applied sciences

MDPI

Article

LiDAR-Based Dense Pedestrian Detection and Tracking

Wenguang Wang [1,*], Xiyuan Chang [1], Jihuang Yang [1] and Gaofei Xu [2]

[1] School of Electronic and Information Engineering, Beihang University, Beijing 100191, China; changxiyuan@buaa.edu.cn (X.C.); yangjihuang@buaa.edu.cn (J.Y.)
[2] Institute of Deep-Sea Science and Engineering, Chinese Academy of Sciences, Sanya 572000, China; xugf@idsse.ac.cn
* Correspondence: wwenguang@buaa.edu.cn

Abstract: LiDAR-based pedestrian detection and tracking (PDT) with high-resolution sensing capability plays an important role in real-world applications such as security monitoring, human behavior analysis, and intelligent transportation. The problem of LiDAR-based PDT suffers from the complex gathering movements and the phenomenon of self- and inter-object occlusions. In this paper, the detection and tracking of dense pedestrians using three-dimensional (3D) real-measured LiDAR point clouds in surveillance applications is studied. To deal with the problem of undersegmentation of dense pedestrian point clouds, the kernel density estimation (KDE) is used for pedestrians center estimation which further leads to a pedestrian segmentation method. Three novel features are defined and used for further PDT performance improvements, which takes advantage of the pedestrians' posture and body proportion. Finally, a new track management strategy for dense pedestrians is presented to deal with the tracking instability caused by dense pedestrians occlusion. The performance of the proposed method is validated with experiments on the KITTI dataset. The experiment shows that the proposed method can significantly increase F1 score from 0.5122 to 0.7829 compared with the STM-KDE. In addition, compared with AB3DMOT and EagerMOT, the tracking trajectories from the proposed method have the longest average survival time of 36.17 frames.

Keywords: LiDAR; pedestrian detection; tracking; segmentation

Citation: Wang, W.; Chang, X.; Yang, J.; Xu, G. LiDAR-Based Dense Pedestrian Detection and Tracking. *Appl. Sci.* **2022**, *12*, 1799. https://doi.org/10.3390/app12041799

Academic Editor: João Miguel Fernandes Rodrigues

Received: 15 January 2022
Accepted: 1 February 2022
Published: 9 February 2022

1. Introduction

LiDAR is widely used in various fields due to its high resolution and robustness against time-varying illumination conditions. In engineering, surveying, and mapping, it can be used for point cloud reconstruction to achieve 3D modeling, disaster assessment, and target recognition. In intelligent transportation systems, it can be used for environmental awareness to realize automatic driving and safety monitoring.

People are the most important part of our society. Pedestrians appear in all scenes of our social life. Pedestrian detection and tracking (PDT) is an important part of environmental perception, and is still a challenging and popular issue in academic research, including the areas of security monitoring, automatic driving, etc., due to the diversity of dense pedestrian postures and mutual occlusion. Pedestrian detection and tracking based on visual images has been widely studied during the last decades [1–3]. Note that the performance of vision-based PDT methods seriously relies on the illumination conditions, which, however, can be easily affected by environmental factors (e.g., rain or snow). In addition, distance information of pedestrians of interest (PoIs) is missing in two-dimensional (2D) visual detection, which can potentially degrade the PDT accuracy. Compared with vision-based detection, 3D point clouds obtained from LiDAR can provide precise position, motion status, and movement features of pedestrians, which is an alternative choice in the problem of PDT.

At present, many scholars are carrying out research about pedestrian detection and tracking. In [4], the occupancy grid method and Kalman filter are adopted to detect and track pedestrians on urban roads. The occupancy gird method is simple and intuitive to

apply; however, this method faces the problem of large false alarms caused by obstacles. Reference [5] presents an approach of pedestrian detection and tracking using in-vehicle LiDAR. In this approach, pedestrians are detected according to the features of reflected signals and geometrical distribution; for tracking, the states are estimated by the interactive multi-model (IMM) and the auction algorithm. Reference [6] proposed a simultaneous detection and tracking (SDAT) method for pedestrian trajectory recovery. In this method, only moving objects are of interest. Instead of detecting the individual object, the SDAT detects its trajectory directly by assigning the point data to a trajectory hypothesis. This idea can be regarded as a kind of track-before-detect method. However, it has the drawback of time inefficiency and is not suited to online tracking. Reference [7] proposed a distance-aware projection method, mapping a 3D point cloud to a 2D plane to detect pedestrians. Again based on 3D projection, reference [8] selected four features to train an SVM classifier to detect pedestrians and then obtain their trajectories via a Kalman filter. The 3D projection method is effective for close pedestrian detection with dense point cloud. Each object has its own unique geometry. The model-based box fitting method is used to distinguish pedestrians and vehicles in [9], meanwhile, the box fitting can provide pose estimation, which will be used for moving targets tracking. Based on a 3D bounding box, PointRCNN is proposed for 3D object detection in [10], which can generate 3D proposals from raw point cloud in a bottom-up manner. In regards to the 3D multi-object tracking (MOT), a baseline and new evaluation metrics were proposed in [11] based on a 3D Kalman filter. In addition, camera technology and LiDAR can be used together to improve the surveillance performance. Reference [12] fuses the detection results of the two sensors and then performs two-stage data association to achieve pedestrian tracking. Reference [13] uses a deep neural network to detect pedestrians in RGB images and depth images converted by point clouds. Its novelty lies in the use of the Kalman filter for both sensor fusion and tracking. Although the above multi-sensor fusion methods have high precision, they are complicated.

Unlike color discrimination, point clouds usually reflect target differences through feature extraction. For example, to distinguish obstacles such as low trees and traffic signs from pedestrians in the road scene. It is well known that the more effective the extracted features are, the more accurate the pedestrian detection effect will be. Ten 3D point cloud features, including normalized moment of inertia tensor, 2D histogram, and reflection intensity distribution, are summarized in reference [7]. In [14], a density enhancement method is proposed to improve the quality of sparse point clouds by radial basis function (RBF)-based interpolation. In [15], kernel density estimation (KDE) is used for pedestrian clustering, and point clouds are projected onto the main plane to extract local adaptive regression kernel (LARK) features. Then, single template matching (STM) is used for pedestrian detection. Note that the geometric information is widely used in the above mentioned literature to distinguish pedestrians from clutters. As a widely used pedestrian surveillance method, the KDE is also used for pedestrian segmentation, pedestrian mobility configuration, and pedestrian densities analysis in references [16–18].

With the development of artificial intelligence technology, 3D point cloud target detection based on deep learning has emerged in recent years. Unlike VoxNet [19], which requires a voxel processing first, PointNet [20] and PointNet++ [21] could learn features directly from the original point cloud. On this basis, PointRCNN [10] and PointPillars [22] were proposed for targets detection. In addition, the YOLO structure, which can be used for real-time detection, has been widely investigated. YOLO3D [23] and Complexer-YOLO [24] are proposed for pedestrian detection on point clouds. WYs-WM [25] uses a weighted mean scheme to unite the detection results based on YOLO framework on images and point clouds. The experimental results on the KITTI dataset show that they can detect both pedestrians and vehicles. Furthermore, the CenterPoint framework is proposed in [26] and applied on NUSCENES dataset to detect pedestrians. The detection performance of deep learning method is heavily dependent on training data. Although currently available datasets for pedestrian detection include Pandaset, Apollo, etc., the 3D point cloud dataset is still incomplete due to the complexity of the real scene.

Another challenge of the PDT problem is occlusion among dense pedestrians, which may cause a model to unexpectedly miss pedestrians and further degrades the robustness of the whole framework. A decentralized multi-target tracking algorithm is proposed in [27], which reduces occlusions using a LiDAR sensor network. However, using a distributed network with multiple sensors always leads to a complex and expensive system. In [28], a part-based model is used to handle the missed detections and partial occlusion, where a camera is used to capture the individual information in high quantity and quality. However, it is hard to accurately detect occluded pedestrians with only partial LiDAR points, and false detections emerging randomly make the problem more challenging. The track management method is an important part of object tracking for better state estimates in the presence of missed detections and false alarms [29]. A simple track management method initializes tracks from a few consecutive associated detections or deletes tracks from a few consecutive missed detections [30]. A complex trajectory management strategy was proposed in [31]. It contains six trajectory states: Init-Active, Init-Inactive, Identified-Active, Identified-Inactive, Deleted, and Archived. State switches are accomplished through the Long-Term Assignment (LTA) and the Short-Term Assignment (STA). From this perspective, track management considering occlusion is another way to improve the performance of PDT.

To high-resolution LiDAR, clustering is the basis of pedestrian detection and tracking, and when the pedestrians are close, clustering undersegmentation becomes a challenging problem. For dense pedestrians, their point clouds are adjacent to each other or even directly connected, while the point cloud density at the torso is higher than that at the junction between pedestrians. Therefore, it is possible to improve the clustering performance of dense pedestrians by properly modeling the difference of point cloud density between torso and junction. Note that standing pedestrians are almost perpendicular to the ground, and there are certain geometric constraints between the width and height of human body. These characteristics can provide discrimination for pedestrian detection. For the occlusion problem, occluded targets can be found by dividing the area into free, occupied, and occluded subareas. Then, adjusting the track management strategy for occluded targets can provide more accurate and reliable tracking results. The main contributions of this paper are as follows:

- A new KDE-based method for dense pedestrian segmentation is proposed, where the KDE method is used to estimate the density distribution of pedestrian point clouds on the horizontal plane. In the presented method, each of the pedestrians can be represented by a local maximum, which can improve the segmentation performance of pedestrian clustering.
- Three novel features are defined in this paper according to pedestrians' posture and body proportion using eigenvalues and eigenvectors extracted from the covariance matrix of the point cloud. The first feature shows whether the main direction of candidate point cloud is upright or not, which is useful to distinguish upright pedestrians from targets that are not upright. The other two features can describe the human outline, which is used to distinguish pedestrians from targets that do not satisfy human body proportions.
- A new track management strategy for dense pedestrians is presented aiming at the problem of tracking instability caused by occlusion among dense pedestrians. Four trajectory states, candidate trajectory, matched trajectory, occluded trajectory, and terminated trajectory, are defined, and the states transition processes are given in the proposed track management framework.

Among the above three contributions, the first is just applying the existing KDE to dense pedestrian segmentation, and the other two belong to method design innovation. The rest of the paper is arranged as follows: Section 2 gives solutions and improved methods for LiDAR pedestrian detection and tracking. Firstly, the framework of the proposed pedestrian detection and tracking method is introduced. Then, the implementation process is introduced in detail, including fine segmentation of dense pedestrians, pedestrian detection based on features, and pedestrian tracking considering occlusion. Section 3

shows the verification of proposed methods, mainly based on the KITTI dataset, and some experimental results are given and analyzed. Finally, the concluding remarks are given in Section 4.

2. Methodology

2.1. Overall Framework

The framework of the proposed dense pedestrian detection and tracking method is shown in Figure 1, which mainly contains the following modules: ground elimination, clustering, fine segmentation, pedestrian detection, and pedestrian tracking. The ground elimination is the process of removing point clouds from ground. Random Sample Consensus (RANSAC) algorithm [32] can be used to fit the ground plane for ground point clouds elimination. After filtering out the ground points, the remaining point clouds can be clustered to separate pedestrians from each other. It is common to obtain undersegmented clustering results using existing clustering methods such as the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [33] and the Radially Bounded Nearest Neighbor (RBNN) algorithm [34] when pedestrians gather together and their point clouds are close to each other.

In this section, kernel density estimation is used for the fine segmentation of dense pedestrians based on RBNN. Subsequently, preprocessing is performed to remove high walls and large buildings, and then the extracted features related to the posture and figure of the human body are used for pedestrian detection. According to the detection results and the geometric relationship between pedestrians, the tracking strategy of the occlusion area is designed to improve the tracking performance. The specific implementation of fine segmentation, dense pedestrian detection, and dense pedestrian tracking are introduced in Sections 2.2–2.4.

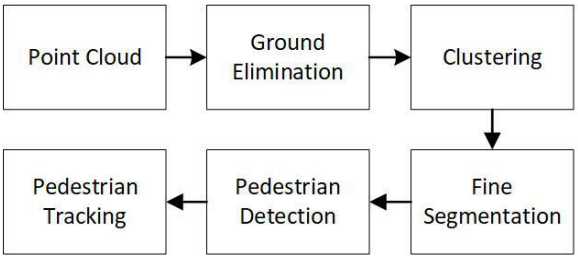


Figure 1. Dense pedestrian detection and tracking flow.

2.2. Fine Segmentation of Dense Pedestrians

After RBNN clustering, the objects in the scene are segmented into different groups. However, adjacent pedestrians might be assigned the same group if their point clouds are really close or even connected. As a result, the dense pedestrians are often undersegmented. As the foundation, the RBNN clustering is used as a preliminary result, which allows dense pedestrian point clouds to be divided into the same cluster. The preliminary clustering needs to be further refined to separate the adjacent pedestrians.

To obtain the fine segmentation of dense pedestrians, the result of RBNN clustering is projected onto the horizontal plane. Point cloud density is highly correlated with the height of pedestrians within the LiDAR illumination range. To a person, whose point cloud may spread in large area, the points from the body will focus on the center. Thus the point cloud from the torso will be denser than that of the adjacent area. Kernel density estimation can be used to estimate the density distribution of pedestrian point clouds on the horizontal projection surface, and the local maximum of the density distribution can be used as the clustering center of a single pedestrian. Then, the undersegmented pedestrian point clouds can be finely segmented according to the center of the pedestrian.

The coordinate of the point cloud set P in the XOY plane is $P_{xoy} = p_1, \ldots, p_n$, and the point cloud density at position p can be obtained using kernel density estimation, as shown in Equation (1).

$$f(p) = \frac{1}{n}\sum_{i=1}^{n} K(p - p_i, \tau) \tag{1}$$

where n is the point number of the point cloud set, τ is the window width, which is related to the width of the pedestrian, p_i is the position of the ith point of the set, and $K(\bullet)$ is the Gaussian core function, as shown in the Equation (2).

$$K(p - p_i, \tau) = \frac{1}{2\pi\tau\Sigma^{1/2}} \exp\{-\frac{(p - p_i)^T \Sigma^{-1}(p - p_i)}{2\tau^2}\} \tag{2}$$

where Σ is the covariance matrix of Gaussian distribution and T means matrix transpose. As the 3D coordinates are uncorrelated, this paper uses an identity matrix as the covariance matrix. Combining Equations (1) and (2), and then ignoring the constant coefficient, we find

$$f(p) = \frac{1}{n}\sum_{i=1}^{n} \exp\{-\frac{(p - p_i)^T (p - p_i)}{2\tau^2}\} \tag{3}$$

Equation (3) can be used to describe the density distribution of the point cloud on the horizontal plane. Local extreme values will be generated somewhere in the pedestrian's torso, and the local maximum point of the density distribution can be used as the cluster center of a single pedestrian.

After obtaining the cluster centers, the distances from each point to cluster centers on the horizontal projection plane can be used to assign it to the nearest cluster center. The fine segmentation is to divide confused points into isolated clusters.

2.3. Dense Pedestrian Detection

After the processing of clustering and segmentation, some geometric features, such as the width and height of a target, can be used to remove high walls and large buildings. As we all know, pedestrians are non-rigid, and their postures are time-varying. However, pedestrians are generally upright, and the geometric features, such as height and width, will maintain a certain proportion. By extracting these features, pedestrians can be distinguished from obstacles such as trees, traffic lights, and fire hydrants. During the fine segmentation, a large target, such as a vehicle, may be divided into several false pedestrians, in which case the geometric features will play a role in improving the detection performance.

Suppose $P = \{d_1, d_2, \ldots, d_n\}$ is a point cloud after fine segmentation, where d_i is the 3D position of a single point. The covariance matrix of the point cloud coordinates contains the positional relationship among the points. After eigendecomposition, the eigenvalues and eigenvectors of the covariance matrix can be used to characterize pedestrians. The process of eigendecomposition is given below:

(1) Centralizing all point clouds, $d_i' \leftarrow d_i - \frac{1}{n}\sum_{i=1}^{n} d_i$;
(2) Calculate the covariance matrix $X = \frac{1}{n-1}DD^T$, where $D = [d_1', d_2', \ldots, d_n']$;
(3) Based on eigen-decomposition of the covariance matrix, we can find three eigenvalues $\lambda_1, \geq \lambda_2$, and $\geq \lambda_3$ and the corresponding eigenvectors w_1, w_2, and w_3.

The eigenvector w_1 corresponds to the maximum eigenvalue, the point cloud has the greatest variance along the vector w_1 than other directions, and the variance is λ_1. The eigenvector w_2 corresponds to the second eigenvalue, which is orthogonal to w_1 and the variance along the vector w_2 is λ_2. Similarly, w_3 is orthogonal to w_1 and w_2, and the corresponding variance is λ_3. Based on the eigenvalues and eigenvectors, we can define three features to describe pedestrians.

When the pedestrian is upright, then the maximum variance direction of the point cloud should be perpendicular to the ground, which means the eigenvector w_1 should be perpendicular to the ground. The ground normal vector e can be obtained in the ground elimination process using RANSAC. We can take the angle between eigenvector w_1 and the vector e as the first feature. The calculation of the angle θ is shown in (4). The distribution

of angle θ can be modeled by Rayleigh distribution, as shown in (5). The distribution described in (5) can be fitted using real data so as to obtain the parameter σ_θ.

$$\theta = \text{acos}\left(\frac{w_1 \cdot e}{\|w_1\|\|e\|}\right) \quad 0 < \theta < \frac{\pi}{2} \tag{4}$$

$$p(\theta, \sigma_\theta) = \frac{\theta}{\sigma_\theta^2} \exp\left(-\frac{\theta^2}{2\sigma_\theta^2}\right) \tag{5}$$

Based on the report [35], which presents the results of a comprehensive anthropometric survey of U.S. Marines, λ_1 is the variance of the point cloud projected in the direction w_1, which is positively correlated with the height of pedestrians. Similarly, λ_2 and λ_3 can reflect the width of the human body. However, λ_1, λ_2, and λ_3 vary as the number of points changes. The two ratios shown in (6) and (7) are useful to describe human outline.

$$r_1 = \lambda_1 / \lambda_2 \tag{6}$$

$$r_2 = \lambda_2 / \lambda_3 \tag{7}$$

When LiDAR rays are uniformly distributed in space, assuming that the pedestrian's height is H, shoulder width is W, and chest depth is D, we use (8)–(10) to estimate λ_1, λ_2, and λ_3, respectively.

$$\hat{\lambda}_1 = \frac{1}{H} \int_0^H (x - H/2)^2 dx = \frac{H^2}{12} \tag{8}$$

$$\hat{\lambda}_2 = \frac{1}{W} \int_0^W (x - W/2)^2 dx = \frac{W^2}{12} \tag{9}$$

$$\hat{\lambda}_3 = \frac{1}{D} \int_0^D (x - D/2)^2 dx = \frac{D^2}{12} \tag{10}$$

Thus, we can obtain the estimated $\hat{r}_1$ and $\hat{r}_2$ with (11) and (12).

$$\hat{r}_1 = \hat{\lambda}_1 / \hat{\lambda}_2 = \frac{H^2}{W^2} \tag{11}$$

$$\hat{r}_2 = \hat{\lambda}_2 / \hat{\lambda}_3 = \frac{W^2}{D^2} \tag{12}$$

The estimated $\hat{r}_1$ and $\hat{r}_2$ provide the reference for r_1 and r_2. On the basis of fine clustering, pedestrian detection can be realized based on (13) by extracting the three features, where θ_{thr}, r_{tl1}, r_{tr1}, r_{tl2}, and r_{tr2} are threshold parameters. A cluster satisfying Equation (13) can be detected as a pedestrian.

$$\begin{cases} \theta < \theta_{thr} \\ r_{tl1} < r_1 < r_{tr1} \\ r_{tl2} < r_2 < r_{tr2} \end{cases} \tag{13}$$

2.4. Dense Pedestrian Tracking

Pedestrian tracking can provide not only trajectories, but also valuable information for behavior analysis. In addition, in occlusion areas, the motion prediction also provides useful information for pedestrian detection. The state estimation of pedestrians is introduced in Section 2.4.1, data association process is given in Section 2.4.2, and trajectory management considering occlusion is given in Section 2.4.3.

2.4.1. State Estimation

The walking of pedestrians on the ground can be modeled as a two-dimensional movement. The LiDAR scanning frequency is usually high. For Velodyne HDL-64E, its typical scanning frequency is 10 Hz, and pedestrian movement is relatively slow, thus pedestrian velocity can be seen as constant. During the LiDAR scan, the state of pedestrians can be modeled as $X_k(x, y, v_x, v_y)$, where (x, y) is the centroid and (v_x, v_y) is the velocity. The pedestrian state transition is modeled as follows:

$$X_{k|k-1} = FX_{k-1} + w_{k-1}, \quad w_{k-1} \sim N(0, Q_{k-1}) \tag{14}$$

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_2 \tag{15}$$

$$Q_{k-1} = \sigma_v^2 \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} \\ \frac{T^3}{2} & T^2 \end{bmatrix} \otimes I_2 \tag{16}$$

where F is the transition matrix and w_{k-1} is the process noise which obeys Gaussian distribution with a covariance matrix Q_{k-1}. T is the time interval and σ_v is the standard deviation of the process noise. I_2 denotes the 2×2 identity matrix. $\otimes$ means Kronecker tensor product. Note that many other appropriate models (e.g., discrete walking model [36], terrain-constrained model [37], and social force-based model [38]) also can be used here to represent the pedestrian behaviors in realistic applications.

Each person within the LiDAR detection range will produce a set of measurements rather than a single point. Suppose $y_k = \{y_k^1, y_k^2, \ldots, y_k^n\}$ is a set of LiDAR points originating from a person detected by using the method proposed in Section 2.3. As shown in (17), it is simple and effective to use the centroid of the set to represent the person.

$$Y_k = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{17}$$

When a person is represented by its centroid, the observation model can be written as follows:

$$\hat{Y}_k = HX_{k|k-1} + e_k, \quad e_k \sim N(0, R_k) \tag{18}$$

$$H = \begin{bmatrix} I_2 & 0_2 \end{bmatrix} \tag{19}$$

$$R_k = \sigma_e^2 I_2 \tag{20}$$

where H is the observation matrix, 0_2 denotes the 2×2 zero matrix, e_k is the observation noise which obeys Gaussian distribution with a covariance matrix R and σ_e is the standard deviation of the observation noise.

The Kalman filter [39] is used to predict and update the pedestrian state in this paper. The prediction of the state and the state covariance can be obtained by

$$X_{k|k-1} = FX_{k-1|k-1} \tag{21}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q \tag{22}$$

where $X_{k-1|k-1}$ and $P_{k-1|k-1}$ are the target state and state covariance matrix at time $k-1$, respectively. F is the state transition matrix and Q is the process noise covariance matrix. According to (21) and (22), we can gain the predicted state and the predicted state covariance matrix at time k.

After the data association, the centroid of each target can be regarded as observation $Y_k(x, y)$. The pedestrian status will be updated by (23)–(25):

$$K = \frac{P_{k|k-1}H^T}{\left(HP_{k|k-1}H^T + R\right)^{-1}} \tag{23}$$

$$X_{k|k} = X_{k|k-1} + K(Y_k - HX_{k|k-1}) \tag{24}$$

$$P_{k|k} = (I - KH)P_{k|k-1} \tag{25}$$

where K is the Kalman filter gain and $X_{k|k}$ and $P_{k|k}$ are the updated state and its covariance matrix, respectively. Equations (21)–(25): constitute the complete Kalman filtering process.

2.4.2. Data Association

Data association is the key part of the traditional multi-target tracking problem [40–42]. Assuming that the targets' trajectories at time $k-1$ are $T^{k-1} = \{t_1^{k-1}, t_2^{k-1}, \ldots, t_N^{k-1}\}$, the predicted target trajectories at time k are $\hat{T}^k = \{\hat{t}_1^k, \hat{t}_2^k, \ldots, \hat{t}_N^k\}$ according to Kalman filtering. If M pedestrians $Z^k = \{z_1^k, z_2^k, \ldots, z_M^k\}$ are detected at time k, the purpose of the association is to match the measurements Z^k with the predicted trajectories $\hat{T}^k$.

Assignment matrix $A = [a_{ij}]_{N \times M}, a_{ij} \in \{0, 1\}$ can be used to describe the relationship between the trajectories and the measurements. $a_{ij} = 1$ indicates that trajectory i is matched to the measurement j and $a_{ij} = 0$ indicates that they are not matched. In tracking, location is the basis of target association. At the same time, the LiDAR echo intensity is related to the color and material of clothes, so the echo intensity and position can be combined to construct the cost matrix $C = [c_{ij}]_{N \times M}$. The association cost between trajectory i and measurement j is as follows:

$$c_{ij} = \begin{cases} \omega_{ij}(v_j - u_i)^T(v_j - u_i), & (v_j - u_i)^T(v_j - u_i) < \tau_c \\ \infty, & \text{Otherwise} \end{cases} \tag{26}$$

$$\omega_{ij} = 1 - \frac{R}{2\sigma_m} \tag{27}$$

where u_i and v_j are the positions of trajectory i and the measurement j respectively. The ω_{ij} represents the different degree of echo intensity. To the trajectory i, we can obtain the range $(I_m - \sigma_m , I_m + \sigma_m)$ of its echo intensity, where I_m and σ_m are the mean and standard deviation of the echo intensity of all the measurements that comprise the trajectory i, respectively. To the measurement j, we can obtain the range $(I_j - \sigma_j , I_j + \sigma_j)$ of its echo intensity, where I_j and σ_j are the mean and standard deviation of the measurement j respectively. The R represents the length of overlap between $(I_j - \sigma_j , I_j + \sigma_j)$ and $(I_m - \sigma_m , I_m + \sigma_m)$.

The total association cost can be represented as Ψ:

$$\Psi = \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij} a_{ij} \tag{28}$$

The data association is to find the optimal assignment matrix A^* that minimizes the Ψ.

$$A^* = \arg\min_{A} \Psi \tag{29}$$

Equation (29) can be solved using the Hungarian algorithm [43].

2.4.3. Track Management Considering Occlusion

Usually, the purpose of the association is to match detections with predicted trajectories. In traditional trajectory management strategy, a new trajectory will be initialized if a target is detected in successive frames. An existing trajectory survives when there is detection matched with it. Alternatively, an existing trajectory will be terminated if there is not any detection matched with it for a long time. However, when a target is occluded and not detected for a long time, its corresponding trajectory will be terminated, and when the target reappears, a new trajectory will be initialized, which will cause the trajectory of the occluded target to be unstable. In order to solve this problem, we make some improvements via trajectory management considering occlusion. There are four trajectory states

for the occluded trajectory: candidate trajectory, matched trajectory, occluded trajectory, and terminated trajectory. As shown in Figure 2, these states can transfer to each other. The corresponding conditions in Figure 2 are as follows:

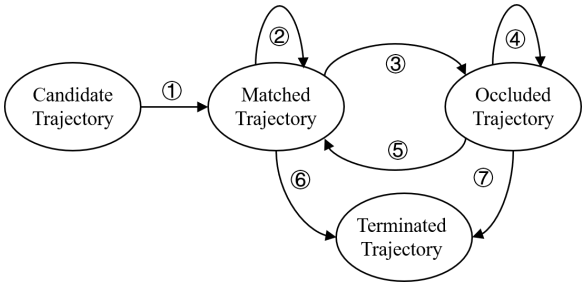


Figure 2. Trajectory state transition.

① If a pedestrian is continuously detected in T_1 successive frames, initiate a new trajectory and add it to the matched trajectories.
② If a matched trajectory has matched detection in the current frame, it keeps the state of matched trajectory.
③ If a matched trajectory can not match with any detection in the current frame, and its predicted position is in occlusion, it turns to be an occluded trajectory.
④ An occluded trajectory keeps its state if there is no detection matching with it in a short time, such as no more than T_2 frames.
⑤ An occluded trajectory turns to a matched trajectory once it is matched with a detection.
⑥ A matched trajectory will be terminated if it has not matched with any detection in T_3 consecutive frames and it was not occluded.
⑦ An occluded trajectory will be terminated if it has not matched with any detection for T_2 successive frames.

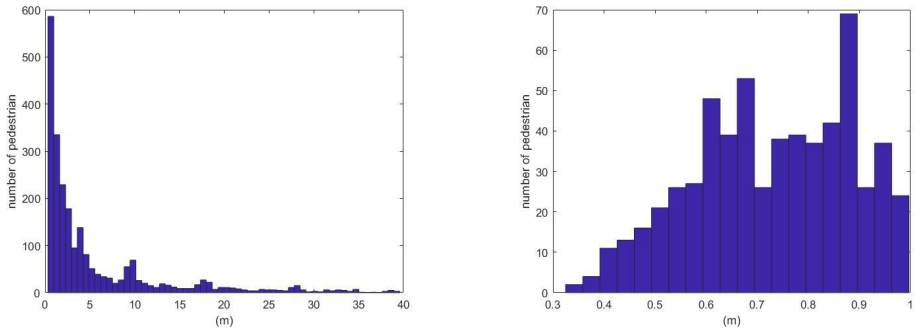
The virtual scan can be used to detect occlusion [44]. After ground elimination and gridding, the scanning area is divided into free, occupied, and occluded sub-regions. When a target falls into the occluded area, it means that it will be blocked or partially blocked. If a pedestrian is missed in the occluded area, the parameter T_2 will be enabled to judge the transition of tracking state, otherwise T_3 is enabled.

3. Experimental Analysis

The experiment includes the following three aspects: segmentation performance evaluation (Section 3.1), detection performance evaluation (Section 3.2), and tracking performance validation (Section 3.3). In Section 3.1, the fine segmentation method is compared with RBNN. The STM-KDE [15] is compared with the proposed detection method in Section 3.2. In Section 3.3, the proposed tracking method is mainly compared with AB3DMOT [11] and EagerMOT [12]; both of them are excellent methods proposed recently for point cloud processing.

The KITTI datasets are used to verify and analyze the performance of the proposed method. The 2011_09_28_drive_0016 is collected at an intersection on campus with 186 frames. These data are used for Sections 3.1 and 3.2. The 0016 and 0017 data of the KITTI MOT benchmark training set are collected in the campus with 209 frames and the urban area with 145 frames, respectively. These data are used for Section 3.3. The above three experimental datasets contain dense pedestrians with self- and inter-target occlusion. In addition, we have acquired data in Beihang University using Livox Horizon LiDAR. We can call these Beihang data in this paper. Beihang data include seven nonlinear moving persons. Their wandering in smaller areas produces serious occlusion, which will be applied to verify the tracking performance for dense nonlinear moving pedestrians in Section 3.3. Taking 2011_09_28_drive_0016 as an example, to represent the density of pedestrians, the statistical results of the nearest distance among pedestrians are shown in Figure 3. It can be seen that the maximum proportion of pedestrian distance is less than

1 m. Figure 3b shows the distribution of pedestrians distance within 1 m. Therefore, the experimental scenery belongs to the typical area.



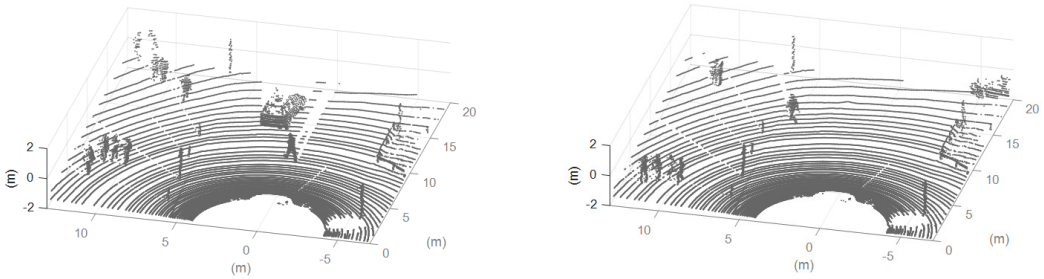
(**a**) (**b**)

Figure 3. Histogram of the nearest separating distance among pedestrians in experimental data. (**a**) Distance distribution among pedestrians. (**b**) The zoom in of the distance within 1 m.

3.1. Segmentation Performance Evaluation

To verify the performance of fine segmentation of dense pedestrians, an area close to the LiDAR, with dense point clouds and many pedestrians gathering, is chosen from the 2011_09_28_drive_0016 dataset.

Frame 1 and frame 38 of the experimental dataset are shown in Figure 4. Before clustering, we use RANSAC to remove ground points. The clustering results are shown in Figure 5. Figure 5a,d are the clustering results of RBNN. Figure 5b,e are the kernel density estimation of RBNN clustering results, which are the basis of fine segmentation. Figure 5c,f are the clustering results after fine segmentation.



(**a**) (**b**)

Figure 4. Two frames of original point clouds. (**a**) Frame 1. (**b**) Frame 38.

As shown in Figure 5a,d, for the dense pedestrian point clouds in the red box, the RBNN clustering appears undersegmentation. However, in Figure 5c,f, the proposed fine segmentation method can segment them well. The reason for the clustering difference is that the RBNN method only uses the distance among points to cluster different pedestrians. However, fine segmentation method uses the distance and the density of point cloud to distinguish multiple adjacent pedestrians. As shown in Figure 5b,e, the local extreme value appears at the position of upright pedestrians, and the pedestrians centers can be extracted effectively, resulting in a better separation performance of the dense pedestrians.

However, neither the RBNN nor the proposed fine segmentation method can segment the two pedestrians in the black box of frame 38 accurately as they are close to each other and occluded seriously. Their optical image is shown in the red box in Figure 6a. Figure 6b,c show their LiDAR point cloud and the kernel density estimation, respectively. It can be seen that the kernel density estimation of the point cloud, just from one single peak caused by the very close distance, exhibits the wrong result of clustering.

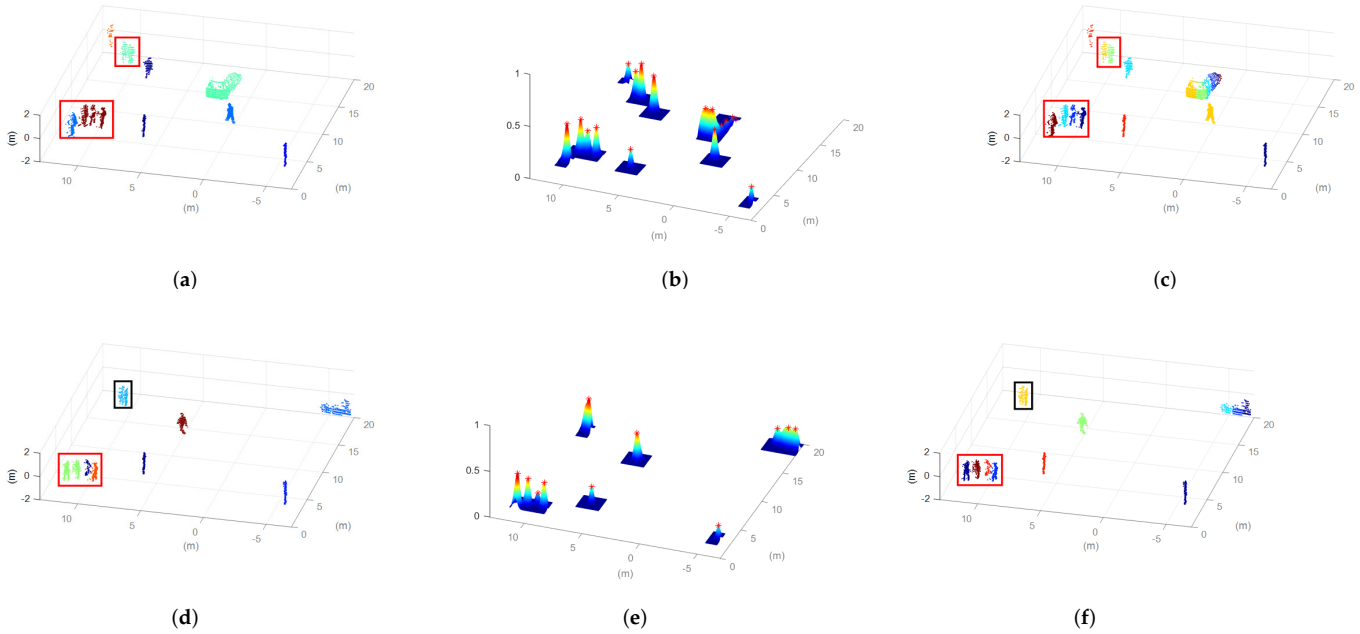


Figure 5. Clustering results of frame 1 and frame 38. (**a**) RBNN clustering result of frame 1. (**b**) Kernel density estimation of frame 1. (**c**) Fine segmentation result of frame 1. (**d**) RBNN clustering result of frame 38. (**e**) Kernel density estimation of frame 38. (**f**) Fine segmentation result of frame 38.

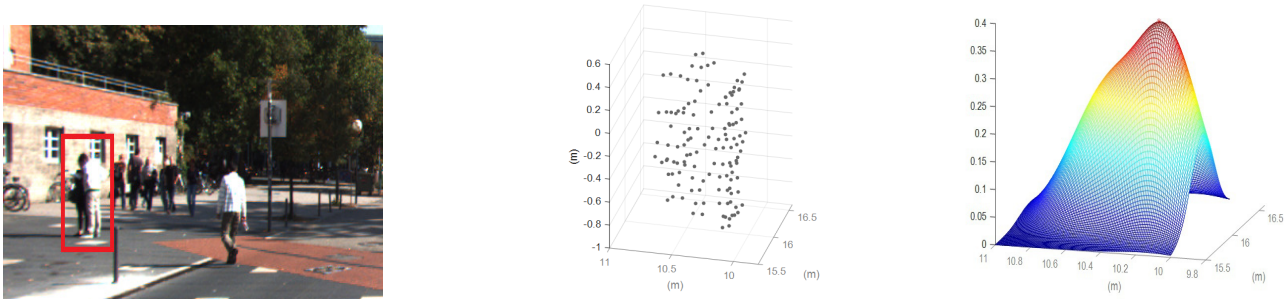
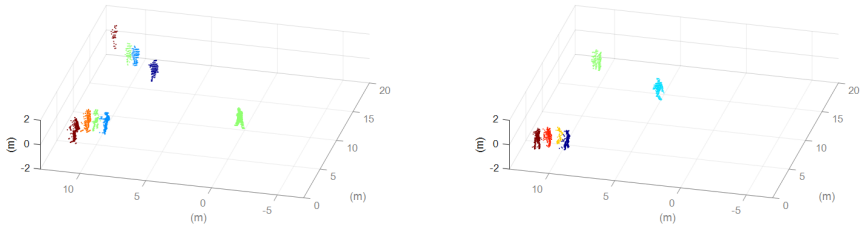


Figure 6. Different descriptions of two very close pedestrians. (**a**) Optical image. (**b**) LiDAR point cloud. (**c**) Kernel density estimation result.

3.2. Detection Performance Evaluation

To validate the detection performance of the proposed method, the 2011_09_28_drive_0016 data in the KITTI dataset are used.

After the preprocessing, such as the clustering and segmentation, we can use the three novel features to detect pedestrians. Figure 7a,b are, respectively, the pedestrian detection results after features-based discrimination of frame 1 and frame 38 under the parameters $\theta_{thr} = 0.35$, $r_{tl1} = 1$, $r_{tr1} = 20$, $r_{tl2} = 1$, and $r_{tr2} = 50$. By comparing the fine segmentation clustering results in Figure 5c,f with the detection results in Figure 7a,b, it can be seen that the vehicle and traffic poles in the clustering results have been successfully removed. That means the three novel features defined in this paper can effectively distinguish pedestrians from other objects, such as vehicles, traffic poles, and trees.



(**a**) (**b**)

Figure 7. Pedestrian detection results of frame 1 and frame 38 after features-based discrimination. (**a**) Frame 1. (**b**) Frame 38.

The STM-KDE is used as the benchmark for the performance evaluation. In [15], kernel density estimation is used for pedestrian clustering and LARK features are extracted for pedestrian detection. The dataset is also used in [15] and pedestrian labels are available. In the dataset, there are 2174 pedestrians marked in 186 frames, but some of the partially occluded pedestrians are not marked. Based on the labels in [15], we additionally mark the partially occluded pedestrians and obtain 2545 pedestrians in 186 frames. Given by (30) the accuracy rate, recall rate, and F1 score are used to measure the detection effect.

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \quad \mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \quad \mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P}+\mathrm{R}} \tag{30}$$

where TP is the number of true positives, FP is the number of false positives, FN is the number of false negatives, P and R are the accuracy rate and recall rate, respectively, and the F1 score is the harmonic average of P and R. The F1 score is widely used as the performance evaluation index, which means that the higher the F1 score is, the better the detection performance of the method is.

Table 1 shows the detection results of the STM-KDE and the proposed method on the 2011_09_28_drive_0016 dataset. It can be found that the proposed method not only improves the detection accuracy, but also significantly improves the recall rate of pedestrian detection. The F1 score is also improved from 0.5122 to 0.7838. Figure 8 shows the ROC curves of different detection methods. The blue line shows the performance of the proposed method. The red line shows the performance of the STM-KDE. It can be seen that, under the same false-alarm number, the detection rate of the proposed method is higher than STM-KDE. The point cloud from long-distance pedestrian is sparse. STM-KDE uses the unique template based on LARK feature to match targets, which is difficult to use effectively for pedestrians at different distances. Perhaps using multiple templates from different distances could improve the detection performance of STM-KDE. In the proposed method, the three novel features are related to the proportion of pedestrian body, so they are robust to different distances and can obtain better detection performance.

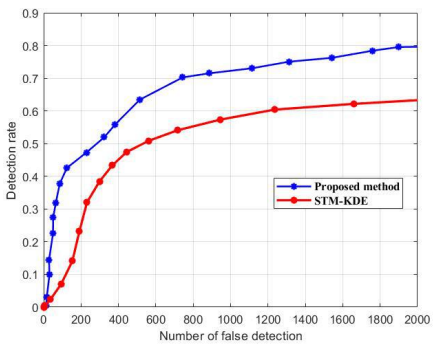
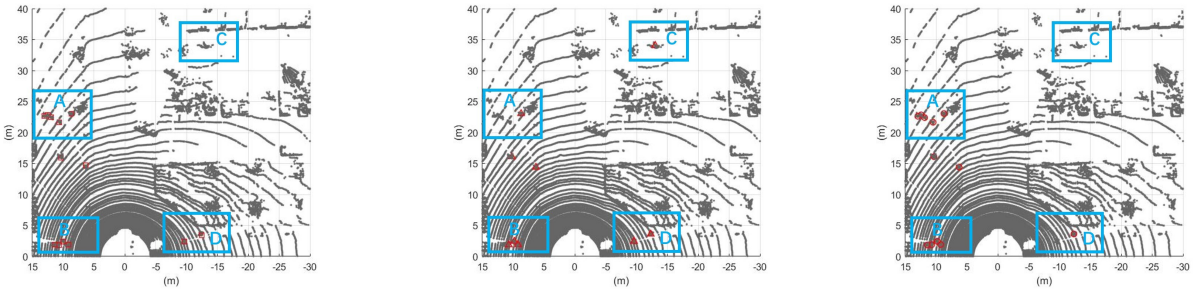


Figure 8. ROC curves of different methods for pedestrians detection.

Table 1. Pedestrian detection results on the 2011_09_28_drive_0016 dataset.

Method	Total Num	True Detection	False Detection	Precision	Recall	F1
STM-KDE	2545	979	299	0.7660	0.3847	0.5122
Proposed method	2545	**1659**	**34**	**0.9799**	**0.6519**	**0.7838**

To show the detection performance of the proposed method more clearly, the detection results of frame 50 are taken as an example, which is shown in Figure 9. The pedestrians in the scene are marked by a red box in Figure 9a, which is used as ground truth. Figure 9b is the detection result of STM-KDE, and Figure 9c is the result of the proposed method.



(**a**) (**b**) (**c**)

Figure 9. Comparison of pedestrian detection result between STM-KDE and proposed method at frame 50. (**a**) Ground truth. (**b**) Result of STM-KDE. (**c**) Result of proposed method.

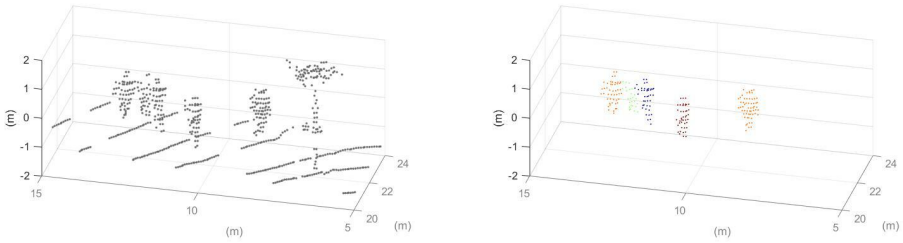
Note that the detection performance is analyzed in four different areas A, B, C, and D for a complete performance comparison. As shown in Table 2, there are five pedestrians in area A; one pedestrian is detected by STM-KDE, while all five pedestrians are successfully detected by the proposed method. There are four pedestrians in area B; three pedestrians are detected by STM-KDE, while all of them are successfully detected by the proposed method. One false detection appears in area C from STM-KDE. There are two pedestrians in Area D, both are successfully detected by STM-KDE, and one of them is detected by the proposed method.

Table 2. Pedestrian detection results at frame 50.

TP/FP Area / Method	A	B	C	D
Ground Truth	5	4	0	2
STM-KDE	1/0	3/0	0/1	2/0
Proposed method	5/0	4/0	0/0	1/0

In the areas A and B where pedestrians are dense, five targets are missed by the STM-KDE. The origin point clouds of area A is given in Figure 10a, and Figure 10b shows the detection results of proposed method.

Note that the proposed method can provide a better segmentation and detection performance compared with STM-KDE, which is based on single template matching. In area C, there is a false target from STM-KDE. As shown in Table 1, there are many more false detections than the proposed method. The reason is the STM-KDE uses a single template to match targets, which have to set a lower threshold to adapt to the non-rigid shape of pedestrians. The proposed method has a missing alarm in area D, which is caused by the difference in point cloud proportion between pedestrians carrying backpacks and ordinary pedestrians. Thus, the extracted features exceeded the threshold, which caused the detection of this pedestrian to be invalid, while the STM-KDE detects the pedestrian.



(**a**) (**b**)

Figure 10. Detection result of the proposed method in area A. (**a**) Origin point clouds. (**b**) Detected pedestrians using proposed method.

3.3. Tracking Performance Validation

Aa KITTI only provides 2D ground truth, we project the LiDAR point clouds to the camera imaging plane through a transformation matrix, and evaluate the performance based on the bounding box. The evaluation metrics include MOTA (Multiple Object Tracking Accuracy), MOTP (Multiple Object Tracking Precision), R (Recall), P (Precision), F1, MT (number of mostly tracked trajectories), ML (number of mostly lost trajectories), and IDS (number of identity switches). The definitions of MOTA and MOTP are as follows [45]:

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{MME}_t)}{\sum_t \text{GT}_t} \tag{31}$$

$$\text{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{32}$$

where FN_t, FP_t, MME_t, and GT_t are the number of false negatives, false positives, mismatches, and true targets, respectively, for time t. $d_{t,i}$ is the error between estimated position and the true position of target i for time t. c_t is the total matches for time t.

Table 3 shows the values of each of the metrics. In the table, ↑ indicates that the larger the metric, the better the tracking performance, and ↓ means quite the opposite. In the tracking experiment, we set T1 = 3, T2 = 20, and T3 = 3. Both the AB3DMOT method and the new proposed method use LiDAR as the unique sensor. It can be seen from the MOTA and F1 scores that the proposed method can obtain better overall performance than AB3DMOT. The EagerMOT is a kind of 2D+3D method, which uses camera and LiDAR together. It may result in richer information to obtain better performance due to the multi-sensor fusion. However, the proposed method can still obtain similar or even better results than EagerMOT on the terms of MOTP, Precision, ML, and IDS.

Table 3. Tracking performance comparison.

Methods	MOTA ↑	MOTP ↑	Recall ↑	Precision ↑	F1 ↑	MT ↑	ML ↓	IDS ↓
Proposed method	0.6042	0.6763	0.7510	0.8603	0.8019	0.4643	0	55
AB3DMOT	0.5671	0.6604	0.6661	0.8914	0.7624	0.2143	0.1071	37
EagerMOT	0.7449	0.6745	0.8873	0.8838	0.8856	0.8571	0	65

In order to verify the effect of our new trajectory management, this paper uses the average survival time of trajectory to evaluate the stability of the trajectory. The average survival time is defined as follows:

$$L = \frac{\sum_t Tar_t}{Num_{ID}} \tag{33}$$

where Tar_t is the number of tracked targets at frame t and Num_{ID} is the total number of trajectories.

Table 4 shows the number of detections associated successfully, the number of trajectories, and the average survival time of the proposed method to the 0016 dataset. It can be seen from the table that the average survival time of the proposed method is longer than other methods. As the occluded trajectory can turn to matched trajectory as shown in Figure 2, it effectively reduces the rebirth and death caused by occlusion. This shows that the stability of the trajectory has been improved after adopting the new management strategy.

To show the tracking performance of the new trajectory management strategy clearly, the tracking trajectories of the dataset 0016 and the Beihang data are shown in Figure 11 and Figure 12, respectively. Figures 11a and 12a show the trajectories with traditional strategy. Figures 11b and 12b show the trajectories with new strategy. These bold curves show the difference between the tracking trajectories obtained by two different tracking strategies. The nine trajectories bolded in Figure 11a actually correspond to only four pedestrians, and the five trajectories bolded in Figure 12a actually correspond to only two pedestrians. Due to occlusion, many tracks are frequently terminated and started. Through the special processing of the occluded track, the dense pedestrians' trajectories are more stable and continuous.

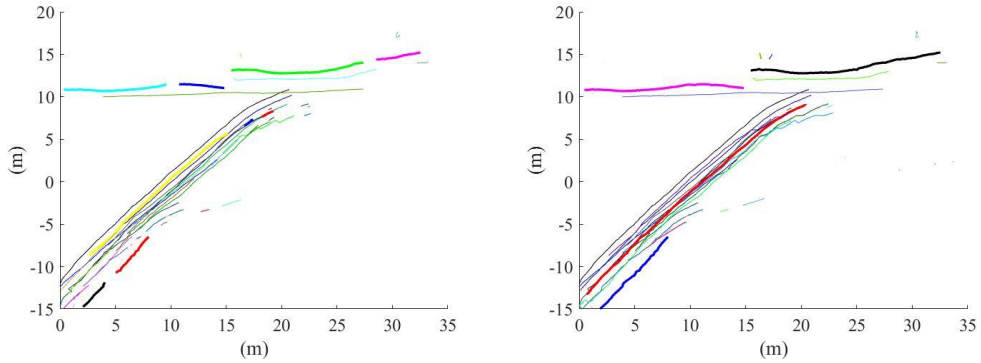


Figure 11. Trajectories with different strategies from the dataset 0016. (**a**) Using traditional strategy. (**b**) Using new strategy.

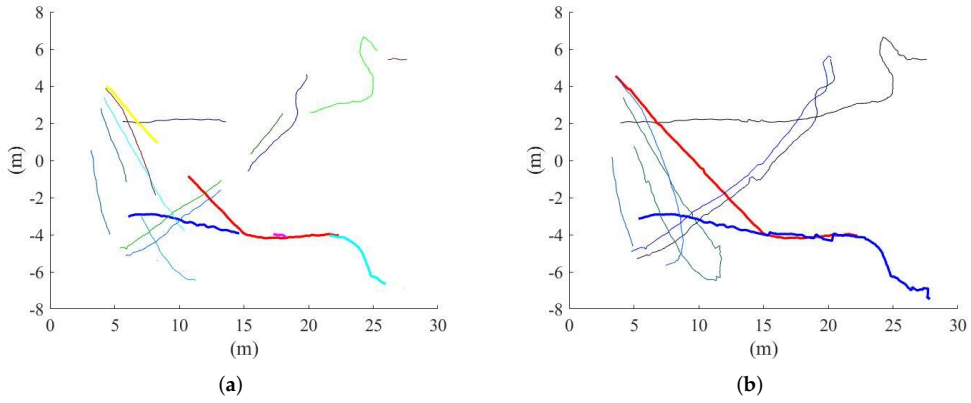


Figure 12. Trajectories with different strategies from the Beihang data. (**a**) Using traditional strategy. (**b**) Using new strategy.

Table 4. Average tracking length of different method.

Methods	Number of Detections Associationed Successfully	Num of Track	Average Tracking Length
Proposed method	1736	48	**36.17**
AB3DMOT	1464	46	31.83
EagerMOT	2056	57	36.07
Tracking using traditional strategy	1692	55	30.76

4. Conclusions

LiDAR 3D point clouds-based detection and tracking of dense pedestrians is studied in this paper. First, for the problem of undersegmentation of dense pedestrians, a fine segmentation method based on kernel density estimation was proposed. The proposed method fits the distribution of pedestrian point clouds on the horizontal projection plane, and finds the local maxima as the clustering center of pedestrians, so as to segment the point cloud more finely. Subsequently, three features that can characterize the posture and figure of the pedestrians are defined and applied to pedestrian detection. For the problem of trajectory interruption caused by occlusion in dense pedestrian tracking, a new trajectory management strategy including the occlusion state is given. In addition, the echo intensity and distance are combined to obtain the association cost to improve the stability of the trajectory.

The dense pedestrian detection and tracking method proposed in this paper can be widely used in intelligent monitoring systems and intelligent transportation systems. For example, in an intelligent monitoring system, pedestrian behavior can be analyzed after PDT, and then abnormal behavior can be detected to ensure public safety. In addition, PDT can be used to estimate the flow of people in the transportation system, so as to rationally allocate traffic resources and alleviate traffic congestion.

Author Contributions: W.W.: conceptualization, methodology, research, writing, and supervision; X.C.: conceptualization, methodology, validation, research, and writing; J.Y.: conceptualization, methodology, validation, and research; and G.X.: conceptualization and research. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundations of China under Grant 62073334 and Grant 61771028.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xie, H.; Zheng, W.; Shin, H. Occluded Pedestrian Detection Techniques by Deformable Attention-Guided Network (DAGN). *Appl. Sci.* **2021**, *11*, 6025. [CrossRef]
2. Chen, Y.; Shin, H. Pedestrian detection at night in infrared images using an attention-guided encoder-decoder convolutional neural network. *Appl. Sci.* **2020**, *10*, 809. [CrossRef]
3. Ciaparrone, G.; Sánchez, F.L.; Tabik, S.; Troiano, L.; Tagliaferri, R.; Herrera, F. Deep learning in video multi-object tracking: A survey. *Neurocomputing* **2020**, *381*, 61–88. [CrossRef]
4. Sato, S.; Hashimoto, M.; Takita, M.; Takagi, K.; Ogawa, T. Multilayer lidar-based pedestrian tracking in urban environments. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, La Jolla, CA, USA, 21–24 June 2010; pp. 849–854.
5. Ogawa, T.; Sakai, H.; Suzuki, Y.; Takagi, K.; Morikawa, K. Pedestrian detection and tracking using in-vehicle lidar for automotive application. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 734–739.
6. Xiao, W.; Vallet, B.; Schindler, K.; Paparoditis, N. Simultaneous detection and tracking of pedestrian from velodyne laser scanning data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 295–302. [CrossRef]

7. Tang, H.L.; Chien, S.C.; Cheng, W.H.; Chen, Y.Y.; Hua, K.L. Multi-cue pedestrian detection from 3D point cloud data. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 1279–1284.
8. Wang, H.; Wang, B.; Liu, B.; Meng, X.; Yang, G. Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle. *Robot. Auton. Syst.* **2017**, *88*, 71–78. [CrossRef]
9. Sualeh, M.; Kim, G.W. Dynamic multi-lidar based multiple object detection and tracking. *Sensors* **2019**, *19*, 1474. [CrossRef]
10. Shi, S.; Wang, X.; Li, H. Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 770–779.
11. Weng, X.; Wang, J.; Held, D.; Kitani, K. 3d multi-object tracking: A baseline and new evaluation metrics. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 10359–10366.
12. Kim, A.; Ošep, A.; Leal-Taixé, L. EagerMOT: 3D Multi-Object Tracking via Sensor Fusion. *arXiv* **2021**, arXiv:2104.14682.
13. Islam, M.M.; Newaz, A.A.R.; Karimoddini, A. A Pedestrian Detection and Tracking Framework for Autonomous Cars: Efficient Fusion of Camera and LiDAR Data. In Proceedings of the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia, 17–20 October 2021; pp. 1287–1292.
14. Li, K.; Wang, X.; Xu, Y.; Wang, J. Density enhancement-based long-range pedestrian detection using 3-D range data. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 1368–1380. [CrossRef]
15. Liu, K.; Wang, W.; Wang, J. Pedestrian detection with LiDAR point clouds based on single template matching. *Electronics* **2019**, *8*, 780. [CrossRef]
16. Hsieh, J.; Chen, S.; Chuang, C.; Chen, Y.; Guo, Z.; Fan, K. Pedestrian segmentation using deformable triangulation and kernel density estimation. In Proceedings of the 2009 International Conference on Machine Learning and Cybernetics, Baoding, China, 12–15 July 2009; Volume 6, pp. 3270–3274.
17. Delso, J.; Martín, B.; Ortega, E. A new procedure using network analysis and kernel density estimations to evaluate the effect of urban configurations on pedestrian mobility. The case study of Vitoria -Gasteiz. *J. Transp. Geogr.* **2018**, *67*, 61–72. [CrossRef]
18. Petrasova, A.; Hipp, J.A.; Mitasova, H. Visualization of Pedestrian Density Dynamics Using Data Extracted from Public Webcams. *Int. J. Geo-Inf.* **2019**, *8*, 559. [CrossRef]
19. Maturana, D.; Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
20. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
21. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–14.
22. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
23. Ali, W.; Abdelkarim, S.; Zidan, M.; Zahran, M.; El Sallab, A. Yolo3d: End-to-end real-time 3d oriented object bounding box detection from lidar point cloud. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018; pp. 716–728.
24. Simon, M.; Amende, K.; Kraus, A.; Honer, J.; Samann, T.; Kaulbersch, H.; Milz, S.; Michael Gross, H. Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–20 June 2019; pp. 1190–1199.
25. Kim, J.; Cho, J. Exploring a multimodal mixture-of-YOLOs framework for advanced real-time object detection. *Appl. Sci.* **2020**, *10*, 612. [CrossRef]
26. Yin, T.; Zhou, X.; Krahenbuhl, P. Center-Based 3D Object Detection and Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 11784–11793.
27. Wenzl, K.; Ruser, H.; Kargel, C. Performance evaluation of a decentralized multitarget-tracking algorithm using a LIDAR sensor network with stationary beams. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 1174–1182. [CrossRef]
28. Shu, G.; Dehghan, A.; Oreifej, O.; Hand, E.; Shah, M. Part-based multiple-person tracking with partial occlusion handling. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1815–1821.
29. Shin, S.G.; Ahn, D.R.; Lee, H.K. Occlusion handling and track management method of high-level sensor fusion for robust pedestrian tracking. In Proceedings of the 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, Korea, 16–18 November 2017; pp. 233–238.
30. Lim, Y.C.; Lee, C.H.; Kwon, S.; Kim, J. Event-driven track management method for robust multi-vehicle tracking. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 189–194.
31. Benedek, C. 3D people surveillance on range data sequences of a rotating Lidar. *Pattern Recognit. Lett.* **2014**, *50*, 149–158. [CrossRef]

32. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
33. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*; AAAI Press: Palo Alto, CA, USA, 1996; Volume 96, pp. 226–231.
34. Klasing, K.; Wollherr, D.; Buss, M. A clustering method for efficient segmentation of 3D laser data. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 4043–4048.
35. Gordon, C.C.; Blackwell, C.L.; Bradtmiller, B.; Parham, J.L.; Hotzman, J.; Paquette, S.P.; Corner, B.D.; Hodge, B.M. *2010 Anthropometric Survey of US Marine Corps Personnel: Methods and Summary Statistics*; Technical Report; Army Natick Soldier Research Development and Engineering Center: Natick, MA, USA, 2013.
36. Huijing Zhao.; Shibasaki, R. A novel system for tracking pedestrians using multiple single-row laser-range scanners. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2005**, *35*, 283–291. [CrossRef]
37. Liu, B.; Tharmarasa, R.; Jassemi, R.; Brown, D.; Kirubarajan, T. Extended Target Tracking With Multipath Detections, Terrain-Constrained Motion Model and Clutter. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7056–7072. [CrossRef]
38. Feng, P.; Wang, W.; Dlay, S.; Naqvi, S.M.; Chambers, J. Social force model-based MCMC-OCSVM particle PHD filter for multiple human tracking. *IEEE Trans. Multimed.* **2016**, *19*, 725–739. [CrossRef]
39. Kalman, R.E. A New Approach To Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
40. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
41. Brekke, E.; Hallingstad, O.; Glattetre, J. Improved target tracking in the presence of wakes. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 1005–1017. [CrossRef]
42. Liu, B.; Tang, X.; Tharmarasa, R.; Kirubarajan, T.; Jassemi, R.; Hallé, S. Underwater Target Tracking in Uncertain Multipath Ocean Environments. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4899–4915. [CrossRef]
43. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
44. Chen, T.; Wang, R.; Dai, B.; Liu, D.; Song, J. Likelihood-field-model-based dynamic vehicle detection and tracking for self-driving. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3142–3158. [CrossRef]
45. Bernardin, K.; Stiefelhagen, R. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP J. Image Video Process.* **2008**, *2008*, 246309. [CrossRef]