit or miss) each occupied cell's probability and estimating the target center.

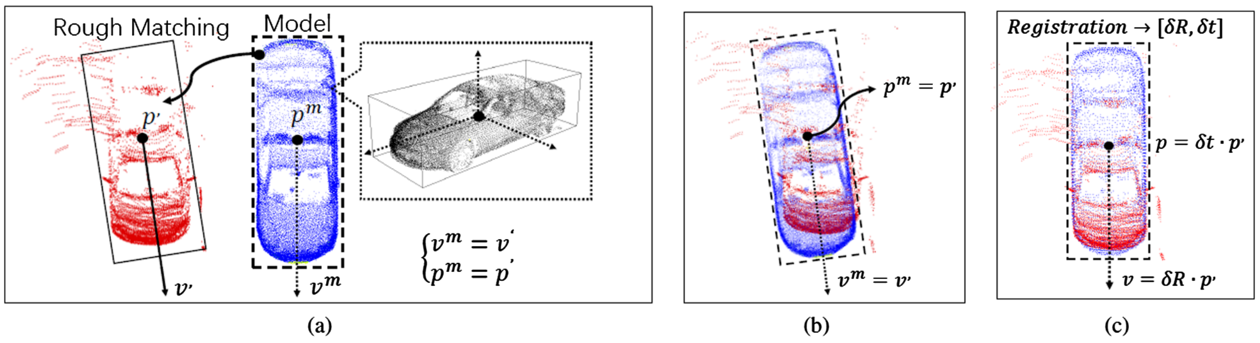

Fig. 5. Illustration of the fine-tuning step: red: object point cloud. blue: model point cloud. (a) Transform the model (set $v^m = v^0$, $p^m = p^0$) close to the object, where pose v^0 and center p^0 is rough estimate by PCA and SDTM, respectively. (b) Rough transformed model, and (c) Fine-tuning the final pose v and center$p$ by g-ICP registration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

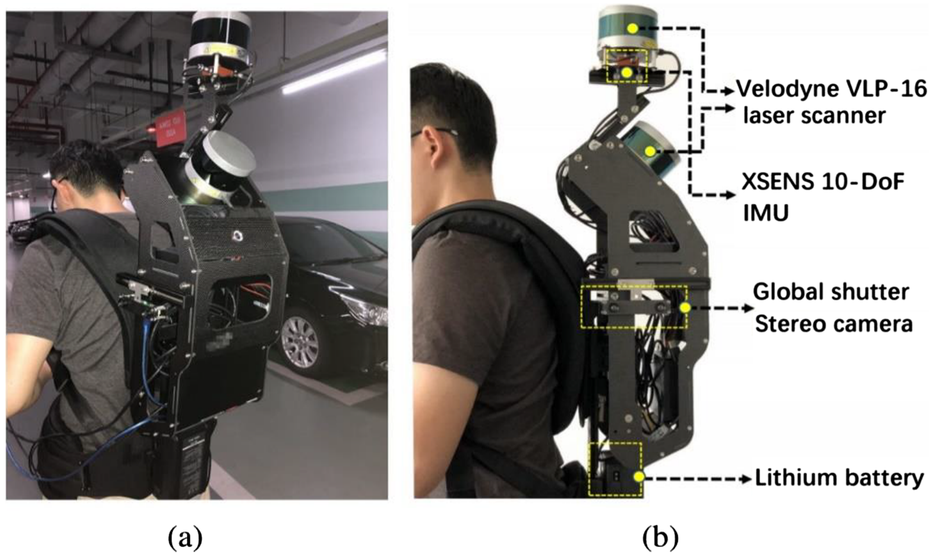

Fig. 6. (a) Experimental environment in an underground parking lot. (b) Our self-design BLS consisting of (1) two Velodyne VLP-16 laser scanners, (2) XSENS 10-DoF IMU and (3) a stereo camera with a global shutter sensor.

Table 1
Real world dataset acquired by our BLS system.

Data-set	Dimensions	No. of Scans (15 Hz)	No. of Images (20 Hz)	IMU (200 Hz)
PL#0	34 × 64 × 6.5	3364	4238	√
PL#1	34 × 64 × 6.5	3091	3897	√
PL#2	25 × 80 × 7.5	1601	2017	√
PL#3	30 × 30 × 7.5	1269	1599	√

$$P(n) = L(n|z^l_{1:T}) = L(n|z^l_{1:T-1}) + L(n|z^l_T) \tag{3}$$

$$where\ L(n) = log(\frac{P(n)}{1-P(n)})$$

$$and\ L(n|z^l_{1:T}) = L(n|z^l_{T,hit}) + L(n|z^l_{T,miss})$$

Here, z^l_T denotes cells that belong to the T^{th} frustum, V, that hits the l^{th} object, $T = \{1, 2, \cdots, k_l\}$; k_l is the number of V's that hit the l^{th} object; n denotes the n^{th} cell; L is the log-odds value calculated by the logic function. Note that $L(n|z^l_{1:T})$actually consists of two parts (Eq. (3)): the first part, $L(n|z^l_{T,hit})$, is the positive value of cells that are hit by V. And the second part, $L(n|z^l_{T,miss})$, is the negative value of the occupied cells that are not within the region of V at the current frame. By a hit/miss operation, we iteratively increase/decrease the probability of object/non-object points (foreground/background noise) occupying a cell. Then, given the threshold, τ, we extract the l^{th} candidate cells, C^l_T, of the target as follows:

Table 2
3D object detection AP_{3D} on the BLS dataset in 0.5/0.7 IoU. AVOD (J. Ku et al., 2017) and F-PointNet (Qi et al., 2017) (previous opensource state-of-the-art) are based on 2D and 3D CNNs.

Method	IoU = 70%			IoU = 50%		
	$AP_{vehicle}$	AP_{pillar}	AP_{QRcode}	$AP_{vehicle}$	AP_{pillar}	AP_{QRcode}
AVOD (%)	58.52	53.03	32.30	78.30	72.52	49.21
F-PointNet(%)	**62.71**	58.45	39.11	**78.80**	75.60	50.72
Ours (%)	61.15	52.33	**66.06**	65.03	73.48	**75.40**
Ours (fine-tuning)	61.97	**67.52**	61.55	76.82	**81.31**	70.04

$$C^l_T = \{c^l_j | P(c^l_j) = L\left(c^l_j | V^l_{1:k_l}\right) > \tau. \tag{4}$$

Next, we estimate the target center, P^j_{center}, through the LiDAR point cloud, p_t, inside the frustum region. To avoid the noise from non-targets, we use an *iterative weighted mean value strategy* to estimate the center point. We calculate the initial candidate center, P'_{center}, from point cloud, p_c, in the frustum region, project p_c onto the octomap, and calculate the occupation probability, f_i. When the next frame appears, we perform a nearest neighbor check for each incoming point to the candidate center point set $P[P^1_t, P^2_t, \cdots, P^j_t]$. If no neighbor is found, then the region is considered as a candidate center point and joins the candidate center point queue; henceforth, this region is considered activated. If a neighbor is found, i.e. when the distance between the new center, P^t_{center}, and the candidate center, P^{t-1}_{center} , is less than the threshold, ϕ, we lock

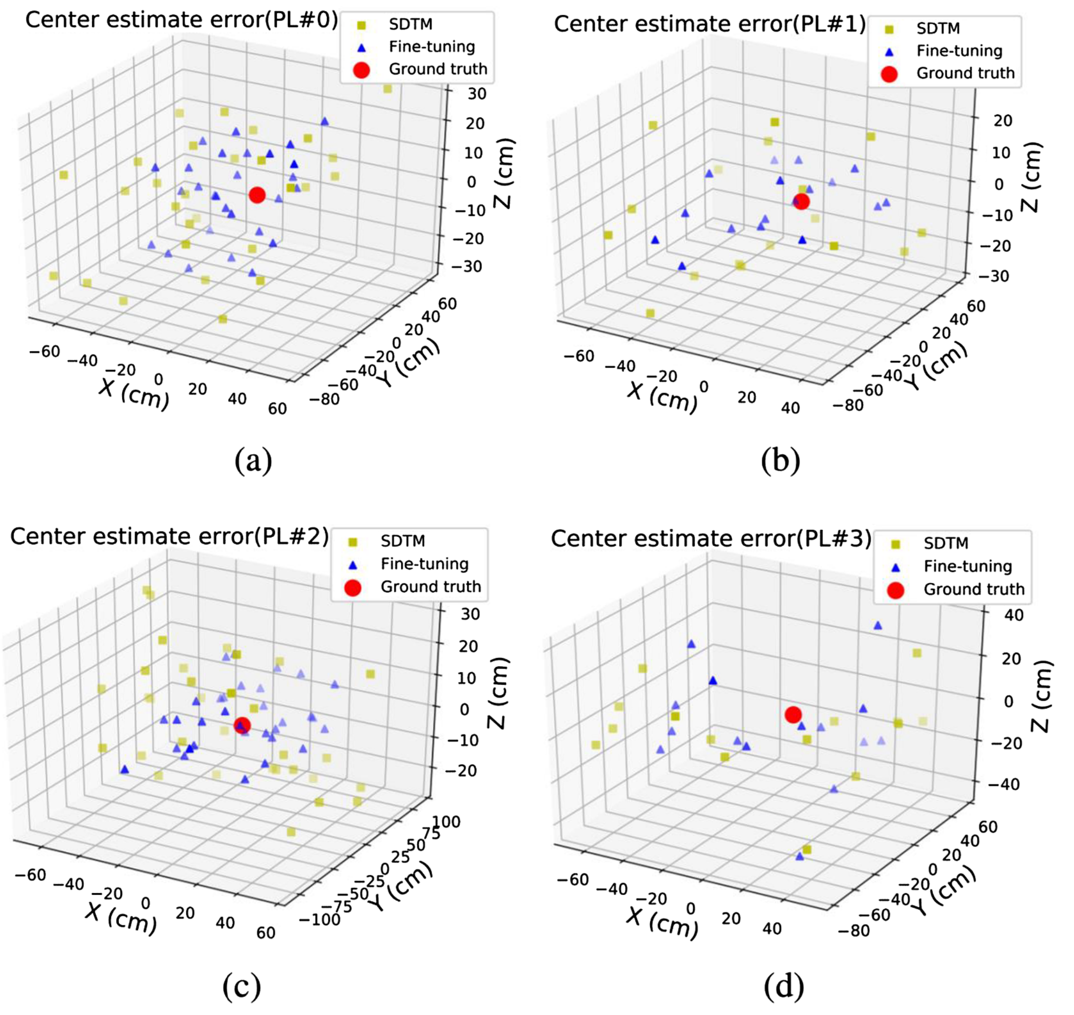

Fig. 7. Visualization of the error distribution of object center points localization in our BLS data-set.

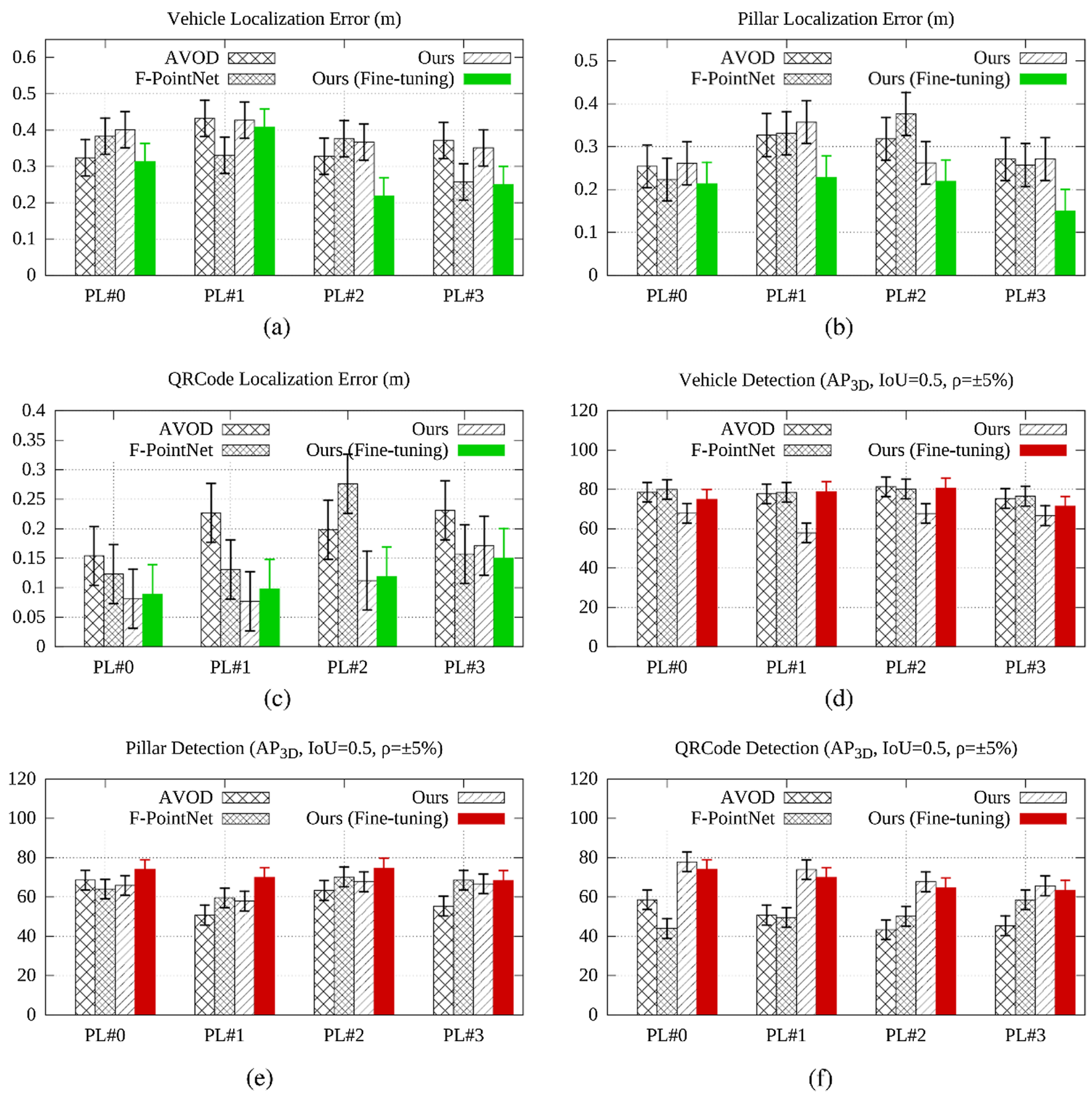

Fig. 8. Comparison of localization errors (a to c), and AP_{3D} scores (d to f) generated by AVOD/F-PointNet and our method on four datasets (PL_0 *to* PL_3).

the target and perform target association and tracking, and iteratively update the center point, P^t_{center}, and region probability by using a weighted mean value strategy (Eq. (5)).

The center of the j^{th} object is calculated as follows:

$$P^t_{center_j} = \frac{1}{n}\cdot\sum_{i=0}^{n}[(\sigma\cdot\frac{1}{d_i}\cdot f_i)\cdot p^t_c]$$
$$= \frac{1}{n}\sum_{i=0}^{n}[(\sigma\cdot\frac{1}{d_i}\cdot[L(n|v^l_{1:T-1}) + L(n|v^l_T)])\cdot p^t_c] \quad (5)$$

where

$$d_i = \sqrt{P^{t-1}_{center_j} - P^t_{center_j}},\ p^t_i \in C^j_t = C^j_{t-1}\cup C^j_t$$

Here, we combine the occupation probability, f_i, and the distance, d_i, between the current point cloud and the old center, $P^{t-1}_{center_j}$, as the weight; σ is a normalization parameter. The proposed algorithm is summarized as follows.

Algorithm 1: $\langle P^t_{Cj}, o^t_j, L_j\rangle$ = $\boldsymbol{SDTM}(V^j_t, q_t, P^{t-1}_{cj})$: Iteratively estimate $\boldsymbol{j^{th}}$ object $\boldsymbol{o^t_j}$'s center $\boldsymbol{P^t_{Cj}}$ at time $\boldsymbol{t}$ Here $\boldsymbol{P^t_{Center_j}}$ is abbreviated as $\boldsymbol{P^t_{Cj}}$

```
Input: $V^j_t$, $q_t$, $P^{t-1}_{cj}$
Output: $P^t_{Cj}$, $o^t_j$, $L_j$
  for j = 0; j ≤ s; j! = s do
    $q^t_j \leftarrow Frustum\_Filter(V^j_t, q_t)$;
    $P^t_{Cj} \leftarrow \frac{1}{n}\bullet\sum_{i=0}^{s} q^t_j$;
  if t = =0 then
    $Push(centerQueue[\ ], P^t_{Cj})$;
  else
    for j = 0; j ≤ s; j! = s do
      if $Distance(centerQueue[\ ], P^t_{Cj}) < \varphi$ then
        $d_i \leftarrow \sqrt{P^{t-1}_{cj} - P^t_{cj}}$;
        $p^t_c \leftarrow p^t_c + p^{t-1}_c$;
        $P^t_{Cj} = \frac{1}{n}\sum_{i=0}^{n}[(\sigma\bullet\frac{1}{d_i}\bullet f_i)\bullet p^t_c]$;
        $updateCenter(centerQueue[j], P^t_{Cj})$;
        $objectAssociation(p^t_c, L_j)$;
      else
        $Push(centerQueue[\ ], P^t_{Cj})$;
```

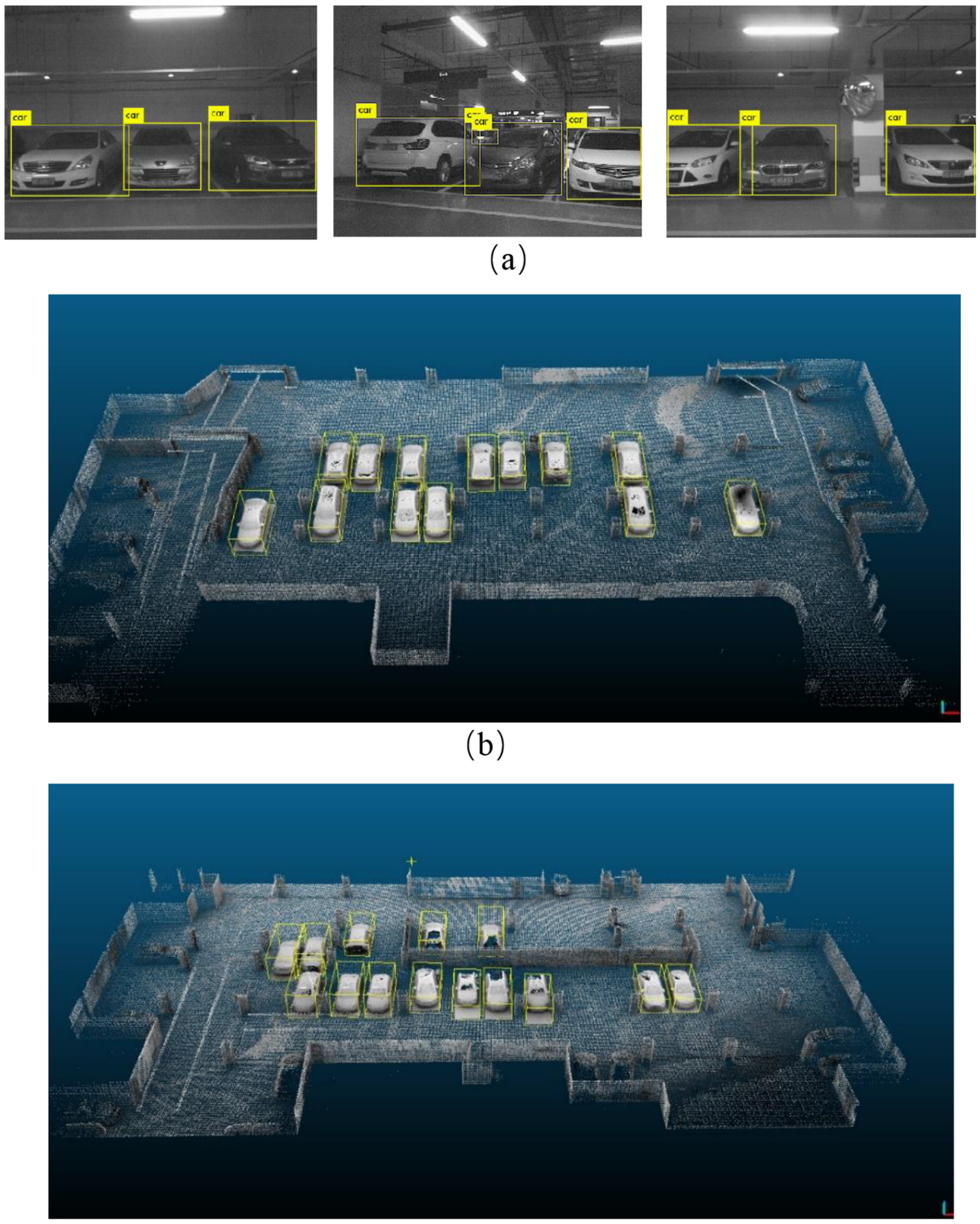

Fig. 9. (a) 2D detection results. (b) and (c) Examples of SLAM and detection results of our method on BLS dataset.

Table 3
3D object detection AP_{3D}, AP_{BEV} on KITTI dataset in 0.5/0.7 IoU. AVOD (J. Ku et al., 2017) and F-PointNet (Qi et al., 2017) algorithms that are based on 2D and 3D CNNs.

Method	Data_0009			Data_0022		
	$AP_{3D}0.7$	$AP_{3D}0.5$	AP_{BEV}	$AP_{3D}0.7$	$AP_{3D}0.5$	AP_{BEV}
AVOD (%)	65.05	73.53	78.82	58.39	**71.52**	78.21
F-PointNet (%)	67.75	**76.45**	80.11	57.22	69.60	74.72
Ours (%)	66.15	72.33	76.06	55.56	67.48	77.40
Ours (fine-tuning)	**68.97**	75.52	**81.55**	**61.12**	71.31	**80.04**

```
return ⟨$P^t_{Cj}$, $o^t_j$, $L_j$⟩
```

This strategy, which naturally filters the far points, assumes that the probable highest occupation point is most likely to be calculated as the center, thereby avoiding the central calculation error caused by foreground/background noise. From the nearest neighbor relationship, the context association can be determined, i.e. the LiDAR point cloud objects in the front and rear frames within a frustum region can be labeled and associated. Algorithm.1 gives the specifics for SDTM as follows: using frustum, V_t^j , point cloud, q_t, and last center, $P^{t-1}_{centerj}$,as input, iteratively estimate the center, $P^t_{center_j}$, of the j^{th} object, o_t^j, at time, t, and then association by L_j.

3.4. 3D bounding box estimation

To estimate the Orientation Bounding Box (OBB) of an object in a scene with high noise, we propose a Generalized-ICP-based (Segal et al., 2009) Probability Orientation (GPO) estimation method, which combines Principal Components Analysis (PCA) (Jolliffe and Cadima, 2016) for rough pose estimation and probabilistic model registration for pose-position fine-tuning. Here the size of the bounding box is set by a priori number.

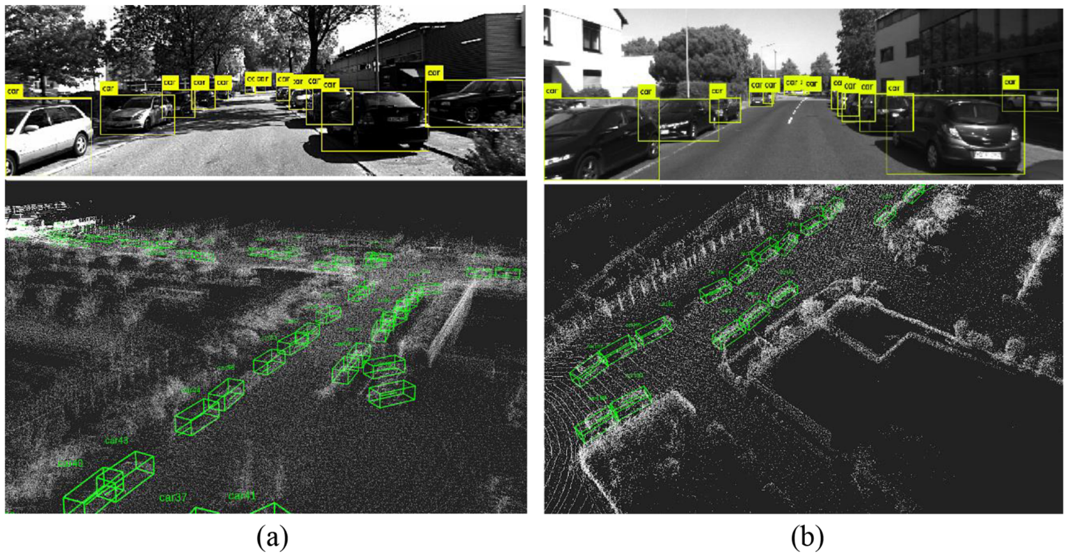

Fig. 10. Example of SLAM and 3D detection results of our method on KITTI benchmark in (a) 2011-09-26-drive-0009 dataset and (b) 2011-09-26-drive-0022 dataset.

(a) **Object association:** Because we use frustums to associate and track objects in 3D space, we can merge and label the point cloud information from each frame of a frustum. Assuming q_t^j is the j^{th}frustum point cloud of each frame, t, according to information in previous section, we obtain the frustum tracking results to associate each point cloud, q_t^j,with 3D object, F_j, using label L_j. Therefore, using the SLAM trajectory, T_t, we merge these associated point clouds into the global coordinate system to obtain the complete target point cloud, F_j, and the corresponding label, L_j, as follows:

$$F_j = < q^j, L_j > = < \sum^{t} (T_t \cdot q_t^j), L_j > . \tag{6}$$

(b) Rough oriented estimate:

Given the central point, P_j^c, and complete point cloud information, F_j, of an object, we apply the classical PCA to roughly estimate the orientation of the 3D box. PCA is a technique that uses an orthogonal transformation (a rotation) to convert a set of observations of possibly correlated variables, into a set of values of linearly uncorrelated variables called principal components. In doing so, PCA accomplishes three things: (i) isolates noise, (ii) eliminates effects of rotation, and (iii) separates out the redundant degrees of freedom. In statistics, the covariance between the two variables measures the degree of the linear relationship between the two variables. Small values indicate that the variables are independent. Eq. (7) describes the rough OBB estimation:

$$F_j \sim \Phi_j = \mathrm{diag}(x_i, y_i, z_i) = Q^{-1}\Phi_j Q$$

$$v_{obj} = \max\{v_i\};\ i \in (1\sim3) \tag{7}$$

Given the covariance matrix, Φ_j, of the target point cloud, F_j, the large diagonal elements correspond to strong signals; the off-diagonal elements indicate the covariances between the variables. Large off-diagonal elements correspond to distortions in the data. In order to minimize distortion, the covariance matrix should be changed, such that the off-diagonal elements are close to zero. Hence, we diagonalize the covariance matrix Φ_j.

Let $V_j$denote the j^{th}column of eigenvector matrix, Q. After diagonalizing Φ_j, the maximum eigenvectors $v_{obj}$of the covariance matrix compose the rough orientation of our OBB.

(c) **Fine-tuning:** Because of foreground and background noise and data-deficiency problems appear in a SLAM scene, poor geometric features easily affect the stability of the above algorithm. To achieve a more accurate and robust estimation, inspired by the g-ICP (Segal et al., 2009) algorithm, we introduce a probabilistic framework (Eq. (9)) to the point cloud and model alignment by using the local point cloud surface variance as the weight of registration, which modifies the PCA estimated error caused by the lack of features in the sparse point clouds.

Specifically, first we transform our object model, m_k,into the neighbor of F_j by the center position, P_{cj}^t, of F_j and rough pose, v_{obj}. After registering the nearest neighbor models, m_k and F_j, we fine-tune the estimated error to further improve the estimated attitude results of the bounding box (Fig. 5). The algorithm is described in detail as follows:

According to the probabilistic model of the g-ICP, we consider the noise of m_k and F_j, which are then generated according to $m_k \sim$ N (m_k, C_k^{mk})and $F_j \sim$ N (F_j, C_j^{Fj}). In this case, $C_k^{mk}$and $C_j^{Fj}$are covariance matrices associated with measured point clouds, $m_k$and F_j. Here, considering a small transform, ΔT_i , between model, m_k,and target point cloud, F_j, and the distribution from which $d_j^T$is drawn, we define d_j^T *as*follows: $d_j^T = m_k - \mathrm{T} * F_j$; $m_k$and F_j are assumed to be drawn from independent gaussians; thus, we have the following:

$$d_j^T \sim \mathcal{N}\left(m_k - (T)F_j,\ C_j^{F_j} + (T)C_k^{m_k}(T)^T\right)$$

$$= \mathcal{N}(0,\ C_j^{F_i}(T)C_k^{m_k}(T)^T). \tag{8}$$

Now MLE is used to iteratively compute ΔT as follows:

$$\Delta T_i = \underset{\Delta T_i}{\arg\min} \sum_{j=1}^{n} d_j^{(\Delta T_i)^T}(C_k^{m_k} + \Delta T_i C_j^{F_j} \Delta T_i^T)^{-1} d_j^{(\Delta T_i)} \tag{9}$$

This defines the key step of the fine-tuning. The nearest twenty points are used as the local neighborhood to calculate the variance of every point.

4. Experiments and discussion

To compare our method with current state-of-the-art methods, experiments were conducted with two different types of datasets: (i) our own datasets were captured by our backpack system in underground parking lots, which contain three target objects, (vehicles, pillars and QRCodes), (ii) KITTI (Geiger et al., 2012) benchmark datasets, which have only vehicles. To do the experiments, the algorithms were run on a computer with 3.1 GHz quad cores and 6 GiB memory, with Linux Ubuntu 16.04 distribution and ROS Kinetics (Quigley et al., 2009). Our quantitative results demonstrate the accuracy of our method for localization, and analysis the *AP/mAP* of *3D IoU* on above 50% and 70%, respectively.

(a) **BLS dataset:** The first experiment was performed with the dataset collected by our own laser scanning systems: a self-designed BLS consisting of a camera, dual VLP-16 LiDAR, a stereo camera and an IMU (Fig. 6). The system uses two Velodyne VLP-16 laser scanners, each composed of 16 laser-detector pairs individually aimed in 2° increments over the 30° (-15° to -15°) field of view of the sensor. One laser scanner is placed horizontally and the other is mounted at 45° below it. A stereo camera with two global shutter sensors is mounted on the middle of the backpack (For algorithm validation and reduced complexity, we installed only one stereo camera on the right side of the backpack).

Using the data described in Table 1, experiments were conducted in four different underground parking lots. The target-free automatic calibration algorithm (Gong et al., 2018) was used to calibrate VLP-16; the target-based semi-automatic method (Dhall et al., 2017) was used to calibrate the camera and VLP-16. We synchronized the dataset by using the timestamp before we process. The entire coordinate system of the sensors was calibrated to the horizontal LiDAR. LOAM (Zhang and Singh, 2014) and YOLOv3 (Redmon and Farhadi, 2017), two rapid and robust algorithms were used for mapping and detecting 2D objects, respectively. The average point density of the underground garage data is about 1,800 points/m^2, and the precision of the data is 3–5 cm. Similar with KITTI, we manually labeled the objects in the form of 3D bounding boxes as ground truth. The ground truth data were saved as binary matrices.

The results of the experiments of four underground parking lot datasets (see Fig. 7) indicate the accuracy of center point estimation. For each group, SDTM and fine-tuning results are compared with ground truth. As expected, the iterative weighted mean method (SDTM) locates targets well and better approaches the actual center point after fine tuning. Table 2 details the numerical analysis for each dataset. For the BLS dataset, the average positioning accuracy of our algorithm is 33.1 cm, which compared with F-PointNet is an improvement. Fig. 8(a)–(f) shows the results of the proposed algorithm in real-time simultaneous mapping and target location/detection for the BLS dataset.

IoU is defined as

$$IoU = \frac{DetectionResult \cap GroundTruth}{DetectionResult \cup GroundTruth}. \tag{10}$$

In Table 2, the object detection average accuracy (AP_{3D}) of the state-of-art algorithms are compared with our methods at 50% and 70% IoU thresholds, respectively. IoU was used as the evaluation criteria and calculated according to Eq. (10). Benefitting from a comprehensive estimation of multi-frame data for different measurement views, our algorithm attains an AP of 81.31% in AP_{pillar}; the overall AP_{QRCode} is also over 75.40%. Current state-of-art algorithms focus mainly on unmanned scenes, which use single frame information for object detection. Whereas, our algorithm uses a multi-frame estimation in the SLAM scenario. Thus, because of the different scenarios and algorithm structures, results from our experiments cannot be compared with the unified standard. For F-PointNet, the detection results of each frame are used to calculate IoU and then calculate the average IoU as the final result for the entire multi-frame scene; in our algorithm, IoU is calculated after SLAM. Fig. 9(a), (b) show the vehicle detection results of our method on BLS dataset. Our method without 3D CNNs outperforms the existing methods.

(b) **KITTI benchmark for vehicle detection:** This benchmark has different kinds of datasets for different tasks (2/3D object detection, odometry, object tracking, etc.). Many researchers used this dataset as a benchmark. However, this dataset is a discrete random static scene, which is unsuitable for our continuous scene fusion and estimation problems. Thus, two scenes (2011-09-26-drive-0009 and 2011–09-26-drive-0022) from the raw dataset were used for comparison, which contains 2/3D bounding box labels (tracklets) in images and point clouds. The dataset properties are given on Geiger et al. (2012). Here, the standard test method (Geiger et al., 2012) was used for AVOD and F-PointNet tests. In our method, the ground truth was transformed by ***tf*** (GPS data) to the global coordinate system. To verify the AP/mAP, IoU was calculated for each object after SLAM.

Table 3 compares the vehicle detection AP_{3D} and AP_{BEV} (BEV: Bird's Eye View) with different methods. Since the objects were detected by a moving observer, occlusion did not exist. The detection results were not obtained at three different levels (easy, moderate, hard). Rather, *mAP* was verified after completing the SLAM operation. Our object detection AP_{3D}/AP_{BEV} has achieved 75.52% and 81.55% accuracy, respectively. Fig. 10(a) and (b) show the vehicle detection results using the KITTI dataset. Results show that without 3D CNNs, the proposed algorithm significantly better performs than the existing methods.

5. Conclusion

We dealt the problems associated with multi-sensor-based 3D object detection algorithms in a SLAM environment. Since point clouds acquired by MLS/BLS systems are usually sparse, and incomplete due to limited scanning position and routes, and occluded by obstacles, such problems often lead to nonrobust results. In addition, structural ambiguity and the requirement for massive labeled data limit the application of 3D-feature-based methods for efficient object detection.

Point cloud features are often ambiguous and unreliable. Manual labeling of objects in huge amount of point clouds is expensive, laborious and sometimes even impossible. We investigated the above problems of point cloud data, and explored the possibility of detecting 3D objects based on a probabilistic framework, and finally propose a three-stage approach to automate locating, bounding, and fine-tuning of 3D objects in noisy, unstructured, sparse 3D MLS/BLS point clouds.

At the locating stage, we develop a frustum-based probabilistic framework (SDTM) to locate 3D objects. Results of the experiments demonstrate that by using only 2D object detection results, our method automatically locates 3D objects in large scenes with the localization error of 33.1 cm. In particular, based on 3D feature descriptors the new algorithm is superior to current methods for detecting small and sparse objects (such as pillars and QRCodes).

Later, we develop a PCA-based rough estimate algorithm and G-PO fine-tuning to estimate 3D bounding boxes. Results show that, without using the feature method, performance of the proposed algorithm is close to and even better than that of state-of-the-art methods. Our method, achieving 75.52% AP_{3D} and 81.55% AP_{BEV} for 3D car detection under 50% IoU in KITTI benchmark, which does not require a large amount of training data, is universally strong and can be applied directly to detect various targets.

The proposed method has a few limitations: (i) the PCA rough match used here is sometimes not robust because of noise in the frustum, which often leads to a lower IoU score, (ii) the developed framework is suitable only for detecting static objects, rather than moving objects. To resolve the above limitations and to improve the performance of 3D object detection, future work will focus on combining appropriate model evaluation methods with unsupervised learning.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants No. 41871380, 41471379, and

61771413. The first author acknowledges the China Scholarships Council for providing doctoral scholarship (No. 201806310155).

References

Breckon, T.P., Fisher, R.B., 2005. Amodal volume completion:3D visual completion. Comput. Vis. Image Underst. 99 (3), 499–526.
Broggi, A., Buzzoni, M., Debattisti, S., Grisleri, P., Laghi, M.C., Medici, P., Versari, P., 2013. Extensive tests of autonomous driving technologies. IEEE Trans. Intell. Transp. Syst. 14 (13), 1403–1415.
Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R., 2015. 3D object proposals for accurate object class detection. In: Advances in Neural Information Processing Systems. pp. 424-432.
Chen, X., Ma, H., Wan, J., Li, B., Xia, T., 2017. Multi-view 3D object detection network for autonomous driving. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1907-1915.
Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: 2005 IEEE Conference on Computer Vision and Pattern Recognition, pp. 886-893.
Deng, Z., Latecki, L. J., 2017. Amodal detection of 3D objects: Inferring 3D bounding boxes from 2D ones in RGB-depth images. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2.
Dhall, A., Chelani, K., Radhakrishnan, V., Krishna, K. M., 2017. LiDAR-camera calibration using 3D-3D point correspondences. arXiv preprint: 1705.09785.pp.
Dong, Z., Yang, B., Liu, Y., Liang, F., Li, B., Zang, Y., 2017. A novel binary shape context for 3D local surface description. ISPRS J. Photogramm. Remote Sens. 130, 431–452.
Engel, J., Koltun, V., Cremers, D., 2017. Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. 40 (3), 611–625.
Felzenszwalb, P., McAllester, D., Ramanan, D., 2008. A discriminatively trained, multi-scale, deformable part model. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1-8.
Forster, C., Pizzoli, M., Scaramuzza, D., 2014. SVO: Fast semi-direct monocular visual odometry. In: IEEE International Conference on Robotics and Automation, pp. 15-22.
Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors. In: 2004 European Conference on Computer Vision. Springer, pp. 224-237.
Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354-3361.
Girsshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587.
Gong, Z., Wen, C., Wang, C., Li, J., 2018. A target-free automatic self-calibration approach for multibeam laser scanners. IEEE Trans. Instrum. Meas. 67 (1), 238–240.
Guo, Y., Sohel, F., Bennamoun, M., Lu, M., Wan, J., 2013. Rotational projection statistics for 3D local surface description and object recognition. Int. J. Comput. Vision 105 (1), 63–86.
Gupta, S., Grishick, R., Arbelae, P., Malik, J., 2014. Learning rich features from RGB-D images for object detection and segmentation. In: 2014 European Conference on Computer Vision. Springer, pp. 345-360.
Hess, W., Kohler, D., Rapp, H., Andor, D., 2016. Real-time loop closure in 2D LiDAR SLAM. In: 2016 IEEE International Conference on Robotics and Automation, pp. 1271-1278.
Johnson, A.E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. 21 (5), 433–449.
Jolliffe, I.T., Cadima, J., 2016. Principal component analysis: A review and recent developments. Philos. Trans. Roy. Soc. A: Math., Phys. Eng. Sci. 374 (2065).
Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S., 2017. Joint 3D proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1-8.
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C-Y., Berg, A.C., 2016. SSD: Single shot multibox detector. In: 2016 European Conference on Computer Vision. pp. 21-37.
Luo, Z., Li, J., Xiao, Z., Mou, G.Z., Cai, X., Wang, C., 2019. Learning high-level features by fusing multi-view representation of MLS point clouds for 3D object recognition in road environments. ISPRS J. Photogramm. Remote Sens. 150, 44–58.
Ma, L., Li, J., Li, Y., Zhong, Z., Chapman, M., 2019. Generation of horizontally curved driving lines in HD maps using mobile laser scanning point clouds. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. https://doi.org/10.1109/JSTARS.2019.2904514.
Ma, L., Li, Y., Li, J., Wang, C., Wang, R., Chapman, M.A., 2018. Mobile laser scanned point-clouds for road object detection and extraction: A review. Remote Sens. 10 (10), 1531.
Munoz, D., Bagnell, J.A., Vandapel, N., Hebert, M., 2009. Contextual classification with functional max-margin Markov networks. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 975-982.
Mur-Artal, R., Montiel, J.M.M., Tardos, J.D., 2015. ORB-SLAM: a versatile and accurate monocular slam system. IEEE Trans. Rob. 31 (5), 1147–1163.
Nuchter, A., Lingemann, K., Hertzberg, J., Surmann, H., 2007. 6D SLAM 3D mapping outdoor environments. J. Field Rob. 24 (8–9), 699–722.
Nurunnabi A., Belton, D., West, G., 2016. Robust segmentation for large volumes of laser scanning 3D point cloud data. IEEE Trans. Geosci. Remote Sens. 54(8), pp. 4790–4805.
Qi, C.R., Su, H., Mo, K., Guibas, L. J., 2017a. Frustum PointNets for 3D object detection from RGB-D data. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 918-927.
Qi. C.R., Su, H., Mo, K., Guibas, L.J., 2017b. PointNet: Deep learning on point sets for 3D classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 652-660.
Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., 2009. Ros: An open-source robot operating system. In: ICRA Workshop on Open Source Software. Vol. 3. Kobe, Japan, pp. 5.
Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. pp. 779-788.
Redmon, J., Farhadi, A., 2017. YOLO 9000: Better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. pp. 7263-7271.
Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing System. pp. 91-99.
Rusu, R. B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation. pp. 3212-3217.
Schreiber, M., Knoppel, C., Franke, U., 2013. LaneLoc: Lane marking based localization using highly accurate, maps. In: 2013 IEEE Intelligent Vehicles Symposium, pp. 449-454.
Segal, A., Haehnel, D., Thrun, S., 2009.Generalized-icp. In: Robotics: Science and Systems. Vol. 2. pp. 435.
Seo, Y.W., Lee, J., Zhang, W., Werrergreen, D., 2015. Recognition of highway work zones for reliable autonomous driving. IEEE Trans. Intell. Transport. Syst. 16 (2), 708–718.
Song, S., Xiao, J., 2016. Deep sliding shapes for Amodal 3D object detection in RGB-D images. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. pp. 808-816.
Sukno, F.M., Waddington, J.L., Whelan, P.F., 2013. Rotationally invariant 3D shape contexts using asymmetry patterns. Proc. GRAPP. 2, 335.
Taskar, B., Guestrin, C., Koller, D., 2004. Max-margin Markov networks. In: Advances in Neural Information Processing System. pp. 25-32.
Tombari, F., Salti, S., Di Stefano, L., 2010. Unique shape context for 3D data description. In: ACM Workshop on 3D Object Retrieval, pp. 57-62.
Trevor, A. J., Gedikli, S., Rusu, R. B., Christensen, H. I., 2013. Efficient organized point cloud segmentation with connected components. In: Semantic Perception Mapping and Exploration. pp.1-6.
Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In: 2001 IEEE Conference on Computer Vision and Pattern Recognition. pp. 21-27.
Wang, C., Wen, C., Hou, S., Gong, Z., Li, Q., Sun, X., Li, J., 2018. Semantic line framework-based indoor building modeling using backpacked laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 143, 150–166.
Xie, S., Liu, S., Chen, Z., Tu, Z., 2018. Attentional shape context net for point cloud recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4606-4615.
Zai, D., Li, J., Guo, Y., Cheng, M., Huang, P., Cao, X., Wang, C., 2017. Pairwise registration of TLS point clouds using covariance descriptors and a non-cooperative game. ISPRS J. Photogramm. Remote Sens. 134, 15–29.
Zai, D., Li, J., Guo, Y., Cheng, M., Lin, Y., Luo, H., Wang, C., 2018. 3D road boundary extraction from mobile laser scanning data via super voxels and graph cuts. IEEE Trans. Intell. Transport. Syst. 19 (3), 802–813.
Zhang, J., Singh, S., 2014. LOAM: Lidar odometry and mapping in real-time. In: Robotic: Science and System. pp. 9.