

基于三维激光雷达的道路障碍物目标位姿检测算法

胡杰^{1,2,3*}, 刘汉^{1,2,3}, 徐文才^{1,2,3}, 赵亮^{1,2,3}

¹武汉理工大学现代汽车零部件技术湖北省重点实验室, 湖北 武汉 430070;

²武汉理工大学汽车零部件技术湖北省协同创新中心, 湖北 武汉 430070;

³武汉理工大学湖北省新能源与智能网联车工程技术研究中心, 湖北 武汉 430070

摘要 提出一种基于随机采样一致(RANSAC)的道路障碍物目标位姿检测算法,利用这一算法对道路障碍物目标进行检测,构建目标障碍物的盒模型,输出盒模型的位姿信息。引入扇形盒分割方法,改进地面分割模块,使用基于激光雷达参数的自适应阈值方法改进聚类分割模块。基于三种激光雷达数据集进行实验验证,本文算法能够准确地对道路障碍物进行位姿信息检测,且具有较高的实时性。相较于最小包围矩形法及三点估计法,本文算法的准确度分别提升 8.14%、6.57%,检测时间分别缩短 23.81%、24.71%。

关键词 遥感; 自动驾驶; 激光雷达; 点云目标检测; 位姿检测

中图分类号 TN958.98; TN249

文献标志码 A

doi: 10.3788/CJL202148.2410001

1 引言

在自动驾驶汽车领域中,智能化驾驶决策高度依赖于准确的道路障碍物检测,车辆能够准确知道道路障碍物信息是实现自动驾驶的前提。三维激光雷达具有分辨率高、探测精度高、全天候工作及抗干扰能力强等特点,能够准确地为车辆提供环境信息。通过三维激光雷达产生的三维点云对道路障碍物进行检测时,需要对三维激光雷达产生的点云数据进行处理,将点云分割成独立子集,每个子集对应含有不同物理意义的目标。三维激光雷达产生的数据为大量的无序点集合,数据量大,算法处理的运行时间较长,这对算法的实时性提出了很高的要求。

目前,国内外针对三维点云的目标检测方法主要分为两类:一类是数据聚类方法,如 K-means 聚类^[1]、DBSCAN 聚类^[2]、层次聚类^[3]、欧氏聚类^[4]等,即利用投射到物体上的点云较为集中的特点,将目标物体上的点云聚类分割为点云簇,通过点云簇对目标物体进行检测;二类是深度学习方法,例 PointNet^[5]、VoxelNet^[6]、SECOND^[7]、PointPillars^[8]、PointRCNN^[9]等,即通过学习目标

物体的点云特征进行检测。深度学习方法具有学习能力强、准确率高、覆盖范围广等优点,但深度学习方法需要预先准备训练数据集,其神经网络模型泛化性差、所需训练时间长、硬件需求高、推理过程计算量大、模型鲁棒性差^[10]。聚类算法具有计算量小、硬件条件需求较低等特点,但聚类算法依赖于参数设置,参数设置对聚类算法精度的影响较大。对于自动驾驶的应用场景,聚类算法因其对硬件的需求低,更能满足自动驾驶汽车的需求。聚类算法中, K-means 聚类需要预先确定目标物的数量来确定 k 值,但在实际环境中目标物的数量是不断变化的,算法适应性较差。而 DBSCAN 聚类以及层次聚类方法的时间复杂度较高,此两种算法会使检测算法的运行时间大大增加,难以满足自动驾驶汽车的实时性要求。欧氏聚类算法对数据的适应性良好,能胜任大多情况下的点云分割任务,但需要预先设定距离阈值,固定距离阈值对于激光点云的近密远疏特性适应程度低。因此,研究欧氏聚类的距离阈值动态设定的方法具有很大的价值。

基于三维点云的目标检测一般分为预处理、地面分割、点云聚类、目标物输出等模块,本研究在此

收稿日期: 2021-04-15; 修回日期: 2021-05-06; 录用日期: 2021-05-17

基金项目: 湖北省技术创新专项(2019AEA169)、湖北省科技重大专项(2020AAA001)

通信作者: *auto_hj@163.com

基础上进一步对目标物的位姿信息进行检测。针对地面点云分割需要高准确性与高实时性的要求,提出一种基于扇形盒的地面点云分割方法,并对地面点云与目标物点云进行分割。针对欧氏聚类的固定距离阈值对于激光雷达近密远疏的特性适应性差的问题,提出一种自适应聚类阈值的欧氏聚类方法,改善了聚类算法在不同距离下的聚类精度。针对目标物位姿信息检测的需求,提出一种基于随机抽样一致(RANSAC)算法的边界直线拟合方法,确定了目标物体的盒模型,通过方向拟合得到目标物体的

方向角,通过坐标系旋转得到目标物体的尺寸,减少了运算量,增强了算法实时性。经实验验证可知,该方法能够准确地对道路障碍物进行位姿检测,相较于最小矩形包围法及三点估计法,本文方法具有更高的准确度,且具有良好的实时性。

2 基本原理

目标位姿检测算法分为点云预处理、地面点云分割、自适应欧氏聚类、目标位姿检测四个模块,目标位姿检测算法的流程图如图 1 所示。

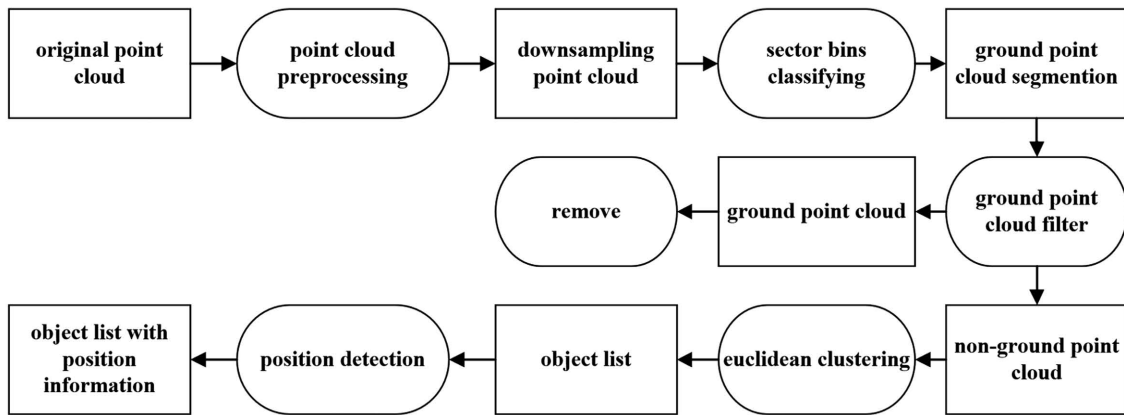


图 1 目标位姿检测算法流程图

Fig. 1 Flow chart of objection position detection algorithm

2.1 激光点云预处理

由于激光雷达产生的点云数据量较大,故在进行聚类之前需要对点云数据进行预处理。激光雷达的探测过程存在过高的建筑物遮挡以及树木遮挡现象,此部分点云会对道路障碍物检测形成干扰,在进行障碍物聚类检测之前需要对这一部分点云进行滤除。点云预处理过程如下:

1) 对点云进行体素化滤波,得到降采样后的点云,降低了点云数据量。为了去除自身车辆及支架上的点云,对与激光雷达距离小于 1.0 m 的点进行滤除。

2) 针对过高的建筑物以及树木遮挡情况,将点云中点的高度 z 超过一定阈值 H_{th} 的点滤除。此处 H_{th} 取 4.0 m, Z 方向零值平面为地面平面,即将高于地面 4.0 m 的点云进行滤除。

2.2 地面点云分割

激光雷达的非目标点云数据中大部分点云属于地面点云。因此,将地面点云分割并滤除是减少数据量、提高目标检测算法精度的有效手段。目前针对地面点云分割算法的研究主要可以分为基于坡度^[11]、基于形态学^[12]、基于分割^[13-16]、基于机器学习^[17]等地面点云分割方法。基于坡度的地面点云分割方法依

赖阈值设定,并需要搜索临近点,故计算量较大,易导致算法运行速度下降;基于形态学的方法通过设定一定的高度阈值,对地面点云进行滤除,虽然算法简单,但无法处理坡道以及起伏路面;基于分割的方法采用随机采样一致算法对地面进行分块拟合,但此种方法运算量大、实时性不高;基于机器学习的方法准确度高,算法实时性较差,模型泛化性较差。

结合分块处理以及坡度阈值检测等理论^[18],提出一种扇形盒地面点云分割算法。如图 2 所示,将初步过滤后的点云索引分布到圆柱形极坐标网格

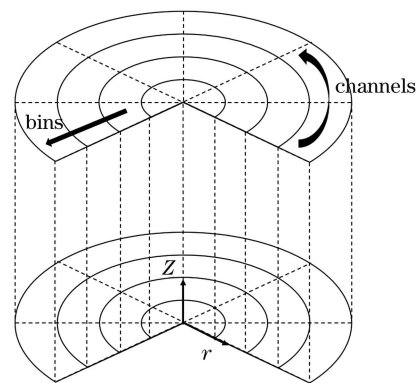


图 2 扇形盒分布及极坐标系

Fig. 2 Sector bins distribution and polar coordinate system

中,极坐标网格由分割出的通道(channels)组成,每一条通道用从原点到尽可能远的点的分段垂直切片来表征,最后每一条通道被进一步分割成盒(bins),形成扇形盒(sector bin)。

从中心扇形盒开始,沿索引升序方向遍历所有扇形盒,在每一个有点云存在的扇形盒中选取 Z 方

向坐标最低的 n 点为种子点,选取种子点的平均值为当前扇形盒的地面高度,并进行邻近及全局坡度的阈值检测。若当前地面高度处于坡度阈值之内,则此高度被视为有效地面高度。根据高度阈值分割当前扇形盒中的地面点。坡度阈值检测范围可表示为

$$\begin{cases} \bar{Z}_{ij} \geq \max [\bar{Z}_{i(j-1)} - (r_{ij} - r_{i(j-1)}) \cdot \cos \theta_n, -r_j \cdot \cos \theta_a] \\ \bar{Z}_{ij} \leq \min [\bar{Z}_{i(j-1)} + (r_{ij} - r_{i(j-1)}) \cdot \cos \theta_n, r_j \cdot \cos \theta_a] \end{cases}, \quad (1)$$

式中: \bar{Z}_{ij} 为扇形盒 Q_{ij} 中所选取种子点 Z 方向的坐标平均值; r_{ij} 为扇形盒 Q_{ij} 中心与坐标原点的距离(Z 方向距离不计入); θ_n 为邻格坡度阈值; θ_a 为全局坡度阈值。 Q_{ij} 为 channels 索引为 i 、bin 索引为 j 的扇形盒。channels 索引排序规则为:俯视图视角的正前方向为起点,沿顺时针方向进行排序;bins 排序规则为:以坐标原点为起点,沿 channel 方向向外,只记录内部存在点云的扇形盒。算法的伪代码见图 3,图中: $Q.push(p)$ 表示对 Q 使用 $push(\cdot)$ 函数, $push(\cdot)$ 函数作用为将括号中对象添加至函数使用对象,即将 p 添加进 Q 中; $Q.sort(p, z)$ 表示对 Q 中的 p 依其 z 值大小进行升序排序; $Q.size(\cdot)$ 表示返回数据结构 Q 的大小,即其中储存 p 的个数。

Algorithm 1 Segmentation algorithm of ground point cloud

Input: P_k vector of point, point P_i include 5 dimensions x, y, z, r, θ

Output: ground point vector P_{ground} and non-ground point vector

P_{non_ground}

1. Procedure Sector Bins Distribute (P_k)

2. for $l=1$ to k , P_k do

3. $Q_{ij}.push(p_i)$ (Q_{ij} :bin vector with index i, j)

4. for $i=1$ to $M, j=1$ to N, Q_{MN} do

5. $Q_{ij}.sort(p, z)$

6. $\bar{Z}_{ij} = \frac{\sum_{i=1}^n p_i z}{n}$

7. **bool** $G_{ij} = \left\{ \bar{Z}_{ij} \leq \min [\bar{Z}_{i(j-1)} + (r_{ij} - r_{i(j-1)}) \cdot \cos \theta_n, r_j \cdot \cos \theta_a] \right.$ and $\left. \bar{Z}_{ij} \geq \max [\bar{Z}_{i(j-1)} - (r_{ij} - r_{i(j-1)}) \cdot \cos \theta_n, -r_j \cdot \cos \theta_a] \right\}$

8. **if** (G_{ij} is True) **then**

9. for $l=1$ to $Q_{ij}.size(\cdot), p_l$ do

10. **if** ($p_l z < \bar{Z}_{ij} + T_h$) **then**

11. $P_{ground}.push(p_l)$

12. **else**

13. $P_{non_ground}.push(p_l)$

图 3 地面分割算法伪代码框图

Fig. 3 Pseudo-code block diagram of ground segmentation algorithm

三种激光雷达数据集实验中,取扇形通道角度值 θ_c 为 5° (共 72 通道),扇形盒长度为 0.5 m,邻格坡度阈值 θ_n 为 12° ,全局坡度阈值 θ_a 为 20° 。

2.3 改进的自适应欧氏聚类算法

聚类算法是通过将点云中满足一定条件(如密度、距离等)的一堆点分为一个类别来聚合出相互独立类别的分类算法^[19-22]。欧氏聚类算法在确定了距离指标之后,以固定的阈值使用 KD-Tree 方法在整个点云数据里进行搜寻查找,故聚类阈值的选取是重中之重。

为了充分适应点云的近密远疏特性,减少距离原因导致目标点云离散程度不同的影响^[23],使用一种根据深度信息(距离值)来调整聚类阈值的方法,根据激光雷达参数对不同距离的点进行阈值调整。

如图 4 所示, P_1 为聚类算法前点, P_1 到激光雷达的直线距离为 r_n ,图中 δ_h 为激光雷达的水平分辨率, δ_v 为激光雷达的垂直分辨率。 P_2 为 P_1 的一个水平邻点, P_3 为 P_1 的一个垂直邻点,则 P_1 到 P_2 的距离可以近似为 $r_n \delta_h \pi / 180$, P_1 到 P_3 的距离可以近似为 $r_n \delta_v \pi / 180$, P_2 到 P_3 的距离可近似为 $r_n \pi \sqrt{(\delta_v)^2 + (\delta_h)^2} / 180$ 。因此将自适应聚类距离阈值确定为

$$d = \frac{(1 + \lambda) \pi r_n}{180} \sqrt{\sin^2 \delta_v + \sin^2 \delta_h} + 3\sigma, \quad (2)$$

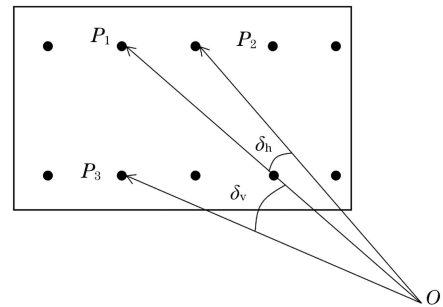


图 4 激光雷达投射点云示意图

Fig. 4 Schematic diagram of point cloud projected by LiDAR

式中 σ 为激光雷达测量误差, λ 为聚类阈值补偿参数。基于三种激光雷达数据集进行实验, 结果发现 λ 取 0.75 时聚类算法在三种激光雷达的数据集上表现最佳。由 (2) 式可以看出, 聚类的阈值随点到激光雷达的距离成正比, 能够良好地适应激光点云距离越远离散程度越高的趋势, 利于改善各种距离下目标的聚类效果。

2.4 目标位姿检测算法

对于目标物体的位姿检测, 通常建立包围点云簇的矩形盒模型, 然后根据矩形盒模型的方向角、尺寸、中心点位置等参数确定目标物体的位姿信息。目前, 矩形盒模型的建立方法主要有三种: 一是边界提取, 通过最小包围矩形 (MBR) 法计算点云簇的最小包围矩形, 建立矩形盒模型^[3-4, 24]; 二是通过主成分分析 (PCA) 法对点云簇进行分析, 计算点云簇协方差矩阵、特征值与特征向量等, 寻找主成分方向, 根据主成分方向计算点云簇尺寸, 建立矩形盒模型^[25]; 三是三点估计法 (Three Point Estimation), 寻找点云簇中距

离最远的两点及距离此两点所形成直线的最远点, 由这三点计算盒模型顶点, 估计矩形盒模型^[17]。MBR 方法受到点云边界的噪声干扰较大; PCA 方法的抗噪声干扰能力较差, 点云噪声会使主成分方向出现偏差; 由于物体尺寸由点云投影到该方向的最远点确定, 最远点的点云噪声对尺寸影响较大, 故 TPE 算法对于聚类分割准确度及点云簇形状较为依赖。

车辆行驶过程中的障碍物主要有车辆、行人、非机动车等, 目标物体的点云特征在于侧面近似垂直于地面, 因此目标物体的方向、尺寸等可以由目标物点云在地面上的投影来进行识别。基于此特点, 提出一种基于随机采样一致 (RANSAC) 的目标位姿检测 (RDME) 算法。RANSAC 算法能够有效地减少噪声点干扰, 准确拟合出目标物体的方向。

首先将目标物点云向地面平面投影, 即在车辆坐标系中将目标物点云向 XOY 平面投影, 如图 5 所示, 可以看出物体边界处的点云密集程度明显高于其他位置。

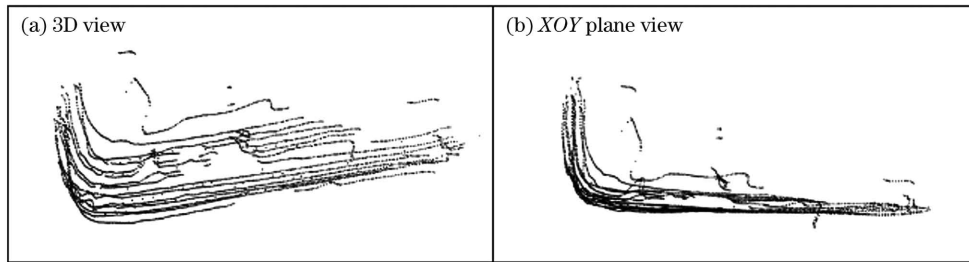


图 5 不同视角点云对比。(a)3D 视角;(b)XOY 平面视角

Fig. 5 Comparison of point clouds from different perspectives. (a) 3D view; (b) XOY plane view

根据目标物点云的这一特性, 利用 RANSAC 算法对点云的长边进行拟合。RANSAC 算法的优点是可以从噪声点中拟合出所需的模型。在此对目标点云的长边进行拟合, 并以拟合出来的直线方向作为目标物的方向角方向。由于目标物点云长边的直线模型特征较为明显, 无需较大的迭代次数, 为了保证 RANSAC 算法的时间效率, 设置最大迭代次数 $k = 120$ 。经本地数据集实验验证, 该值能够在不对算法精度造成较大影响的前提下有效缩短算法的运行时间。

如图 6 所示, 经算法拟合后可以得到 l 的直线公式 $Ax + By + C = 0$, 根据直线公式计算目标物体的方位角 θ_h , 选取的车辆坐标系中 X 方向向前, Y 方向向右, 方位角 θ_h 为直线 l 与 X 方向的夹角, 顺时针方向为正。方向角计算公式为

$$\theta_h = \arctan(-A/B), \quad (3)$$

式中, A 为标准直线公式 x 的系数, B 为标准直线公式 y 的系数。

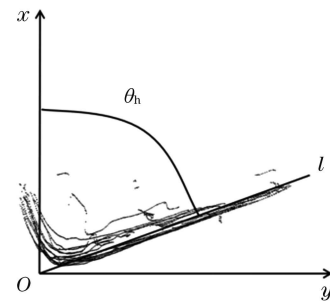


图 6 目标方向角检测

Fig. 6 Object direction angle detection

由 RANSAC 算法得到目标物方向角, 根据目标物方向角将坐标系进行旋转, 如图 7 所示, 使方向角直线平行于 X 轴, Z 方向不变。点云坐标变换公式为

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} \cos \theta_h & \sin \theta_h \\ -\sin \theta_h & \cos \theta_h \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \quad (4)$$

再选取旋转后点云中 X' 、 Y' 方向坐标的最大最小值 x'_{\max} 、 x'_{\min} 、 y'_{\max} 、 y'_{\min} , 计算目标物盒模型的长度 $L =$

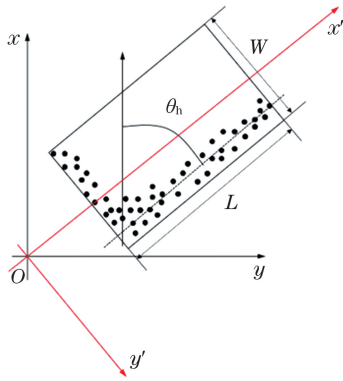


图 7 坐标系旋转过程

Fig. 7 Coordinate system rotation process

$x'_{\max} - x'_{\min}$ 与宽度 $W = y'_{\max} - y'_{\min}$; 选取目标物点云簇在 Z 方向的最大最小值 z_{\max}, z_{\min} , 计算目标物盒模型高度 $H = z_{\max} - z_{\min}$ 。通过此方式能够得到目标物体的尺寸信息。目标物中心点坐标的计算公式为

$$\begin{cases} x'_{\text{mid}} = (x'_{\max} + x'_{\min}) / 2 \\ y'_{\text{mid}} = (y'_{\max} + y'_{\min}) / 2 \\ z_{\text{mid}} = (z_{\max} + z_{\min}) / 2 \end{cases} \quad (5)$$

得到旋转后坐标系下中心点坐标后将中心点坐标通过转换矩阵逆变换得到原坐标系下中心点坐标。算法具体实现伪代码如图 8 所示。

3 实验分析与讨论

为了验证上述检测算法的工作性能,在自动驾驶实验车辆上采用不同型号的激光雷达采集数据,



图 9 车辆传感器安装图

Fig. 9 Installation drawing of vehicle sensor

3.1 实验设备

实验采用 3 种不同型号不同线束的激光雷达 (雷达型号为 VelodyneVLP16, 镭神 C32151L, ZVISION ML-30S), 探究目标检测算法在不同点云

Algorithm2 Position detection algorithm

Input: vector of object O_k ; point vector P_m in O_k ; P_k include five dimensions x, y, z, r, θ

Output: label $\{(P_{\text{mid}})_k, L_k, W_k, H_k, (\theta_h)_k\}$ to O_k .

1. Procedure Position Detection (O_k)
2. for $l=1$ to k, O_k do
3. for $i=1$ to $m, j=i+1$ to m, P_m do
4. $l[i][j]: A_j x + B_j y + C_j = 0$
5. for $n=1$ to m, P_m do
6. $d_n = \frac{|A_j x_m + B_j y_m + C_j|}{\sqrt{A_j^2 + B_j^2}}$
7. if $(d_n < t)$ then
8. $O_j.\text{num}[i][j]++$
9. $O_i.\text{num}[I][J] = \max(O_i.\text{num}[i][j])$
10. $(\theta_h)_i = \arctan(-A_{ij} / B_{ij})$
11. $L_i, W_i, H_i, (P_{\text{mid}})_i = \text{Measurement Detection}(O_i, (\theta_h)_i)$
12. label $\{(P_{\text{mid}})_i, L_i, W_i, H_i, (\theta_h)_i\}$ to O_i .
13. Function Measurement Detection ($O_i, (\theta_h)_i$)
14. for $i=1$ to m, P_m do
15. $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta_h & \sin \theta_h \\ -\sin \theta_h & \cos \theta_h \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
16. $[x'_{\max} \ y'_{\max} \ z'_{\max}]^T = [\max(x') \ \max(y') \ \max(z)]^T$
17. $[x'_{\min} \ y'_{\min} \ z'_{\min}]^T = [\min(x') \ \min(y') \ \min(z)]^T$
18. $[L_i \ W_i \ H_i]^T = [x'_{\max} - x'_{\min} \ y'_{\max} - y'_{\min} \ z'_{\max} - z'_{\min}]^T$
19. $(P_{\text{mid}})_i = [(x_{\text{mid}})_i, (y_{\text{mid}})_i, (z_{\text{mid}})_i]^T$
20. return $L_i, W_i, H_i, (P_{\text{mid}})_i$

图 8 位姿检测算法伪代码框图

Fig. 8 Pseudo code block diagram of direction and size detection algorithm

并从数据中截取部分点云帧进行算法验证。自动驾驶实验车辆顶部的激光雷达采用支架固定、水平安装,安装图如图 9 所示。

特征下的表现。计算所用的计算机 CPU 型号为 Inter i7-9700k, GPU 型号为 NVIDIA GeForce RTX 2080Ti。各激光雷达参数如表 1 所示。

表 1 激光雷达性能参数表

Table 1 Performance parameters of LiDAR

Performance parameters	Velodyne VLP16	LEISHEN C32151L	ZVISION ML-30S
Number of lines	16	32	140
Measuring range / m	100	150	30
Measuring accuracy / cm	±3	±2	±3
Vertical angle range / (°)	30	30	70
Vertical angular resolution / (°)	2	1	1
Horizontal angle range / (°)	360	360	150
Horizontal angular resolution / (°)	0.2	0.5	0.3
Scanning frequency / Hz	10	10	10

3.2 实验方法与结果分析

实验采用真实场景的实验数据集,即将激光雷达安装在车辆的水平支架上,使运行车辆在道路上进行 S 线行驶,采用机器人操作系统(ROS)平台进行数据传输及录制,同时录制激光雷达以及摄像头的信息,使其生成包含激光雷达信息以及摄像头信息的数据包。车辆行驶速度约 20 km/h。实验数据采集的主要目标为车辆、行人、非机动车等。采用 S 线行驶可以使路边的车辆、行人等道路障碍物相对于车辆的方向角不断变化,因其点云特征不同,故可针对不同状态下的目标障碍物检测算法的准确率。

从三种激光雷达的数据包中各随机抽取 100 帧

点云进行算法验证。统计所有单帧点云中出现的目标物数量以及聚类算法与位姿检测算法正确检测到的目标物数量,记录每一帧的算法处理时间。Velodyne 激光雷达原始点云单帧点数约为 20600,算法处理后目标物平均点数约为 260,所有抽取点云帧目标物总计 421 个;镭神激光雷达原始点云单帧点数约为 50600,算法处理后目标物平均点数约为 510,所有抽取点云帧目标物总计 441 个;ZVISION 激光雷达原始点云单帧点数约为 51200,算法处理后目标物平均点数约为 580,所有抽取点云帧目标物总计 209 个。算法检测效果以及同步图像如图 10 所示。

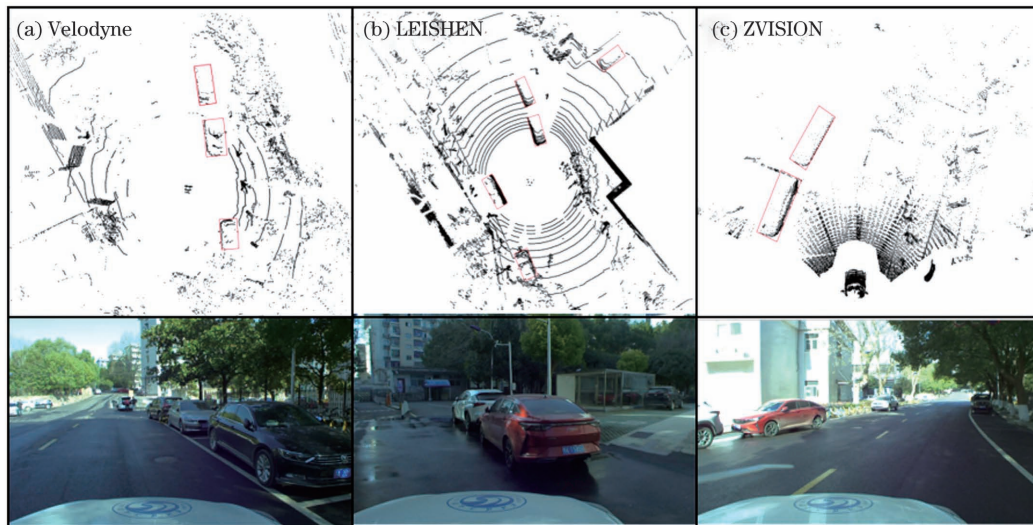


图 10 算法检测效果。(a)Velodyne VLP16;(b)镭神 C32151L;(c)ZVISION ML-30S

Fig. 10 Detection performance of the algorithm. (a) Velodyne VLP16; (b) LEISHEN C32151L; (c) ZVISION ML-30S

检测算法运行过程中存在以下情况:1)目标物体实际存在,可被聚类算法检测到,且可以正确估计出其方向尺寸;2)目标物体实际存在,可被聚类算法检测到,但其方向尺寸未能检测获得;3)目标物体实际存在,但未被聚类算法检测到;4)目标物体实际不存在,但被聚类算法误检。针对以上几种情况,采用

以下指标评价算法效果:①正检率(TPA),即目标物体被聚类算法正确检测的比率 A_{TP} ;②错检率(FNA),即聚类算法误检的比率 A_{FN} ;③完全正检率(TTPA),即目标物体被聚类算法及位姿检测算法同时正确检测的比率 A_{TTP} ;④位姿正检率(PPA),即位姿检测算法正确检测目标数量占聚类

算法正确检测目标数量的比率 A_{PP} 。

$$A_{TP} = \frac{n_{rd} + n_{fd}}{N_g} \times 100\%, \quad (6)$$

$$A_{FN} = \frac{n_f}{N_p} \times 100\%, \quad (7)$$

$$A_{TTP} = \frac{n_{rd}}{N_g} \times 100\%, \quad (8)$$

$$A_{PP} = \frac{n_{rd}}{n_{rd} + n_{fd}} \times 100\%, \quad (9)$$

式中： N_g 为所有实际存在的目标数量； N_p 为所有被聚类算法检测出的目标数量； n_{rd} 为目标物体被聚类算法正确检测且方向尺寸被正确估计的目标数量； n_{fd} 为目标物体被聚类算法正确检测但方向尺寸被错误估计的目标数量； n_f 为聚类算法误检测的目标数量。

所有评价参数的计算均以采集数据集中手工标记点云包围框信息为真实值，将算法检测结果与真实值进行比较，比较方法如下：1) 真实值中目标物数量计入 N_g ，算法检测结果中目标物数量计入 N_p ；2) 将算法检测结果中目标物中心点坐标值与手工标记目标点云中心点坐标值进行最近点匹配，将匹配后两目标物中心点欧氏距离小于等于 0.5 m 的算法检测结果中的目标物数量计入 N_r ，其余检测结果

中的目标物计入 n_r ；3) 将 N_r 中算法检测结果中目标物的方向角及长宽高三维尺寸与匹配真实值中目标物的方向角及长宽高三维尺寸进行求差，若满足

$$\begin{cases} \Delta\theta_h \leq 15^\circ \\ \Delta L + \Delta W + \Delta H \leq 0.2(L_t + W_t + H_t) \end{cases}, \quad (10)$$

则算法检测结果中的目标物计入 n_{rd} ，其余算法检测结果中的目标物计入 n_{fd} ，其中： $\Delta\theta_h$ 为算法检测结果中目标物方向角与真实值差值的绝对值； ΔL 、 ΔW 、 ΔH 分别为检测结果目标物长、宽、高与真实值差值的绝对值； L_t 、 W_t 、 H_t 为目标物长、宽、高真实值。

根据评价标准，将最小外包矩形算法^[3-4, 24] (MBR)、三点估计算法^[17] (TPE) 与位姿检测算法 (RDME) 在不同的激光雷达数据下进行了对比实验。MBR 算法是通过寻找点云簇轮廓最小外包矩形对盒模型包围框进行拟合；三点估计算法是通过求取点云簇中距离最远的两点，进一步寻找距离过此两点的直线最远的点作为第三点，使用此三点作为顶点对点云簇盒模型进行拟合。两种算法常用于三维点云目标跟踪、目标分类任务的前处理，具有一定的典型性及先进性。算法效果评估数据见表 2。

表 2 算法效果评估表

Table 2 Algorithm effect evaluation table

unit: %

LiDAR	Algorithm	TPA	FNA	TTPA	PPA
Velodyne VLP16	MBR	81.24	5.38	69.56	84.44
	TPE	81.24	5.38	68.76	83.13
	RDME	81.24	5.38	70.03	85.56
LEISHEN C32151L	MBR	81.80	6.84	70.83	86.93
	TPE	81.80	6.84	71.34	87.28
	RDME	81.80	6.84	74.23	90.91
ZVISION ML-30S	MBR	80.37	5.56	69.16	86.05
	TPE	80.37	5.56	70.17	87.62
	RDME	80.37	5.56	73.68	94.19

由于 MBR、TPE 与 RDME 采取同样的聚类分割算法，故这两种算法在正检率 (TPA) 及错检率 (FNA) 两项聚类算法评估参数下的表现相同。由表 2 数据可见，聚类算法的表现良好，具有较高的正检率 (TPA) 与较低的错检率 (FNA)。RDME 算法相较于 MBR 以及 TPE 算法在三种雷达下的表现更佳。当激光雷达线束质量提升、点云较为密集时，RDME 算法的 TTPA 最大提升 4.52 个百分

点。在 PPA 这一评价参数上，RDME 算法的性能最大提升 8.14 个百分点。RDME 算法可以有效地对目标物体的方向尺寸进行估计，并且根据激光雷达参数确定的自适应阈值聚类方法能够有效地对不同距离下的目标物体进行识别，提高聚类算法的准确率。RDME 算法、MBR 算法、TPE 算法在不同激光雷达点云数据集下的算法运行时间如图 11 所示。

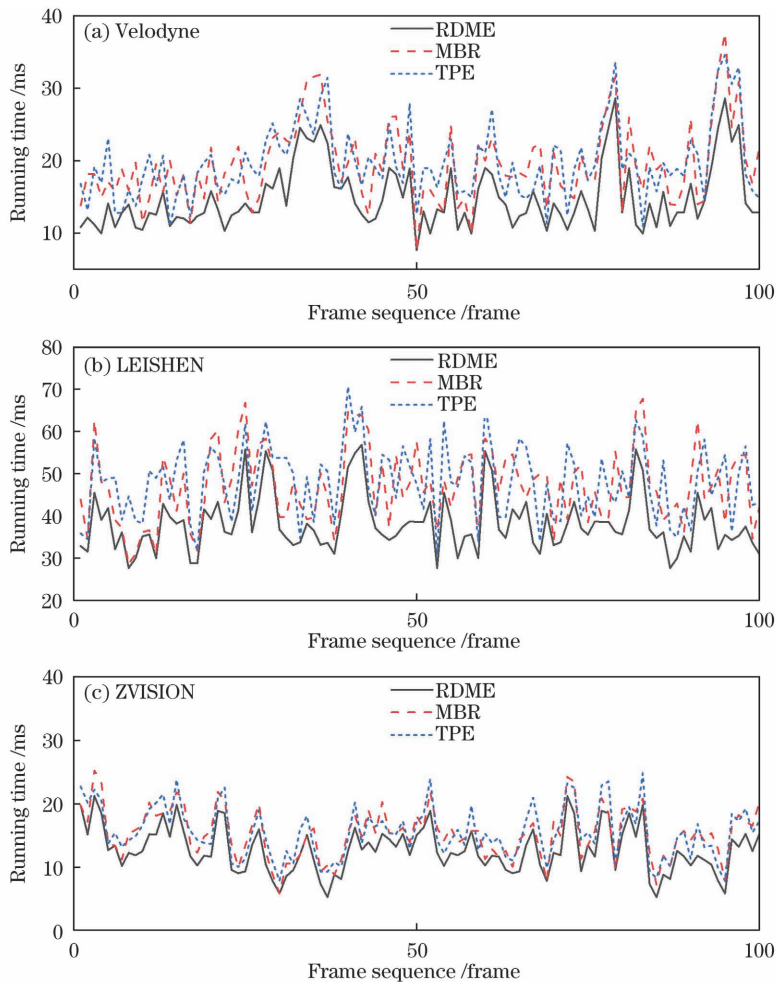


图 11 不同激光雷达下算法检测时间。(a)Velodyne VLP16;(b)镭神 C32151L;(c)ZVISION ML-30S

Fig. 11 Detection time of different LiDARs. (a) Velodyne VLP16; (b) LEISHEN C32151L; (c) ZVISION ML-30S

RDME 算法、MBR 算法、TPE 算法在相同帧序下呈现出不同的检测速度,在 Velodyne VLP16 激光雷达下平均检测时间分别为 14.84 ms、19.48 ms、19.71 ms(不计入数据传输等环节时间),RDME 的检测时间相较于 MBR 缩短 23.81%,相较于 TPE 缩短 24.71%;在镭神 C32151L 激光雷达下平均检测时间分别为 38.10 ms、47.02 ms、48.07 ms, RDME 的检测时间相较于 MBR 缩短 18.97%,相较于 TPE 缩短 20.74%;在 ZVISION ML-30S 激光雷达下平均检测时间分别为 12.78 ms、15.35 ms、15.56 ms, RDME 的检测时间相较于 MBR 缩短 16.74%,相较于 TPE 缩短 17.86%。从图中结果分析可得, RDME 算法较 MBR 算法及 TPE 算法所用的检测时间更少,实时性更好。

检测环节的耗时与当前场景帧下的点云数据量以及被检测出的所有目标数量有关,场景越复杂,目标物体数量越多,这会导致检测算法的计算量增加,从而导致检测时间增加。另外,激光雷达本身的数

据量对算法的检测速度也有重大的影响。实验过程中, RDME 算法最多可检测出 11 个目标物体,算法的检测时间最长可达 56.81 ms。在最大情况下算法检测速度仍然满足车辆实时性要求,故 RDME 算法能够在真实场景内对道路障碍物进行位姿检测。

4 结 论

提出一种依据扇形盒及坡度阈值的地面分割方法,较好地满足了地面点云分割的实时性与准确性需求。提出一种根据激光雷达性能参数确定聚类阈值的算法,利用该算法对点云进行处理,较好地克服了欧氏聚类较为依赖设置距离阈值的问题,能够使不同距离下的障碍物达到较好的聚类效果。利用点云投影后在目标物体表面分布较为集中的特点,提出一种位姿检测(RDME)算法,基于该算法对目标障碍物进行位姿信息检测。通过三种激光雷达实验,将位姿检测(RDME)算法与最小包围矩形(MBR)算法及三点估计(TPE)算法进行比较。

RDME 算法相较于 MBR 算法,在三种激光雷达下检测准确度分别提升 1.12%、3.98%、8.14%,检测时间分别缩短 23.81%、18.97%、16.74%;RDME 算法相较于 TPE 算法,在三种激光雷达下检测准确度分别提升 2.43%、3.63%、6.57%,检测时间分别缩短 24.71%、20.74%、17.86%,在检测精度及实时性上均有着明显提升。算法最长平均检测时间为 47.02 ms,小于 10 Hz 激光雷达实时性要求 (0.1 s),能够满足智能车实时性需求,是一种较为有效的方法。

参 考 文 献

- [1] He Y B, Chen R L, Wu K, et al. Point cloud simplification method based on k-means clustering [J]. *Laser & Optoelectronics Progress*, 2019, 56(9): 091002.
贺一波,陈冉丽,吴侃,等. 基于 k-means 聚类的点云精简方法[J]. *激光与光电子学进展*, 2019, 56(9): 091002.
- [2] Zhang C H. Clustering and tracking method research of lidar-based obstacle in urban environment [D]. Hefei: University of Science and Technology of China, 2019.
张彩红. 城区环境下基于激光雷达的障碍物聚类 and 跟踪方法研究[D]. 合肥: 中国科学技术大学, 2019.
- [3] Yang C. A study on obstacle detection method with unpiloted vehicle [D]. Xi'an: Xi'an Technological University, 2015.
杨成. 无人驾驶智能车障碍检测方法研究[D]. 西安: 西安工业大学, 2015.
- [4] Xu G Y, Niu H, Guo C Y, et al. Research on target recognition and tracking based on 3D laser point cloud [J]. *Automotive Engineering*, 2020, 42(1): 38-46.
徐国艳,牛欢,郭宸阳,等. 基于三维激光点云的目标识别与跟踪研究[J]. *汽车工程*, 2020, 42(1): 38-46.
- [5] Charles R Q, Hao S, Mo K C, et al. PointNet: deep learning on point sets for 3D classification and segmentation [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 77-85.
- [6] Zhou Y, Tuzel O. VoxelNet: end-to-end learning for point cloud based 3D object detection [C] // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 4490-4499.
- [7] Yan Y, Mao Y X, Li B. SECOND: sparsely embedded convolutional detection[J]. *Sensors*, 2018, 18(10): 3337.
- [8] Lang A H, Vora S, Caesar H, et al. PointPillars: fast encoders for object detection from point clouds [C] // 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 12689-12697.
- [9] Shi S S, Wang X G, Li H S. PointRCNN: 3D object proposal generation and detection from point cloud [C] // 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 770-779.
- [10] Lou X Y, Wang H, Cai Y F, et al. A research on an algorithm for real-time detection and classification of road obstacle by using 64-line lidar [J]. *Automotive Engineering*, 2019, 41(7): 779-784.
娄新雨,王海,蔡英凤,等. 采用 64 线激光雷达的实时道路障碍物检测与分类算法的研究[J]. *汽车工程*, 2019, 41(7): 779-784.
- [11] Chu P, Cho S, Sim S, et al. A fast ground segmentation method for 3D point cloud [J]. *Journal of Information Processing Systems*, 2017, 13(3): 491-499.
- [12] Wang X, Wang J Q, Li K Q, et al. Fast segmentation of 3-D point clouds for intelligent vehicles [J]. *Journal of Tsinghua University (Science and Technology)*, 2014, 54(11): 1440-1446.
王肖,王建强,李克强,等. 智能车辆 3-D 点云快速分割方法[J]. *清华大学学报(自然科学版)*, 2014, 54(11): 1440-1446.
- [13] Cortinhal T, Tzelepis G, Erdal A E. SalsaNext: fast, uncertainty-aware semantic segmentation of LiDAR point clouds for autonomous driving [EB/OL]. (2020-03-07)[2021-04-10]. <https://arxiv.org/abs/2003.03653>.
- [14] Fan X H, Xu G L, Li W L, et al. Target segmentation method for three-dimensional LiDAR point cloud based on depth image [J]. *Chinese Journal of Lasers*, 2019, 46(7): 0710002.
范小辉,许国良,李万林,等. 基于深度图的三维激光雷达点云目标分割方法[J]. *中国激光*, 2019, 46(7): 0710002.
- [15] Li J, Zhao K, Bai R, et al. Urban ground segmentation algorithm based on ray slope threshold [J]. *Acta Optica Sinica*, 2019, 39(9): 0928004.
李炯,赵凯,白睿,等. 基于射线坡度阈值的城市地面分割算法[J]. *光学学报*, 2019, 39(9): 0928004.
- [16] Shao J T, Du C Q, Zou B. Lidar ground segmentation method based on point cloud cluster combination feature [J]. *Laser & Optoelectronics*

- Progress, 2021, 58(4): 0428001.
- 邵靖滔, 杜常清, 邹斌. 基于点云簇组合特征的激光雷达地面分割方法[J]. 激光与光电子学进展, 2021, 58(4): 0428001.
- [17] Sualeh M, Kim G W. Dynamic multi-LiDAR based multiple object detection and tracking[J]. Sensors, 2019, 19(6): 1474.
- [18] Ye Y T, Fu L M, Li B J. Object detection and tracking using multi-layer laser for autonomous urban driving[C]//2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), November 1-4, 2016, Rio de Janeiro, Brazil. New York: IEEE Press, 2016: 259-264.
- [19] Zermas D, Izzat I, Papanikolopoulos N. Fast segmentation of 3D point clouds: a paradigm on LiDAR data for autonomous vehicle applications[C]//2017 IEEE International Conference on Robotics and Automation (ICRA), May 29-June 3, 2017, Singapore. New York: IEEE Press, 2017: 5067-5073.
- [20] Douillard B, Underwood J, Vlaskine V, et al. A pipeline for the segmentation and classification of 3D point clouds [M] // Experimental robotics. Heidelberg: Springer, 2014: 585-600.
- [21] Lu X H, Yao J, Tu J G, et al. Pairwise linkage for point cloud segmentation [J]. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016, III-3: 201-208.
- [22] Börcs A, Nagy B, Benedek C. Fast 3D urban object detection on streaming point clouds[M]//Agapito L, Bronstein M M, Rother C. Computer vision-ECCV 2014 Workshops. Lecture notes in computer science. Cham: Springer, 2015, 8926: 628-639.
- [23] Lehtomäki M, Jaakkola A, Hyyppä J, et al. Object classification and recognition from mobile laser scanning point clouds in a road environment [J]. IEEE Transactions on Geoscience and Remote Sensing, 2016, 54(2): 1226-1239.
- [24] Ye Y T, Li B J, Fu L M. Fast object detection and tracking in laser data for autonomous driving [J]. Geomatics and Information Science of Wuhan University, 2019, 44(1): 139-144, 152.
- 叶语同, 李必军, 付黎明. 智能驾驶中点云目标快速检测与跟踪[J]. 武汉大学学报·信息科学版, 2019, 44(1): 139-144, 152.
- [25] Li W M, Liu C. Research on improved PCA-based ICP point cloud registration algorithm[J]. Industrial Control Computer, 2020, 33(6): 11-13.
- 李为民, 刘超. 基于 PCA 的 ICP 点云配准算法的改进研究[J]. 工业控制计算机, 2020, 33(6): 11-13.

Position Detection Algorithm of Road Obstacles Based on 3D LiDAR

Hu Jie^{1,2,3*}, Liu Han^{1,2,3}, Xu Wencai^{1,2,3}, Zhao Liang^{1,2,3}

¹Hubei Key Laboratory of Advanced Technology for Automotive Components, Wuhan University of Technology, Wuhan, Hubei 430070, China;

²Hubei Collaborative Innovation Center for Automotive Components Technology, Wuhan University of Technology, Wuhan, Hubei 430070, China;

³Hubei Research Center for New Energy & Intelligent Connected Vehicle, Wuhan University of Technology, Wuhan, Hubei 430070, China

Abstract

Objective In the field of autopilot, the intelligent driving decision is highly dependent on the accurate detection of road obstacles, and the accurate information of road obstacles is the prerequisite for automatic driving. Three dimensional (3D) LiDAR has a high resolution, high detection accuracy, all-weather operation, and strong anti-jamming ability, allowing it to provide accurate environmental information for vehicles. To detect road obstacles using a 3D point cloud generated by 3D LiDAR, the point cloud data must be processed. The point cloud is divided into independent subsets, each of which corresponds to targets with distinct physical meanings. The target detection of a point cloud is divided into several modules, the most important of which are preprocessing, ground segmentation, point cloud clustering, target output, and other modules that lack the target's pose information. Pose information is critical for tracking, classification, and other tasks involving target obstacles, so it is critical to accurately detect the target's pose.

Methods The target detection based on 3D point cloud is generally divided into preprocessing, ground

segmentation, point cloud clustering, target output, and other modules. On this basis, the target's pose information is determined further. A segmentation method based on a fan-shaped box is proposed to segment the ground point cloud and the target point cloud to meet the requirements of high accuracy and high real-time for ground point cloud segmentation. To address the issue of Euclidean clustering's fixed distance threshold's poor adaptability to the characteristics of LiDAR close dense far sparse, a Euclidean clustering method with adaptive clustering threshold is proposed to improve the clustering accuracy of the clustering algorithm at different distances. A boundary line fitting method based on the random sample consensus (RANSAC) algorithm is proposed to determine the target object's box model to meet the requirement of pose information detection. The direction angle is obtained by direction fitting, and the dimension is obtained by coordinate system rotation, which reduces the amount of computation and enhances the real-time performance of the algorithm. RANSAC direction and measurement estimation is the name of the pose detection algorithm used in this paper (RDME). To validate the algorithm, 100 frames point clouds are randomly selected from each of the three LiDAR datasets during the experiments. Statistics of the number of objects in all single frame point clouds and the number of targets detected by the clustering algorithm and pose detection algorithm record the processing time of each frame and use four indicators to evaluate the algorithm effect, and the algorithm is compared with the minimum bounding rectangle (MBR) algorithm and the Three-Point Estimation (TPE) algorithm to verify the algorithm's effectiveness and advanced nature.

Results and Discussions The clustering algorithm performs well, with high true positive accuracy (TPA) and low false negative accuracy (FNA). RDME algorithm performs better than MBR and TPE algorithms in three LiDAR datasets. The RDME algorithm performs better when the number of LiDAR laser rays is increased and the point cloud is dense, and the total true positive accuracy (TTPA) is improved by up to 4.52 percentage points. For PPA, the performance of RDME algorithm is improved by up to 8.14 percentage points. The RDME algorithm can effectively estimate the direction of the target object, and the adaptive threshold clustering method based on LiDAR parameters can effectively identify the target objects at different distances and improve the clustering algorithm's accuracy (Table 2). In the same frame sequence, the detection speeds of the RDME, MBR, and TPE algorithms differ. The average detection time of the RDME algorithm is 14.84 ms, 19.48 ms, and 19.71 ms respectively (excluding the link time of data transmission) in the first LiDAR datasets, and RDME algorithm detection time is 23.81% lower than the MBR algorithm, and 24.71% less than TPE algorithm; the average detection time of the RDME algorithm is 38.10 ms, 47.02 ms and 48.07 ms respectively in second LiDAR datasets, and the RDME algorithm detection time is 18.97% lower than MBR algorithm and 20.74% less than TPE algorithm; the average detection time of the RDME algorithm is 12.78 ms, 15.35 ms and 15.56 ms respectively in the third LiDAR datasets, and the RDME algorithm detection time is 16.74% lower than MBR algorithm and 17.86% less than TPE algorithm (Fig. 11). The results show that the RDME algorithm has less detection time and better real-time performance than the MBR algorithm and TPE algorithm.

Conclusions A ground segmentation method based on sector bin box and slope threshold is proposed, which can meet the real-time and accuracy requirements of ground point cloud segmentation. This paper proposes an algorithm for determining the clustering threshold based on LiDAR performance parameters to process the point cloud, overcoming the problem that European clustering relies on setting the distance threshold. For obstacles at different distances, the clustering algorithm can achieve a better clustering effect. A pose detection (RDME) algorithm is proposed to detect the pose information of the target obstacle based on the feature that the point cloud is concentrated on the surface of the target object after projection. The position detection (RDME) algorithm is compared to the MBR algorithm and the TPE algorithm using three different types of LiDAR datasets. Compared with the MBR algorithm, the RDME algorithm improves detection accuracy by 1.12%, 3.98%, and 8.14%, and reduces detection time by 23.81%, 18.97%, and 16.74%; compared with the TPE algorithm, the detection accuracy of the RDME algorithm is improved by 2.43%, 3.63%, and 6.57% respectively, and the detection time is reduced by 24.71%, 20.74%, and 17.86% respectively. The RDME algorithm's detection accuracy and real-time performance have been significantly improved. The algorithm's maximum average detection time is 47.02 ms, which is less than 0.1 s of the 10 Hz LiDAR's real-time requirement. It can meet the real-time needs of intelligent vehicles and is a more efficient method.

Key words remote sensing; autonomous vehicle; LiDAR; point cloud object detection; position detection

OCIS codes 280.3640; 100.4996; 150.1135; 150.6910