Semantic Segmentation of 3D Point Clouds in Dynamic Scene Using Semi-supervised Learning

Jilin Mei, *Member, IEEE,* Biao Gao, *Member, IEEE,* Huijing Zhao, *Member, IEEE,* and Hongbin Zha, *Member, IEEE*

***Abstract*—This work studies semantic segmentation of 3D point clouds in dynamic scene. To reduce the large requirement of fine annotations, we propose a new method that integrates semi-supervised learning and neural network for driving applications. Based on range image and the motion of platform, data units are associated between adjacent frames to generate pairwise constraint. Furthermore, a special loss function is designed to encourage constraint data to be assigned with the same semantic label. The performance of our method is illustrated on a new 3D point cloud dataset collected from the driving platform, which will be released together with this publication. Qualitative and quantitative experiments show that combining a few annotations and large constraints data significantly enhances the effectiveness and scene adaptability of the classifier, which brings more than 10% improvement.**

I. Introduction

Scene understanding is crucial for autonomous platforms operating in dynamic environments, as it guides the platforms making more efficient and safe navigation strategy. One of the key topics is semantic segmentation which tells the existence of objects in the surroundings using camera or LiDAR data.

The semantic segmentation is to find a semantic label for each unit in the scenarios. In the context of the 3D point clouds collected from driving platforms, it is a point-wise classification [1], and the label could be people, car, cyclist, tree and so on, as shown in Fig. 1. The traditional pipeline [2] of this task contains (1) preprocessing, e.g., voxlization; (2) feature design, e.g., geometric features extraction; (3) classifier selection, e.g., Random Forest; and (4) postprocessing, e.g., Conditional Random Field. The traditional methods depend too much on discriminative features [1], and their adaptability is not enough to cope with dynamic scenes.

Recently, the success of deep learning in image semantic segmentation brings new approaches [3]. These methods remove the dependence on handcrafted features in an end-to-end manner. However, these methods have a large demand for the fine labeled data [4]. On one hand, point-wise annotation is time-consuming; on the other hand, there are few public data set with point level annotation supporting autonomous driving application. How to ensure the performance of semantic segmentation under the condition of using a few annotations is an interesting topic, which is formally named semi-supervised learning in machine learning field.

J. Mei, B. Gao, H. Zhao, and H. Zha are with the Peking University, with the Key Laboratory of Machine Perception (MOE), and also with the School of Electronics Engineering and Computer Science, Beijing 100871, China (e-mail: zhaohj@cis.pku.edu.cn)

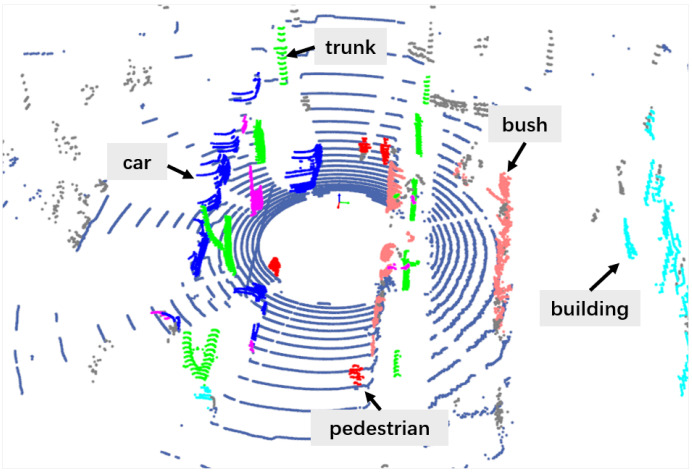

Fig. 1. The dynamic campus scene.

There are many researches discuss semi-supervised learning which usually integrates labeled and unlabeled data [5], [6]. Pairwise constraints are collected from unlabeled data to describe the probability of two samples having the same label. For example, the temporal constraint will tell a high probability if one object appears in adjacent samples. The constraint only indicates that two samples would be the same label although the exact label is not known. Constraints/priors, like the object size and motion information, are very instructive for 3D point clouds segmentation. Therefore, if we explicitly introduce constraints/priors during the training phase of neural network rather than only relying on the amount of labeled data, we can have a great expectation that the parameters of neural network also converge to a good state with a small amount of labeled samples.

In this paper, we propose a new 3D point clouds semantic segmentation method that integrates semi-supervised learning and deep learning for driving scenes. Based on range image and the motion of platform, data units are associated between adjacent frames to generate pairwise constraint. Different with the previous method, we consider the must-link and ignore must-not-link. Moreover, a special loss function is designed to encourage constraint data to be assigned with the same semantic label. We collect and annotate a 3D point cloud dataset in dynamic scene using the LiDAR sensor. Qualitative and quantitative experiments show that combining a few labeled data and large constraints data significantly improve the effectiveness and scene adaptability of the classifier.

This work provides the following contributions: (1) A new

semi-supervised semantic segmentation method are proposed for LiDAR data collected from driving platform. (2) The pairwise constraint is introduced in the training phase to reduce the large requirement of labeled data. (3) the performance of our method is demonstrated on a new dataset, which will be provided together with this publication.

The remainder of this paper is organized as follows. The related works are discussed in Sect. II. In Sect. III, the proposed method is presented. In Sect. IV, the implementation details are discussed. Then, we present the experimental results in Sect. V, and draw some conclusions in Sect. VI.

II. Related Works

Numerous researches have been done in semantic segmentation, the recent progress for image and RGB-D data is well reviewed in [3]. In this section, we mainly focus on methods specified for 3D point clouds(i.e., from LiDAR sensors) in dynamic outdoor scene. Moreover, these works can be broadly divided into three aspects: feature-based methods, deep learning methods, semi-supervised learning methods.

A. Feature-based Methods

Feature-based methods belong to traditional machine learning category here, and the general pipeline of these ones consists of feature selection, classifier design and graphical model description.

The straightforward way is to convert semantic segmentation into a point-wise classification which includes extracting features on each unit, concatenating features as a vector and then determining the label via a well trained classifier. [7] presents a common pipeline from feature selection to classifier training. Due to the irregular arrangement property of point clouds, the authors test 5 different neighborhood definitions to achieve better representation. Similar research is conducted in [8] and it shows the ability to tackle varying density of data. One point cloud usually contains millions of points so evaluating the label for each point is typically computationally expensive(a matter of minutes reported in [8]).

[2] proposes an efficient approach where speed and accuracy are satisfied at the same time, furthermore, the average classification time could be reduced less than 1s. [9] represents the raw point cloud as 2D range image, and proposes a framework of simultaneous segmentation and classification of range image, where both 2D range image and 3D raw data are involved. Straightforward approaches assume that each data unit is independent and ignore the spatial and contextual relations between units. Consequently, they can produce good results based on distinctive features. However, when the features are not discriminative, the point-wise classification will be noisy and locally inconsistent [1].

In order to make the segmentation results spatially smooth, the neighbor elements are taken into account. For this purpose, graphical models such as Markov Random Field (MRF) and Conditional Random Field (CRF) are usually exploited to encode the spatial relationships. In [10], the node potentials and edge potentials are both formulated with parametric linear model where the functional max-margin learning is used to find the optimal weights. [11] proposes a simplified Markov networks to inference the contextual relations between points. Instead of learning all weights for node and edge potentials in graphical models, the node potentials are calculated from a point-wise classifier, and the edge potentials are determined by the physical distance between points.

In the following works, researchers mainly focus on how to define a node and how to build the graph. In [12], both local and non-local descriptors are designed, furthermore, the scene is partitioned into geometrically-homogeneous segments, then the adjacency graph is build and a CRF classifier is designed to capture the high-order spatial relations. The previous graphical models here only encourage neighboring nodes to have the same label, however, [13] introduce a new higher-order model for 3D point cloud classification that takes into account the non-associative geometric context between different labels. Graphical model is not the only way to incorporating contextual information. [14] adopts a sequenced/hierarchical prediction method at different scales so that the contextual information can be transfered between points and regions.

The performance of the above ones largely depends on the handcrafted features. These methods are effective in fixed or regular scenarios, as for dynamic scenes, the features are empirically designed and the performance decreases.

B. Deep Learning Methods

Deep learning, especially the convolution neural network (CNN) without handcrafted features, has shown effectual performance on 2D image segmentation [3]. However, the semantic segmentation of 3D point clouds(i.e., from LiDAR sensors) is still an open research problem [15], due to the irregular, not grid-aligned properties. Therefore, the recent researches project the point clouds into 2D views, concurrently, some of them attempt to directly ways, for example, volumetric/voxel representations.

Inspired by the success of CNN in image segmentation, the state-of-the-art image-based algorithms could be directly used after rendering 2D views from the 3D raw data. [20] projects point clouds into virtual 2D RGB images with Katz projection. Then, a pre-trained CNN is used to semantically classify the images. However, this projection will remove all points that are not visible, for example, if a car is projected, all points behind it are removed. [22] unwraps 360° 3D LiDAR data onto a spherical 2D plane without the point loss. The spherical projection also is applied in SqueezeSeg [25], where the CNN directly outputs the point-wise label of the transformed LiDAR data, then a CRF is applied to refine the outputs. [26] uses cylindrical projection to create the depth and reflectivity images. In [27], the point clouds are encoded by top-view images and a simple fully convolutional neural network (FCN) is used. This method can be real-time due to only elevation and density features are extracted. In [28], the input point cloud is projected into multiple views, such as color, depth and surface normal images.

Another kind of methods model the raw data in direct ways. [29] proposes SEGCloud where the raw 3D point cloud is pre-processed into voxelized point cloud with fixed grid size.

TABLE I
APPROACHES FOR 3D POINT CLOUDS SEMANTIC SEGMENTATION

Research	Learning Method	Input	Classifier	Dataset	Scene
Munoz, 2009, [1]	sup.	L	MRF	-	rural
Hu, 2013, [2]	sup.	L	KLR	VMR-Oakland,Freiburg [16]	urban
Weinmann, 2014 [7]	sup.	L	RF	VMR-Oakland [10],Pairs-rue-Madame [17]	urban
Hackel, 2016 [8]	sup.	L	RF	Pairs-rue-Madame	urban
Zhao, 2010, [9]	sup.	L	SVM	-	campus
Munoz, 2009, [10]	sup.	L/V	MRF	-	-
Lu, 2012, [11]	sup.	L	CRF,SVM	VMR-Oakland	urban
Engelmann, 2017, [15]	sup.	L/V	DL	S3DIS [18],vKITTI [19],KITTI	indoor,urban,urban
Tosteberg, 2017, [20]	sup.	L	DL	Semantic3D,VPS [21]	urban,indoor
Dewan, 2017, [22]	sup.	L	DL	KITTI [23]	urban
Hackel, 2017, [24]	sup.	L	DL	Semantic3D [24]	urban/rural
Wu, 2017, [25]	sup.	L	DL	KITTI	urban
Piewak, 2018, [26]	sup.	L/V	DL	-	urban/rural/highway
Caltagirone, 2017, [27]	sup.	L	DL	KITTI	urban
Lawin, 2017, [28]	sup.	L	DL	Semantic3D	urban/rural
Tchapmi, 2017, [29]	sup.	L/V	DL	NYU V2 [30],Semantic3D,S3DIS,KITTI,	indoor,urban/rural,indoor,urban
Riegler, 2017, [31]	sup.	L	DL	ModelNet10 [32]	CAD model
Qi, 2017, [33]	sup.	L/V	DL	ShapeNet [34],Stanford3D [35]	CAD model,indoor
Landrieu, 2017, [36]	sup.	L/V	DL	Semantic3D,S3DIS	urban,rural/indoor
Bearman, 2016, [4]	semi-sup.	V	DL	PASCAL VOC	-
Yan, 2006, [37]	semi-sup.	V	KLR,SVM	-	nursing home
Bauml, 2013 [38]	semi-sup.	V	KLR,SVM	-	TV series
Hong, 2015, [39]	semi-sup.	V	DL	PASCAL VOC [40]	-
Papandreou, 2015, [41]	semi-sup.	V	DL	PASCAL VOC	-
Lin, 2016, [42]	semi-sup.	V	DL	PASCAL VOC	-
Cour, 2009, [43]	weakly-sup.	V	linear classifier	-	TV series
Pathak, 2015, [44]	weakly-sup.	V	DL	PASCAL VOC	-
Xu, 2015, [45]	weakly-sup.	V	linear classifier	Siftflow [46]	-
Dai, 2015, [47]	weakly-sup.	V	DL	PASCAL VOC	-

sup. : supervised; L : LiDAR Data; V : Camera Data; RF : Random Forest; KLR : Kernel Linear Regression; DL : Deep Learning.

Although [29] is simple and effective, how to set the size of voxel is a problem in large-scale scene. Thus the scene is voxelized at five different resolutions in [24], and each of the five scales is handled separately with CNN. Instead of using fixed grid size, [31] proposes OctNet where the hybrid grid-octree data structure is applied to represent the raw 3D data and each leaf of the octree stores a pooled feature representation. PointNet [33] demonstrates a unified architecture that directly takes raw point clouds as input and outputs the label of each point. The scene is divided into blocks. Then points in each block are passed through a series of multilayer perceptrons (MLP) so that the local features and global features are extracted. Based on [33], [15] makes an extension to incorporate larger-scale spatial context, and the improved results are reported in both indoor and outdoor scenarios. [36] proposes more elegant architecture to capture contextual relations. The first step is to partition the raw point cloud into geometrically simple shapes, called super-points. All super-points are embedded by PointNet [33].

The semantic segmentation with deep learning usually is implemented in the supervised way, that means largely detailed annotations are required. However, making point-wise annotations for 3D point clouds is labor-intensive and time-consuming. Furthermore, there are few public datasets supporting this level annotations.

C. Semi-supervised Learning Methods

Considering the large demand of detailed annotations, many researches study the semi-supervised learning methods where few labeled data and large unlabeled data are integrated, the weakly-supervised learning methods where multiple ambiguous labels are used. Our research belongs to the semi-supervised category, please refer to TABLE I for weakly-supervised methods.

The early work [5] of semi-supervised learning specifies the priori knowledge of unlabeled data via pairwise constraints. A pairwise must-link constraint means the two objects should be the same label although the label is unknown, whereas two

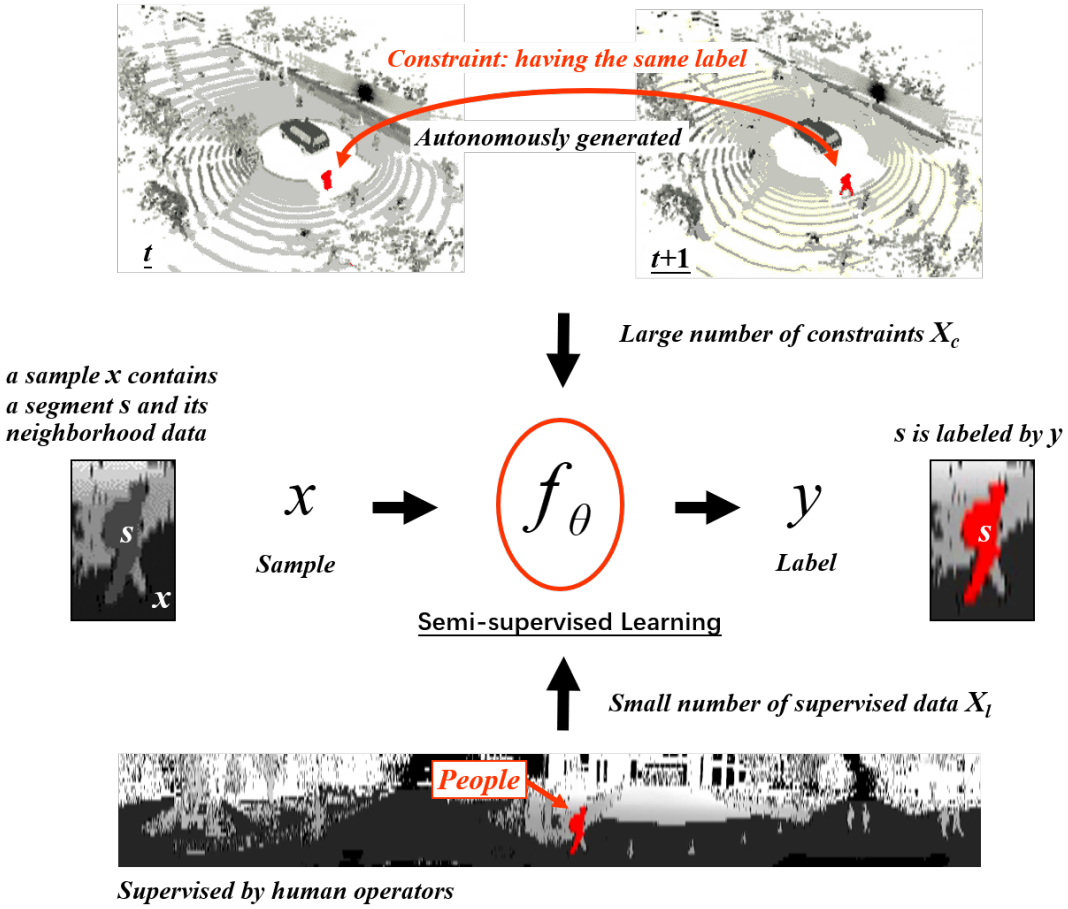

Fig. 2. The framework of semi-supervised learning for 3D point clouds semantic segmentation.

objects associated with must-not-link means that they should have different labels. Both labeled and constraint data are used for model fitting and the authors model the constraint information by the maximum entropy principle. Similar idea is demonstrated in [6]. The pairwise constraints are incorporated in the clustering of Gaussian mixture model (GMM). [5], [6] give the foundation of semi-supervised learning with pairwise constraints.

[37] extends it for video object classification where temporal relations between frames, multi-modalities, such as face and voice, and human feed back (manual annotation) are all considered and formulated in a unified framework. [38] applies constraints for person identification in multimedia data. It achieves state-of-the-art performance on two diverse TV series. Recently, semi-supervised learning is also integrated with deep neural networks (DNN). [39] decouples the semantic segmentation into classification and segmentation using a large number of image-level object labels and a few number of pixel-wise annotations respectively. The classification network tells the class-specific activation maps which are transfered into segmentation network with bridge layers. Then the segmentation network only needs few pixel-wise annotations to train the model, e.g., 5 or 10 strong annotations per class. [41] designs a expectation-maximization (EM) training methods for semantic image segmentation combining bounding boxes, image-level labels and few strongly labeled images.

Semi-supervised learning has successfully applied in image segmentation [39] and video analysis [37]. However, to the author's knowledge, few researches discuss it for the 3D point clouds semantic segmentation. In this paper, we try to combine the semi-supervised learning and neural network to solve the problem that how to do the 3D point clouds semantic segmentation with insufficient point-wise annotations.

III. METHODOLOGY

A. Problem Definition

Let s denotes a small segment of a 3D lidar data, which is extracted by examining consistency of 3D points with their neighborhood on the range image frame using e.g., a region growing method. Without loss of generality, we assume that the 3D points of s are the measurements of a single object. However it is common that s might represents only a part of the object, e.g. the upper body of a pedestrian, due to the nature of lidar measurements. Hence for each s, a data sample x is generated centered at s, containing s as the foreground and its neighborhood data as the background. The problem of this work is formulated as learning a multi-class classifier f_θ that maps x to a label $y \in \{1, ..., K\}$, and subsequently associate y to the 3D points of s.

$$f_\theta : x \to y \in \{1, ..., K\} \tag{1}$$

Given a set of supervised data samples $X_l = \{x_i, y_i\}_{i=1}^M$, where $\{y_i\}$ are annotated manually by human operators for each of the $\{x_i\}$, a common way of learning a classifier f_θ is to find the best θ^* that minimize a loss function L as below.

$$\theta^* = \arg\max_\theta L(X_l; \theta) \tag{2}$$

However the problem of generating a large amount of supervised data is not trivial. This research learns f_θ with a small set of costly supervised data X_l, and additional large set of autonomously generated constraints $X_c = \{< x_i, x_j >_n\}_{n=1}^N$,

$$\theta^* = \arg\max_\theta L(X_l, X_c; \theta) \tag{3}$$

where a constraint $< x_i, x_j >$ denotes that x_i and x_j have the same label, i.e. $y_i = y_j$, which is generated in this research autonomously by associating the data segments along sequential frames according to their locations after ego-motion compensation.

This semi-supervised loss function L is a combined one, containing those on supervised data X_l and constraints X_c respectively.

$$L(X_l, X_c; \theta) = L_l(X_l; \theta) + L_c(X_c; \theta) \tag{4}$$

B. Supervised Loss

For supervised data X_l, we follow the widely used definition on cross entropy, e.g. [33], and define loss L_l as below:

$$L_l(X_l; \theta) = -\frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} y_i ln(P_\theta^k(x_i)) \tag{5}$$

where y_i is a one-hot vector, and $P_\theta^k(x_i)$ is the probability that x_i be assigned a label k by a classifier with the set of parameters θ.

C. Constraints Loss

For each constraint $< x_i, x_j >$, let y_i and y_j denote the labels x_i and x_j respectively by using a learnt classifier f_θ, a penalty is cast if $y_i \neq y_j$. Subsequently a loss is estimated as below for each constraint on the probability that the constrained data samples are assigned of different labels.

$$\begin{aligned} P(y_i \neq y_j) &= \sum_{k=1}^{K} \sum_{\substack{l=1 \\ l \neq k}}^{K} P_\theta^k(x_i) P_\theta^l(x_j) \\ &= 1 - \sum_{k=1}^{K} P_\theta^k(x_i) P_\theta^k(x_j) \end{aligned} \tag{6}$$

Hence we define the loss on constraints L_c as below:

$$\begin{aligned} L_c(X_c; \theta) &= \frac{\gamma}{N} \sum_{n=1}^{N} P(y_i \neq y_j) \\ &= \frac{\gamma}{N} \sum_{n=1}^{N} (1 - \sum_{k=1}^{K} P_\theta^k(x_i) P_\theta^k(x_j)), \gamma \in [0, 1] \end{aligned} \tag{7}$$

where γ is a weighting factor. Obviously, L_c describes the unsupervised learning procedure.

D. Semi-supervised Loss

Combining the losses on supervised data X_l and constraints X_c, the semi-supervised loss is defined as below.

$$\begin{aligned} L(\theta; X_l, Y_l, X_c) &= L_l(X_l, Y_l; \theta) + L_c(X_c; \theta) \\ &= -\frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} 1[y_i = k] ln(P_\theta^k(x_i)) + \\ &\frac{\gamma}{N} \sum_{n=1}^{N} (1 - \sum_{k=1}^{K} P_\theta^k(x_i) P_\theta^k(x_j)), \gamma \in [0, 1]. \end{aligned} \tag{8}$$

IV. Implementation Details

A. Preprocessing

It is time-consuming to classify the raw data point-by-point. So in this work, the raw data is firstly converted into 2D range image, then the oversegmentation is conducted based on geometric consistency and the range image is partitioned into regions [9]. In this way, we only need to evaluate the label of each region.

For each point p_k in the raw point cloud P, $p_k = \langle x_w^k, y_w^k, z_w^k \rangle$. The first step of preprocessing is to convert the raw 3D point cloud into 2D range image. We use cylindrical projection to unwrap the 360° LiDAR data, obtaining the range image with height 32 and width 2160 (for Velodyne32 [48]). In this way, the sensor spins one circle, one range image denoted as $R = \{r_k\}_{k=1}^M$ is generated. The two data structure P and R can be converted into each other equally. Formally, the value of unit in range image R is defined as $v(r_k)$:

$$\begin{aligned} v(r_k) &= \sqrt{{x_w^k}^2 + {y_w^k}^2 + {z_w^k}^2}, \\ v(r_k) &\subset [0, 255]; r_k = \langle x_r^k, y_r^k \rangle. \end{aligned} \tag{9}$$

The second step is oversegmentation. The pseudo code is outlined in APPENDIX. Based on the range image R and raw data P, the coarse segmentation is applied where the road region is extracted and the point is marked as background if the distance to road is too large. Next, edge points are determined by comparing with the 4-neighbor in range image. After that, the region grow method for range image is used to separate the range image into regions, as depicted in Fig.3.(c). Thus, we totally obtain N segments/regions, denoted as $S = \{s_i\}_{i=1}^N$.

B. Sample Definition

After preprocessing, Each segments is represented with three attributes: id, number of points, the coordinate of center, that is, $s_i = \langle id, pnum, cpt \rangle$. The point number of each segments is largely varied during the platform moving. When the object is near the sensor, it has dense point distribution, otherwise, the distribution becomes sparse. If one segments has few points, it is very difficult to identify the label. Therefore, we can not directly use all segments for sample generation.

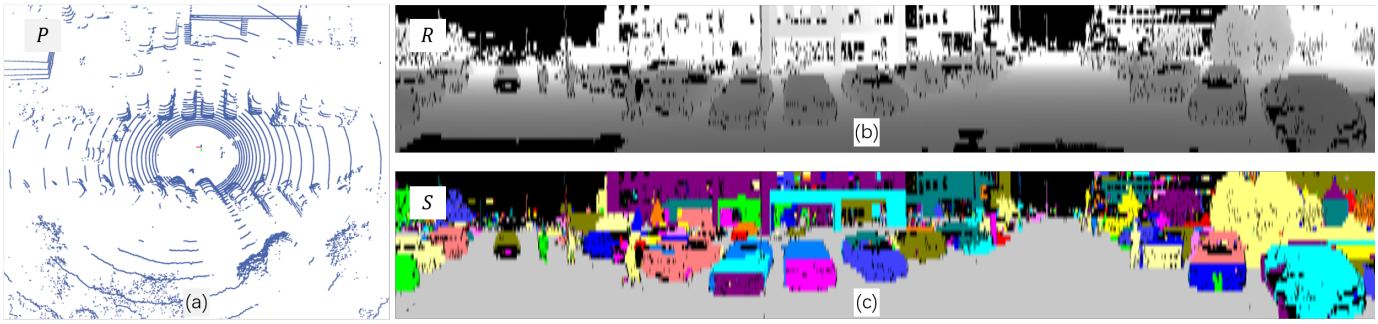

Fig. 3. The oversegmentation of preprocessing. (a) the raw point cloud. (b) the range image. (c) the result of oversegmentation, one color means one unique segment.

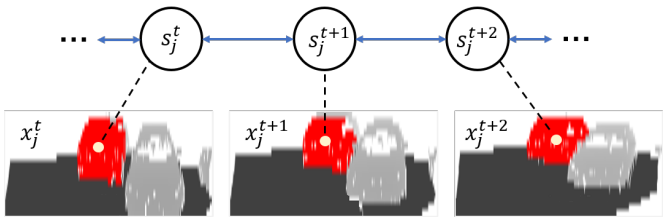

Fig. 4. The procedure of constraint generation. After RegionAssociation, the pairs $\langle x_j^t, x_j^{t+1}\rangle$ and $\langle x_j^{t+1}, x_j^{t+2}\rangle$ are two constraints.

Two criteria are designed to filter ill regions. The candidate segments s_i should satisfy both criteria:

$$\frac{pnum}{\sqrt{cpt.x^2 + cpt.y^2 + cpt.z^2}} > \rho \cap pnum > \sigma \tag{10}$$

where ρ and σ are the parameters.

The straightforward way of sample generation only uses candidate segments, and one region represents one sample. This direct method brings confusions because one object may be separated into multiple regions during oversegmention, for example, the wheel of a cyclist sometimes is alone segmented, but assigning a correct label for the wheel is very hard if we do not consider the background.

In this work, a sample x_i consists of one segment s_i and the background. Human usually combine both the foreground and background to identify objects in dynamic scene. Inspired by this observation, the candidate segment corresponds to the foreground; the background is defined with its neighbor points inside a cuboid centered at s_i, and the cuboid size is 2mx5mx5m(height,width,length). Then, all cropped points are projected on the range image to generate one sample. Currently, one sample contains three channels, height feature map, range feature map and intensity feature map.

C. Constraint Generation

Although manual annotation for unlabeled data is time-consuming and expensive, the unlabeled data can be efficiently used after constraint generation. The constraint, actually being pairwise constraint, describes the temporal relationship between two frames; for example, the s_i^t is one region at frame t, and the motion of platform can also be retrieved (e.g. from GPS); then the r_i^t is transformed into $t+1$ frame to match the nearest region s_i^{t+1}; finally, a series of pairwise regions are collected. The aforementioned process is named RegionAssociation illustrated in Algorithm. 1. Regions and samples are corresponding, thus the samples are linked after RegionAssociation as shown in Fig. 4 where $\langle x_j^t, x_j^{t+1}\rangle$ consists of one constraint, and $\langle x_j^{t+1}, x_j^{t+2}\rangle$ means another one. The constraint is automatically generated and we can produce a lot of it in a cheap way.

Algorithm 1 RegionAssociation

Input: S^t, S^{t+1}, ΔR,ΔT
Output: the constraint set Φ

```
Initialize: Φ ← ∅
for all s_i^t in S^t do
    ŝ_i^{t+1} ← s_i^t · ΔR + ΔT
    s_i^{t+1} ← findnearest(ŝ_i^{t+1}, S^{t+1})
    Φ ← Φ+ < s_i^{t+1}, s_i^t >
end for
return Φ
```

D. Classifier Implementation

The base classifier in Fig.2 is implemented with a CNN. The sample size is 256x256. The γ in Equation. (8) is 1.0. The parameters ρ and σ in Equation. (10) are set with 8.0 and 30 respectively. The base classifier is implemented with 3 convolution layers and 2 fully-connected layers. The number of filters are 32, 32 and 64 for the convolution layers with the kernel size [3,3]. The number of neurons in fully-connected layers are set with 128. The network is implemented with TensorFlow where the learning rate is 1e-4 and the AdamOptimizer is used. The CNN is trained and tested on the computer with a Intel I7-7700 CPU and a Nvidia GTX 1060 GPU.

V. EXPERIMENTAL RESULTS

A. Data Set

The performance of the proposed method is evaluated on a dynamic campus data set collected by an instrumented vehicle, which has a GPS/IMU suite and a Velodyne-HDL32, as shown in Fig. 5. The total route is about 890 meters. All sensor data are collected with each data frame being associated with a

TABLE II
THE TRAINING DATA.

-			people	car	cyclist	trunk	bush	building	unknown
training data	constraint		2272	6322	2562	3253	10893	8167	0
	annotation	training set	1715	4851	531	2445	10103	6591	5650
		validation set	736	2080	228	1049	4330	2826	2426

TABLE III
THE TESTING DATA.

-		people	car	cyclist	trunk	bush	building	unknown
testing data	constraint	925	3542	142	62	2440	1254	0
	annotation	962	4157	169	89	3773	1549	1667

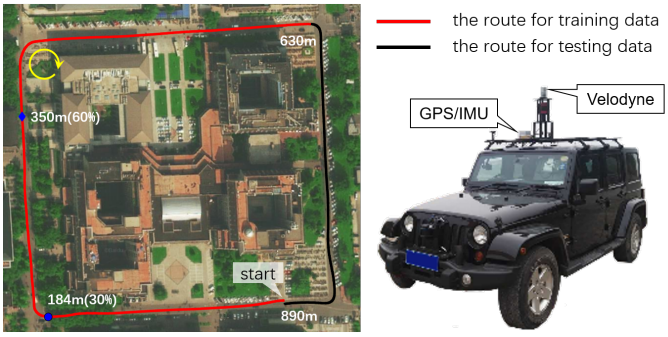

Fig. 5. The routes of data collection and the platform configuration. The 30%/60% in the left means 30%/60% travel of the route for training data.

time log for synchronization. The GPS/IMU data are logged at 100Hz. The LiDAR data are recorded at 10Hz, containing 1373 frames for training (red line in Fig. 5) data and 465 frames for testing data (black line in Fig. 5). One frame can produce multiple samples, for example, we get 6931 car samples from 1373 frame training data in TABLE .II.

Manual annotation is necessary for quantitative evaluation. Instead of working on the raw point cloud, the manual annotation is conducted on the range image where regions are associated with the ones in adjacent frames. We only need click the mouse on one region and assign a category, then a series of associated ones are marked with the same label. Although sometimes the data association brings errors, it largely reduce the annotation time.

The data for experiments are listed in TABLE. II and TABLE. III. Each table has constraint and annotation data. For training data, we randomly select 70% annotations for training set and 30% for validation set. We set six kinds of labels, i.e., people, car, cyclist, trunk, bush and building; these categories are important for driving applications in campus; others, like the pole and cone, are marked as the unknown. Especially, we don't generate constraints for the unknown.

B. Effectiveness on Training Data

There are five experimental settings on training data as shown in TABLE. IV. The baseline_min only uses a few annotations (i.e., 350) and without the constraint, while the baseline_max uses all annotations. Thus, the baseline method works in a supervised way. In general, the baseline_min will show the low border of the performance and baseline_max for the high border. The constraint30 use the same annotations as the baseline_min, but with additional constraints; the tail 30 means all constraints during 30% of the travel are used, i.e., from the start to the 184m(30%) in Fig. 5. The constraint60 and constraint100 have the similar configuration with constraint30, except the amount of constraint data.

According to the settings, five classifiers are offline learned separately. All classifiers have the same network architecture detailed in Set. IV.D. The difference lies in the loss function, where baseline_min and baseline_max only use $L_l(X_l, Y; \theta)$ in Equation. (8), but other settings use both $L_l(X_l, Y; \theta)$ and $L_c(X_c; \theta)$. For offline learning, the classifier are saved with fixed training steps; then we choose the ones whose loss is less than 1e-4; finally, the classifier achieving best performance on the validation set is selected. For online inference, all classifier work in the same way.

The qualitative results are shown in Fig. 6. As the constraint increases, the cars in rectangles are successfully classified with constraint100, even if the occlusions occur. Our method still produces errors, for example, when the person walks near the bush, the constraint100 classifier wrongly annotates it with bush. The main reason is that one sample contains both foreground and background; if the background occupies more information than the foreground, the classifier is likely to assign this sample with the label of the background.

The F-Measure is adopted for quantitative evaluations, which is written as:

$$F - Measure = \frac{2 * recall * precision}{recall + precision} \cdot 100\%. \qquad (11)$$

We use the five classifiers to annotate the training set and the validation set, then the quantitative results are shown in TABLE. V.

Comparing the baseline_min and baseline_max, it shows that large annotations are very important for supervised learning. Comparing the baseline_min and constraint100, it shows that the semi-supervised learning is effective; and the F-Score of constraint100 goes up by 15% on average. Furthermore,

TABLE IV
THE EXPERIMENTAL SETTINGS ON TRAINING DATA.

settings		people	car	cyclist	trunk	bush	building	unknown
baseline_min	constraint	0	0	0	0	0	0	0
	annotation	350	350	350	350	350	350	5650
baseline_max	constraint	0	0	0	0	0	0	0
	annotation	1715	4851	531	2445	10103	6591	5650
constriant30	constraint	682	1897	196	976	3268	2450	0
	annotation	350	350	350	350	350	350	5650
constriant60	constraint	1363	3793	391	1952	6536	4900	0
	annotation	350	350	350	350	350	350	5650
constriant100	constraint	2272	6322	652	3253	10893	8167	0
	annotation	350	350	350	350	350	350	5650

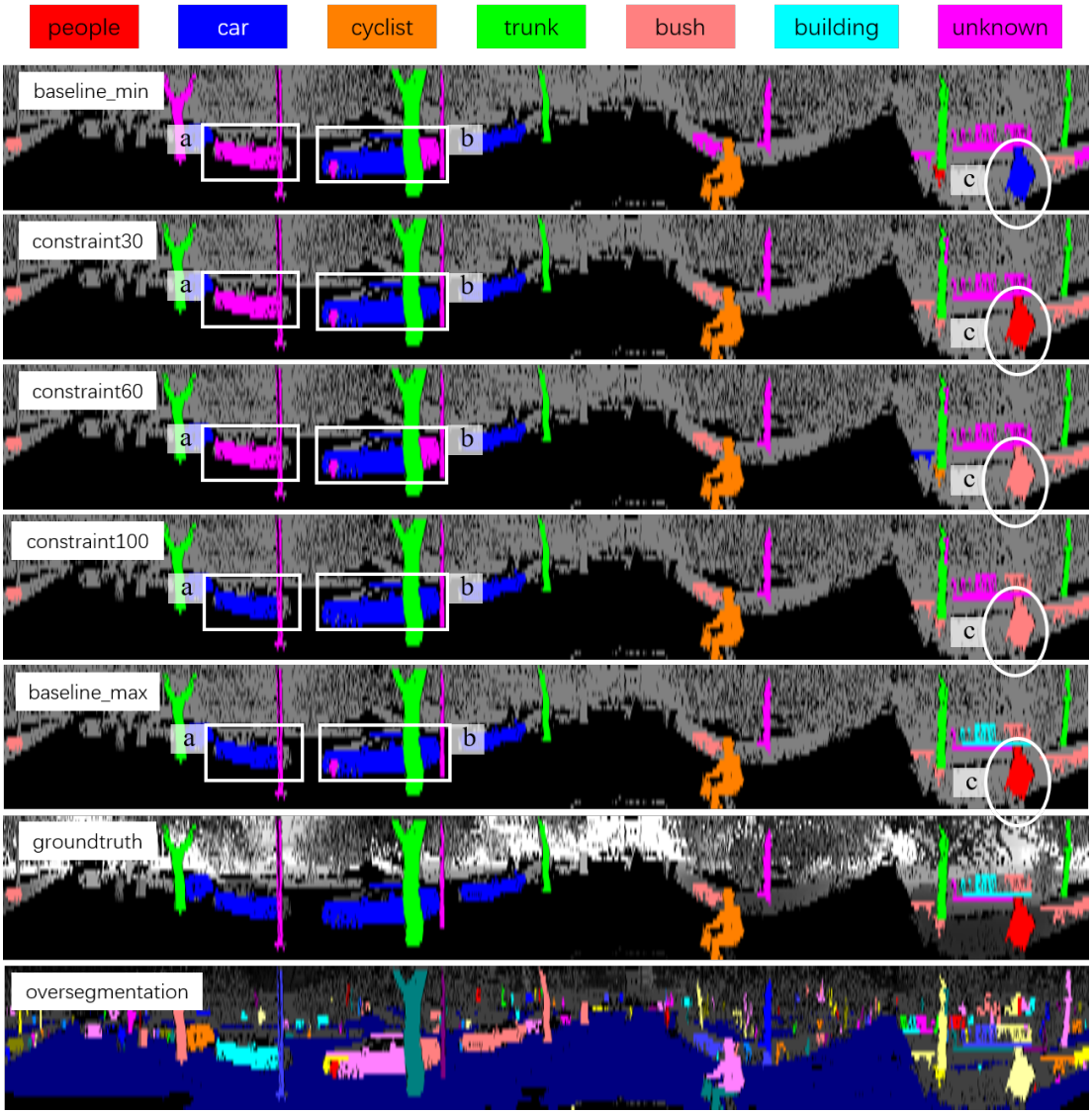

Fig. 6. The qualitative results on training data. The rectangles a and b show that the constraint100 has better performance on car, and the ellipse c shows that the constraint100 makes an error classification when the people near the bush.

TABLE V
F MEASURE ON TRAINING DATA.

learning method	classifier	people	car	cyclist	trunk	bush	building	unknown
sup.	baseline_min	77.2	68.5	66.1	81.0	56.2	78.7	33.9
semi-sup.	constraint30	77.2	83.0	69.6	86.6	61.9	81.8	36.8
semi-sup.	constraint60	81.5	85.8	71.3	90.4	80.7	86.6	48.3
semi-sup.	constraint100	**87.0**	**90.4**	**77.8**	**90.6**	**81.2**	**87.7**	**52.2**
sup.	baseline_max	**93.3**	**96.4**	**83.8**	**96.3**	**92.3**	**94.3**	**80.0**

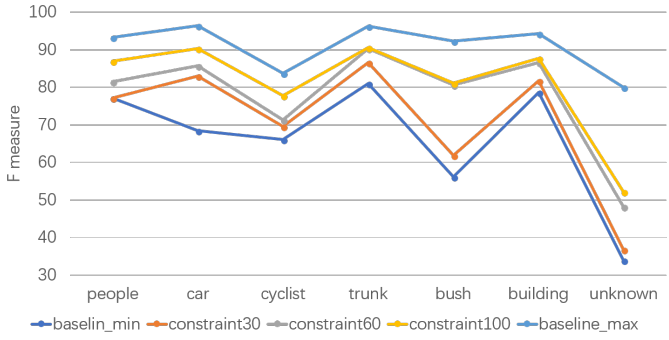

Fig. 7. The quantitative comparison on training data.

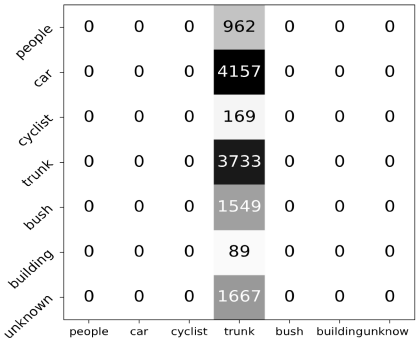

Fig. 8. The confusion matrix of unsupervised learning on testing data.

as the amount of constraint increases, the performance of the classifier is enhanced. A more intuitive comparison is illustrated in Fig. 7. Although the constraint100 is not as good as the baseline_max where all annotations are used, it shows promising results that adding constraint improves the performance of the classifier and reduces the need for annotations. In a short conclusion, semi-supervised learning where only a few annotations are used is effective on the 3D point cloud semantic segmentation.

C. Adaptability on Testing Data

In the fixed scene or data set, the classifier based on supervised learning can easily achieve high performance due to the overfitting; when it is applied in a new scene, additional annotations are necessary to prevent the performance decreasing. However, it is difficult to make largely fine annotations in each new scene for driving applications. Thus, how to enhance the adaptability with a few or without new annotations is crucial for practical applications. We detail three experiments in the following to demonstrate the adaptability of the proposed semi-supervised method. For the quantitative comparison, the F-measure in Equation (11) is used.

1) The pre-trained result: The pre-trained classifiers based on training data are directly applied on the testing data, and the result is shown in TABLE. VI. Comparing the baseline_min and constraint100, it shows that adding constraint improves the adaptability to the new scene; and the F-Score of the constraint100 goes up by 9% on average. Comparing the constraint100 and baseline_max, the constraint100 has higher scores at cyclist and trunk categories, although it is lower in all categories on training data (TABLE. V).

2) The unsupervised result: For the new scene, it is an interesting attempt if only constraints, i.e., without new annotations, are used to improve the pre-trained classifier. Thus, the loss function in Equation. (8) is rewritten as:

$$L(\theta; X_l, Y, X_c) = L_c(X_c; \theta) = \frac{1}{N}\sum_{i=1}^{N}(1 - \sum_{k=1}^{K} P_\theta^k(x_j^t) P_\theta^k(x_j^{t+1})). \quad (12)$$

Here, the pre-trained constraint100 is treated as the initial classifier, then only constraints are used to re-train the model. After this unsupervised learning procedure, the result is shown in Fig. 8. We find that no matter what samples are taken, the classifier always erroneously assigns them as the trunk. We can explain this situation with the loss function in Equation. (12): the constraint only penalizes classifier when it gives x_j^t and x_j^{t+1} with different labels, so the unsupervised learning fails.

3) The fine tuning result: After finding the unsupervised learning does not work for our method, we attempt to add a few new annotations. Specially, the new annotations are generated in an interactive way. The pre-trained constraint100 is also treated as the initial classifier, then it produces the classification result on testing data. Although the pre-trained results show a low F-Score in TABLE. VI, a few new annotations can be selected by human confirmation. In this way, we get 20 annotations for each category and 100 for the unknown as shown in TABLE. VII, where the new annotation is renamed as anchor sample.

Combining the anchor sample and constraint, the pre-trained constraint100 model is fine tuned. the final quantitative result is shown in TABLE. VIII and Fig. 9. Comparing the

TABLE VI
THE PRE-TRAINED CLASSIFIERS ON TESTING DATA.

learning method	classifier	people	car	cyclist	trunk	bush	building	unknown
sup.	baseline_min	57.6	64.0	26.3	37.5	33.2	49.7	33.0
semi-sup.	constraint100	**69.4**	**81.6**	**39.8**	**42.1**	**37.3**	**55.7**	**39.0**
sup.	baseline_max	**73.7**	**88.8**	**30.0**	**30.8**	**71.0**	**65.4**	**53.0**

TABLE VII
THE SETTING OF FINE TUNING

-	people	car	cyclist	trunk	bush	building	unknown
anchor sample	20	20	20	20	20	20	100
constraint	925	3542	142	62	2440	1254	0

TABLE VIII
F MEASURE ON TESTING DATA

learning method	classifier	people	car	cyclist	trunk	bush	building	unknown
sup.	baseline_min	57.6	64.0	26.3	37.5	33.2	49.7	33.0
semi-sup.	constraint100	69.4	81.6	39.8	42.1	37.3	55.7	39.0
semi-sup.	constraint100+fine tuning	**77.1**	**90.7**	**60.2**	**55.7**	**58.0**	**62.3**	**54.3**
sup.	baseline_max	**73.7**	**88.8**	**30.0**	**30.8**	**71.0**	**65.4**	**53.0**

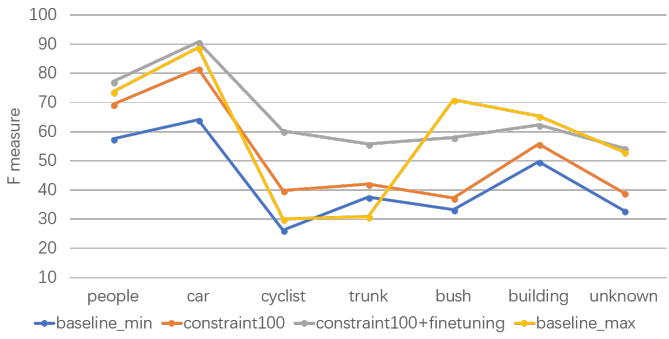

Fig. 9. The quantitative comparison on testing data.

constraint100 and the fine tuning version, the F-Score of the latter one goes up by 13% on average, which shows that fine tuning is effective to improve the adaptability even in a semi-supervised way. Comparing the baseline_max and the fine tuning version, the latter one has higher scores except on bush and building. The qualitative result is shown in Fig. 10. In a short conclusion, fine tuning with anchor sample makes the classifier better adaptability to the new scene.

VI. CONCLUSION AND FUTURE WORK

A semantic segmentation method for 3D point clouds(i.e., from LiDAR senor) is developed in this research, where the semi-supervised learning and neural network are combined to reduce the large requirement of fine annotations. The pairwise constraints between adjacent frames are generated with temporal information, and a special loss function is designed to encourage the constraint data obtaining the same label. This method is examined extensively on a new dataset. Superior results show the improvements on effectiveness and adaptability. Future work will be addressed on how to define the sample, as containing both foreground and background in the sample sometimes makes the classifier confused. In addition, introducing new constraints also will be studied.

REFERENCES

[1] D. Munoz, N. Vandapel, and M. Hebert, "Onboard contextual classification of 3-d point clouds with learned high-order markov random fields," in *IEEE International Conference on Robotics and Automation*. IEEE, 2009.

[2] H. Hu, D. Munoz, J. A. Bagnell, and M. Hebert, "Efficient 3-d scene analysis from streaming data," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2297–2304.

[3] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.

[4] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "Whats the point: Semantic segmentation with point supervision," in *European Conference on Computer Vision*. Springer, 2016, pp. 549–565.

[5] T. Lange, M. H. Law, A. K. Jain, and J. M. Buhmann, "Learning with constrained and unlabelled data," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 731–738.

[6] Z. Lu and T. K. Leen, "Semi-supervised learning with penalized probabilistic clustering," in *Advances in Neural Information Processing Systems*, 2005, pp. 849–856.

[7] M. Weinmann, B. Jutzi, and C. Mallet, "Semantic 3d scene interpretation: A framework combining optimal neighborhood size selection with relevant features," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3, pp. 181–188, 2014.

[8] T. Hackel, J. D. Wegner, and K. Schindler, "Fast semantic segmentation of 3d point clouds with strongly varying density." *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III-3, pp. 177–184, 2016.

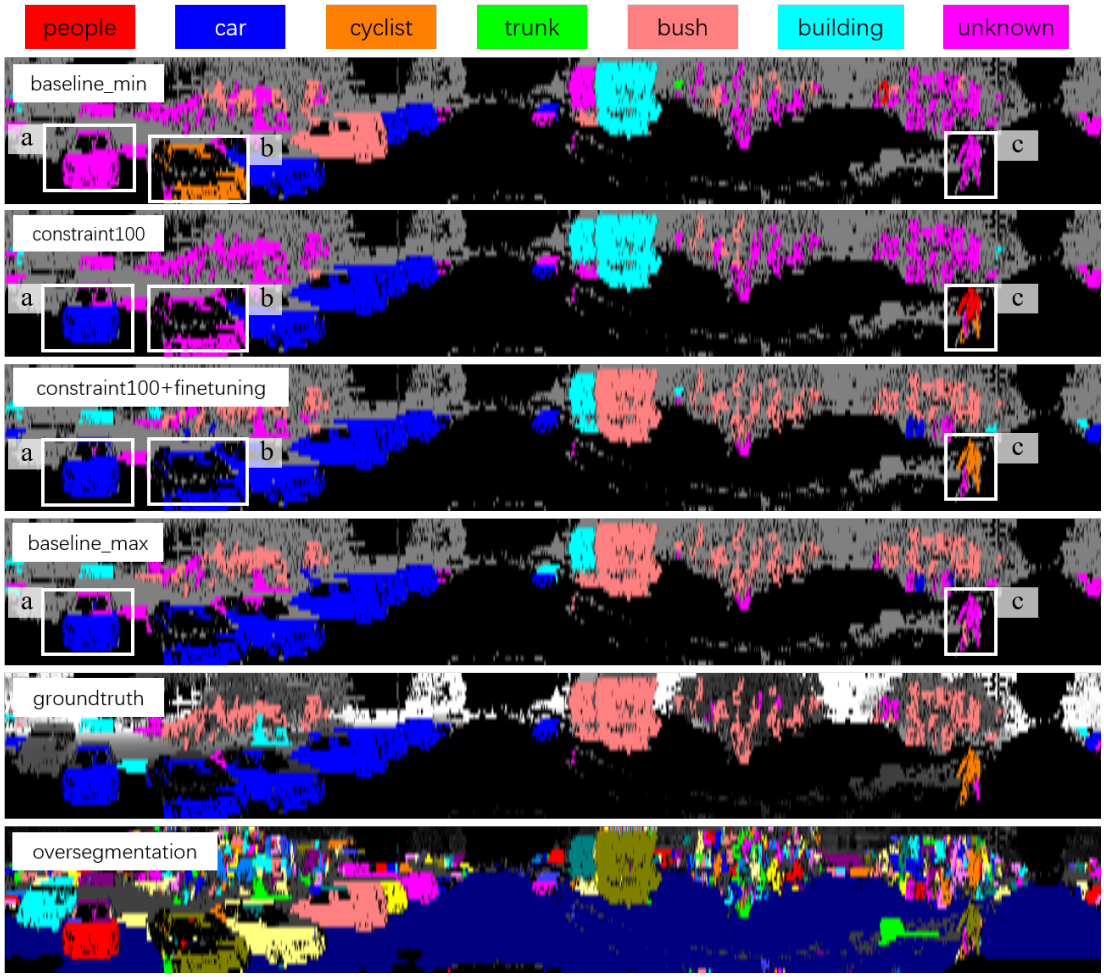

Fig. 10. The qualitative results on testing data. The a and b show the comparison on car and c shows that the cyclist is successfully classified after fine-tuning.

[9] H. Zhao, Y. Liu, X. Zhu, Y. Zhao, and H. Zha, "Scene understanding in a large dynamic environment through a laser-based sensing," in *IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 127–133.

[10] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 975–982.

[11] Y. Lu and C. Rasmussen, "Simplified markov random fields for efficient semantic labeling of 3d point clouds," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2690–2697.

[12] S. Guinard and L. Landrieu, "Weakly supervised segmentation-aided classification of urban scenes from 3d lidar point clouds," in *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. XLII-1/W1, 2017, pp. 151–157.

[13] M. Najafi, S. T. Namin, M. Salzmann, and L. Petersson, "Non-associative higher-order markov networks for point cloud classification," in *European Conference on Computer Vision*. Springer, 2014, pp. 500–515.

[14] X. Xiong, D. Munoz, J. A. Bagnell, and M. Hebert, "3-d scene analysis via sequenced predictions over points and regions," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2609–2616.

[15] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3d semantic segmentation of point clouds," in *IEEE International Conference on Computer Vision Workshops*. IEEE, 2017, pp. 716–724.

[16] J. Behley, V. Steinhage, and A. B. Cremers, "Performance of histogram descriptors for the classification of 3d laser range data in urban environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4391–4398.

[17] A. Serna, B. Marcotegui, F. Goulette, and J.-E. Deschaud, "Paris-rue-madame database: a 3d mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods," in *International Conference on Pattern Recognition, Applications and Methods*, 2014.

[18] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *International Conference on Computer Vision and Pattern Recognition*. IEEE, 2016.

[19] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *International Conference on Computer Vision and Pattern Recognition*. IEEE, 2016.

[20] P. Tosteberg, "Semantic segmentation of point clouds using deep learning," Master's thesis, Linköping University, 2017.

[21] L. University, "Virtual photo sets," http://www.hdrv.org/vps/, accessed June 10, 2018.

[22] G. L. O. a. B. Ayush Dewan, "Deep semantic classification for 3d lidar data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017, pp. 3544–3549.

[23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[24] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "SEMANTIC3D.NET: A new large-scale point cloud classification benchmark," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1-W1, pp. 91–98, 2017.

[25] B. Wu, A. Wan, X. Yue, and K. Keutzer, "Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud," *arXiv preprint arXiv:1710.07368*, 2017.

[26] F. Piewak, P. Pinggera, M. Schäfer, D. Peter, B. Schwarz, N. Schneider,

D. Pfeiffer, M. Enzweiler, and M. Zöllner, "Boosting lidar-based semantic labeling by cross-modal training data generation," *arXiv preprint arXiv:1804.09915*, 2018.
[27] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using fully convolutional neural networks," in *IEEE Intelligent Vehicles Symposium*. IEEE, 2017, pp. 1019–1024.
[28] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg, "Deep projective 3d semantic segmentation," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2017, pp. 95–107.
[29] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," *arXiv preprint arXiv:1710.07563*, 2017.
[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.
[31] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 3. IEEE, 2017, pp. 6620–6629.
[32] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *IEEE conference on computer vision and pattern recognition*. IEEE, 2015, pp. 1912–1920.
[33] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 77–85.
[34] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas *et al.*, "A scalable active framework for region annotation in 3d shape collections," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 210, 2016.
[35] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 1534–1543.
[36] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," *arXiv preprint arXiv:1711.09869*, 2017.
[37] R. Yan, J. Zhang, J. Yang, and A. G. Hauptmann, "A discriminative learning framework with pairwise constraints for video object classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 578–593, 2006.
[38] M. Bäuml, M. Tapaswi, and R. Stiefelhagen, "Semi-supervised learning with constraints for person identification in multimedia data," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 3602–3609.
[39] S. Hong, H. Noh, and B. Han, "Decoupled deep neural network for semi-supervised semantic segmentation," in *Advances in Neural Information Processing Systems*, 2015, pp. 1495–1503.
[40] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
[41] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *IEEE International Conference on Computer Vision*. IEEE, 2015, pp. 1742–1750.
[42] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 3159–3167.
[43] T. Cour, B. Sapp, C. Jordan, and B. Taskar, "Learning from ambiguously labeled images," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 919–926.
[44] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *IEEE International Conference on Computer Vision*. IEEE, 2015, pp. 1796–1804.
[45] J. Xu, A. G. Schwing, and R. Urtasun, "Learning to segment under various forms of weak supervision," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 3781–3790.
[46] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2368–2382, 2011.
[47] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *IEEE International Conference on Computer Vision*. IEEE, 2015, pp. 1635–1643.
[48] "velodyne lidar," http://www.velodynelidar.com/, accessed June 12, 2018.

APPENDIX

Algorithm 2 Oversegmentation

Input: D, A, $maxnum$, gnd_thres,bkg_thres,$mindst$, α
Output: the label matrix B

$L = \{Unvalid, Ground, Background, Unknown, EdgePt\}$
$S \leftarrow CoarseSeg(D, gnd\_thres, bkg\_thres, L)$
$B \leftarrow EdgeExtraction(A, S, mindst, \alpha, L)$
$B \leftarrow RegionGrow(B, L, maxnum)$
return B

Algorithm 3 CoarseSeg

Input: D, gnd_thres, bkg_thres, L
Output: the indicator matrix S

Initialize S with $Unknown$
$G \leftarrow GroundExtraction(D)$
for all d_k **in** D **do**
  $d \leftarrow dis(d_k, G)$
  if $d < gnd\_thres$ **then**
    $S(s_k) \leftarrow Ground$
  else if $d > bkg\_thres$ **then**
    $S(s_k) \leftarrow Background$
  else if I_w^k**is**0 **then**
    $S(s_k) \leftarrow Unvalid$
  end if
end for
return S

Algorithm 4 EdgeExtraction

Input: A, S, $mindst$, α, L
Output: the label matrix B

Initialize B with $Unknown$
for all a_k **in** A **do**
  the 4-neighbour of a_k is N^k
  $valid = 0, d = -1$
  for all p_n^k **in** N^k **do**
    if s_k **in** $\{Unvalid, Ground, Background\}$ **then**
      $valid++$
    else
      $d \leftarrow max[d, dis(p_n^k, a_k)]$
    end if
  end for
  if $valid > 1$ **or** $d >= max(mindst, \alpha * rng_k)$ **then**
    $B(b_k) \leftarrow EdgePt$
  end if
end for
return B