Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 1583129, 12 pages
https://doi.org/10.1155/2020/1583129

Research Article

Moving Object Classification Using 3D Point Cloud in Urban Traffic Environment

MingFang Zhang,[1] **Rui Fu**,[2] **YingShi Guo**,[2] **and Li Wang**[1]

[1]*Beijing Key Lab of Urban Intelligent Traffic Control Technology, North China University of Technology, Beijing 100144, China*
[2]*Key Lab of Vehicle Transportation Safety Technology, Ministry of Transport, Chang'an University, Xi'an 710064, China*

Correspondence should be addressed to Rui Fu; furui@chd.edu.cn

Received 3 August 2019; Revised 2 February 2020; Accepted 11 February 2020; Published 17 March 2020

Academic Editor: Jaeyoung Lee

Copyright © 2020 MingFang Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Moving object classification is essential for autonomous vehicle to complete high-level tasks like scene understanding and motion planning. In this paper, we propose a novel approach for classifying moving objects into four classes of interest using 3D point cloud in urban traffic environment. Unlike most existing work on object recognition which involves dense point cloud, our approach combines extensive feature extraction with the multiframe classification optimization to solve the classification task when partial occlusion occurs. First, the point cloud of moving object is segmented by a data preprocessing procedure. Then, the efficient features are selected via Gini index criterion applied to the extended feature set. Next, Bayes Decision Theory (BDT) is employed to incorporate the preliminary results from posterior probability Support Vector Machine (SVM) classifier at consecutive frames. The point cloud data acquired from our own LIDAR as well as public KITTI dataset is used to validate the proposed moving object classification method in the experiments. The results show that the proposed SVM-BDT classifier based on 18 selected features can effectively recognize the moving objects.

1. Introduction

Autonomous driving has become an increasingly popular domain for intelligent transportation system [1, 2]. Moving object classification is a critical step to achieve reliable planning of driving trajectories for autonomous vehicles in dynamic environment, and the prior knowledge of the category attribute helps to build an appropriate dynamic model for moving objects [3–5]. The most commonly used sensors for object recognition are camera and LIDAR. Compared with cameras, LIDAR can obtain the accurate 3D measurements and it is invulnerable to weather and illumination. Extensive research bends the efforts to object recognition using LIDAR. Conventional techniques can be coarsely divided into two categories.

The first category of the methods determines the object semantics through calculating the similarity between the scanned object and the predefined template. Simple geometric or motion model is constructed to classify rigid objects, while it is difficult to recognize pedestrians. Fang and Duan [6] employ iterative endpoint fitting algorithm to fit the segmented point cloud and calculate the number and size of line segments to determine whether the object is a vehicle or not. Petrovskaya and Thrun [7] combine the rectangular model of point cloud in 2D occupancy grid map with the motion model established by Rao-Blackwellized particle filter to improve the vehicle classification accuracy when partial occlusion temporarily occurs.

The second class of the methods mainly focuses on the effective feature descriptors of the object of interest as well as training specific classifiers [8, 9]. For vehicle recognition, Yang and Dong [10] calculated the geometric features based on the optimal neighborhood size of each point, and classified the segments using SVMs. Lee and Coifman [11] checked the shape feature of each point cloud cluster and classified the vehicle into six classes of vehicles. For pedestrian recognition, Kim et al. [12] used SVM classifier with 31 layer-based features for pedestrian recognition. Arras et al. [13] defined 14 static features including roundness and compactness to train the pedestrian classifier based on the point cloud of the legs. Although this method can generate good recognition result at indoor environment, it is not suitable for outdoor. For multiclass object recognition, Azim and Aycard [14] used the simple ratio characteristics of 3D bounding box based on point cloud to recognize the vehicles and pedestrians, such as width-height ratio and length-height ratio. However, frequent occlusion in real traffic environment leads to false size ratio so that the recognition performance is poor. Wang et al. [15] proposed 120-dimensional feature set including rotating images, shape factors, point normal vectors, and Euclidean distances and employed SVM classifier to recognize multiclass moving objects. Teichman et al. [16] constructed boosting classifier to recognize moving vehicles, pedestrians, and bicycles by integrating geometric features and motion features. Moreover, occupancy-grid-based methods [17, 18] are presented to detect moving objects efficiently, but they only estimate the kinematic state of the object without the need to classify the category.

Conclusion drawn from the abovementioned related literatures can be summarized as follows. First, the existing moving object classification methods based on LIDAR are designed for the relatively dense point cloud returned from the scanned object. Second, temporary or partial occlusion at consecutive frames is seldom considered in the majority of object classification schemes, and the effectiveness of the extracted features has not been analyzed from the perspective of object category. Furthermore, most of the aforementioned classification methods are proposed to recognize common moving objects including vehicles, pedestrians, and bicycles. In real traffic scenarios, the pedestrians often appear as independent individuals or a small crowd. When two pedestrians are too close, it is so hard to segment the returned point cloud clearly that it is difficult to identify individual pedestrian. As shown in Figure 1, the pedestrians marked as A and B are segmented as a whole wrongly. It is very common that two or three human get together, and the crowd composed of two or three pedestrians may be regarded as other class of moving object due to point cloud under-segmentation. Motivated by the abovementioned analysis, we proposed a LIDAR-based classification method in this paper for four categories of moving objects, namely, vehicle, pedestrian, bicycle, and crowd. Velodyne HDL-64E LIDAR is adopted to collect 3D point cloud of the surrounding environment. Our method for moving object classification uses raw point cloud as follows. First, the points measured on moving objects are segmented from the rest of 3D point cloud. This process consists of ground segmentation, the clustering of nonground points, and moving object detection. Second, both global- and layer-based features are extracted to describe the geometric characteristics, and Gini index criterion is utilized to select the effective features based on the category attributes of training samples. Next, posteriori probability SVM classifier is employed to obtain the classification result at each frame, and BDT algorithm is further used to optimize the classification result of the tracked object at consecutive frames. Finally, the proposed SVM-BDT-based classification method is validated using the point cloud dataset collected by our own LIDAR as well as public KITTI dataset.

The contributions of this paper are two-fold. First, we describe a novel approach for classifying moving objects into vehicle, pedestrian, bicycle, and crowd, since the point cloud segment of the crowd may be confused with other types of objects and even reduces the accuracy of object recognition. This approach makes progress towards the application goal of moving object classification in real traffic environment for autonomous vehicle. Second, we adapt the idea of SVM-BDT classifier to incorporate multiframe classification results based on the effective features, and moving object classification is transformed into maximum posteriori probability solution problem.

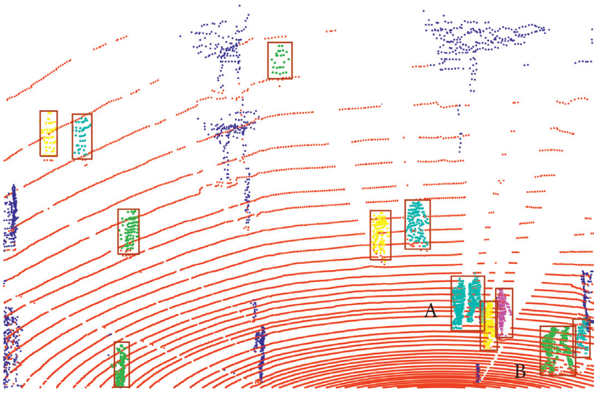


FIGURE 1: Segmentation of moving object point cloud at one frame.

The remainder of this paper is organized as follows. Section 2 introduces the point cloud preprocessing. Section 3 presents feature extraction. Section 4 describes the classification method. Section 5 demonstrates experimental results. Finally, Section 6 offers conclusions and future works.

2. Point Cloud Preprocessing

The point cloud is characterized by the coordinates in the world coordinate system, and LIDAR position is marked as the origin of the world coordinate system. In this section, ground points are removed from 3D raw point cloud using the ground segmentation method in [19] which combines Markov random field models with loopy belief propagation algorithm. Then, nonground points should be divided into independent clusters. Since the number of point cloud clusters of moving objects in surrounding environment is unknown and the density of point cloud varies with the range, mean shift clustering algorithm in [20] is selected. In order to reduce the influence of fixed bandwidth on the stability of clustering results, an improved mean shift clustering algorithm based on adaptive bandwidth is proposed as follows:

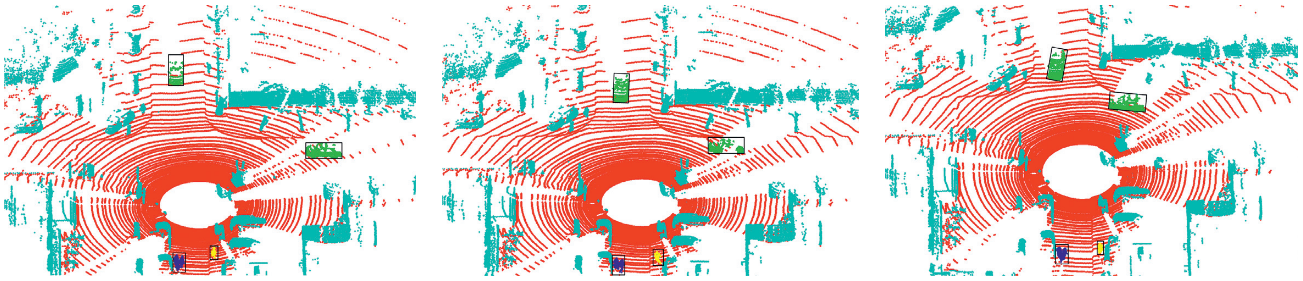
(1) The nonground points are represented by

$$\begin{aligned} P &= (p_1, \ldots, p_n), \\ p_i &= (x_i, y_i), \quad \text{for } i = 1, \ldots, n. \end{aligned} \tag{1}$$

Given initial kernel radius bandwidth h_0, Gauss kernel function $G(p)$ and the tolerance ε, the initial kernel density estimation is calculated by

$$\tilde{f}(p) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h^d} k\left(\left\| \frac{p - p_i}{h} \right\|^2 \right), \tag{2}$$

where d is the dimension of the data space.



(a)

(b)

(c)

Figure 2: Top view of the scene with moving clusters. (a) Frame 149, (b) Frame 157, and (c) Frame 165.

(2) The adaptive bandwidth is calculated for each point by $h(p_i) = h_0\sqrt{\lambda/\tilde{f}(p_i)}$, where λ is a proportional constant value and it can be calculated by $\log\lambda = n^{-1}\sum_{i=1}^{n}\log\tilde{f}(p_i)$.

(3) The initial kernel centroid is marked as q_1, and the weighted mean value at q_j is computed using kernel functions G and the weight $\{h^{d+2}(p_i)\}^{-1}$:

$$q_{j+1} = m_h(q_j) = \frac{\sum_{i=1}^{n} p_i/h^{d+2}(p_i)g\left(\left\|(q_j - p_i)/(h(p_i))\right\|^2\right)}{\sum_{i=1}^{n} 1/h^{d+2}(p_i)g\left(\left\|(q_j - p_i)/(h(p_i))\right\|^2\right)}, \quad j = 1, 2, \ldots, n-1, \tag{3}$$

where $m_h(q_j)$ is the mean shift vector. Note that q_j is iteratively calculated until the gradient of the convergence is zero, i.e., $\|m_h(q_j) - q_j\| < \varepsilon$.

Next, in order to detect moving objects, a local grid map is constructed using 3D occupancy grid algorithm in [6] to divide the surrounding environment into occupied, free, and unknown voxels. When new measurements return, the dynamic voxels are detected in the consistent grid map based on the inconsistencies between occupied space and free space. Then, all the moving clusters in the dynamic voxels are extracted, as shown in Figure 2.

3. Feature Extraction

In general, the height of the point cloud clusters of moving objects including vehicle, pedestrian, bicycle, and crowd ranges from 1 m and 4 m. When partial occlusion occurs, the height of the clusters can be less than 1 m. Since the layer-based features describe a more detailed level of local shape characteristics than the global features, we divide the point cloud cluster into eight layers along the vertical direction of the horizontal plane. 2D features at each layer are employed to supplement the description of 3D geometric features and reduce the disturbance of partial occlusion. In addition to collecting the existing feature descriptors in the literatures, the differences of point cloud characteristics among four classes of moving objects are analyzed, and number-of-point-based features, shape features, and statistical features are selected, as shown in Tables 1–3.

In order to remove the features that make no significant difference on the object classification results, Gini index criterion of CART decision tree algorithm [21] is used for feature selection. The forward search mechanism of the feature subset is combined with the subset evaluation mechanism to select the efficient features in order of priority, so that all samples falling at the subnodes belong to the same category, namely the highest purity is achieved at each subnode. Define that the proportion of the k-class sample in the training sample set U is u_k ($k = 1, 2, 3, 4$), and Gini index is used to represent the purity probability distribution of the sample set U:

$$\mathrm{Gini}(U) = \sum_{k=1}^{|y|}\sum_{k'\neq k} u_k u_{k'} = 1 - \sum_{k=1}^{|y|} u_k^2. \tag{4}$$

The attribute $w = \{w_1, w_2, \ldots, w_V\}$ is used to divide the sample set U; thus, V branch nodes are obtained. The samples with the attribute w_V at the vth branch node are denoted as U_V, and the weight of the branch node is set as $|UV|/|U|$. Given the attribute w, the Gini index of the sample set U is defined by

$$\mathrm{Gini\_index}(U, w) = \sum_{v=1}^{V}\frac{U^v}{U}\mathrm{Gini}(U^v). \tag{5}$$

The attribute $w_* = \arg\max \mathrm{Gini\_index}(U, w)$ with the minimum Gini index is selected as the optimal boundary of the features. Based on the optimal attribute, the features are allocated for two subnodes generated from the current node. The abovementioned calculation is carried out recursively until Gini index is less than the preset the threshold. The category attributes of training sample sets are divided, respectively, for vehicle, pedestrian, bicycle, and crowd. Four decision trees of the hierarchical features for four categories of moving objects are obtained, as shown in Figure 3. In this figure, the solid line denotes yes and the dotted line indicates

Table 1: Number-of-point-based features.

Index	Description	Formula
f_1	Number of points of the cluster	n
f_2	The slope of the line fitted by the number of points at each layer [13]	$y = kx + c_1$
f_3-f_4	The first order and second order coefficient of the quadratic curve are fitted by the number of points at each layer [13]	$y = bx^2 + ax + c_2$
f_5	The product of the number of points and the minimum distance between the scan point and the origin point	$n^* d_{\min}$

Table 2: Shape-based features.

Index	Description	Formula
f_6	The sum of the area of the fitting rectangles of the horizontal projection points at each layer	$A_1 + A_2 + \cdots + A_8$
f_7	The average of the area of the fitting rectangles of the horizontal projection points at each layer	$s_A/8$
f_8–f_{15}	The density of the point cloud at each layer	n_{L_i}/A_i
f_{16}	The area of the fitting rectangle of the horizontal projection points	$L^*\ W$
f_{17}-f_{18}	The length and width of the fitting rectangle of the horizontal projection points	—
f_{19}	The standard Cartesian dimension of the horizontal projection points	$\sqrt{\Delta x^2 + \Delta y^2}$
f_{20}–f_{27}	The mean of inscribed angles at each layer (the proximity to a circle)	$\overline{\alpha}$
f_{28}–f_{35}	The standard variance of inscribed angles at each layer	$\sqrt{(1/np)\sum_{i=1}^{np}(\alpha - \overline{\alpha})^2}$
f_{36}–f_{43}	The circle fitting level at each layer (sum of squared residuals of the vertical distances between the points and the least square fitting circle)	$(1/np)\sum_{i=1}^{np}(p_{xy,i} - \widehat{p}_{c,i})^2$
f_{44}–f_{51}	The radius of the circle fitted by the horizontal projection of the points at each layer	Radius $r = \sqrt{D^2 + E^2 - 4F}/2$ (circle $x^2 + y^2 + Dx + Ey + F = 0$)
f_{52}–f_{59}	The line fitting level at each layer (sum of squared residuals of the vertical distances between the points and the least square fitting line) [12]	$1/np\sum_{i=1}^{np}(p_{xy,i} - \widehat{p}_{l,i})^2$

Table 3: Statistical features.

Index	Description	Formula
f_{60}	Standard deviation between 3D points and mean center μ_{3D} [13]	$\sqrt{(1/np)\sum_{i=1}^{np}\lVert p_{3\,D,i} - \mu_{3\,D}\rVert}$
f_{61}	Standard deviation between the horizontal projection points and mean center $\overline{p}$	$\sqrt{(1/np)\sum_{i=1}^{np}\lVert p_{xy,i} - \overline{p}\rVert}$
f_{62}	Mean deviation between the horizontal projection points and median point $\widetilde{p}$	$(1/np)\sum_{i=1}^{np}\lVert p_{xy,i} - \widetilde{p}\rVert$
f_{63}	Second central moment at horizontal plane (with the mean center μ_p)	$(1/np)\sum_{i=1}^{np}(p_{xy,i} - \mu_p)^2$
f_{64}	Third central moment at horizontal plane (with the mean center μ_p)	$(1/np)\sum_{i=1}^{np}(p_{xy,i} - \mu_p)^3$
f_{65}	Fourth central moment at horizontal plane (with the mean center μ_p)	$(1/np)\sum_{i=1}^{np}(p_{xy,i} - \mu_p)^4$
f_{66}	Mean reflection intensity	$M_{\text{intensity}}$
f_{67}	Standard deviation of reflection intensity	$S_{\text{intensity}}$
f_{68}	The residual ratio of mean distances between the points at each block area and the long axis of the fitting rectangle	$\overline{d}_k/\overline{D}_k$

no. Based on the comprehensive analysis of four hierarchical feature decision trees, 18 effective features are selected from the initial 68 features, namely f_1, f_2, f_4, f_6, f_{11}, f_{16}, f_{19}, f_{24}, f_{40}, f_{43}, f_{47}, f_{55}, f_{59}, f_{62}, f_{64}, f_{65}, f_{66}, and f_{68}.

4. Classification

4.1. Training SVM Classifier. SVM classifier based on posterior probability is employed to calculate the probability of the point cloud cluster belonging to each categories of moving objects. First, the standard nonlinear SVM classifier is used as the basic classification function:

$$y = \operatorname{sign}(f(x)) = \operatorname{sign}\left(w^T x + b\right), \\ = \operatorname{sign}\left(\sum_{i=1}^{l} y_i \alpha_i (K(x_i, x) + b)\right). \quad (6)$$

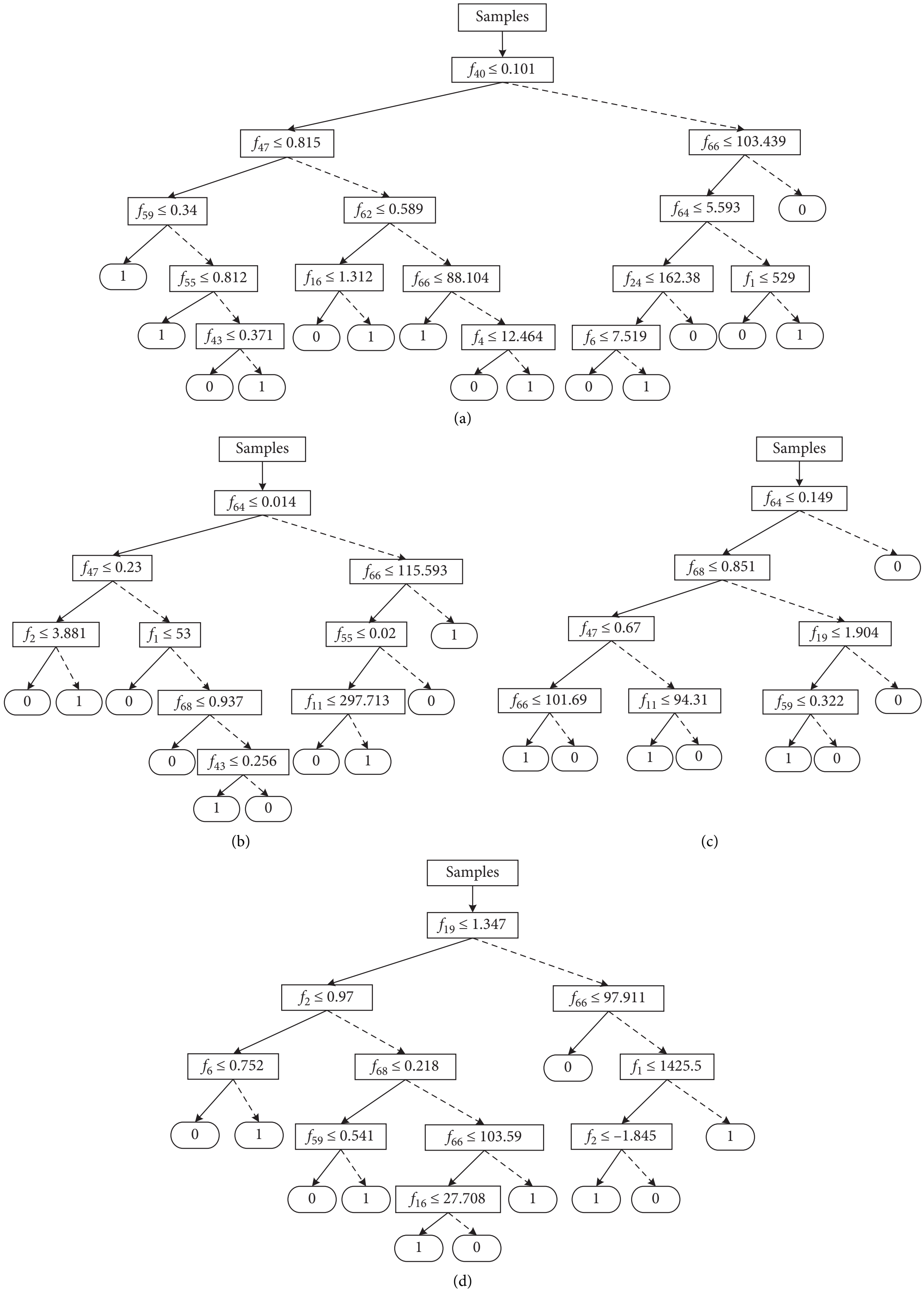


Figure 3: Decision tree at feature level. (a) Vehicle, (b) pedestrian, (c) bicycle, (d) crowd.

The standard SVM classifier only determines the probability value as 1 or 0. In order to ensure the sparsity in support vectors of SVM classifier and the accuracy of classification results, Sigmoid function is used to convert the output of standard SVM into a posterior probability [22]:

$$P(y=1|f)=\frac{1}{1+\exp(Af+B)}, \tag{7}$$

where $P(y=1\mid f)$ denotes the probability value of correct classification when the output is f and A and B are the parameters to be fitted. Define the training set as (f_i, t_i), and the output of the probability value is set as $t_i=(y_i+1)/2$, where y_i is the sample category, $y_i=\{-1,+1\}$. The optimization strategy of the parameters A and B can be solved by minimizing the negative log likelihood function on the training set:

$$\min_{z=(A,B)} F(z)=-\sum_{i=1}^{l}\left(t_i\log(p_i)+(1-t_i)\log(1-p_i)\right), \tag{8}$$

where $p_i=[1+\exp(Af_i+B)]^{-1}$, $f_i=f(x_i)$, $t_i=\{0,+1\}$, $i=1,2,\ldots,l$.

4.2. BDT-Based Classification Optimization. The point cloud of moving object is associated with the tracker at the consecutive frames and the tracker is updated based on the association result. The location information of each moving object at the next frame is predicted using linear Kalman filter. The moving object model is denoted as $\{I, L, W, x, y, if, t_I, G_m\}$, where I denotes the object index; L and W denote the fitting rectangular size of the point cloud cluster; x and y denote the center location of the point cloud cluster; if denotes whether the object has an associated tracker, the initial value is set as 0 and indicates no tracker; t_I denotes the associated tracker index; and G_m denotes the minimum value of the cost function between the object and the associated tracker. The tracker model is denoted as $\{I, L, W, x, y, v_x, v_y, \text{ifLost}, o_I\}$, where I denotes the tracker index; L and W denote the fitting rectangular size of moving object matched with the tracker at the last frame; x, y, v_x, and v_y denote the position and speed prediction of the filtered tracker at the current frame; ifLost denotes whether the tracked object at the current frame is lost or not, the initial value is set as 1 and indicates that the tracked object is lost; and o_I indicates the serial number of moving object corresponding to the tracker at the current frame when the tracked object has not been lost.

The deterministic data association algorithm based on the fusion of multiple features is used to associate the moving object with the tracker. The location and geometric features of moving object are utilized as the primary and secondary constraints respectively. The objects are associated with the trackers by minimizing the cost function. Assuming that m moving objects are generated from the point cloud pre-processing procedure at $t+1$th frame and n trackers exist, the cost equation based on the association between the ith moving object and the jth tracker at $t+1$th frame is established:

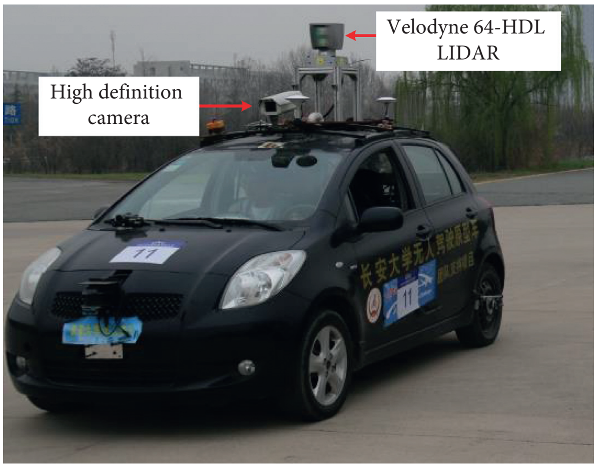


Figure 4: Autonomous vehicle.

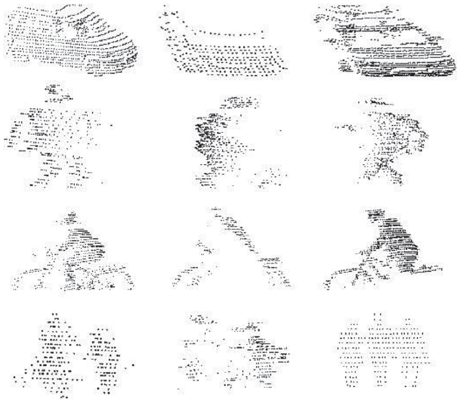


Figure 5: Moving object samples.

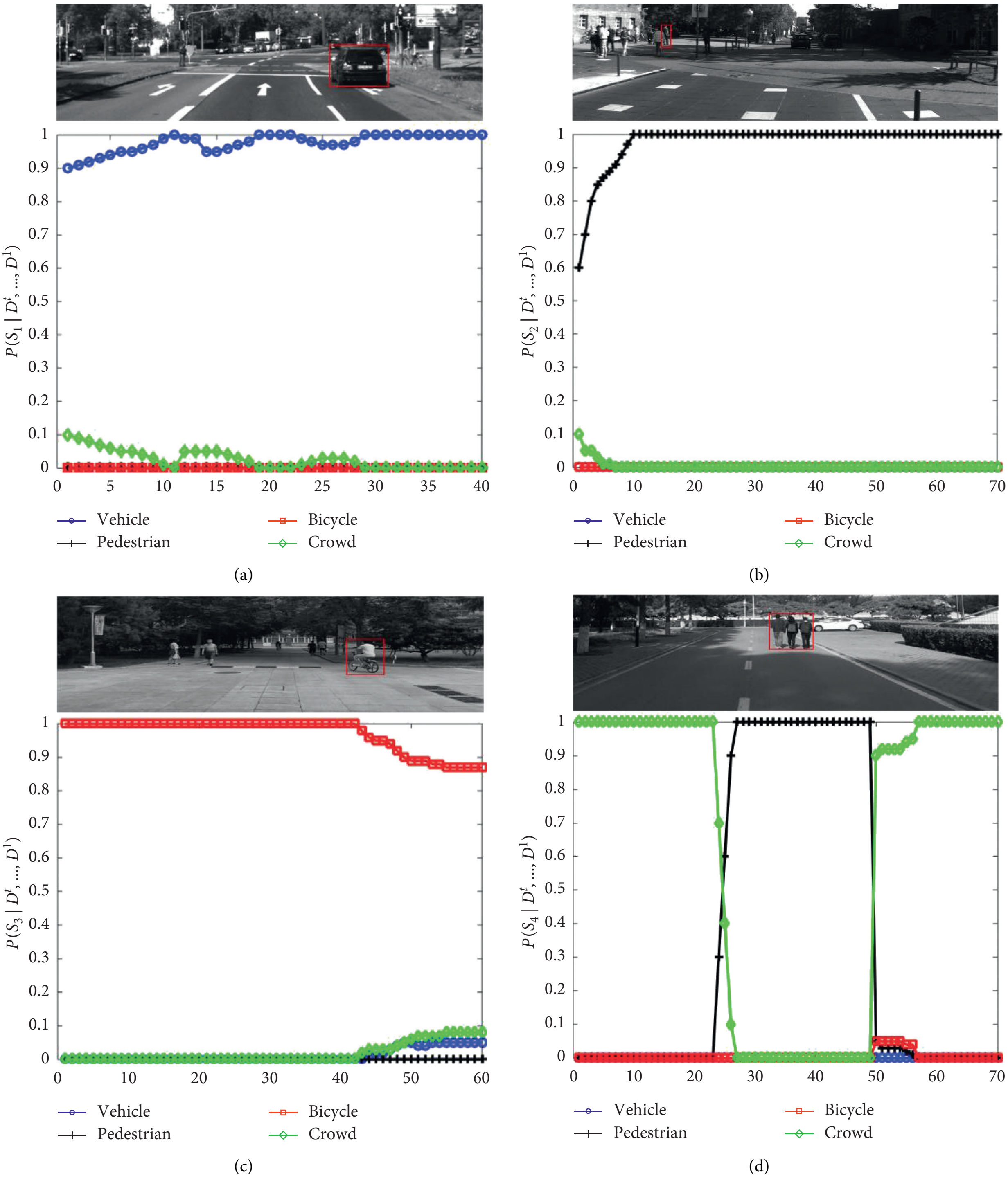
$$G(i,j)=g_1\cdot\text{pos}(i,j)+g_2\cdot\text{box}(i,j),$$

$$\text{pos}(i,j)=\frac{\sqrt{\left(x_{t+1}^{i}-x_t^{j}\right)^2+\left(y_{t+1}^{i}-y_t^{j}\right)^2}}{\max_n\sqrt{\left(x_{t+1}^{i}-x_t^{n}\right)^2+\left(y_{t+1}^{i}-y_t^{n}\right)^2}},$$

$$\text{box}(i,j)=\frac{\left|\sqrt{\left(L_{t+1}^{i}\right)^2+\left(W_{t+1}^{i}\right)^2}-\sqrt{\left(L_t^{j}\right)^2+\left(W_t^{j}\right)^2}\right|}{\max_n\left|\sqrt{\left(L_{t+1}^{i}\right)^2+\left(W_{t+1}^{i}\right)^2}-\sqrt{\left(L_t^{n}\right)^2+\left(W_t^{n}\right)^2}\right|}, \tag{9}$$

where pos (i,j) denotes the cost component between the position of moving object and the position of the tracker; box (i,j) denotes the cost component between the size of the fitting rectangle of moving object and the size of the tracker; g_1 and g_2 denote the weight of the position and the weight of the size, respectively, (define that $g_1+g_2=1$, the weight of the position is higher than that of the position; thus, $g_1=0.7$); $\max_n|\bullet|$ represents the maximum of the association values between the ith moving object and n trackers at the $t+1$th frame.

When both the number of points and the shape vary constantly for the point cloud of the same moving object at



(a)

(b)

(c)

(d)

Figure 6: The variation of posterior probability that point cloud of moving object belongs to each category S_i in tracking process. (a) Vehicle, (b) pedestrian, (c) bicycle, and (d) crowd.

continuous frames and partial occlusion may exists in a few frames, the posterior probability of the category estimation will change frame by frame. Based on the prior probability distribution of the object recognition result at each individual frame, the posterior distribution model of the multiframe classification is established, and the maximum output of posterior distribution is recursively solved by the maximum likelihood method to generate the optimal classification result at the current frame. The classification results at 10 previous consecutive frames are selected to estimate the optimal category at the current frame. Define the object category as $S_i \in S = \{\text{vehicle, pedestrian, bicycle, crowd}\}$, $i = 1, 2, \ldots, 4$, and there are J moving clusters $\{C_1, \ldots, C_J\}$ at the tth frame. According to equation (6), the result of the category decision for the kth cluster C^k $(1 < k < J)$ is generated as $\{d_i^k\}$, $i = 1, 2, \ldots, 4$. The category decision vector for the kth cluster which is tracked by the tracker T_t at the tth frame is denoted as $Dt^k = [d_1^k, \ldots, d_4^k]^T$. Assuming that the observation D_t at the tth frame is only related to the given category S_i, the posterior distribution probability of the kth moving object cluster belonging to each category S_i at the tth frame is updated by the state at the $t-1$th frame:

$$P(S_i \mid D_t, \ldots, D_1) = \frac{P(D_t \mid S_i) \cdot P(S_i \mid D_{t-1}, \ldots, D_1)}{\sum_{i=1} P(S_i \mid D_{t-1}, \ldots, D_1) P(D_t \mid S_i)}, \tag{10}$$

where the likelihood function at the tth frame $P(D_t \mid S_i) = \mathrm{sign}(di^t)$. At the end, the maximum posteriori probability is used to estimate the category of the kth cluster tracked by the tracker at the tth frame.

5. Experimental Results

5.1. Data Collection. In order to test the performance of the proposed moving object classification method, four categories of the point cloud samples including vehicle, pedestrian, bicycle, and crowd are collected using Velodyne HDL-64E LIDAR equipped on our autonomous vehicle (Figure 4). The videos generated by 3 external cameras on the autonomous vehicle are acquired synchronously to manually label the real categories for the samples. Meanwhile, 3D LIDAR data in public KITTI dataset [23] is also used to supplement the point cloud samples. The point cloud clusters of moving objects are extracted with the data preprocessing procedure. Note that the extracted clusters of moving objects with the same category are sensed from different view directions and distances. Figure 5 shows a few examples of moving object. All the experiments are processed on an Intel i7-4700, 3.20 GHz core processor with 8 GB RAM using C++ code.

5.2. SVM-BDT-Based Classification Results. The framework for moving object classification is tested on the task of calculating the posterior probability that the point cloud cluster belongs to each category of moving objects at consecutive frames. We run the proposed SVM-BDT classification method using the 18 selected features, and the output of the posterior probability in multiple scenarios are shown in Figure 6. In each subgraph, the upper picture is the scene image captured synchronously and the rectangle represents one moving object tracked by 3D LIDAR, and the bottom picture shows the variation of the posterior probability frame by frame. As shown in Figure 6(b), the pedestrian in the red rectangle suddenly appears and becomes from partially occluded to be completely exposed in the point cloud environment. The posteriori probability that the cluster belongs to the pedestrian increases rapidly after several initial frames, and then it maintains at the maximum value. Figure 6(c) shows that as the bicycle goes far gradually, the number of points returned from the bicycle decreases gradually, and the posterior probability that the point cloud cluster belongs to the bicycle decreases correspondingly. As shown in Figure 6(d), multiple pedestrians walk away from the LIDAR. At first, the pedestrians walk so close that the posterior probability of the category belonging to the crowd is the highest. Then, the distance among the pedestrians increases gradually, the point cloud of the crowd is successfully segmented into multiple pedestrians, and the posterior probability of the category belonging to the pedestrian increases accordingly. Later, the distances among the pedestrians decrease, and the posterior probability of the category belonging to the crowd increases again.

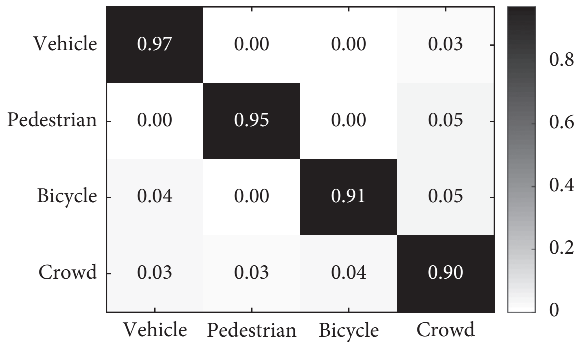


Figure 7: Confusion matrix of recognition results based on SVM-BDT classifier.

To evaluate the performance of the proposed SVM-BDT method quantitatively, 2000 groups of the point cloud samples are selected for each category of moving object, and 5-fold cross validation is conducted. Each group contains 10 consecutive frames. Figure 7 is the confusion matrix of the recognition results. We can see that the recognition accuracies of vehicle, pedestrian, bicycle, and crowd are 97%, 95%, 91%, and 90%, respectively. The proposed classification method shows the best recognition performance on moving vehicles, and the point cloud cluster of the crowd is the most likely to get confused with the other types of moving objects. Overall, the average recognition accuracy of SVM-BDT method is 93.25%, and the recognition result satisfies the requirements of autonomous vehicle on the recognition of surrounding moving obstacles. The total running time of the proposed SVM-BDT method increases with the number of moving objects in the traffic scenario, especially the time cost of BDT-based classification optimization stage, since the point cloud of each moving object is associated with each tracker at the consecutive frames and the trackers are updated based on the association result.

5.3. Classification Performance Comparison. To demonstrate that both the number of features and various classifiers affect the recognition performance, we compare four categories of moving object classification methods, namely, (1) 18 features + SVM, (2) 68 features + SVM, (3) 18 features + SVM + BDT, and (4) 68 features + SVM + BDT. All methods are tested with 5-fold cross validation using 2400 groups of point cloud samples, and each category of moving object has 600 groups of point cloud samples. Note that each group of point cloud sample contains 10 consecutive frames and partial occlusion

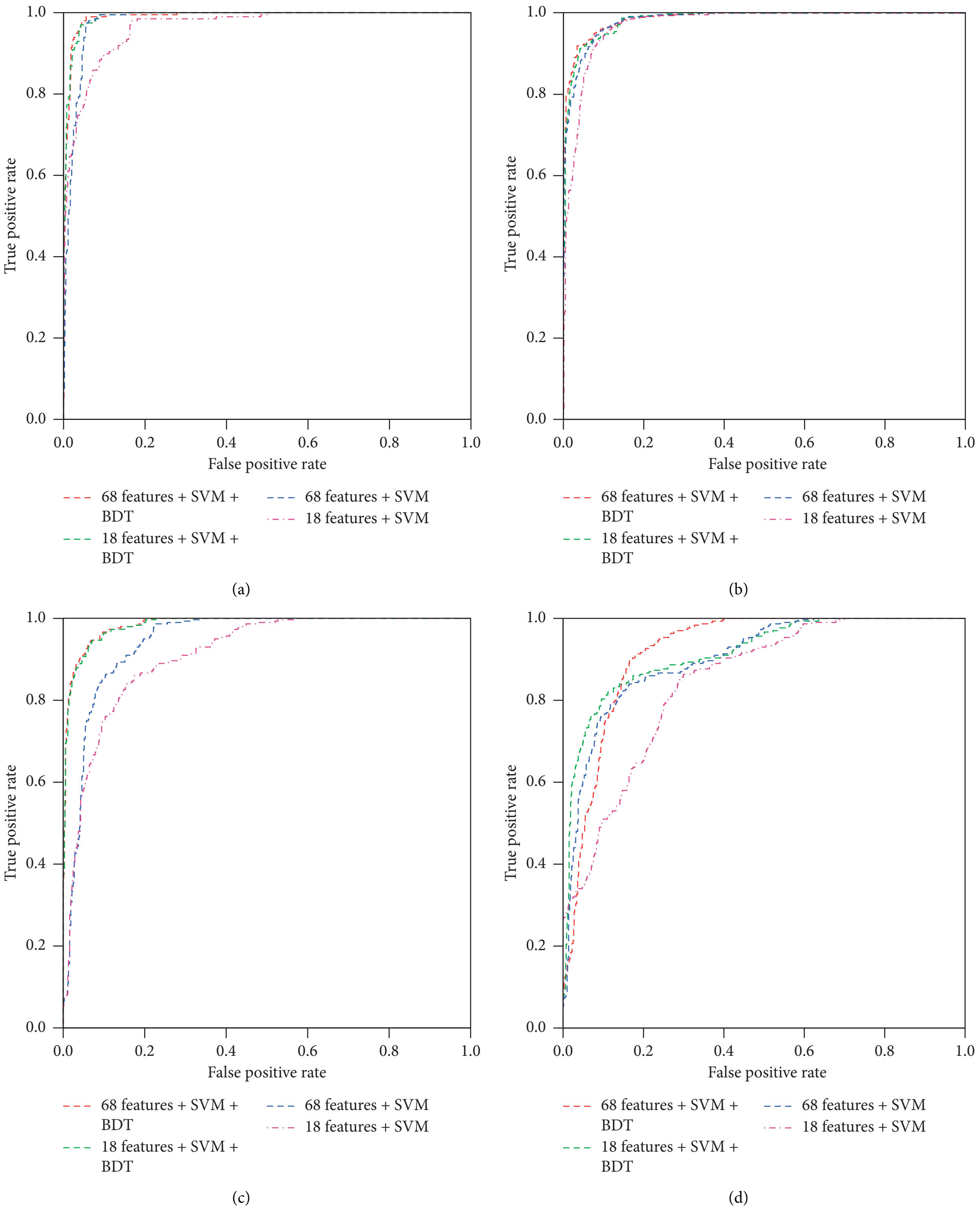


Figure 8: ROC curves of moving object classification results. (a) Vehicle, (b) pedestrian, (c) bicycle, and (d) crowd.

occurs at several frames. The Receiver Operating Characteristic (ROC) curves for four methods are shown in Figure 8. The larger the area under the ROC curve (AUC) is, the better is the performance of the classification method, and vice versa. AUC values and the run time of four methods are listed in Table 4. The run time denotes the total time cost of both the feature extraction and classification stages. We can see that in terms of the SVM-

Table 4: Performance comparison of moving object classification algorithms.

Object	Features	Classifier	AUC	Run time per object (ms)
Vehicle	18	SVM	0.950	40.67
		SVM-BDT	0.989	77.89
	68	SVM	0.980	60.45
		SVM-BDT	0.991	88.38
Pedestrian	18	SVM	0.953	39.77
		SVM-BDT	0.976	72.38
	68	SVM	0.969	57.42
		SVM-BDT	0.985	83.93
Bicycle	18	SVM	0.859	42.94
		SVM-BDT	0.958	75.83
	68	SVM	0.902	59.94
		SVM-BDT	0.968	85.01
Crowd	18	SVM	0.845	41.70
		SVM-BDT	0.931	75.33
	68	SVM	0.923	58.87
		SVM-BDT	0.933	84.15

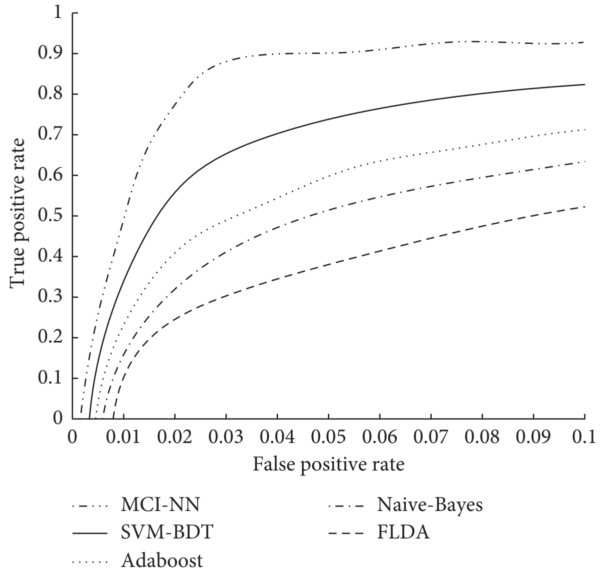


Figure 9: Roc curves of crowd recognition results.

BDT-based moving object classification method, AUC value obtained by 18 features is close to that obtained by 68 features; thus, the characteristics of four categories of moving objects can be explained by 18 selected features well. For the same feature set, the SVM-BDT-based classification method outperforms the SVM-based method. It demonstrates the BDT algorithm using the consecutive frames optimizes the classification result at individual frame effectively.

It even overcomes the partial occlusion. Moreover, for the same classifier, the method using 18 features run less time than the one using 68 features. Considering the recognition accuracy, computation complexity, and operational efficiency, it can be concluded that SVM-BDT classifier based on 18 features is the best choice to achieve the recognition of moving objects including vehicle, pedestrian, bicycle, and crowd.

The crowd is regarded as a special moving object which is different from the single pedestrian in this paper. To further validate the crowd recognition performance of the proposed SVM-BDT-based method, several commonly used recognition methods are compared. The ROC curves of the classification results are shown in Figure 9, and the results clearly show the superiority of the proposed SVM-BDT-based method against Adaboost algorithm [24], Naive Bayes algorithm [25], and FLDA algorithm [26]. Although the crowd recognition accuracy of MCI-NN algorithm [27] outperforms the proposed SVM-BDT-based method, MCI-NN algorithm consumes more memory and takes more operation time due to using Markov kernel function. Therefore, considering the recognition accuracy and efficiency, the proposed SVM-BDT-based method shows better crowd recognition performance.

6. Conclusions and Future Works

In this paper, we propose an approach for moving object classification using 3D point cloud in urban traffic environment. This approach classifies moving objects into four classes, namely vehicle, pedestrian, bicycle, and crowd. The accurate modeling of moving object classification using 3D point cloud consists of several procedures that all affect the final classification results. To obtain the effective features of moving objects, unlike the application of simple feature description, Gini index criterion is employed in this work based on the characteristics of each category of moving objects to select from the extracted features including number-of-points-based features, shape features, and statistical features. In the classification procedure, unlike previous works where the classifier is modeled with the point cloud at single frame, the moving object is recognized based on SVM-BDT classifier to incorporate multiframe classification results. The presented method has three benefits. First, our method can classify the common moving objects in urban environment even if the pedestrians walk close or partial occlusion occurs. Second, this method digs deep into the point cloud distribution based on the category attribute to recognize moving object efficiently. Moreover, the BDT-based classification optimization is conducted on the results of the posterior probability SVM classifier at consecutive frames to improve the moving object classification performance. The method is tested using the point cloud dataset collected by our own LIDAR as well as public KITTI dataset in the experiment. The results reveal that the proposed SVM-BDT method based on 18 features can achieve better classification accuracy for vehicle, pedestrian, bicycle, and crowd, compared with several other methods. Note that the point cloud samples collected by our method are within 40 meters, beyond which the declining resolution of the point cloud caused many mistakes. The proposed method has a limitation on classifying the moving objects at long range; thus, this challenge is treated as a subject of our future work. Another aspect of future work is the deep understanding of the behaviour using 3D LIDAR by integrating the motion cues with the classification results of the surrounding objects in urban traffic environment.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by grants from National Key R&D Program of China (2018YFB1600500), National Natural Science Foundation of China (51905007 and 51775053), the Great Wall Scholar Program (CIT&TCD20190304), Ability Construction of Science and Beijing Key Lab Construction Fund (PXM2017-014212-000033), and NCUT Start-Up Fund.

References

[1] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car-part II: a case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5119–5132, 2015.

[2] D. Li and H. Gao, "A hardware platform framework for an intelligent vehicle based on a driving brain," *Engineering*, vol. 4, no. 4, pp. 464–470, 2018.

[3] H. Gao, G. Shi, G. Xie, and B. Cheng, "Car-following method based on inverse reinforcement learning for autonomous vehicle decision-making," *International Journal of Advanced Robotic Systems*, vol. 15, no. 6, pp. 1–11, 2018.

[4] O. Toro, T. Becsi, S. Aradi, and P. Gaspar, "Sensitivity and performance evaluation of multiple-model state estimation algorithms for autonomous vehicle functions," *Journal of Advanced Transportation*, vol. 2019, Article ID 7496017, 13 pages, 2019.

[5] H. Wang, X. Lou, Y. Cai, Y. Li, and L. Chen, "Real-time vehicle detection algorithm based on vision and lidar point cloud fusion," *Journal of Sensors*, vol. 2019, Article ID 8473980, 9 pages, 2019.

[6] Z. Fang and J. Duan, "Obstacle detection for intelligent vehicle based on LabVIEW and laser measurement system," in *Proceedings of IEEE International Conference on Information and Automation*, pp. 202–206, Karlsruhe, Germany, 2013.

[7] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[8] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4224–4231, 2018.

[9] J. Behley, V. Steinhage, and A. B. Cremers, "Performance of histogram descriptors for the classification of 3D laser range data in urban environments," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4391–4398, Saint Paul, MN, USA, May 2012.

[10] B. Yang and Z. Dong, "A shape-based segmentation method for mobile laser scanning point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 81, pp. 19–30, 2013.

[11] H. Lee and B. Coifman, "Side-fire lidar-based vehicle classification," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2308, no. 1, pp. 173–183, 2012.

[12] B. Kim, B. Choi, S. Park, and K. Kim, "Pedestrian/vehicle detection using a 2.5-D multi-layer laser scanner," *IEEE Sensors Journal*, vol. 16, no. 2, pp. 400–408, 2016.

[13] K. O. Arras, O. M. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2D range data," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1002, pp. 3402–3407, Rome, Italy, April 2007.

[14] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 802–807, Madrid, Spain, 2012.

[15] D. Z. Wang, I. Posner, and P. Newman, "What could move? Finding cars, pedestrians and bicyclists in 3d laser data," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4038–4044, San Francisco, CA, USA, 2012.

[16] A. Teichman, J. Levinson, and S. Thrun, "Towards 3D object recognition via classification of arbitrary object tracks," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4034–4041, Shanghai, China, May 2011.

[17] D. Z. Wang, I. Posner, and P. Newman, "Model-free detection and tracking of dynamic objects with 2D lidar," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1039–1063, 2015.

[18] D. Stumper, F. Gies, S. Hoermann, and K. Dietmayer, "Offline object extraction from dynamic occupancy grid map sequences," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 389–396, Changshu, China, June 2018.

[19] M. Zhang, D. D. Morris, and F. Rui, "Ground segmentation based on loopy belief propagation for sparse 3D point clouds," in *Proceedings of IEEE International Conference on 3D Vision*, pp. 1–9, Lyon, France, October 2015.

[20] T. Duong, G. Beck, H. Azzag, and M. Lebbah, "Nearest neighbour estimators of density derivatives, with application to mean shift clustering," *Pattern Recognition Letters*, vol. 80, pp. 224–230, 2016.

[21] W. Chen, X. Xie, J. Wang et al., "A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility," *Catena*, vol. 151, pp. 147–160, 2017.

[22] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[23] A. Geiger, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, Providence, RI, USA, June 2012.

[24] A. Azim and O. Aycard, "Layer-based supervised classification of moving objects in outdoor dynamic environment using 3D laser scanner," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 1408–1414, Dearborn, MI, USA, June 2014.

[25] C. Premebida, O. Ludwig, and U. Nunes, "Exploiting LIDAR-based features on pedestrian detection in urban scenarios," in *Proceedings of IEEE International Conference on Intelligent*

Transportation Systems, pp. 1–6, St. Louis, MO, USA, October 2009.

[26] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian, "Local Fisher discriminant analysis for pedestrian Re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3318–3325, Portland, OR, USA, June 2013.

[27] O. Ludwig and U. Nunes, "Improving the generalization properties of neural networks: an application to vehicle detection," in *Proceedings of IEEE International Conference on Intelligent Transportation Systems*, pp. 310–315, Beijing, China, October 2008.