

基于 LiDAR 点云的仓库用无人消防车 障碍物检测算法研究

郭昊琰^{1, 2}, 陈志成^{1, 2}, 杨晨^{1, 2}

(1. 中国船舶集团有限公司 第七一三研究所, 郑州 450000;

2. 郑州市特种场所火灾防护技术重点实验室, 郑州 450000)

摘要: 利用三维点云数据进行道路信息获取对于提高仓库用无人消防车的智能化水平至关重要, 在无人消防车的功能架构中, 障碍物检测承担着重要作用, 可有效提升无人消防车灭火效率, 实现对火源的及时扑灭; 分析总结了基于机械式激光雷达点云数据的障碍物检测算法, 针对障碍物检测的实时性要求以及无人消防车运动导致的点云数据畸变问题, 通过点云预处理提高了算法的运行速度, 采用畸变补偿算法减小了由车辆运动引起的失真; 改进了传统的聚类算法, 实现了既可以准确检测距离较近或较远的障碍物, 又可以减少噪声和异常点对聚类效果的影响; 将所述算法应用于原始数据集的处理, 对算法的有效性进行评估, 结果表明障碍物聚类精度提高了 5.1%, 聚类速度提高了 4.9%, 能够实现对多种障碍物的准确、快速检测, 对仓库用无人消防车避障能力的提升具有一定的指导意义。

关键词: 激光雷达; 无人消防车; 点云聚类; 障碍物检测

Research on Obstacle Detection Algorithm of Warehouse-Used Unmanned Fire-Fighting Vehicles Based on LiDAR Point Cloud

GUO Haoyan^{1, 2}, CHEN Zhicheng^{1, 2}, YANG Chen^{1, 2}

(1. The 713th Research Institute of CSSC, Zhengzhou 450000, China;

2. Zhengzhou Key Laboratory of Fire Protection Technology in Special Places, Zhengzhou 450000, China)

Abstract: By using three-dimensional point cloud data, it is of great importance for road information acquisition to improve the intelligence level of warehouse-used unmanned fire-fighting vehicles, and unmanned fire-fighting vehicles play an important role in obstacle detection, which can improve the efficiency of unmanned fire-fighting vehicles and extinguish fire sources in a timely manner. This paper analyzes and summarizes the obstacle detection algorithm based on mechanical LiDAR point cloud data. Aiming at the real-time requirement of obstacle detection and the point cloud data distortion caused by the movement of unmanned fire-fighting vehicles, the algorithm improves the running speed by point cloud preprocessing, and the distortion correction algorithm is used to reduce the distortion caused by the movement of the vehicle. Improving the traditional clustering algorithm to accurately detect close or far obstacles and reduce the influence of the noise and outliers on the clustering effect. The algorithm is applied to process original data set and evaluate its effectiveness. The results show that the obstacle clustering accuracy is increased by 5.1%, and the clustering speed by 4.9%, which can realize the accurate and rapid detection of various obstacles, and it has a certain guiding significance for warehouse-used unmanned fire vehicles to improve the ability of obstacle avoidance.

Keywords: LiDAR; unmanned fire-fighting vehicle; point cloud clustering; obstacle detection

0 引言

仓库火场救援存在救援难度大、效率低、危险系数高等问题^[1], 灾害现场往往伴随着易燃易爆、缺氧、有

毒等危险状况, 使消防救援人员难以深入现场^[2-3]。因此世界各国都在积极进行仓库用无人消防车的研制, 将自动驾驶技术应用于消防车, 以帮助或代替消防救援人员在危险区域进行环境侦察和消防工作。

收稿日期: 2024-06-30; 修回日期: 2024-08-07。

作者简介: 郭昊琰(1995-), 男, 硕士, 助理工程师。

引用格式: 郭昊琰, 陈志成, 杨晨, 等. 基于 LiDAR 点云的仓库用无人消防车障碍物检测算法研究[J]. 计算机测量与控制, 2025, 33(8): 242-248.

无人消防车实现自动驾驶的技术路线主要有两种: 一种是基于智能化道路基础设施的车辆—道路协同方案, 另一种是基于车载传感器的环境感知方案。前者需要建设配套的系统化设施进行路径引导, 在情况不明的灾害现场其可行性和机动性不高; 而后者具有可操作性强、时间成本低、环境适应能力强等优势^[4]。随着相关技术研究的不断深入, 人们逐渐认识到快速感知道路信息对于提高无人消防车救援效能的重要性^[5-6]。近年来, 车载传感器的硬件素质不断提高, 数据信息越来越准确, 为目标检测算法提供了可靠的决策信息^[7]。车载环境感知设备中, 传统摄像头在昏暗的室内环境下获取的环境图像信息十分有限^[8], 而激光雷达可以在光照不足的情况下持续稳定工作, 全天候捕获道路障碍物的三维点云数据, 表征目标尺寸并精确定位^[9], 已广泛应用于无人消防车的环境识别与路径规划, 能够有效提升消防救援效能^[10], 因此该领域的研究具有重要的现实意义。然而车载激光雷达点云会因车辆运动产生畸变, 传统点云处理算法存在处理速度慢、难以准确检测距离较近或较远障碍物、易受到噪声和异常点干扰的问题, 影响了障碍物检测的精度和可靠性^[11-12]。

本文对仓库用无人消防车车载激光雷达的障碍物检测问题进行研究, 通过点云预处理算法消除畸变并提高了运算速度, 对传统聚类算法的搜索半径和聚类方式进行改进, 提高了障碍物点云聚类的精度和速度, 实现了对障碍物的准确、快速检测。

1 点云障碍物检测算法基本原理

基于点云数据的障碍物检测是指从点云数据中提取出障碍物的方位、尺寸、运动方向等信息^[13], 检测算法流程如图 1 所示, 主要由以下 3 部分组成:

1) 点云预处理: 车辆自身的运动会造成车载激光雷达获取的点云数据产生畸变, 传统点云处理算法存在处理速度慢的问题, 通过点云预处理算法消除畸变信息, 并减少无效点云对算法性能的影响^[14], 从而提高运算速度。

2) 点云聚类: 通过将点云数据聚类为一系列具有共同特征的群, 实现对障碍物的识别和提取, 是实现障碍物检测的关键^[15]。

3) 包络矩形的建立: 对每一系列聚类, 确定一个能够将点云簇包裹的矩形框, 通过计算矩形框的尺寸、

朝向等获取障碍物的特征信息。

算法主要用于处理车载机械式激光雷达采集的点云数据。机械式激光雷达是一种发展最早、技术较成熟的激光雷达, 其具有旋转工作部件, 以激光作为工作光束^[16], 采用回波分析和 ToF (Time of Flight) 技术原理得到大量无拓扑结构的离散空间坐标点, 称为点云, 进而获得周围环境物体的距离和轮廓信息^[17-18], 基本组成和工作原理如图 2 所示。

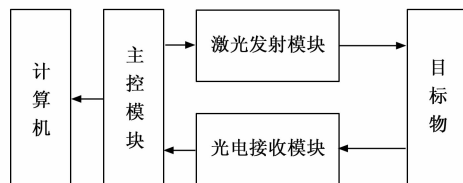


图 2 激光雷达基本组成和工作原理框图

1.1 点云预处理

点云预处理算法主要由车身点云移除、地面点云分割及畸变补偿 3 部分组成。车载激光雷达每秒能够采集大量点云数据, 处理这些数据需要占用大量的计算资源和存储空间^[19]。为了满足消防车障碍物检测及避障的实时性要求, 需要首先滤除车身点云和地面点云等无用数据; 同时, 激光雷达点云会因目标与车辆本身的相对运动产生畸变^[20], 因此还要进行畸变补偿。

1.1.1 车身点云移除

车载机械式激光雷达旋转角度可达到 360° , 且发射的激光束具有一定的照射范围, 工作过程中部分激光束会照射到自身车身上, 若不滤除这些数据点, 后续的处理算法会将其识别为障碍物。本文采用算法主要根据车身尺寸以及激光雷达安装高度设置相应的三维坐标阈值, 并删除阈值范围内的点云数据。以激光雷达中心为原点, 以车辆前进方向为 x 轴正方向, 以雷达轴向为 z 轴, 建立右手空间坐标系, 则三维坐标满足式 (1) ~ (3) 的点云应删除:

$$-\frac{l}{2}-d \leq x \leq \frac{l}{2}+d \quad (1)$$

$$-\frac{w}{2} \leq y \leq \frac{w}{2} \quad (2)$$

$$-h \leq z \leq 0 \quad (3)$$

式中, l 、 w 分别为车身的长度跟宽度, h 为车载激光雷达的安装高度, d 为激光雷达中心到车辆长度中心的距离。

1.1.2 地面点云分割

原始点云中还包含大量的地面点, 其存在会干扰障碍物的检测, 降低运算效率。本文采用 Ray Ground Filter 算法进行地面点云分割并滤除, 基本思想是利用射线 (ray) 的形式组织点云: 首先根据点云的高度, 过

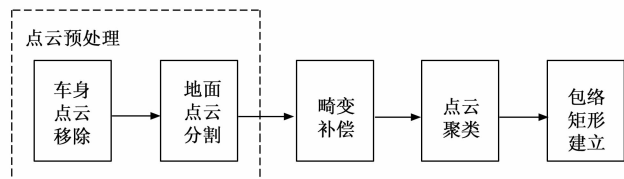


图 1 基于 LiDAR 点云的障碍物检测算法流程图

滤掉那些过高或靠近车身的点；然后将点云投影到水平面上，将三维点云数据降维至 $o-xy$ 平面的二维数据；再计算每个点所在线束与车辆前进方向的平面夹角，并对同一夹角线束上的点按照半径大小进行排序；最后计算相邻两点相对于水平面的平面坡度，如果坡度低于设定的阈值，则判断该点为地面点^[21]，如图 3 所示。这种算法可以减少数据维度、提高运算速度。

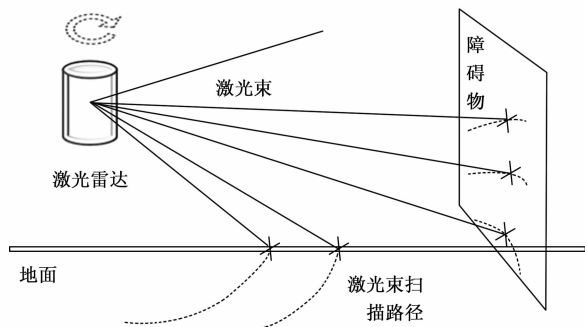


图 3 地面点云分割示意图

1.1.3 畸变补偿

车辆自身的运动会产生点云数据畸变，将畸变数据直接用于障碍物检测，可能得到错误的检测结果。畸变补偿算法步骤如下：

- 1) 将同一帧点云的坐标从车辆坐标转换为该点的地面坐标；
- 2) 将转换后的地面坐标平移来消除激光雷达在采集周期内车辆直行过程产生的点云畸变；
- 3) 通过地面坐标旋转消除激光雷达旋转产生的畸变；
- 4) 将地面坐标转换为车辆坐标方便对矫正后的点云进行后续处理。

1.2 障碍物点云聚类

完成点云预处理后，需要采用聚类算法对点云进行分组，将障碍物点云提取出来。常用的聚类方法包括基于划分的聚类、基于层次的聚类、基于栅格的聚类和基于密度的聚类。其中基于密度的聚类方法的显著优点在于，它不需要预先知道点云数据集中的簇数量，还能够有效处理噪声和异常点，非常适用于车载激光雷达点云的障碍物检测。常见的基于密度的聚类算法如 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)，能够在特征密度较高的数据集中聚类形成任意形状的簇，且不易受噪声的干扰^[22]。DBSCAN 算法包含搜索半径 (Eps) 和密度阈值 (MinPts) 两个关键参数，处理过程如下：

- 1) 首先从数据集中选择一个任意点。如果附近由搜索半径定义的点的数量大于或等于密度阈值，则这些

点和初始点一起归入一个簇中；

- 2) 然后通过迭代检查候选点的邻接点来扩大集群，如果它们符合由密度阈值规定的密度标准，则将它们纳入簇中。

上述过程一直持续到形成一个完整的簇，任何邻域密度低于密度阈值的点都被认为是一个噪声点。如果一个候选点的邻接点少于密度阈值，它就被标记为一个边界点，不能进一步促进簇的增长。

1.3 最小包围矩形的建立

要想直观地获得道路障碍物的语义信息并发送给后续决策部分，以完成避障，就需要对聚类后的点云进行包围矩形的建立^[23]，通过计算外部包围框得到相应的障碍物信息。本文采用基于点云凸包的最小包围矩形算法，首先从障碍物点云中提取凸包点，然后生成包含这些凸包点的外部矩形，将所有外部矩形中面积最小的确定为该障碍物的最小包围矩形，再把二维矩形扩展为三维边框，用以代替障碍物并表征其尺寸、行进方向、空间位置等特征信息。

1.3.1 点云凸包的提取

凸包指能够包含点集内所有点的凸多边形的交集，即点集最外层点连起来形成的凸多边形。点云凸包的提取是建立包围矩形的关键步骤。本文采用的点云凸包提取算法基于 Graham 扫描法^[24]。首先找到障碍物点云中 y 坐标最小的点，记为 P_0 ，以 P_0 为极点，将所有点根据极角 α 大小进行排序，排序好的点记为 $P_1, P_2, \dots$ ，如图 4 (a) 所示。然后将最低点 P_0 和第一个点 P_1 推入栈，对待入栈的点 P_2 进行判断，若点 P_2 位于向量 $\overrightarrow{P_0P_1}$ 左侧，即顺序三点 $P_0P_1P_2$ 为逆时针方向，则将点 P_2 推入栈；若点 P_2 位于向量 $\overrightarrow{P_0P_1}$ 右侧，即顺序三点 $P_0P_1P_2$ 为顺时针，则先将当前栈顶元素 P_1 推出，再将点 P_2 推入栈，接着对下一个待入栈点 P_3 进行判断。按顺序遍历完所有点，最终栈内剩余元素就是障碍物的凸包点，如图 4 (b) 所示。上述逆/顺时针关系可由向量叉积进行数学描述，如式 (4) 所示：

$$P_{n-1}P_n \times P_{n-1}P_{n+1} = \begin{cases} < 0, & \text{逆时针} \\ > 0, & \text{顺时针} \end{cases} \quad (4)$$

1.3.2 包围矩形的拟合

首先确定候选的包围矩形。从凸包点中任选两个相邻点形成矩形的一条边，将其作为主方向；随后将其余点投射到这条边上，计算这些投影点的最大间距，从而确定矩形的长；矩形的宽是通过计算离这条边最远点的距离确定的。按上述过程遍历所有的凸包点，得到多个候选包围矩形，如图 5 所示。

最后，通过计算面积确定候选包围矩形中的最小包围矩形，再沿 z 轴方向把二维矩形拓展成三维，使用三维边框代替障碍物，获取障碍物点云的几何尺寸、朝

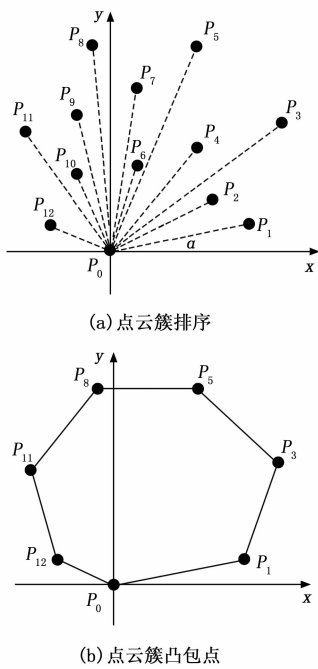


图 4 凸包点提取过程示意图

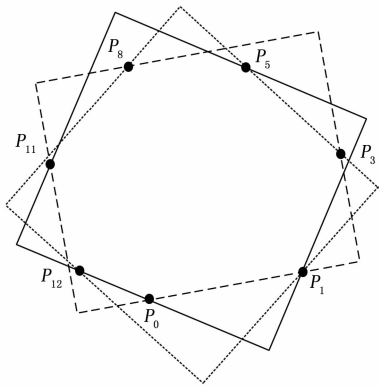


图 5 基于凸包点的候选包围矩形

向、方位等特征信息, 实现障碍物检测。

2 DBSCAN 聚类算法改进

若将 DBSCAN 直接用于点云聚类, 会存在参数自适应性和聚类效果较差的缺点。为解决这一问题, 对传统 DBSCAN 聚类算法从以下两个方面进行了改进。

2.1 自适应搜索半径

根据激光雷达的测距原理, 在各角度分辨率相同的情况下, 越远的相邻点间距越大, 即距离激光雷达近的物体, 其表面点云密度大于远处的物体, 若搜索半径 (Eps) 和密度阈值 ($MinPts$) 均为固定值, 将会造成分割不足或分割过当: 阈值过大, 可能导致近处的两个物体被分为一类, 即分割不足; 阈值过小, 可能导致远处的同一物体被分割开, 即分割过当^[25]。因此 Eps 和 $MinPts$ 的取值应随不同距离处点云的分布而变化。另

外, 激光雷达的分辨率在水平和垂直方向上也存在差异, 当试图用一个固定的 Eps 分割紧临物体时, 较大或较小的 Eps 都不能进行有效分割。因此, Eps 在水平和垂直方向上也要有所变化, 这种自适应的搜索半径方法在点云数据中形成一个动态的搜索椭圆, 如图 6 所示, 黑白两色圆点代表不同物体表面的数据点。

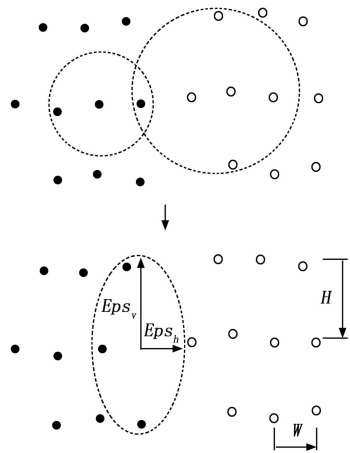


图 6 自适应搜索半径示意图

如图 6 所示, 由于激光雷达分辨率在水平、垂直方向上存在差异, 使用参数 W 和 H 分别表示在一个给定的障碍物位置上两个相邻点之间的水平和垂直距离, 用于确定水平搜索半径 Eps_h 和垂直搜索半径 Eps_v , Eps_h 和 Eps_v 分别对应了搜索椭圆的半短轴和半长轴, W 和 H 的计算见式 (5):

$$\begin{cases} W = 2d \cdot \tan(\theta_h/2) \\ H = 2d \cdot \tan(\theta_v/2) \end{cases} \quad (5)$$

式中, d 为该位置点到激光雷达中心的距离, θ_h 和 θ_v 分别表示激光雷达在水平和竖直方向上的角分辨率, 查阅用户手册可知。为了能够搜索到同一物体表面的相邻点, 搜索半径应不小于该位置两相邻点间距。 Eps_h 和 Eps_v 的计算方式如式 (6) 所示:

$$\begin{cases} Eps_h = (a - 1) \cdot W \\ Eps_v = (b - 1) \cdot H \end{cases}, \quad a \geq 2, b \geq 2 \quad (6)$$

式中, a 、 b 分别表示搜索半径在水平、垂直方向上跨越的激光束数量, 应至少跨越两个激光束, 否则无法聚类。

2.2 距离—密度簇合并方法

引入随距离变化的自适应参数, 可以在一定程度上优化聚类效果, 但对密度不同的簇进行处理时, 还可能出现大量的噪声点和离群点。为解决这一问题, 本文考虑结合距离和密度两个参数对点云数据进行簇合并, 以提高聚类的准确性和可靠性。算法的基本流程如下, 当两个簇的密度和距离同时满足一定条件时才合并成一个新的簇:

1) 输入点云数据 C 并通过传统 DBSCAN 算法得到原始的簇划分结果;

2) 计算簇间的距离并将距离小于预设阈值的簇进行合并。对于两个簇 C_i 和 C_j , 他们之间的距离可以用欧式距离公式计算得出, 如式 (7) 所示:

$$d(C_i, C_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{7}$$

如果距离小于搜索半径, 则将两个簇合并为更大的簇 C_{ij} , 新簇的质心由加权平均的方式计算得出, 如式 (8) 所示:

$$c_{ij} = \frac{1}{n_i + n_j} \sum_{k=1}^p (n_i c_{ik} + n_j c_{jk}) e_k \tag{8}$$

式中, p 代表特征向量的维数, e_k 代表第 k 个基向量, n_i 和 n_j 表示第 i 和第 j 个数据点的权重, c_{ik} 和 c_{jk} 分别表示第 i 和第 j 个数据点在第 k 维的特征值;

3) 计算簇内密度, 并将密度小于阈值的点从簇内删除。对于每一个簇 C_k , 由式 (9) 计算其平均距离:

$$\rho(C_k) = \frac{1}{n_k} \sum_{p \in C_k} dist(p, C_k) = \frac{1}{n_k} \sum_{p \in C_k} \|p - C_k\| \tag{9}$$

式中, n_k 表示簇 C_k 的数据量, $dist(p, C_k)$ 表示数据点 p 到簇 C_k 聚类中心的距离;

4) 重复执行上述 2) 和 3), 直到无法合并新簇或删除簇为止。

通过上述方式进一步优化聚类效果, 能够有效减少噪声点和离群点, 提高聚类准确性。

3 实验结果和分析

3.1 实验环境配置

基于 Ubuntu18.04.6 系统进行实验环境配置, 使用 Python 语言按照前述点云障碍物检测算法编写程序, 运行平台处理器主频为 2.4 GHz, 运行内存为 8 GB, 对公开的机械式激光雷达原始点云数据集进行处理, 对比分析处理结果, 验证算法的有效性。

3.2 点云预处理结果分析

3.2.1 地面点云分割算法结果分析

如图 7 (a) 所示, 原始数据包含了其他车辆、行人、树木和建筑物等多种物体信息。采用 Ray Ground Filter 算法对地面点云进行分割, 得到了图 (b) 所示的一帧点云图, 可以看到地面点被完整地去除。无效点云的剔除, 进一步保障了后续数据处理的运算速度和实时性。

3.2.2 畸变补偿算法结果分析

为了评估畸变补偿效果, 提取了经地面点云分割处理后的包含行人信息的点云, 然后进行畸变补偿并对感兴趣域局部放大。图 8 (a) 为车辆静止时获得的目标点云数据, 图 8 (b) 为车辆运动时获得的目标点云数据,

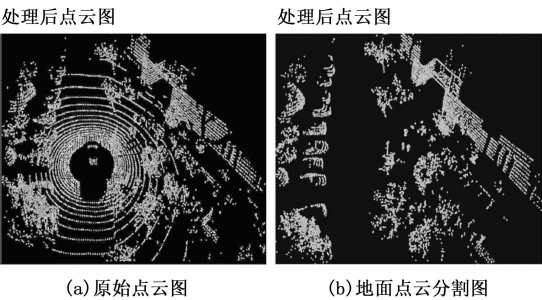


图 7 地面点云分割结果

表 1 总结了畸变补偿前后点云数据各项特征变化情况, 可以看出车辆运动时, 包含行人信息的点云出现一定程度的失真, 畸变主要体现在目标点云的宽度有所增加。图 8 (c) 展示了畸变补偿后的点云数据, 表 2 为畸变补偿前后目标点云长度和宽度的变化, 可以看出宽度畸变得到有效控制, 而长度基本不变, 效果符合预期。

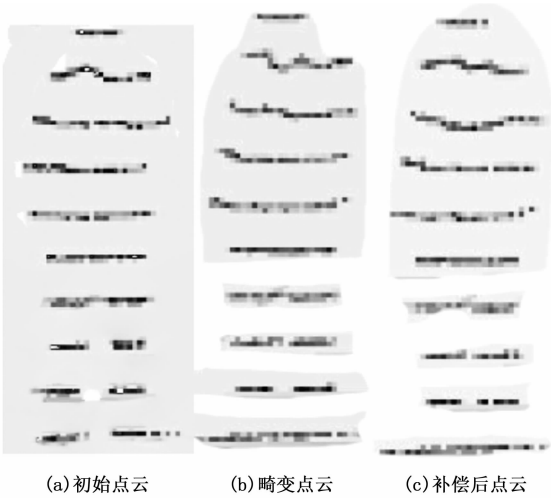


图 8 包含行人信息的点云数据处理结果

表 1 包含行人信息的点云数据变化情况

点云信息	图 8(a)	图 8(b)	图 8(c)
点云数目/点	365	353	353
聚类宽度/m	0.239	0.289	0.243
聚类长度/m	0.584	0.582	0.582
聚类高度/m	1.697	1.689	1.689
聚类体积/m³	0.237	0.279	0.253
检测距离/m	6.196	6.135	6.155

表 2 校正前后行人点云数据尺寸变化

点云尺寸	长度/m	宽度/m
原始尺寸	0.584	0.239
点云畸变后尺寸	0.582	0.289
校正后尺寸	0.582	0.243

3.3 障碍物聚类结果分析

为了验证改进的聚类算法的聚类效果, 以及最小包

络矩形算法的处理结果，采用两种聚类算法对包含多种障碍物的点云数据进行处理，并对聚类结果建立最小包络矩形，检测结果如图 9、图 10 所示。

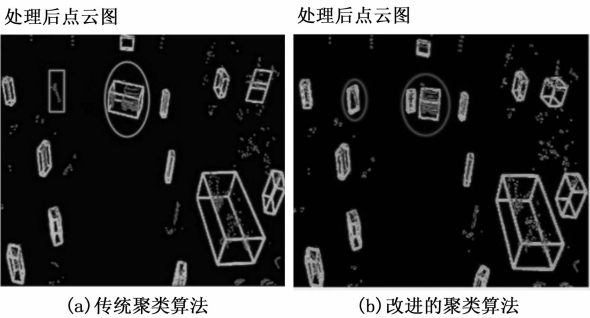


图 9 第一组聚类结果比较

图 9 所示的原始激光雷达点云数据中主要包含车辆、行人及树木等障碍物。如图 9 (a) 中矩形框所示，传统聚类算法的聚类参数固定，未能将距离较远处狭长的树干聚类出来并形成有效的障碍物检测结果，聚类结果不准确；而改进的聚类算法能够检测到相应位置的树干障碍物，如图 9 (b) 左侧椭圆框所示。如图 9 (a) 椭圆框所示，相邻的车辆与行人未能被区分开来，这是因为该位置车辆和行人都距离激光雷达较近且间距较小，聚类结果将其视为同一个障碍物；而改进的聚类算法能够将该位置临近的车辆和行人区分开，如图 9 (b) 右侧椭圆框所示。由以上结果可知，改进的聚类算法能够有效避免点云分割不足问题，改善障碍物漏检情况。

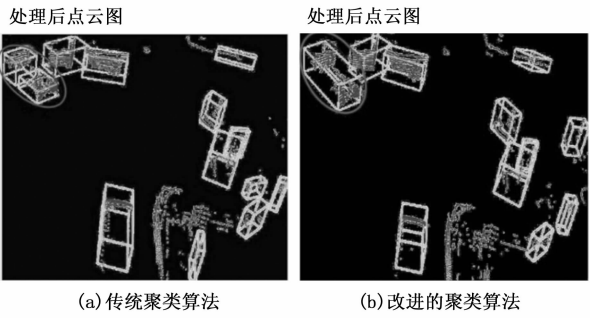


图 10 第二组聚类结果比较

图 10 所示的原始激光雷达点云数据中主要包含货车、普通车辆及树木等障碍物。该组数据中障碍物距离激光雷达较远，导致点云较为稀疏。由于传统聚类算法的聚类参数固定，聚类结果将同一辆货车的车头和车身分割为两个不同的障碍物，如图 10 (a) 椭圆框所示。而改进的聚类方法能够很好地将距离较远处稀疏的障碍物点云进行聚类，从而将大货车表面的点云完整地还原成同一个障碍物，如图 10 (b) 椭圆框所示。由以上结果可知，改进的聚类算法能够有效避免点云分割过当问题，提高聚类准确性。

3.4 准确性与实时性分析

为了进一步验证本文所述障碍物检测算法的准确性

和实时性，按照前述障碍物检测算法流程对 200 帧机械式激光雷达的原始点云数据进行处理：首先进行点云预处理，然后分别采用传统的和改进的聚类算法进行聚类，再建立包络矩形得到障碍物检测结果，最后对检测结果的正检率 (%) 和处理时长 (ms) 进行对比分析，结果如表 3 所示。

表 3 两种聚类算法正检率与处理时间对比

聚类算法	传统聚类算法	改进的聚类算法
正检率/%	82.1	86.3
处理时间/ms	102	97

由表 3 可以看出，改进的聚类算法的障碍物正检率相比于传统聚类算法提高了 5.1%，同时，处理时间相比于传统聚类算法缩短了 4.9%，准确性与实时性均有提高。

4 结束语

障碍物检测是仓库用无人消防车功能架构中的重要环节，利用激光雷达三维点云进行障碍物检测能够有效提升无人消防车的智能化水平，提高灭火效率，及时有效防止灾害的进一步扩大。本文对基于机械式激光雷达点云的障碍物检测算法进行了分析和总结，对点云障碍物检测关键技术进行了研究。针对障碍物点云聚类的实时性要求和点云畸变问题，首先对原始点云进行预处理，剔除无效点并消除由车辆运动产生的畸变，然后通过改进的点云聚类算法对障碍物进行聚类，并建立最小包络矩形，从而实现障碍物检测。将算法用于原始点云数据处理，对其有效性进行了评估。结果表明，相比于传统算法，改进的聚类算法能够更加准确地检测出距离较近或较远的障碍物，障碍物检测的准确性和实时性均有所提升，在基于激光雷达点云的仓库用无人消防车障碍物检测中具有一定的应用价值。

参考文献：

[1] 向富平，周 雅，邓喜文，等. 基于 STM32 智能无人消防车的设计 [J]. 电脑知识与技术，2024，20 (5): 117 - 120.

[2] 郭亭亭，刘文涛，李希彬，等. 防爆型消防侦察机器人的研发与应用 [J]. 机电信息，2020 (8): 36 - 37.

[3] 贾 硕，张文昌，吴 航，等. 救援机器人研究现状及其发展趋势 [J]. 医疗卫生装备，2019，40 (8): 90 - 95.

[4] 陈 浩，陈 珺，刘 飞. 基于自主探索的移动机器人路径规划研究 [J]. 计算机工程，2025，51 (1): 60 - 70.

[5] JIANG W, CHEN W, SONG C, et al. Obstacle detection and tracking for intelligent agricultural machinery [J]. Computers and Electrical Engineering, 2023: 108.

[6] INDER K, SILVA D V, SHI X. Learning control policies

- of driverless vehicles from UAV video streams in complex urban environments [J]. Remote Sensing, 2019, 11 (23): 2723.
- [7] 郝雯, 王映辉, 宁小娟, 等. 面向点云的三维物体识别方法综述 [J]. 计算机科学, 2017, 44 (9): 11-16.
- [8] HAN J, KIM D, LEE M, et al. Enhanced road boundary and obstacle detection using a downward-looking LIDAR sensor [J]. IEEE Transactions on Vehicular Technology, 2012, 61 (3): 971-985.
- [9] 王世峰, 戴祥, 徐宁, 等. 无人驾驶汽车环境感知技术综述 [J]. 长春理工大学学报 (自然科学版), 2017, 40 (1): 1-6.
- [10] 钟琳, 蒋旭东, 张涛, 等. 消防机器人发展现状和自主行走关键技术探索 [C] //中国消防协会. 2022 中国消防协会科学技术年会论文集. 应急管理部上海消防研究所; 中国地震应急搜救中心, 2022: 8.
- [11] MICHAEL P, PEDRO C, ANTÓNIO S, et al. Obstacle detection for autonomous guided vehicles through point cloud clustering using depth data [J]. Machines, 2022, 10 (5): 332-332.
- [12] SHI T D, ZHONG D Y, BI L. A new challenge: detection of small-scale falling rocks on transportation roads in open-pit mines [J]. Sensors, 2021, 21 (10): 3548-3548.
- [13] 谢德胜, 徐友春, 王任栋, 等. 基于三维激光雷达的无人车障碍物检测与跟踪 [J]. 汽车工程, 2018, 40 (8): 952-959.
- [14] 樊轶铨, 张俊琪, 崔宸, 等. 点云预处理算法综述 [J]. 信息与电脑 (理论版), 2023, 35 (6): 206-209.
- [15] 王子洋, 李琼琼, 张子蕴, 等. 应用于无人驾驶车辆的点云聚类算法研究进展 [J]. 世界科技研究与发展, 2021, 43 (3): 274-285.
- [16] 余杭. 基于激光雷达的 3D 目标检测研究综述 [J]. 汽车文摘, 2024 (2): 18-27.
- [17] 徐红鑫, 裴志伟, 王道林, 等. 三维相机发展趋势综述 [J]. 信息系统工程, 2022 (6): 104-108.
- [18] 党亚南, 田照星, 郭利强. 车载激光雷达点云数据处理关键技术 [J]. 计算机测量与控制, 2022, 30 (1): 234-238.
- [19] LUDA Z, YI H, XING Y, et al. Robust multi-task learning network for complex LiDAR point cloud data preprocessing [J]. Expert Systems With Applications, 2024, 237 (PB): 1-18.
- [20] 陈强, 陈海波, 张沥化. 基于 NDT 配准与轮式里程计的激光雷达运动畸变补偿算法 [J]. 高技术通讯, 2024, 34 (1): 83-91.
- [21] 李玲星, 张小俊. 基于目标检测的激光雷达道路边缘检测算法 [J]. 科学技术与工程, 2024, 24 (9): 3519-3525.
- [22] YU H, CHEN L, YAO J, et al. A three-way clustering method based on an improved DBSCAN algorithm [J]. Physica A: Statistical Mechanics and its Applications, 2019, 535: 122289.
- [23] 王凯, 支煜, 陈浩, 等. 一种基于 Graham 扫描算法的空间点云结构化算法研究 [J]. 现代电子技术, 2018, 41 (14): 139-142.
- [24] SHENG X, RUI SHENG W, HAO W, et al. An optimal hierarchical clustering approach to mobile LiDAR point clouds [J]. IEEE Transactions on Intelligent Transportation Systems, 2019: 1-12.
- [25] 邹斌, 刘康, 王科未. 基于三维激光雷达的动态障碍物检测和追踪方法 [J]. 汽车技术, 2017 (8): 19-25.
- (上接第 196 页)
- [10] 杨亭, 常霞, 粘丹妮. 卫星遥测数据处理软件的 Excel VBA 实现 [J]. 计算机测量与控制, 2014, 22 (8): 2677-2679.
- [11] 任子强. 基于 OPC 技术实现 Wincc 归档数据的 Excel 报表自动生成 [J]. 电子制作, 2015 (22): 22-23.
- [12] 贺晓巧, 王建民, 赵晔, 等. 基于 OPC 和 Wincc 的磨机负荷控制系统 [J]. 计算机测量与控制, 2014, 22 (12): 3954-3956.
- [13] Siemens AG. 如何使用 Excel 通过 OPC 访问 WinCC 的实时数据 [EB/OL]. (2024-02-05) [2025-03-21]. <https://support.industry.siemens.com/cs/document/77467938/Industry-Support-Siemens>.
- [14] 谢光宇. 基于 WinCC WebUX 和 SQL 的制药生产信息化系统设计 [J]. 工业控制计算机, 2021, 34 (7): 133-135.
- [15] 段培永, 王玉红, 李慧. 利用 VB 实现 WinCC 归档数据处理 [J]. 计算机系统应用, 2013, 22 (10): 139-142.
- [16] 汪伟滨. WinCC 基于 VBS 脚本运用 Access 数据库和 Excel 制作的报表查询系统 [J]. 工业控制计算机, 2023, 36 (12): 66-67.
- [17] 强明辉, 张琳琳, 马永炜. 基于 OPC 技术的 WinCC 实时数据归档 [J]. 电气自动化, 2015, 37 (3): 21-23.
- [18] 赵晶晶, 袁庆一, 王伟. Access 数据库在试验软件中的应用 [J]. 工程与试验, 2016, 56 (4): 63-65.
- [19] 王璐烽. 深入探讨 Excel 考试系统定时器的实现方法 [C] //旭日华夏 (北京) 国际科学技术研究院. 首届国际信息化建设学术研讨会论文集 (一). 重庆工业职业技术学院, 2016: 291.
- [20] 朱财良. 基于 CATIA 二次开发的汽车产品数据处理 [J]. 装备制造技术, 2024 (3): 147-150.
- [21] 陈华. 西门子 SIMATIC WinCC 使用指南 (下册) [M]. 北京: 机械工业出版社, 2018.