Proceedings of the 2017 IEEE
International Conference on Robotics and Biomimetics
December 5-8, 2017, Macau SAR, China

Simultaneous Clustering Classification and Tracking on Point Clouds using Bayesian filter

Sukai Wang, Huaiyang Huang, Ming Liu

***Abstract*— Simultaneous Clustering, Classification and Tracking (SCCT) maintains many challenges, especially for point cloud data. SCCT is an essential process to facilitate the autonomous mobile systems. We present a novel unified framework from the object extraction to tracking with real-time performance. The framework can be described as five sub-tasks: ground extraction, clustering, recognition, tracking and representation. We compare the adjacent two frames to solve dense tracking and motion estimation. The state of each clustered object (moving or static) is estimated by using Spatial-Temporal methods. The distinguish objects with different features are extracted. Conditional Random Field and Bayesian filter are adopted to solve the data association problem. All the algorithmic modules have been tested on both outdoor actual environments and indoor simulation situations. The results indicate the efficiency and effectiveness of the proposed method.**

I. INTRODUCTION

A. *Motivation*

To navigate autonomously in urban environments, the vehicle need to be able to control, perceive and especially interact with other traffic and environment. These are essential factors for autonomous execution of various urban driving skills [1] [2]. A complete autonomous driving case is a commonly envisaged and yet challenging goal nowadays, for which a reliable perception of the local environment is crucial [3] [4]. Distinction of the pedestrain, car, bicycle and other obstacles from the whole environment is the key task for intelligent vehicles, to estimate the risks on surrounding space for optimal path choosing, with temporal and spatial modeling of the environment. Compared with other common sensors like camera or ultrasonic, the Laser range scanners (aka. 3D-LiDARs) are more popular in vehicle market. Besides, with the recent development in the key technology of 3D-LiDAR and the appearance of more affordable solid state LiDAR, 3D-LiDAR will be standard equipment in every unmanned vehicle in the near future. 3D-LiDAR sensing is becoming more and more prevalent.

With the advent of low-cost 3D sensing hardware such as Kinect, and continued evolution of advanced point-cloud processing method, 3D perception plays a more significant role in robotics, as well as other fields [5] [6]. This paper presents a framework contains a series of processes, namely Simultaneous Clustering Classification and Tracking (SCCT), on 3D point-cloud data captured by 3D LiDAR.

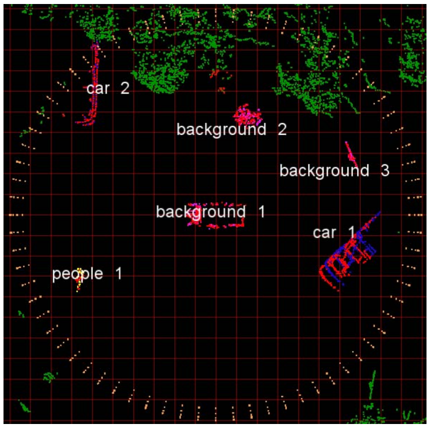

Fig. 1. Clustering and Classification results of a typical 3D-LiDAR point-cloud sequence. Red color points belong to the new frame, and points with other color belong to the old frame which has 0.1 second delay compared with new frame. In the old frame, pedestrian marked in yellow, backgrounds marked in pink, cars marked in blue, and green points are nonsensical points.

SCCT is an essential yet tough problem especially because of the real-time requirement. A related problem namely Simultaneously Clustering and Recognition (SCAR) was presented by us in [7], where a lightweight scene recognition method using color features and geometric information is presented. It enables a mobile robot to solve the localization problem in topological regions real-timely. So, the simultaneous processing method in this paper takes some references to the previous work but form-up a more practical solution to point-cloud data.

Detecting an object in an image or a video is one of the foremost steps before proceeding to understand the semantics of the scene [8]. Tracking every target object over the a series of consecutive frames provides us with relevant information that helps the scenario situation awareness. It is similar to the problem of detection and tracking of moving objects (DTMO) in this paper. However, the processing of the point-cloud is different from the processing of image or video because of the sparse representation. As a result, extracting features from a point-cloud is much more difficult than that in images or videos.

In the past few years, several methods, like feature comparing or Iterative Closest Point (ICP) algorithm [9] have been proposed to solve DTMO problem In this paper, Bayesian filter, SpatioTemporal Conditional Random Field

Sukai Wang is with College of Biomedical Engineering and Instrument Science, Huaiyang Huang (hyhuang1995@gmail.com) is with College of Mechanical Engineering, Zhejiang University, Ming Liu (eelium@ust.hk) is with Robotics and Multi-Perception Lab (RAM-LAB), Department of Electronics and Computer Engineering and the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, Hong Kong SAR, Ming Liu is also with the City University of Hong Kong Shenzhen Research Institute, China

978-1-5386-3742-5/17/$31.00 © 2017 IEEE

(STCRF) [10] [11] [12] and Markov Localization are the main models to solve the core problem.

Compared with the other methods, Bayesian filter has special advantages. First of all, Bayesian filter is more general than Hidden Markov Model (HMM) and it can work with continuous state space. By using Bayesian filter, the features of the whole scene are withdrawn or it can be deemed that the feature influence is weakened so that the difficulty of the feature extraction can be reduced. Except that, the real-time performance of the calculating processing can be achieved with satisfying all the other requirements.

B. *Contributions*

We emphasis the following contributions in this paper:

- *Bayesian filter:* We presented a Bayesian filter-based framework for the tracking and association. It further reduces the inappropriate probability caused by the incorrect judgment due to the inaccuracy of the sensor and defective judgment basis. The probability distribution obtained in the previous time step can be used to improve the accuracy of the current probability distribution.
- *Conditional Random field:* CRF is used to massage the boundary and smooth the discontinuous point-cloud, keeping away from the situation when one object split into two parts or two objects are merged.
- *Simultaneous complete processing:* A complete SCC-T solution is proposed including the processes from clustering to tracking. With raw input of point-cloud sequence, it outputs the position and history track of each objects from its sparse point-cloud representation.

C. *Organization*

In the next section, the main models were designed, which provide a better way to represent the point-cloud, and to segment and estimate the classification of each point. Section III describes the processing pipeline of the whole process chain and represents the experiment results. The analysis and conclusions are presented in section IV. Section V introduces the environment and condition about the supplementary video.

II. Proposed Model

We show the proposed framework in this section. With the point-cloud observations by 3D LiDAR in dynamic environments, we try to extract and track salient objects, with outputting the probability of each object class. Fusing with the LiDAR-based motion estimation, we can estimate the optimal motion update of the actual object as well.

Because the data coming from the LiDAR sensor are affected by measurement errors, obstacle and missing points result in the uncertain probability in a given configuration. Instead of giving a single best estimate of the current configuration, this model represents the robot configuration as a probability distribution over the all possible classification situation.

A. *Representation*

Wang and Ji in [13] proposed a novel approach based on conditional random field (CRF) models that integrate temporal and spatial constraints for object segmentation applying to highly dynamic scenes including both fast camera and object movements.

Inspired by that work, the ground surface is modeled as CRF and represented by a regular 2D lattice on XY plane in this paper. Classification and segmentation are apparently more rational and valuable in three-dimension. In the more general planar motion case the grid-map is a three-dimensional array where each cell contains the probability of the four classes in that cell. In this case, the cell size must be chosen carefully. Because the search region is restricted in a 10 meters radius sphere, we decompose the target space as a $200 \times 200 \times 10$ grid.

With such a grid representation, each node in the plane $N_i = (x_i, y_i)$ associates a random variable $G_i = (Ppe_i, Pca_i, Pba_i, Pot_i)$ representing the four probability of the four classes, pedestrian, car, background and other. And $G = (G_i, \cdots, G_n)$ means the set of all such variables in the lattice, where n is the number of nodes. The LiDAR measurement consists of a set of 3D locations $(x_j, y_j, z_j) \in \mathbb{R}^3$. In spatial level, every measurement N_i is associated with the closest neighbourhood. The set of observation indexes associated with the node N_i is noted C_i. In temporal level, ${G_i}^{t-1}$ represents the random variable probability associated with node N_i at previous time step. Measurement data is z_t, prediction data is u_t. Belief is the probability of being at the current environment situation x_t. Prediction update computes a new belief through previous probability distribution and prediction movement. Measurement update computes a new belief after computing the probability through Spatial Temporal method.

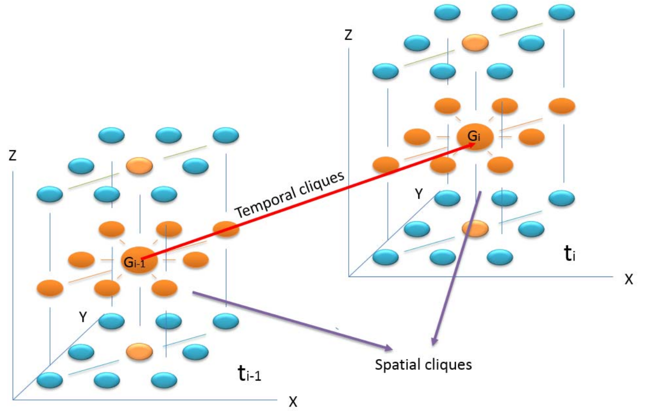

Fig. 2. Three-dimension Spatio Temporal Conditional Random Field model.

A three-dimension CRF approximation is used to model the interaction between the nodes. A variable G_i is considered to only depend on its spatial neighborhood $\{G_k \mid k \in N_i\}$ (spatial clique) and its temporal neighborhood G_i^{t-1} (temporal clique). A graphical representation is shown on Fig. 2, where the potential function over the cliques are defined as:

$$\phi(g, g') = e^{-(d_g - d_{g'})^2} \tag{1}$$

where d is the density of each cell g or g'.

B. Inference over graph with Bayesian Filtering

In general, the state at time t is generated stochastically. Thus, it makes sense to specify the probability distribution from which state x_t is generated. The probabilistic law characterizing the evolution of state might be given by a probability distribution of the following form:

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}) \tag{2}$$

where z is the observations, which is the density of each cell (see: eq.(1)); x is the state of the cell, which includes the object status, including its existence, class, velocity and tracking result.

- measurement update
  In this phase, the LiDAR sensor updates its previous position, in other words, after a series computation, it updates its former belief.
  Depending on the Theorem of Total probability, the probability belief can be described as:

$$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \tag{3}$$

- prediction update
  In this phase the robot corrects its previous position (i.e. its former belief) combining the space probability distribution in the previous times step with the information from the record which recording the objects movement and position, each object can correct its previous position and update the probability distribution.

$$p(x_t \mid z_t) = \frac{p(z_t \mid x_t) p(x_t)}{p(z_t)} \tag{4}$$

For all x_t, do measurement update and prediction update and return the belief x_t names Bayesian Filter in general.

Because of the extreme difficulty in calculating $p(x_t \mid z_{0:t}, u_{0:t})$, by associating Bayesian theorem, $p(x_t \mid z_{0:t}, u_{0:t})$ can be transformed into a series problems which can be solved more easily:

$$\begin{aligned} P(x_t \mid & z_{0:t}, u_{0:t}) \\ & \propto P(z_t \mid x_t) \\ & \times \sum_{x_{t-1}} P(x_t \mid x_{t-1}, u_{t-1}) P(x_{t-1} \mid z_{0:t-1}, u_{0:t-1}) \end{aligned} \tag{5}$$

C. Initialization

In the first frame of the dataset, the information about the classification configuration is null, so the probability distribution satisfies the uniform distribution $\int_{-\infty}^{+\infty} p(x)dx = 1$. After that, in every time step each lattice which contains one kind of label is associated with Gaussian distribution, uniting the closest neighbourhood in spatial and temporal. The distribution can be stated as:

$$p(x) = det(2\pi\Sigma)^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{6}$$

where the mean $\mu = 0$ and the covariance $\Sigma = 2$ empirically.

This model is applied to blur boundary in section 3.3, also it is used as a high-reliability classification method. From the first frame of the data set, the classification probability distribution is continuously updated. Because of the additional influence of the probability distribution in previous time step, the current probability distribution will lead to a more accurate classification result.

III. EXPERIMENTS

A. Experimental Platform

For the experiments, a testing car in Fig. 3 has been equipped with two 3D-LiDARs on the top. One is Velodyne LiDAR VLP-16 (16 laser beams), and the other one is 2D-sick LiDAR, which is a real-time 2D LiDAR for collision avoidance. Meanwhile, IMU, GPS, one occam stereo camera and two LiDARs sensors' data are collected by IPC (i7 intel nuc) with ROS system. The testing vehicle's max speed is 24 km/h.

In this paper, the process was applied to real data set acquired by Velodyne VLP-16 (dense data) and simulation data set acquired by V-REP.

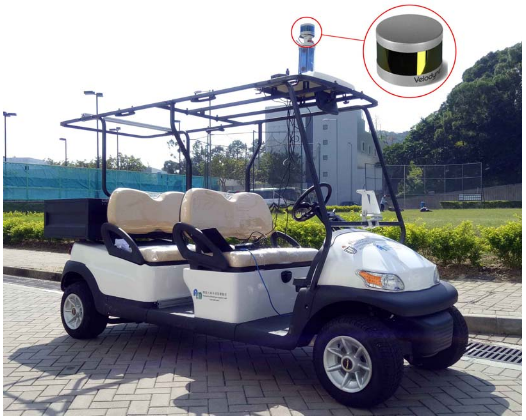

Fig. 3. Experimental Platform: Unmanned vehicle with Velodyne LiDAR VLP-16, 2D-sick LiDAR, IMU, GPS, OCCAM stereo camera and IPC (i7 intel nuc).

B. Process and Solutions

The experiment pipeline is described in Fig. 4. The proposed model is applied over a long series of captured data from outdoor environments. For detail of the dataset, please refer to http://ram-lab.com/download.

Nowadays, object detection methods based on 3D data are so far optimized for either unstructured off-road environments or flat urban environments [14] [15]. Random sample consensus (RANSAC) algorithm (plane model) is used to extract the ground in the first part of the process. The maximum iterations time and distance threshold are the most influential parameters in the algorithm. Because of the

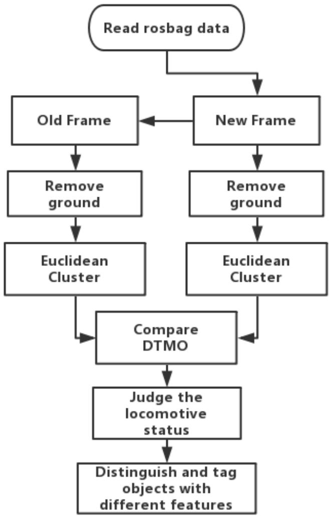

Fig. 4. Flowchart of the whole process.

uneven of the ground, sometimes there are some missing points in the ground which will disturb the clustering result in next steps. So, to optimize the filter result, more than once plane extractions are used in the process. The result of the ground extraction is illustrated in Fig. 5. Only a few points in objects' root part are classified as ground. Almost all points belong to ground are in the right classification.

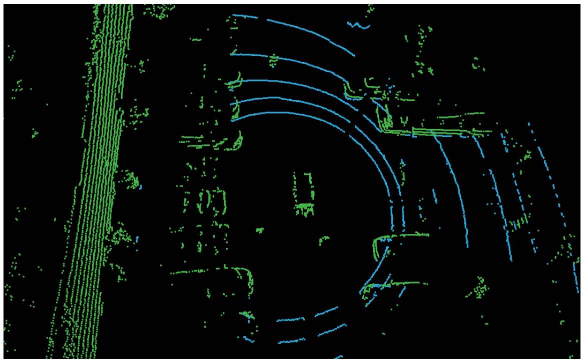

Fig. 5. Segmentation result after extract the ground in a complex environment. Ground points are in blue, non ground points are in green.

Segmentation has been studied in computer vision which is often formulated as graph clustering for several decades. The segmentation of LiDAR point-clouds is a key step for 3D reconstruction of architecture, but it is also a complicated step. Ground extraction is the prerequisite step. The point-clouds obtained by laser scanners involve coordinates and intensity. Since all these data are the components of the Euclidean space, we can use Euclidean clustering method, which utilize the spatial similarity, to segment the point and cluster the raw point set into several different point groups.

Fig. 6 shows the result of Euclidean algorithm. Different independent objects are marked with different colors. Most of the independent objects are separated, though if two independent objects' distance is less than the cluster tolerance distance set in the algorithm, they will be treated as one object.

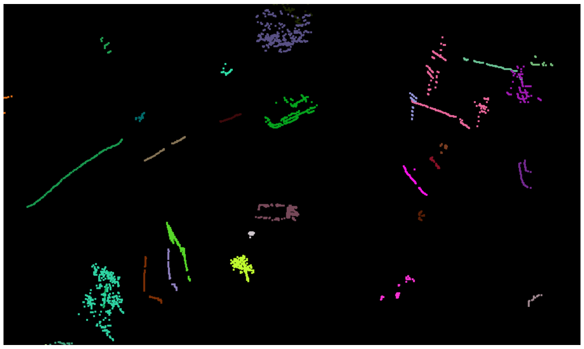

Fig. 6. A clusterization result. Each color represents a separate cluster. The colors do not have any particular meaning, they were assigned randomly.

The index of each object can be attached after segmenting all independent objects above the ground. In this paper, only eligible objects, whose center of gravity is in one meter around the LiDAR sensor center, will be considered and classified. By this way, most of trees and mountains are excluded in the classification process, in case thousands of nonsensical points slow down the speed of processing and disturb the detection and tracking result.

Except for the start time and end time, most of the objects should not appear or disappear suddenly inside the scene. If the sensor's collecting frequency is fast enough, and the objects' relative velocities are not too fast, the same object's position in two adjacent frames will not far apart from each other.

If the rotation velocity of the LiDAR sensor is not too fast, characteristic judgment is enough to find the corresponding point-cloud in two adjacent frames. Otherwise, Iterative Closest Point (ICP) transformation array can give another judgment basis.

However, because of the inaccuracy of the LiDAR sensor and uncertain reflection surface, sometimes, a big object splits into two sub-objects like Fig. 7(b). Also, if the distance between two moving objects is smaller than the Euclidean clustering's threshold like Fig. 7(e)(f)(g), these two objects will be considered as one object. These situations result in mismatching of the objects. The Conditional Random Field and Bayesian filter modeled in section II provide a way to solve this problem, by reducing the influence of error feature judgment.

A four-dimensional vector is created to store the record of each object's trajectory. Using the information in the record and the DTMO result, distinguish and tag different types of objects are the last part of the process. Fig. 1 and Fig. 8 are the screenshots of the running result. In these figures, yellow circle means processing range, only the objects which are in this range will be classified and tagged. Blue means car, yellow means people, pink means background, other-class does not appear in this scene, and green points group means these points are not considered in the process. The red point-clouds are the double images in the new frame, others (with

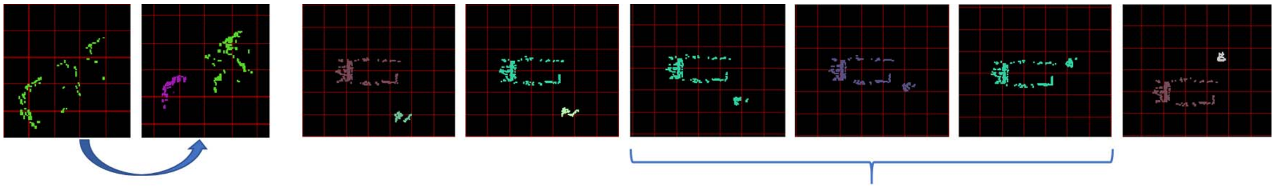

Fig. 7. (a)(b) A integral car split into two parts. (e)(f)(g) A people walk to the side of the car, then regarded as one object during this time (Same color means these points belong to same object).

different color and classification) are the image in the old frame. Because there are two frames in one screenshot, each object has two set of points. One is in the new frame and the other is in the old frame. So, the farther apart of one objects double, the faster of the objects speed.

IV. CONCLUSIONS AND FUTURE WORK

This paper presented a 3D point-cloud classification algorithm, that is capable of distinct people, car and background in outdoor and indoor environments. The space is modeled as three-dimension Spatio-Temporal Conditional Random Field and each lattice of the space is given a probability. By update the environmental probability distribution, Bayesian filter helps to achieve a high-reliability classification. Experimental result presented promising performances in uneven terrain, complex outdoor circumstance with slower angular velocity of rotation, and indoor environment with bigger angular velocity of rotation.

Future work will consist of a 3D point-cloud classification method with labeled points, meant as input for learning methods [16] [17], and applying the classification result to the real control of the unmanned vehicle.

The advantages and disadvantages are summarized as follows:

- Speed
  The presented algorithm has been implemented in C++ without using GPU accelerated. Robot Operating System (ROS) is used to preserve the data collected in field and replay the data while program performing. PCL is used for point-cloud processing, which is a free, open source C++ library that presents an advanced and extensive approach to the subject of 3D perception. And it's meant to provide support for all the common 3D building blocks that applications need [5].

TABLE I

	Frequency
Data package rate	9.8 Hz
Avg. rate of performance	1.65 Hz
Avg. rate of performance Without ICP	2.85 Hz
Avg. rate of performance Without RANSAC	2.7 Hz
Avg. rate of performance Without RANSAC and ICP	8.97 Hz

  Computational performance of the algorithm with different situation, like algorithm without ICP processing or without RANSAC processing, is as shown in Table I. The frequency result shown in the table indicate that the ICP and RANSAC functions consume most of time. In this paper's computational process, PCL library in ROS package has been used. And the PCL library in Libpointmatcher in Github will provide a better and more efficient library functions.
- Accuracy
  In this circumstance with static testing vehicle and moving pedestrian, the algorithm can detect and track almost all objects in about 90 percent of time when the LiDAR sensor is static. (When the LiDAR sensor is moving, the discrimination will lower to about 70 percent.) Nevertheless, the real error discrimination comes with the data have classification tags. So, the next step, KITTI dataset will be used into algorithm like in [18].
- Parameter Update
  Though the parameter in this algorithm is easily modified manually, like adding more rules to judge whether two point-clouds' features are same or not, robustness of this algorithm is also a problem need to fix. If the Accuracy part above can provide the feedback, the parameters can be updated by learning.

V. VIDEO SUPPLEMENT

In the supplementary video, we demonstrate the algorithm's result in various environment and different LiDAR sensor movement status.

1) The first section in the video illustrates the situation that the testing vehicle parked in the export of the parking lot and some pedestrian and car passed it. The rest of the video is recorded while the LiDAR sensor is moving with the testing vehicle.
2) The second section illustrates that the testing car drove along the road and distinguished the car and background.
3) The third section illustrates that the testing vehicle turned a big angle around the corner.
4) The fourth section illustrates that the testing vehicle drove through the parking lot slowly in an intricate environment.

ACKNOWLEDGMENT

This work is supported by Shenzhen Science, Technology and Innovation Comission (SZSTI)

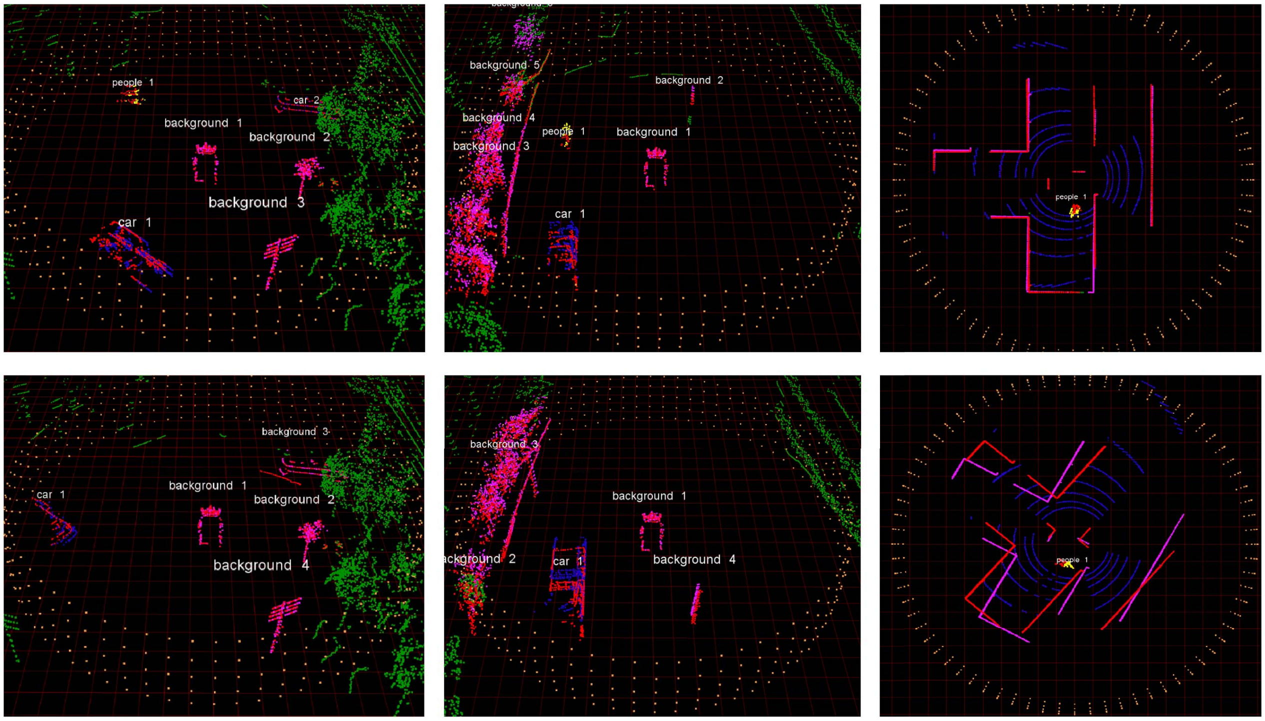

Fig. 8. 3D point-cloud classification results in different environments. Red color points belong to the new frame, and points with other color belong to the old frame. (first column) Static testing vehicle parks in an open rural place, with backgrounds, a car and a trunk. (second column) Driving testing vehicle drives in a broad road, passing car, trunk and bicycle. (three column) Moving TurtleBot robot in indoor situation with big angular velocity of rotation.

JCYJ20160428154842603 and JCYJ20160401100022706, awarded to Prof. Ming Liu; National Natural Science Foundation of China No. 61403325, awarded to Prof. Ming Liu.

References

[1] Ming Liu and R. Siegwart. Navigation on point-cloud - a riemannian metric approach. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4088–4093, May 2014.

[2] Mario Gianni, Panagiotis Papadakis, Fiora Pirri, Ming Liu, Francois Pomerleau, Francis Colas, Karel Zimmermann, Tomas Svoboda, Tomas Petricek, Geert-Jan M Kruijff, et al. A unified framework for planning and execution-monitoring of mobile robots. *Automated action planning for autonomous mobile robots*, 11:09, 2011.

[3] Christoph Stiller, Philippe Martinet, Christian Laugier, Urbano Nunes, and Philippe Bonnifait. Perception and planning for autonomous vehicles [guest editorial]. *IEEE Intelligent Transportation Systems Magazine*, 7(1):6–7, 2015.

[4] Ming Liu, Lujia Wang, and Roland Siegwart. DP-Fusion: A generic framework for online multi sensor recognition. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2012.

[5] M. Liu. Robotic online path planning on point cloud. volume 46, pages 1217–1228, May 2016.

[6] Francis Colas, Srivatsa Mahesh, François Pomerleau, Ming Liu, and Roland Siegwart. 3d path planning and execution for search and rescue ground robots. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 722–727. IEEE, 2013.

[7] Ming Liu and Roland Siegwart. Topological mapping and scene recognition with lightweight color descriptors for an omnidirectional camera. *IEEE Transactions on Robotics*, 30(2):310–324, 2014.

[8] Fatih Porikli and Alper Yilmaz. Object detection and tracking. *Video Analytics for Business Intelligence*, pages 3–41, 2012.

[9] Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 842–849. IEEE, 2003.

[10] Lukas Rummelhard, Anshul Paigwar, Amaury Nègre, and Christian Laugier. Ground estimation and point cloud segmentation using spatiotemporal conditional random field. In *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pages 1105–1110. IEEE, 2017.

[11] Charles Sutton, Andrew McCallum, et al. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373, 2012.

[12] Wei-Lwun Lu, Kevin P Murphy, James J Little, Alla Sheffer, and Hongbo Fu. A hybrid conditional random field for estimating the underlying ground surface from airborne lidar data. *IEEE Transactions on Geoscience and Remote Sensing*, 47(8):2913–2922, 2009.

[13] Yang Wang and Qiang Ji. A dynamic conditional random field model for object segmentation in image sequences. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 264–270. IEEE, 2005.

[14] Frank Moosmann, Oliver Pink, and Christoph Stiller. Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 215–220. IEEE, 2009.

[15] Mingfang Zhang, Daniel D Morris, and Rui Fu. Ground segmentation based on loopy belief propagation for sparse 3d point clouds. In *3D Vision (3DV), 2015 International Conference on*, pages 615–622. IEEE, 2015.

[16] Lei Tai, Shaohua Li, and Ming Liu. A deep-network solution towards model-less obstacle avoidance. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2759–2764, Oct 2016.

[17] Lei Tai, Giuseppe Paolo, and Ming Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017.

[18] Patrick Y Shinzato, Denis F Wolf, and Christoph Stiller. Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 687–692. IEEE, 2014.