Proceedings of The 2018 IEEE International
Conference on Real-time Computing and Robotics
August 1-5, 2018, Kandima, Maldives

Three-Dimensional Real-Time Object Perception Based on a 16-beam LiDAR for an Autonomous Driving Car

Jianyin Fan, Xiaorui Zhu*, and Hao Yang

***Abstract*— Object perception is essential for autonomous driving applications in urban environment. A 64-beam LiDAR is a widely-used solution in this field, but its high price has prevented it from broader applications of autonomous driving technology. An alternative solution is to adopt a 16-beam LiDAR or multiple 16-beam LiDARs. However, 16-beam LiDAR obtains relative sparse data that makes object perception more challenging. In this paper, a new perception method is proposed to tackle problems caused by sparse data obtained from a 16-beam LiDAR. First, a segmentation method is proposed based on 2D grid image where a free space constraint is employed to reduce unreasonable image dilation and some segments are merged based on prior knowledge. Then, selective features of bounding box are employed in association process for a more accurate result given the sparse data. The proposed method is evaluated on an autonomous driving car in real urban scenarios. The results show that segmentation error can be as low as 7.7% with the free space constraint and prior knowledge, and absolute tracking error and the overall classification accuracy are 0.44 m/s and 93.33% respectively.**

I. INTRODUCTION

Autonomous driving technologies have received more and more attention in the last decade. The success of the DARPA Urban Challenges [1] as well as Google's self-driving project evoke intensive research and development of autonomous cars operating in realistic environment. Object perception is one of the important technologies for an autonomous vehicle where categories, positions and velocities of other traffic participants are required, such as vehicles and pedestrians in urban environment.

A 64-beam LiDAR is widely used in autonomous driving research since it can provide more intensive 3D information of the environment with a longer range of detection than other sensors. However, its high price limits broader applications of autonomous driving technologies. In this paper, a new perception method is proposed based on an affordable 16-beam LiDAR for autonomous driving applications. It is observed that the data gathered from a 16-beam LiDAR, Fig. 1, are much sparser than those gathered from a 64-beam LiDAR. Hence, a four-stage perception pipeline is proposed consisting of ground estimation, segmentation, tracking and classification, in which some special treatments are taken to tackle problems caused by sparse point cloud data.

Since ground points account for a large portion of the data gathered from LiDAR, it is important that the ground points could be accurately identified from raw data stream to guarantee accomplishment of the subsequent segmentation tasks. Simple strategies, such as removing ground points according to height or distance difference between adjoining channels, would not work well due to much sparser point cloud data. Hence, a point-wise classifier is employed here based on local features of points to obtain acceptable results of ground point estimation.

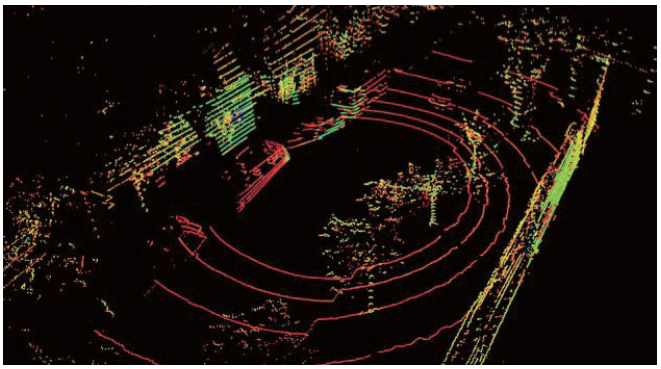


Fig. 1. Raw data gathered from 16-beam LIDAR in urban environment.

Segmentation is to group points belonging to the same object. A traditional strategy is based on proximity, however, it is difficult to find a suitable threshold especially among sparse point cloud data. If threshold is not proper, then one object would be divided into several parts, or multiple objects would be regarded as one. Therefore, a new segmentation method based on grid image is proposed in this paper. First, an image dilation is adopted with the free space constraints to get an initial segmentation result. Then, prior knowledge is employed to further reduce segmentation errors.

In terms of tracking, shape features are used to associate those measurements belonging to one object in different frames, a Kalman filter is adopted to estimate motion states for objects while reducing errors caused by shape change because of different observation position. In classification, some features that are widely used in classifying 3D point cloud measurements are selected to achieve good performance in our application. Then a support vector machine (SVM) classifier is trained to distinguish classes of objects.

Main contributions of this work include a new segmentation method for sparse data obtained from a 16-beam LiDAR and a new association method based on the shape feature of bounding box in order to increase successful rate of association for dynamic objects on road.

A survey of related works is presented in Section II. The system configuration is illustrated in Section III. Details of

The authors are with Harbin Institute of Technology (Shenzhen), Shenzhen 518055, Guangdong, China

*Xiaorui Zhu is the corresponding author, and her mail address is (xiaoruizhu@hit.edu.cn).

978-1-5386-6869-6/18/$31.00 © 2018 IEEE

the perception system are shown in Section IV. Field evaluation is presented and discussed in Section V. Conclusions are summarized in Section VI.

II. Related Work

Ground-points estimation from raw data stream plays a pivotal role in a LiDAR perception system. Some simple strategies based on geometric feature have been developed to identify ground points [2]. However, these geometric feature-based methods are not effective for very sparse data obtained by a 16-beam LiDAR because the distance between two ground points tends to be very large. Some machine learning approaches have been also introduced to handle complex scenarios [3].

Traditionally, 3D data are projected onto a plane for the segmentation task. For instance, planar grid cells could be clustered via the connected components algorithm [2], [4]. Similarly, an 8-connected components analysis could be used to process difference maps which have been generated based on the min map and the max map [5]. These segmentation methods based on connected components worked well for dense data but easily result in errors for sparse data. More recently, several segmentation methods have been implemented on 3D data directly. Moosmann et al. [6] constructed a neighborhood graph based on an ordered point cloud, and then used both locally convex and normal vectors to determine whether or not two surfaces belong to the same object. Shin et al. [7] proposed a method to cluster non-ground points based on Euclidean distance and to handle the over-segmentation problem with GP regression. However, segmentation with Euclidean distance on 3D data could easily fail for sparse data.

In terms of object tracking, Azim and Aycgard [8] adopted Global Nearest Neighbor (GNN) [9] for data association and Bayesian filters for state estimation. Choi et al. [10] introduced a geometric model to further reduce the measurement error Moosmann and Stiller [11] proposed to train a classifier in the tracking process to determine whether or not an object can be associated with an existing tracked one. As for object classification, Azim and Aycgard [8] adopted the size of bounding box to determine the category of an object. 3D point measurements were first converted into fixed-dimensional feature vector then classified via SVM classifiers in [2] and [12]. In addition, Teichman et al. [4] combined the outputs of a segment classifier and a holistic classifier into a final log odds estimator for classifying tracking sequences.

III. System Configuration

Our perception system is designed for tracking and classifying objects detected by a 16-beam LiDAR. The proposed pipeline of the system is shown in Fig. 2. Since the whole process is implemented in Cartesian coordinates, a transformation from a spherical coordinate to a Cartesian coordinate is first performed. The data obtained during a single rotation of the sensor are regarded as one frame and stored in an ordered point cloud database with 16 rows and 1800 columns according to the internal parameters of the sensor.

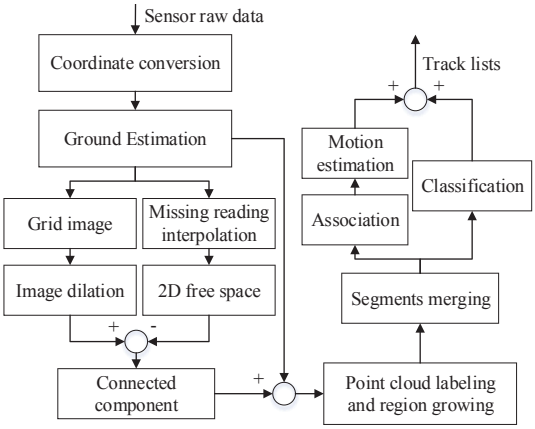


Fig. 2. The architecture of the object perception system.

Then a method similar to [3] is adopted to classify ground points. This method consists of an offline stage and an online stage. In the offline stage, point clouds are automatically labeled for training a classifier. In the online stage, ground points are classified based on their local features.

In the segmentation process, the half of the point cloud obtained from the bottom 8 laser channels are first separated to build a grid image. At the same time, the missing readings in this part of data are interpolated to obtain a reliable free space constraint. According to the constraint, those reasonless occupied pixels are corrected in image dilation process. Then, the grid image is labeled via a connected component algorithm. The labeled grid image is used as a lookup table to label the lower half of point cloud. Subsequently, a region growing algorithm is implemented to label the rest of point cloud on the basis of labeled lower half of point cloud. And some separated segments produced in the initial process are merged based on their shape information.

After the segmentation process, objects need to be tracked and classified respectively. Only the lower part of 3D point measurements of an object are used for tracking. The objects in the current frame are associated with the existing track sequences via an association matrix, and their motion states are estimated through a Kalman filter. In the classification process, the whole 3D point measurements of an object are first converted to a fix-dimensional feature vector and then classified with four one-versus-all SVM classifiers.

IV. Perception Algorithms

A. Segmentation

Objects of interest in each frame are required to be separated for subsequent tracking and classification tasks. In the proposed method, a 2D grid with a free space constraint is first adopted for an initial segmentation, and thereupon some segmentation errors are further eliminated with prior knowledge. The output of the proposed algorithm is a labeled point cloud where each label indicates a disjoint subset of points (segment).

In the beginning, only the data obtained from the bottom 8 laser channels are used to avoid interference of higher-

Fig. 3. White points are occupancy grid, grey points are free space and red points are the points corrected by free space.

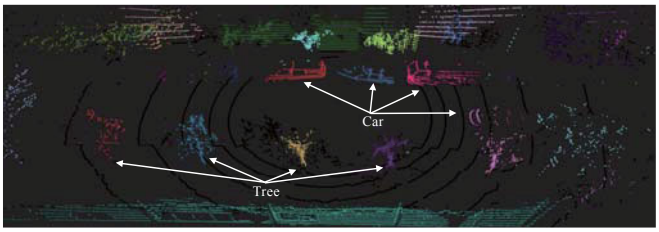


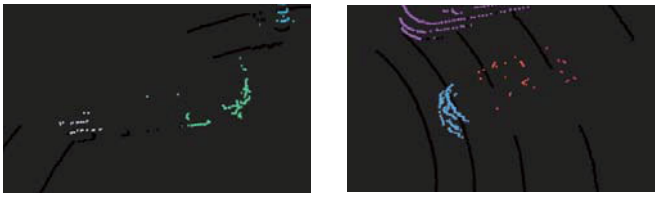
Fig. 4. A labeled point cloud after region growing. Each color indicates a segment.

elevation objects on the roads such as tree leaves. As ground points have been classified, a 2D LiDAR-centered occupancy image with the dimension $100m \times 100m$ is generated by projecting non-ground points onto the X-Y plane of the sensor coordinate system. Each pixel covers a small patch of $0.1m \times 0.1m$ and stores a binary information about whether or not this pixel is occupied by an obstacle. Then an image dilation is performed on the grid image since the data gathered from our sensor are too sparse. However, image dilation may also pose some mistakes, such as grouping two neighboring objects as one. Hence a free space constraint is adopted to tackle this problem in this paper.

For every point detected by the sensor, there is no obstacle lying between the sensor and the point. This feature can be used to obtain the free space constraint. However, in order to obtain a reliable free space, missing readings must be handled first since laser range finders are widely known to have difficulty detecting specular objects. As the point cloud is denser in horizontal direction, the missing points are assumed to be in the same line with their nearest valid points in the horizontal direction and interpolated. Then, the nearest non-ground points in all columns of the point cloud are acquired and converted to the image coordinate system, the area surrounded by these points is the 2D free space shown in Fig. 3.

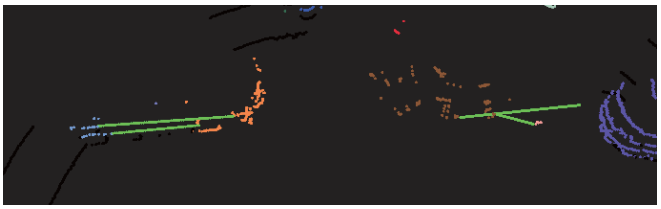
A standard connected component algorithms proposed are applied on the corrected grid image, which assigns each pixel the label of the connected component it belongs to. Then the 3D point cloud is traversed again and labeled using labeled image as a lookup table.

The foregoing process is performed on the data obtained from the bottom 8 laser channels which is sufficient for tracking task. Certainly, upper readings remain remarkably important to distinguish traffic participants and higher-elevation

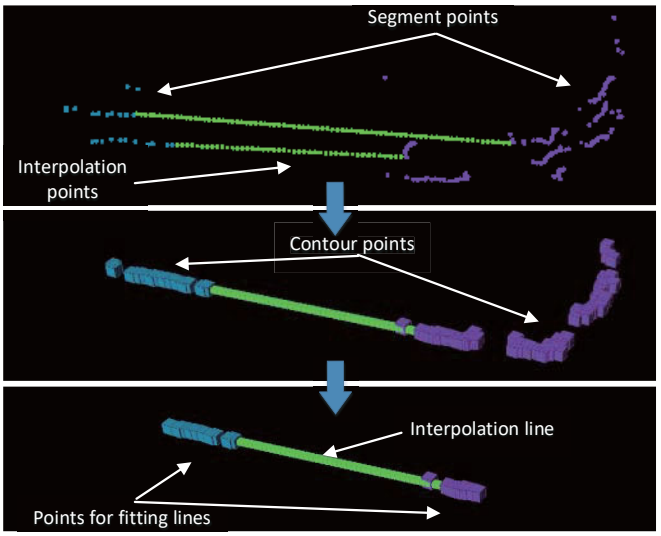


(a) (b)

Fig. 5. (a) Segmentation errors arising from missing reading. (b) Segmentation errors arising from large gap between readings.



(a)



(b)

Fig. 6. (a) The segments linked by green points may belong to the same objects. (b) The process of deciding whether the two segments belong to the same object.

objects like trees and buildings. A region growing algorithm is implemented to label the upper point cloud based on the labeled half point cloud. An example of the labeled point clouds is shown in Fig. 4.

It is obvious that incorrect segments are inevitable. These incorrect segments (Fig. 5) can be corrected according to additional shape information of objects.

The first type of segmentation error occurs when the laser beam hits the undetectable surface, such as windows on the side of vehicle. To locate this kind of error, two segments linked by the interpolation points which are on the boundary of the free space constraint are first found. When the distance between adjoining interpolation points is less than a certain threshold and two segments are close enough, these two segments may belong to one object. Then we employ contour points of these segments to determine

whether the two segments belong to the same object. After finding contour points, 10 contour points closing to another contour are found respectively to fit two lines via a Random Sample Consensus (RANSAC) algorithm. Two segments will be categorized into the same object if these two lines and the interpolation line are roughly parallel. Fig. 6 shows an example of how the segmentation errors can be corrected.

The second type of segmentation error occurs when the gap between readings from different laser channels is too wide. To tackle this problem, the segments floating above the ground plane and containing less number of points than a threshold are first selected. If there is another segment between the floating segment and the sensor, and the distance between these two segments is less than a threshold, then they can be merged to one segment.

B. Tracking

In this part, the association and filtering algorithms are illustrated. The segments pertaining to one object in different frames are referred to as 'track'. After ground estimation and object segmentation, a simple gating function is introduced to filter out invalid objects being too large or too small to be traffic participants. 3D bounding boxes are used to represent objects on account of their compact shape features.

Association is of great significance for tracking because assigning a wrong object to a track could directly result in apparent estimation error. To reduce association errors in complex environments, the GNN algorithm with extra features is adopted in this paper. An association matrix A is used to determine relations between objects in the current frame and the existing tracks. Elements of the matrix are acquired as follow:

$$a_{ij} = \begin{cases} \sum_{i=1}^{4} \gamma_i P_i & if\ 0.5 < P_i < 1 \\ 0 & otherwise \end{cases} \quad (1)$$

where P_i is

$$P_1 = 1 - \left| \frac{\tau - d}{d} \right| \quad (2)$$

$$P_2 = 1 - \left| \frac{S_t - S_{t-1}}{S_{t-1}} \right| \quad (3)$$

$$P_3 = 1 - \left| \frac{B_t}{B_{t-1}} \right| \quad (4)$$

$$P_4 = 1 - \left| \frac{N_t}{\max(N_t, N_{t-1})} \right| \quad (5)$$

The association matrix A is of size $n \times m$ where n refers the number of the existing tracks and m indicates the number of objects in the current frame. a_{ij} is the association index of the i-th existing track and the j-th object in the current frame. P_1, P_2, P_3 and P_4 refer to the difference between objects in the previous and current frames in terms of distance, shape and number of points, and γ_i are weighing factors of these features. τ is the distance threshold while d is the distance between two segments in different frames. S_t is the area of the bounding box, B_t is the aspect ratio and N counts as the number of 3D points.

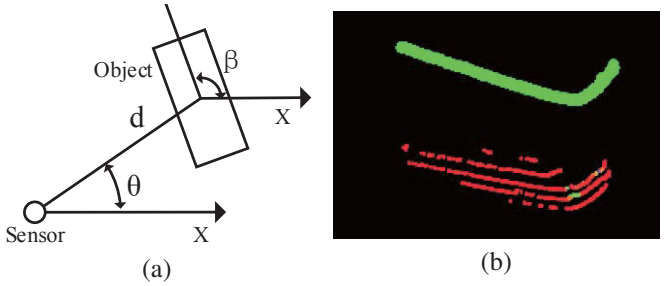


Fig. 7. (a) Size and position feature consists of the length, width and height of the bounding box, d, θ and β. (b) A RANSAC algorithm is adopted to fit the dominant line of contour points(green) and this line is used to fit a bounding box.

Fig. 8. Experiment environment.

The maximum value of each row and column are first found out to determine the tracks and objects that cannot be associated with. If the maximum value of the i-th row is less than a threshold and there is no associated object in the next 10 frames, the corresponding track is going to be removed from the tracking list. Similarly, if the maximum value of the j-th column is less than a threshold, the corresponding object is a new object and it initiates a new track sequence. For each object that can be associated, the maximum value of its corresponding column is used to determine its associated track. However, in some situations, this method is not effective if the object is moving at a large relative velocity or some parts of the object are undetectable. A second association is introduced for remaining objects and tracks in which only the distance feature is used with a distance threshold of 3.3m.

After association, a similar to [10] is adopted to estimate objects' motion. A constant velocity motion model is employed on account of the high frame rate (10Hz in our experiments). Then a Kalman filter is adopted to estimate the motion of objects. A corrected measurement of the reference point is used to update the velocity states.

C. Classification

The first step in the design of the object classifier is to extract meaningful features from 3D point measurements. For each target object, a 42-dimensional feature vector is established through converting 3D point measurement to five sets of invariant descriptors. The descriptors consist of 6-dimensional size and position feature illustrated in Fig. 7, 20-dimensional slice feature [13], 6-dimensional normalized

moment of inertia tensor [14], 4-dimensional intensity factor including maximum and minimum intensity, mean intensity and intensity variance, 6-dimensional covariance matrix [14].

For classifying objects, SVM classifiers with the non-linear Radial Basis Function kernel are adopted. Parameters are trained using cross-validation. Four individual one-versus-all classifiers are trained for cars, trunk, pedestrians and background objects, respectively. Then final decisions are made based on the maximal probability outputs from four classifiers.

V. Experimental Results

A. Experiment Procedures

The proposed algorithm was evaluated using data obtained in some real urban environments (Fig. 8). The sensor is mounted on the top of a vehicle which is 2.1m above the ground. A prototype of our system has been implemented on a computer equipped with a dual core Intel i5 processor (3.2Hz) and 8GB of RAM.

In order to evaluate the segmentation algorithm, 484 points clouds are randomly selected and manually labeled encompassing 2067 objects. The similar method in [15] is adopted to determine whether an object is correctly segmented. An over-segmentation error occurs when a segment is comprised by more than one object, and an over-segmentation error occurs when a single object is divided into multiple segments. The fractions of under-segmentation error and over-segmentation error are acquired as follows:

$$U = \frac{1}{N} \sum \mathbb{I} \left(\frac{|C_s \cap C_{gt}|}{|C_s|} < \tau_u \right) \tag{6}$$

$$O = \frac{1}{N} \sum \mathbb{I} \left(\frac{|C_s \cap C_{gt}|}{|C_{gt}|} < \tau_o \right) \tag{7}$$

where $\mathbb{I}$ is an indicator function that is equal to 1 if the input is true and 0 otherwise. C_{gt} represent the set of ground-truth points of a segment, C_s is the set of points pertaining to the segment from our segmentation algorithm. In our application, $\tau_u = 0.8$ and $\tau_o = 0.8$ are chosen. An overall error rate based on the fractions of over-segmentation error and under-segmentation error is computed as follows:

$$E = U + O \tag{8}$$

To evaluate the association algorithm, the tracks comprised by more than 10 segments are selected among the dataset describing a real urban environment. A track is considered to be correct only when all the segments in this track pertain to the same object. Then the association accuracy rates are manually counted and calculated.

The absolute velocities of stationary objects acquired based on their relative velocities and the vehicle velocity are used to evaluate the tracking algorithm due to unavailability of velocities of most dynamic objects. We obtain the mean absolute velocity and Root Mean Square (RMS) error of our tracking algorithm from 300 stationary objects in real environment.

TABLE I
Details of Classification Data Sets

Class	Car	Truck	Pedestrian	Background
Training set	18804	5062	10471	17803
Test set	4440	1013	2133	8005
Total	23244	6075	12604	25808

TABLE II
Details of Segmentation results

Method	Oversegmentation error (%)	Undersegmentation error (%)	Overall error (%)
Grid only	7.30527	2.32221	9.62748
Adding free space	7.30527	1.69328	8.99855
Adding first correction	6.48283	1.88679	8.36962
Final result	4.88631	2.90276	7.78907

Since most existing online datasets were obtained from 64-beam LiDARs, we need to attain enough datasets from 16-beam LiDARs for training classifiers in which track sequences other than segments are labeled manually [4]. However, only instances of trackable objects comprise these datasets. Hence some non-traceable background segments are augmented into these datasets. Training set and test set are gathered from different roads in different time. Details of datasets are shown in TABLE I.

B. Segmentation

Since the proposed segmentation algorithm encompasses four steps, we compare segmentation fractions in each step and show how these methods improve the segmentation performance. The results are illustrated in TABLE II.

It is evident that adding the free space constraint results in improvements in under-segmentation performance while over-segmentation performance does not change. In fact, the free space constraint has reduced segmentation errors. As the corrections are introduced, the over-segmentation fraction and the overall fraction indicate a decline while the under-segmentation fraction increases slightly. These correction methods can reduce segmentation errors while some under-segmentation errors are introduced.

C. Tracking

The association results are shown in TABLE III. The association of cars has the best accuracy among three categories since the gap between two vehicles is generally wide and the shape feature works well when the bounding box

TABLE III
Details of Association Results

Class	Correct track	Total track	Accuracy(%)
Car	338	373	90.6
Pedestrian	156	187	83.4
Background	490	613	79.9
Total	984	1173	83.9

TABLE IV
TRACKING ACCURACY

Tracking Method	Mean absolute error (m/s)	RMS error (m/s)
Our method	0.44	0.80
Color-augmented grid search with interpolation	0.40	0.86
Color-augmented ICP with interpolation	0.41	1.00
Centroid difference	0.77	1.27

is comparatively large. For the association of pedestrians, invalidity of the shape feature and the small distance among pedestrians result in a comparatively low accuracy. Since background objects are disordered and often obscured by other objects, the unreliable segmentation of background objects results in the lowest accuracy.

The final results of our tracking algorithm are 0.44 m/s for mean absolute velocity and 0.80 ms/s for RMS error. The results of our tracking algorithm are acceptable in contrast with the results in [16] shown in TABLE IV considering only the LiDAR is used here.

D. Classification

Classification accuracies are 94.75% for car, 92.20% for truck, 97.14% for pedestrian and 91.67% for background, the overall classification accuracy is 93.33%. The confusion matrix is shown in Fig. 9. The overall classification accuracy is acceptable compared with 93.1% in [4] and 93.9% in [2]. Since there are a relative small number of truck samples in our datasets, the classification accuracy of truck is the lowest among four categories. Moreover, the most of classification errors occur when background objects are distinguished from other objects because of the variety of background objects. The classification accuracy will significantly increase if the classification process is only implemented on road area.

VI. CONCLUSIONS

A perception system based on a 16-beam LiDAR is proposed and evaluated in this paper. Initially, a segmentation method based on 2D grid image is proposed where a free space constraint and prior knowledge are employed. The segmentation results are extended to the entire point cloud, which is of great importance in objects classification. And the segmentation experiments show that free space constraint and prior knowledge indeed reduce the number of incorrect segments. In addition, the objects in different frames are associated using a GNN method with extra features for a better performance in complex environment, and then the motion states are estimated via a Kalman filter. At the same time, appropriate features are selected for training classifiers to discriminate classes of objects. Experiments show that the acceptable classification accuracy can be achieved.

ACKNOWLEDGMENT

This research was supported by National Natural Science Foundation of China under Grant No. 91648102. Experimental platforms were provided by the RoboSense Inc.

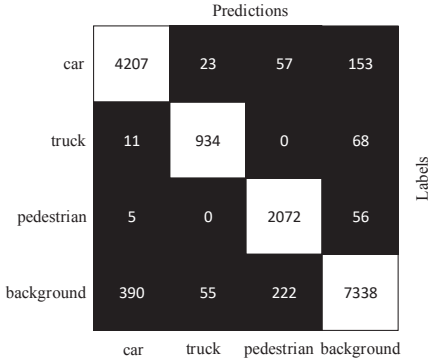


Fig. 9. Confusion matrix for classification.

REFERENCES

[1] C. Urmson et al., "Autonomous driving in urban environments: Boss and the urban challenge," Journal of Field Robotics, vol. 25, no. 8, pp. 425-466, 2008.

[2] M. Himmelsbach, A. Mueller, T. Lttel, and H.-J. Wnsche, "LIDAR-based 3D object perception," in Proceedings of 1st International Workshop on Cognition for Technical Systems, Mnchen, Oct. 2008, vol. 1.

[3] M. Samples and M. R. James, "Learning a real-time 3D point cloud obstacle discriminator via bootstrapping," in Workshop on Robotics and Intelligent Transportation System, Anchorage, Alaska, 2010.

[4] A. Teichman, J. Levinson, and S. Thrun, "Towards 3D object recognition via classification of arbitrary object tracks," in IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 4034-4041.

[5] D. Korchev, S. Cheng, and Y. Owechko, "On real-time lidar data segmentation and classification," in Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), 2013.

[6] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion," in IEEE Intelligent Vehicles Symposium, 2009, pp. 215-220.

[7] M.-O. Shin, G.-M. Oh, S.-W. Kim, and S.-W. Seo, "Real-Time and Accurate Segmentation of 3-D Point Clouds Based on Gaussian Process Regression," IEEE Transactions on Intelligent Transportation Systems, 2017.

[8] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment," in Intelligent Vehicles Symposium (IV), IEEE, 2012, pp. 802-807.

[9] S. S. Blackman, "Multiple-target tracking with radar applications," Norwood, MA, USA: Artech House, 1986.

[10] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer, "Multi-target tracking using a 3d-lidar sensor for autonomous vehicles," in 16th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, 2013, pp. 881-886.

[11] F. Moosmann and C. Stiller, "Joint self-localization and tracking of generic objects in 3D range data," in International Conference on Robotics and Automation (ICRA), IEEE, 2013, pp. 1146-1152.

[12] D. Z. Wang, I. Posner, and P. Newman, "What could move? finding cars, pedestrians and bicyclists in 3d laser data," in International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 4038-4044.

[13] K. Kidono, T. Miyasaka, A. Watanabe, T. Naito, and J. Miura, "Pedestrian recognition using high-definition LIDAR," in Intelligent Vehicles Symposium (IV), IEEE, 2011, pp. 405-410.

[14] L. E. Navarro-Serment, C. Mertz, and M. Hebert, "Pedestrian detection and tracking using three-dimensional ladar data," The International Journal of Robotics Research, vol. 29, no. 12, pp. 1516-1528, 2010.

[15] D. Held, D. Guillory, B. Rebsamen, S. Thrun, and S. Savarese, "A Probabilistic Framework for Real-time 3D Segmentation using Spatial, Temporal, and Semantic Cues," in Robotics: Science and Systems, 2016.

[16] D. Held, J. Levinson, and S. Thrun, "Precision tracking with sparse 3d and dense color 2d data," in IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 1138-1145.