LiDAR-Based Pedestrian Detection Postprint

Authors: Ren Kefei, Zhang Li

Date: 2019-01-28T00:00:00+00:00

Abstract

Among the numerous tasks in the autonomous driving domain, pedestrian detection is an indispensable technology. To address the limitation that image-based pedestrian detection algorithms cannot obtain depth information, we propose a LiDAR-based pedestrian detection algorithm. This algorithm integrates traditional LiDAR-based moving object recognition algorithms with deep learning-based point cloud recognition algorithms, enabling pedestrian perception and detection without reliance on image data, thereby acquiring accurate three-dimensional positions of pedestrians to assist the autonomous driving control system in making reasonable decisions. The algorithm was evaluated on the KITTI 3D object detection benchmark dataset, achieving an average precision of 33.37% on the moderate difficulty test, outperforming other LiDAR-based algorithms and fully demonstrating the effectiveness of the proposed method.

Full Text

Preamble

Vol. 37 No. 4
Application Research of Computers
ChinaXiv Partner Journal
Accepted Paper

Pedestrian Detection Based on LiDAR Data

Ren Kefei, Zhang Li
(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

Abstract: Pedestrian detection is an essential task among the many challenges in autonomous driving. Traditional pedestrian detection algorithms relying on image data cannot obtain depth information. To address this limitation, this paper proposes a pedestrian detection algorithm based on LiDAR data. The algorithm combines conventional moving object detection methods for LiDAR data

with deep learning-based point cloud recognition techniques, enabling pedestrian perception and detection without image data while accurately obtaining 3D pedestrian locations to assist autonomous driving control systems in making reasonable decisions. The algorithm was evaluated on the KITTI 3D object detection benchmark, achieving 33.37% average precision on moderate difficulty tests, outperforming other LiDAR-based methods and demonstrating its effectiveness.

Keywords: pedestrian detection; LiDAR; point cloud; deep learning

Introduction

Pedestrian detection aims to locate pedestrians in complex backgrounds and mark their positions with bounding boxes or 3D cuboids, placing it within the broader domain of object detection. As a long-standing and challenging problem in computer vision, pedestrian detection finds widespread application across various tasks and is particularly critical in autonomous driving. Autonomous driving systems must perceive their surroundings through sensors, identify important objects such as pedestrians and vehicles, and obtain crucial information including position, depth, velocity, and trajectory to assist central control systems in making reasonable decisions. Depth information, representing the distance from objects to the sensor, forms the basis for acquiring 3D position and velocity data. Research on pedestrian detection in autonomous driving contexts helps systems make more informed decisions and avoid traffic accidents and casualties, underscoring its paramount importance.

Traditional pedestrian detection algorithms primarily utilize image data. However, since applications typically involve outdoor scenes, lighting conditions and weather significantly impact image quality and consequently detection accuracy. Dalal et al. [1] proposed a pedestrian detection algorithm based on HOG (Histogram of Oriented Gradients) features and Support Vector Machines (SVM), using edge and gradient intensity information from HOG features to generate pedestrian proposals combined with SVM classification. However, SVM computational complexity scales with the number of support vectors, making it inefficient. Jones et al. [2] addressed this by employing AdaBoost instead of SVM, improving computational speed. The RCNN series algorithms [3,4] achieved remarkable success in object detection through end-to-end multi-target detection models using convolutional neural networks. However, directly applying Faster-RCNN to pedestrian detection yields unsatisfactory results, primarily because pedestrians constitute small targets in most scenes, and multiple convolutional operations can cause target loss and numerous missed detections. Zhang et al. [5] improved Faster-RCNN by incorporating semantic and depth information, enhancing model specificity and reducing missed detections. While image-based pedestrian detection algorithms can effectively utilize image features to locate pedestrians, they cannot obtain depth information. Although some studies have

attempted to estimate depth through deep neural networks [6,7], the accuracy remains limited.

With the widespread adoption of LiDAR, precise depth information can be acquired in outdoor scenes. LiDAR samples the environment through laser pulses to obtain distance measurements, exhibiting minimal sensitivity to weather and lighting conditions. LiDAR data takes the form of point clouds, characterized by disorder, sparsity, and limited features. Directly applying successful image-based convolutional neural networks to point cloud processing does not yield satisfactory results. Asvadi et al. [8] proposed a moving object detection algorithm that converts point clouds into voxels with binary values indicating point presence, then identifies moving objects through temporal voxel accumulation. Maturana et al. [9] introduced VoxNet, which converts point clouds to voxel representations and extends CNNs to 3D convolutional neural networks. However, due to point cloud sparsity, voxel conversion produces numerous zero values that contribute nothing to recognition while consuming substantial computational resources. Qi et al.' s PointNet [10] and PointNet++ [11] represent breakthrough innovations in point cloud classification. PointNet directly processes unordered point clouds, extracting global features through multi-layer perceptrons for classification, while PointNet++ improves upon this by adding hierarchical point cloud regional feature extraction subnetworks (set abstraction networks) to enhance accuracy. However, these methods only work well for small-scale scenes and perform poorly on large-scale point clouds in autonomous driving contexts. Qi et al. [12] proposed using 2D image detectors to identify targets, restricting PointNet' s application to frustums obtained by back-projecting 2D bounding boxes, then cascading PointNet networks for binary point cloud segmentation (target vs. non-target) before regressing object positions. This approach remains dependent on image detectors, limiting final detection results by image detection performance without addressing accuracy degradation due to lighting and adverse weather. BirdNet [13] and TopNet [14] represent image-free vehicle detection methods that estimate the ground plane, map point clouds onto it, convert handcrafted features to bird' s-eye-view representations, and apply Faster-RCNN for detection and recognition.

This paper proposes a LiDAR-based pedestrian detection algorithm that processes point cloud data and detects targets without image dependency, obtaining target categories and position information. The algorithm features three innovations: (a) a pedestrian detection method combining traditional LiDAR target detection with point cloud classification; (b) a density-adaptive ground estimation method that adjusts the algorithm's application scope based on point cloud density; and (c) a point cloud density adjustment algorithm that preprocesses data to better suit traditional clustering methods.

1 Ground Removal

We assume that detection targets (pedestrians, vehicles, trees, buildings, etc.) all lie above ground level, with target point clouds connected through the

ground plane. After removing road points, target point clouds become separated [15,16], enabling pedestrian candidate clusters through clustering algorithms. Traditional ground removal methods [17,18] model ground as a fixed plane or quadratic surface. However, LiDAR motion affects data acquisition, making road surfaces in point clouds deviate from fixed surface models. Reference [8] divides point cloud data into multiple segments based on distance for separate processing, constructing different planar models for ground removal. However, this method uses fixed distance intervals, which becomes meaningless when the sensor' s effective range is much smaller than set intervals (e.g., due to building occlusion), as some intervals may contain only noise or no road points. Therefore, this paper proposes a density-adaptive ground removal algorithm.

1.1 Density-Adaptive Point Cloud Slicing

LiDAR point clouds exhibit non-uniform density, with lower density and higher noise levels at greater distances from the sensor. Dividing point clouds by distance, the nearest segment with highest density yields the most reliable ground plane model. The density-adaptive slicing process proceeds as follows: (a) determine the nearest and farthest slicing distances; (b) set interval parameters and determine slice intervals and quantities; (c) verify whether each slice contains sufficient points (above threshold N), merging with adjacent slices if below threshold and repeating; (d) discard the final slice as noise if it contains insufficient points (below threshold Thresh).

Figure 1 compares the slicing algorithm from reference [8] with our density-adaptive approach, where different colors represent different slices and black points indicate noise. In Figure 1(a), the outermost green points at maximum distance contain substantial noise and are unsuitable for ground estimation, yet reference [8] still attempts ground estimation in subsequent steps. In Figure 1(b), the outermost point cloud is treated as noise when its element count falls below threshold. Our density-adaptive slicing algorithm utilizes point cloud data more effectively, improving efficiency over reference [8].

1.2 Threshold Selection and Ground Fitting

For each slice, we employ quartile-based thresholding. First, the median height Q2 divides points into upper and lower halves. Then, we determine the medians of each half, Q1 and Q3. Points with heights between Q1 and Q3 serve as inliers for ground fitting. This approach reduces processing time by decreasing point quantity while removing most noise to improve efficiency and accuracy. We then apply RANSAC [19] to fit ground planes using inliers from each slice. RANSAC is an iterative optimal model-fitting algorithm more robust than direct least squares for noisy data. Finally, we validate each slice' s ground plane using the verification method from reference [8].

2 Clustering

Clustering algorithms, as unsupervised learning methods, are widely applied in data mining, image processing, and pattern recognition. DBSCAN [20] is a robust density-based clustering algorithm whose primary advantage is not requiring predetermined cluster numbers. However, DBSCAN' s clustering radius and minimum element count are globally fixed, making it suitable only for uniformly dense data. When applied to non-uniform data like LiDAR point clouds, clustering performance degrades significantly. Therefore, we propose preprocessing point clouds to transform their density, achieving uniform density along the height dimension before DBSCAN clustering to improve accuracy.

2.1 Preprocessing

LiDAR scan lines typically distribute at fixed vertical angular intervals, inevitably causing higher density near the sensor and lower density farther away. Figure 2 [Figure 2: see original paper] illustrates the LiDAR scan line model, where plane z=1 represents the sensor plane (usually mounted on the vehicle), blue dashed lines are scan lines, d1 is the minimum point cloud distance, and Δ is the angular interval between adjacent scan lines. Assuming a point exists at distance d2 from the sensor (red point in Figure 2), we adjust point cloud height dimension as follows.

Using similar triangle principles yields Equation (1). After simplification, we obtain Equation (2), where h is the original height coordinate and h_new is the adjusted height coordinate. Figure 3(c) [Figure 3: see original paper] shows the density-adjusted point cloud, demonstrating that only height dimension density changes, making the data more suitable for clustering while preserving geometric features.

2.2 DBSCAN Clustering

DBSCAN requires two global parameters: eps (clustering radius) and MinPts (minimum cluster size). Points separated by distances greater than eps belong to different clusters, while clusters with fewer than MinPts points are treated as noise. Experimentally, we found optimal clustering performance with eps = 0.13 and MinPts = 5.

Figure 3 [Figure 3: see original paper] presents results from the processing pipeline: (a) raw point cloud data (within camera FOV) with pedestrians in red bounding boxes, colored by height (blue for low, yellow for high); (b) remaining points after ground removal; (c) density-adjusted point cloud; (d) DBSCAN clustering results with different colors for different clusters and black for noise. The pedestrian within the red bounding box appears green, indicating it was correctly identified as a candidate cluster, though numerous background clusters also remain, necessitating further filtering.

2.3 Filtering

Pedestrian prior knowledge enables filtering of point cloud clusters to remove most background clusters, reducing computational complexity and alleviating positive-negative sample imbalance. Based on candidate cluster length (l), width (w), and maximum height (h, the highest point' s distance from ground), we apply the following criteria (Equations (3)-(6)) to identify pedestrian candidates, treating others as noise.

3 Cluster Recognition Network

We improved the point cloud segmentation and regression networks from reference [12] for our recognition network. In reference [12], segmentation and regression networks are cascaded: the segmentation network takes point clouds within 2D detection-limited frustums as input, outputs target points, and the regression network takes these target points to output bounding box parameters. Our approach places classification and regression networks in parallel, both taking pedestrian candidate clusters as input. We employ PointNet++ as the feature extraction network, using set abstraction networks for hierarchical local feature extraction to obtain global point cloud feature vectors. Fully connected layers then output final result vectors. The classification network outputs a 2D vector representing background and pedestrian probabilities (summing to 1), classifying candidates as pedestrians when pedestrian probability exceeds 0.6. The regression network predicts bounding box position and orientation, decomposing rotation angle regression into bin classification and residual regression. Thus, the regression network outputs an N+2 dimensional vector, where N dimensions represent rotation bin probabilities and 2 dimensions represent residuals.

4 Experiments

4.1 Sample Generation

For candidate clusters passing the filtering stage, we generate training samples for the recognition network. We count points within ground truth bounding boxes and calculate their proportion of total cluster points. Candidates with proportions exceeding threshold (0.7 in our experiments) are positive samples (pedestrians), while others are negative samples.

4.2 Dataset

We evaluated our method on the KITTI dataset [21,22], a well-known autonomous driving benchmark containing stereo images, LiDAR data, and IMU measurements for evaluating recognition, tracking, visual odometry, optical flow, and segmentation tasks. KITTI' s LiDAR (Velodyne HDL-64E) features 64 scan lines, 120 m maximum range, 360° horizontal FOV with 0.08° angular resolution, and 26.9° vertical FOV with approximately 0.4° angular resolution. Mounted at

approximately 1.73 m height, the dataset contains 7,481 training samples and 7,518 test samples. Test cases are categorized as Easy (bounding box height 40 pixels, fully visible), Moderate (height 25 pixels, partially occluded), and Hard (height 25 pixels, mostly occluded).

Experiments were conducted on an Intel i7-4790K CPU (3.1 GHz), 32 GB RAM, GeForce GTX TITAN X GPU (12 GB), using TensorFlow r1.4. Evaluation metrics include Precision-Recall (P-R) curves and Average Precision (AP), where AP represents the area under the P-R curve.

4.3 Results and Analysis

Our algorithm first separates ground and targets through ground removal, then obtains target clusters via clustering, and finally feeds clusters into a trained deep neural network to acquire category and location information. The method achieves strong results on KITTI, outperforming other LiDAR-only pedestrian detection algorithms.

Table 1 compares our algorithm with BirdNet [13] and TopNet [14] on KITTI. For Easy and Moderate cases, our AP exceeds BirdNet and TopNet by over 20 points, while for Hard cases, our AP is more than 10 points higher, demonstrating superior pedestrian detection performance.

Precision and recall are calculated as:
Precision = TP / (TP + FP)
Recall = TP / (TP + FN)

where TP, FP, and FN represent true positives, false positives, and false negatives, respectively.

Figure 5 [Figure 5: see original paper] shows P-R curve comparisons with references [13,14] for Easy, Moderate, and Hard cases (left to right). Blue curves represent our algorithm, while red and yellow curves show BirdNet [13] and TopNet [14] results, respectively. Our algorithm significantly outperforms both methods across all difficulty levels.

5 Conclusion

This paper proposes a pedestrian detection algorithm combining traditional LiDAR moving object detection with point cloud classification. Using only LiDAR data, the method directly obtains detection results and depth information. The algorithm first separates ground and targets, then acquires target clusters through clustering, and finally inputs clusters into a trained deep neural network to obtain category and location information. Our method achieves excellent results on the KITTI dataset, surpassing other pedestrian detection algorithms that rely solely on LiDAR data.

References

[1] Jones S M, Viola P. Fast multi-view face detection, TR-20003-96 [R]. Mitsubishi Electric Research Lab 2003.
[2] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2014: 580-587.
[3] Girshick R. Fast R-CNN [C]// Proc of IEEE international Conference on Computer Vision. Washington DC: IEEE Computer Society, 2015:
[4] Ren Shaoqing, He Kaiming, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks [C]// Advances in Neural Information Processing Systems. 2015: 91-99.
[5] Zhang Shanshan, Benenson R, Omran M, et al. How far are we from solving pedestrian detection? [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2016: 1259-1267.
[6] Liu Fayao, Shen Chunhua, Lin Guosheng. Deep convolutional neural fields for depth estimation from a single image [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2015: 5162-5170.
[7] Garg R, Vijay Kumar B G, Carneiro G, et al. Unsupervised cnn for single view depth estimation: Geometry to the rescue [C]//Proc of European Conference on Computer Vision. Springer, Cham, 2016:
[8] Asvadi A, Premebida C, Peixoto P, et al. 3D lidar-based static and moving obstacle detection in driving environments: #n approach based on voxels and multi-region ground planes [J]. Robotics and Autonomous Systems, 2016, 83: 299-311.
[9] Maturana D, Scherer S. Voxnet: A 3D convolutional neural network for real-time object recognition [C]//Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2015: 922-928.
[10] Qi C R, Su Hao, Mo Kaichun, et al. PointNet: deep learning on point sets for 3D classification and segmentation [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2017.
[11] Qi C R, Yi Li, Su Hao, et al. PointNet+: deep hierarchical feature learning on point sets in a metric space [C]// Advances in Neural Information Processing Systems. 2017: 5099-5108.
[12] Qi C R, Liu Wei, Wu Chenxia, et al. Frustum PointNets for 3D object detection RGB-D [EB/OL]. (2018-04-13). https://arxiv.org/pdf/1711.08488.pdf.
[13] Beltran J, Guindel C, Moreno F M, et al. BirdNet: a 3D object detection framework from LiDAR information [J]. arXiv preprint arXiv: 1805. 01195, 2018.
[14] Wirges S, Fischer T, Frias J B, et al. Object detection and classification in occupancy grid maps using deep convolutional networks [J]. arXiv preprint arXiv: 1805. 08689, 2018.

[15] Broggi A, Cattani S, Patander M, et al. A full-3D voxel-based dynamic obstacle detection for urban scenario using stereo vision [C]//Proc of the 16th International IEEE Conference on Intelligent Transportation Systems. Piscataway, NJ: IEEE Press, 2013: 71-76.
[16] Sun Pengpeng, Zhao Xiangmo, Xu Zhigang, et al. Urban curb robust detection algorithm based on 3D-LIDAR [J]. Journal of Zhejiang University:Engineering Science, 2018, 52 (3): 504-514.
[17] Oliveira M, Santos V, Sappa A D, et al. Scene representations for autonomous driving: an approach based on polygonal primitives [C]//Proc of the 2nd Iberian Robotics Conference. Cham:Springer, 2016:
[18] Sappa A D, Herrero R, Dornaika F, et al. Road approximation in euclidean and v-disparity space: a comparative study [C]//Proc of International Conference on Computer Aided Systems Theory. Berlin: Springer, 2007: 1105-1112.
[19] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography [J]. Communications of the ACM, 1981, 24 (6): 381-395.
[20] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise [C]// Kdd. 1996, 96 (34): 226-231.
[21] Geiger A, Lenz P, Stiller C, et al. Vision meets robotics: the KITTI dataset [J]. International Journal of Robotics Research, 2013, 32 (11):
[22] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2012: 3354-3361.

Note: Figure translations are in progress. See original paper for figures.