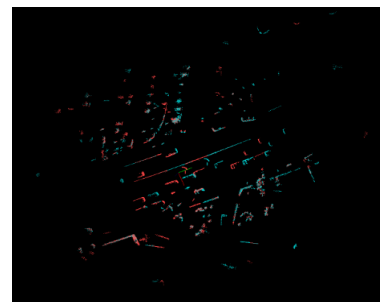




DOI: 10.12086/oe.2018.180266

一种城市环境三维点云配准的预处理方法

赵凯¹, 徐友春^{2*}, 王任栋¹¹陆军军事交通学院, 天津 300161;²军事交通运输研究所, 天津 300161

摘要: 针对城市三维环境下 LiDAR 点云数据密度大、离群噪点多、分布散乱不利于后期点云帧间匹配的问题, 提出一种应用于城市环境下大规模三维 LiDAR 点云帧间匹配的预处理方法。首先, 将点云数据转化为均值高程图, 利用网格之间的高度梯度对点云进行地面分割处理; 然后, 通过三维体素栅格划分的方法改进了 DBSCAN 聚类算法, 用改进后的 VG-DBSCAN 对点云进行聚类, 聚类后目标点云与离群点分离, 从而剔除点云中的离群噪点; 最后, 采用 Voxel Grid 滤波器对点云降采样。实验结果表明, 所提方法可以对点云数据进行实时的预处理, 平均耗时为 132.1 ms; 预处理之后点云帧间匹配的精确度提高了 2 倍, 平均耗时也仅为预处理前的 1/6。

关键词: 点云预处理; 地面分割; 点云去噪; 帧间匹配

中图分类号: O436.3

文献标志码: A

引用格式: 赵凯, 徐友春, 王任栋. 一种城市环境三维点云配准的预处理方法[J]. 光电工程, 2018, 45(12): 180266

A preprocessing method of 3D point clouds registration in urban environments

Zhao Kai¹, Xu Youchun^{2*}, Wang Rendong¹¹Army Military Transportation University, Tianjin 300161, China;²Institute of Military Transportation, Tianjin 300161, China

Abstract: Aiming at the problem that 3D LiDAR point cloud has high data density, outlier noise, and scattered distribution in urban environment, which is not conducive to the matching between point clouds in the later stage, a pre-processing method for large-scale LiDAR point cloud frame matching in urban environments is proposed. First, the point cloud data is transformed into a Mean Elevation Map, and the ground point segmentation processing is performed on the point cloud using the height gradient between the grids; then, the DBSCAN clustering algorithm is improved by the three-dimensional voxel grid division method, and the improved VG-DBSCAN is used to cluster point clouds and separate the target point cloud from the outliers after clustering, thereby, which eliminates outlier noises in the point cloud. Finally, the Voxel Grid filter is used to down sample the point cloud. The experimental results show that the proposed method can perform real-time preprocessing on point cloud data, and the average time is 132.1 ms. After pre-processing, the accuracy of point cloud frame matching is increased by 2 times, and the average time consumption is only 1/6 before pre-processing.

收稿日期: 2018-05-21; 收到修改稿日期: 2018-09-12

基金项目: 国家重点研发计划(2016YFB0101001-6)

作者简介: 赵凯(1994-), 男, 硕士研究生, 主要从事智能车激光雷达定位方面的研究。E-mail: zhikai929@126.com

通信作者: 徐友春(1972-), 男, 博士, 博士生导师, 主要从事智能车辆技术方面的研究。E-mail: xu56419@126.com

Keywords: point cloud preprocessing; ground segmentation; point cloud density clustering; inter-frame matching

Citation: Zhao K, Xu Y C, Wang R D, *et al.* A preprocessing method of 3D point clouds registration in urban environments[J]. *Opto-Electronic Engineering*, 2018, 45(12): 180266

1 引言

城市道路环境以其复杂、多变的特点成为当前自动驾驶领域的研究热点和难点^[1-2]。点云配准是实现障碍物检测与跟踪、高精度地图绘制、同时定位与地图创建(simultaneous localization and mapping, SLAM)等功能不可或缺的技术手段。然而在城市三维环境下,多线 LiDAR 扫描得到的三维点云规模庞大,包含数以万计的数据点,而且其中存在着大量的离群噪点,这些因素大大增加了城市环境中 LiDAR 点云配准的难度,严重影响着点云配准的准确性和实时性。因此,实时高效的三维点云数据预处理方法成为目前研究的首要目标。点云数据预处理的目的是为三维 LiDAR 点云的帧间匹配提供噪点少、规模小的两帧点云数据,以此来保证匹配的准确性和实时性。

城市环境下三维激光雷达扫描得到的点云含有大量地面点,然而地面点云数据的特征不明显,反倒增加了点云规模。这对后续的帧间匹配没有作用甚至带来麻烦,所以需要首先将其滤掉。目前,点云地面分割已被广泛用于自动驾驶汽车的环境感知部分,而且针对不同的应用场合以及不同的传感器提出了多种地面分割方法。Biosca 等^[3]由邻近估计点的法向量提取局部点特征,利用欧氏聚类 and 区域增长算法快速分割地面和非地面,但是选取不同种子点将得到不同的分割区域,导致鲁棒性较差。Himmelsbach 等^[4]将三维数据以极坐标的方式栅格化,并对每个扇形区域建立基于直线拟合的非参数化地面模型,用非参数的方法分离地面与非地面。Moosman 等^[5]提出一种新的局部凸面标准用于分割地面与非地面,其特点是能充分利用所有扫描点的三维信息,但是计算量较大。张名芳等^[6]提出了一种基于三维不规则点云的地面分割算法。在构建连接图的基础上建立了相应的测量代价和平滑性代价函数模型,得到地面高度最优估计值,实现地面点与非地面点的分割,但对于大规模场景该方法计算量较大。

另一方面,由于传感器的局限性、采集设备的固有噪声、场景中的物体表面的反射特性等而使点云数据不可避免地产生稀疏的离群点。噪声点的存在严重影响特征提取、特征匹配,因此,有必要对原始点云

去噪滤波,以获得适合进一步处理的精确点云。Fleishman^[7]等人将双边滤波进行改进拓展到三维网格模型上,提出针对网格降噪的双边滤波技术,该算法具有良好的降噪效果,且能够保持点模型的几何特征,但不能很好地对离群点加以处理,易削去模型最尖锐部分。李仁忠^[8]等人提出一种基于方法库的点云模型去噪与精简算法,依据噪声点离模型主体的距离,利用统计滤波结合半径滤波去除大尺度噪声。苏本跃^[9]等人用 K-means 聚类算法对点云进行聚类,对每个类中的点,根据点到聚类中心的欧氏距离和邻近点曲率变化判断是否为噪声点,进而去除大量外部噪点。然而上述方法均不适用于大规模散乱点云的去噪处理,而且时间开销大。

鉴于上述算法的局限性,本文提出了一种针对大规模散乱点云匹配的预处理方法。该方法首先利用基于均值高程图的方法去除了点云数据中的地面点。然后,通过改进的聚类算法 VG-DBSCAN 聚类目标点,实现目标点和离群点的分离,达到滤除离群点的目的。最后,采用 Voxel Grid 滤波器对点云进行降采样,减小点云的密度。该方法在去除离群噪点和冗余信息降低点云数量规模的同时,又能保留点云的特征信息,不损坏点云的细节信息。预处理提高了后续点云匹配的精确度,减少了时间开销。

2 地面分割

地面分割作为一项重要的预处理步骤,目的将三维点云分成表示地面点和非地面点的两个独立子集,去除点云数据中对于匹配而言冗余的地面点。为此,本文提出一种基于网格平均高度值的分割方法。该方法的具体步骤如下:

Step 1) 将点云数据转化为均值高程图^[10]。按照设定的网格大小 L_c ,将点云划分为二维网格,并且计算每个网格中点云数据的平均高度值。

Step 2) 计算每个网格的高度梯度。如图 1 所示,网格 C 与它相邻的四个网格的高度值做差,保留最大的插值作为该网格 C 的梯度。

Step 3) 设定阈值 g_{Thresh} 对所有网格进行划分。高度梯度大于 g_{Thresh} 的标记为障碍物,小于 g_{Thresh} 的标记

为候选地面。

Step 4) 计算所有标记为地面的网格构成的连通域，面积最大的作为参考地面。将其他连通域和参考地面的高度值进行比较，如果高于参考地面(超过设定的阈值 h_{Thresh})就把这些地面网格重新标记为障碍物。

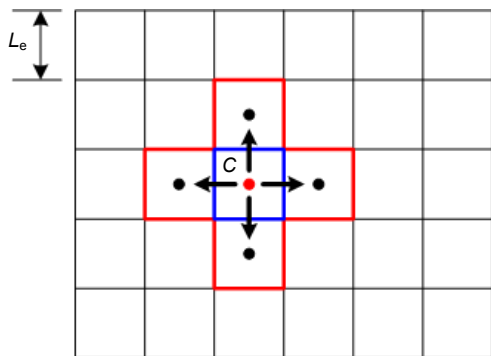


图 1 网格邻域
Fig. 1 Grid neighborhood

Step 5) 校正梯度计算过程中产生的方法误差。如图 2 所示，在进行高度梯度计算时，如果相邻两个网格的高度差(图中蓝色虚线)恰好大于它们各自与邻域内其他网格的高度差值时，这两个网格会具有相同的高度梯度(图中实线箭头)，这就会导致中间的网格被标记为非地面，而事实上它应该被标记为候选地面。校正方法如下：对于每个非地面单元网格 B ，找到相邻的地面网格并计算它们的平局高度差，如果 B 的自身高度接近相邻网络的平均高度值，即它们的高度差小于设定的阈值 h_{Thresh} ，则将 B 标记为地面单元。

Step 6) 重新考察标记为障碍且与地面相邻的网格。由于 Step 5)校正了部分被误划分为障碍物的网格，重新划分其为地面网格，这时应跳转至 Step 4)重新计

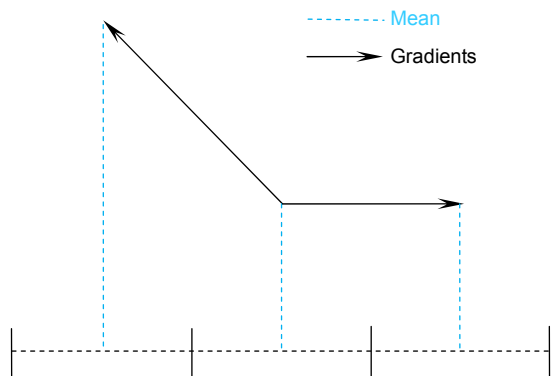


图 2 方法误差产生原因分析
Fig. 2 Analysis of the causes of method errors

算被标记为地面的所网格构成的连通域。

至此，三维点云数据被划分为地面点和非地面点两个部分，去除地面点部分即实现点云的地面分割。

3 离群噪声滤除

在原始的点云数据中，除了目标点云外，还有一些分布比较稀疏或孤立的噪声点。而且经过地面分割之后的点云，由于分割误差，导致一些地面点未被去除而变成离群噪声点。聚类算法可以通过选择合适的阈值将目标点云与离群点分离，达到去除离群点的目的。因此，为了减少离群点对点云匹配产生的不良影响，本文提出一种基于体素栅格的快速密度聚类算法(voxel grid-based fast DBSCAN, VG-DBSCAN)对点云进行去除离群噪声点处理。

3.1 DBSCAN 算法原理

DBSCAN(density-based spatial clustering of application with noise)算法将簇定义为密度相连的点的最大集合，能够把具有足够高密度的区域划分为簇，并可在存在噪声的数据中发现任意形状的聚类，其显著优点是聚类速度快且能有效处理噪声点^[11]。在点云数据的处理中，由于噪声的数目、分布均未知，所以本文将 DBSCAN 算法引入到点云的去噪处理中，利用密度聚类方法对三维点云进行聚类，从而在聚类结果中选择出目标点云主体，实现噪声的去除。

DBSCAN 算法的核心思想总结如下：

该算法有两个参数，即 l_d 和 n_{minpts} ，其中 l_d 表示目标数据点搜索邻域半径， n_{minpts} 表示最小邻域点数。在图 3 中，如果 p 点的 l_d 邻域至少有 n_{minpts} 个点，则 p 被称为核心点。搜索区域中的点将重复聚类，直到只剩下 l_d 邻域内点数小于 n_{minpts} 的数据点。点 b 在点 c

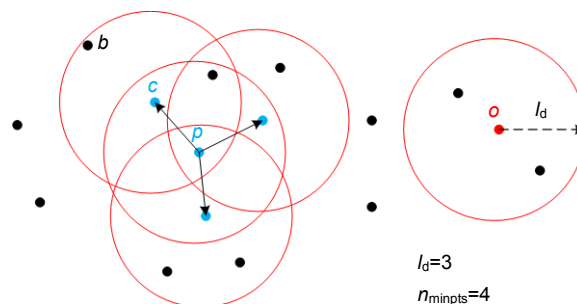


图 3 DBSCAN 算法核心思想
Fig. 3 The core idea of DBSCAN algorithm

的 l_d 邻域内,但点 b 的 l_d 邻域内点数小于 n_{minpts} ,因此, b 点被标记为边界点(b 不是核心点,但落在某个核心点的 l_d 邻域内)。点 o 邻域点数少于 n_{minpts} ,因此不能成为新的簇。

3.2 基于体素栅格的快速密度聚类算法

在 DBSCAN 和基于相对密度的聚类方法中,每个对象必须与数据集中的众多其他对象进行比较。为了减少执行时间,本文对 DBSCAN 算法进行了改进,提出基于三维体素栅格的密度聚类算法 VF-DBSCAN。该算法将三维点云数据按照维度划分为以体素栅格为单元的多个相邻的区间,创建一个由栅格单元组成的集合,以此大大减少每个对象在数据空间中邻域的搜索范围,只要考察当前对象的空间相邻栅格单元就能实现其邻域扫描,快速发现各个聚类。

VG-DBSCAN 算法的具体步骤如下:

Step 1) 划分体素栅格

如图 4(a)所示,构建一个三维体素栅格,并将三维 LiDAR 点云数据的每个数据点划分给它所在的栅格单元。每个单元栅格的体对角线长度是 l_d ,这意味着栅格单元的边长为 $l_d/\sqrt{3}$,如图 4(b)。如此设定单元大小的重要意义在于如果栅格内的点数大于 n_{minpts} ,则能够直接将所有这些包含于该体素栅格单元的标记为核心点。因为体素栅格内任何两点在栅格内的最大距离小于等于 l_d ,所以如果体素栅格内有大于 n_{minpts} 数量的数据点,那么单元内任何点的 l_d 邻域至少包含 n_{minpts} 个点。

在构建栅格之前,首先需要确定六个值,即三维点云数据 X, Y, Z 轴的最大和最小坐标值。通过这六个值和体素栅格边长可以确定栅格的行数 n_{rows} 、列数 n_{cols} 和层数 n_{lays} 。然后为每个数据点计算它所属的体

素栅格。对于体素栅格边界上的点,我们将其分配在右上方的栅格中。如果有一个数据点正好位于栅格的最顶部或最右边的体素栅格边界上,这时应该添加一个额外的行、列、层用以囊括该点,防止将其分配到栅格外部。

对角线长度 l_d 在大城市环境下的三维点云中一般取为 0.8~1.2 较为合适,这一栅格大小可以有效涵盖大部分目标障碍物的尺寸。参数 n_{minpts} 的大小与参数 l_d 的大小相关,当 l_d 取 0.8~1.2 时, n_{minpts} 取 10~20。参数 n_{minpts} 太小时容易将噪声点聚类为目标点云,太大则去噪过度,破坏相关目标障碍物的特征信息。

Step 2) 确定核心点

这一步将遍历所有非空体素栅格。要查找栅格内的所有核心点,首先必须检查该单元格的点数是否大于 n_{minpts} 。如果是,则将所有这些点标记为核心点。如果不是,则必须检查该单元格的每个点,并确定它是否是核心点。要确定点 $p (p \in C)$ 是否是核心点,需要检查该点所在栅格的相邻栅格 $N_d(C)$ 内的一些点。 $N_d(C)$ 定义为某个体素栅格单元 C 及其相邻的栅格单元 C' 的集合,即 $N_d(C) = \{C, C' | dist(C, C') \leq l_d\}$ 。

图 5 为体素栅格某层的二维示意图,由此可以看到,对于三维体素栅格,最多需要检查 124 个栅格单元。计算这些栅格中每个点与 p 之间的距离。当这些单元栅格内的所有点已经被检查过,或者当发现 p 的 l_d 邻域内的点数大于等于 n_{minpts} 时,终止查找。

Step 3) 合并簇

如果两个不同单元中的两个核心点之间的距离小于等于 l_d ,则这两个点属于同一个簇。举个简单的例子,如图 6 所示,相邻的两个栅格单元各含有四个数据点,假设 $n_{minpts} = 3$,则两个栅格单元内的所有点都

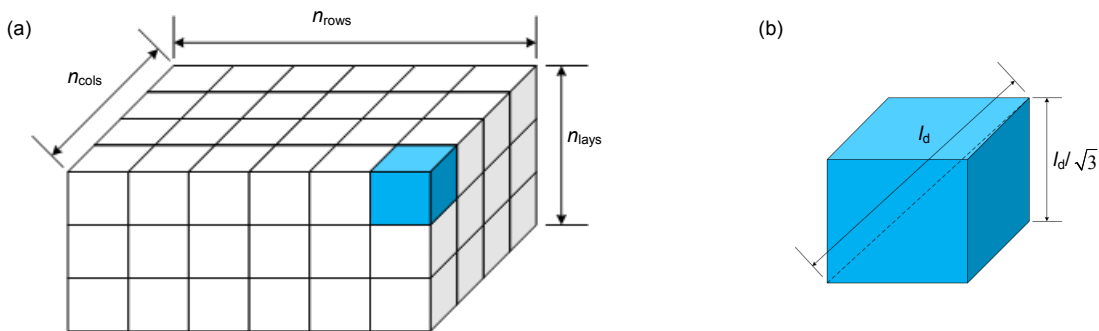


图 4 划分三维体素栅格。(a) 三维体素栅格; (b) 栅格单元
Fig. 4 Dividing a three-dimensional voxel grid. (a) Three-dimensional voxel grid; (b) Grid unit

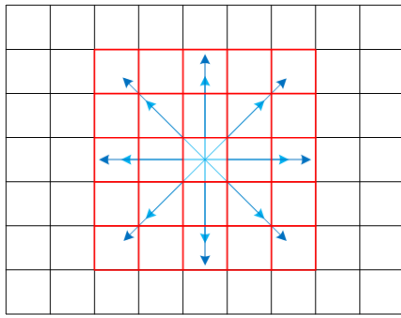


图5 体素栅格的某层的二维示意图

Fig. 5 Two-dimensional illustration of a layer in a voxel grid

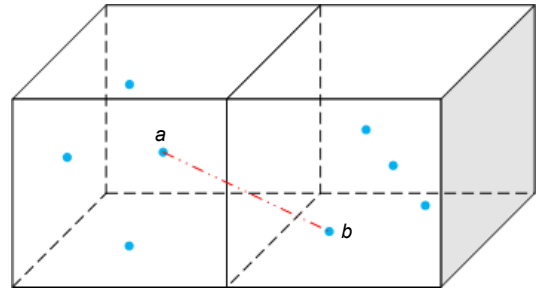


图6 合并簇的简例

Fig. 6 A brief example of a merged cluster

是核心点。如果 a 和 b 之间的距离小于等于 l_d ，则可以得出结论，所有这八个点都属于同一个簇。否则，划分为两个各含有四个点的簇。

Step 4) 确定边界点和离群噪点

最后一步是确定边界点和离群噪点。这一步通过遍历所有非核心点来完成。如果在 $N_{l_d}(C)$ 中存在至少一个核心点并且最靠近 p 的核心点属于簇 clusters_1，则非核心点 p 是簇 clusters_1 的边界点。完成上述四个步骤之后，所有不是核心点或边界点的数据点都标记为离群噪点。

4 点云降采样

经过地面分割和去噪处理之后的三维 LiDAR 点云的数量规模以及点云密度依旧相当大，大量的数据对环境有更加精确的描述，但同时也给算法的运算效率带来了巨大的负担。因此，点云数据预处理的第三步是数据帧的降采样。本文中点云数据主要用于点云帧间匹配，可以在不影响环境特征表示的前提下适当减少点云密度以提高算法效率。

本文采用 Voxel Grid 滤波器^[12]对点云进行降采样，即减少点云的数量规模，同时保持点云的障碍物特征。基本思路是将点云数据在三维体素栅格化的基础上(见第 3.2 节)，用体素栅格内所有点的重心来表示该栅格的环境数据。对于一个含有 N 个点的三维体素，经过 Voxel Grid 滤波器处理后得到表示环境数据的点的计算公式为

$$x = \frac{1}{N} \sum_{i=1}^N x_i, \quad y = \frac{1}{N} \sum_{i=1}^N y_i, \quad z = \frac{1}{N} \sum_{i=1}^N z_i。$$

5 实验结果与分析

为了验证本文所提方法的正确性和有效性，采用

Velodyne HDL-64E 型 64 线激光雷达所采集的城市复杂动态环境下的三维点云数据为研究对象。实验平台为 Intel(R) Core(TM) i7-6700 CPU @3.40 GHz, 16 GB ARM, Unbuntu 16.04 操作系统, Qt Creator5.7 开发环境, 开发语言为 C++。

首先，选取一帧具有代表性的点云对其进行地面分割处理，如图 7(a)所示，所选取的点云中大量的动态障碍物，如机动车、非机动车、行人等，可以代表城市复杂动态环境下的三维 LiDAR 点云。分割结果如图 7(b)、图 7(c)所示。图 7(b)、图 7(c)是地面分割后的整体效果图，可以看出地面点和非地面点能够被正确区别划分，整体分割效果良好；图 7(d)、图 7(e)为局部分割细节结果(图 7(c)中的框选部分)，可以看出该方法有效地避免了过分割和欠分割问题，可以精确地分离障碍物与地面。

经过地面分割之后，再对点云进行滤波去噪处理，这一部分，以开源点云库 PCL-1.8.1.dev 中具有代表性的统计滤波去噪算法和半径滤波去噪算法进行对比实验。如图 8(a)所示，选取一帧经过地面分割处理的点云为对象，滤波结果如图 8(b)、图 8(c)、图 8(d)和表 1 所示。从实验结果看出，三种方法均能在去除离群噪点的同时保留障碍物的局部特征，并且在一定程度上降低了点云的规模。但从整体效果可以看出，本文方法相比于其他两种代表性方法，对点云去噪、简化的效果更好，并且耗时最少。图 9 截取了图 8(d)中的局部处理效果，可以看出所提算法能够很好地聚类主要的目标物体，使其与噪声点分离，达到了滤除离群噪声点的目的。

经过 VG-DBSCAN 方法滤除离群噪点之后，再对点云进行降采样处理。以图 8(a)的点云为对象，体素栅格大小 $V_{leafsize} = 0.6 \times 0.6 \times 0.6$ ，降采样的结果如图 10

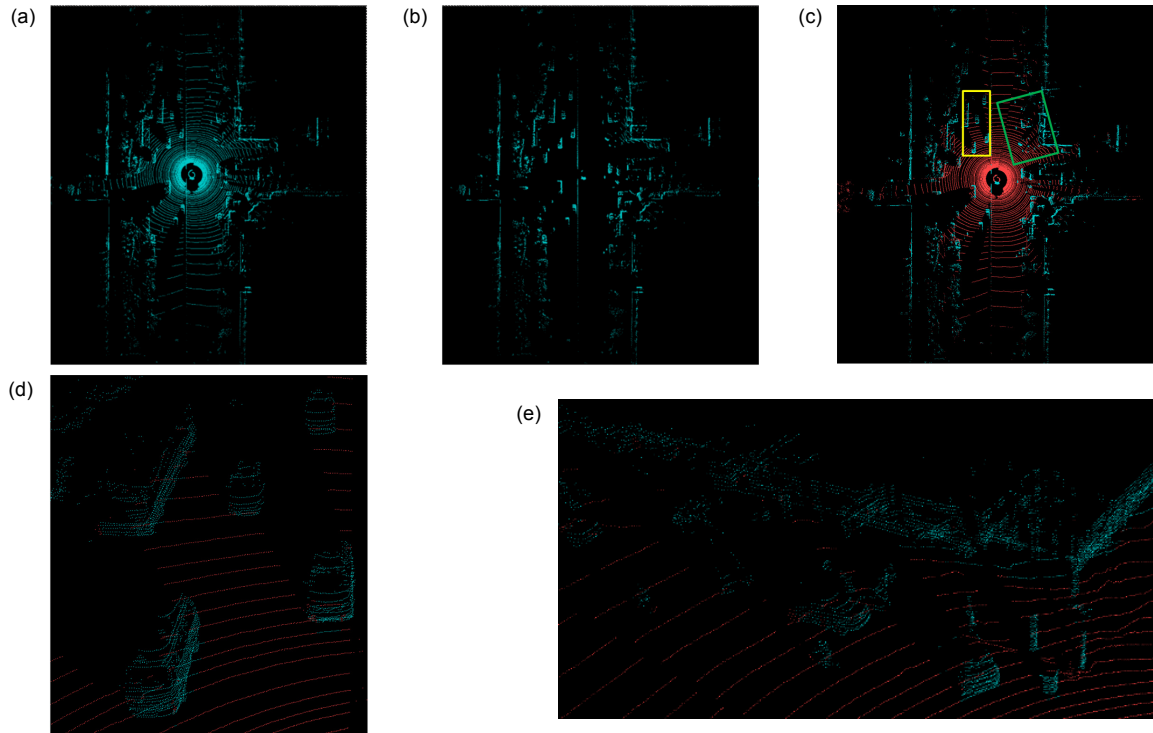


图 7 地面分割结果。(a) 地面分割前; (b) 地面分割后; (c) 分割整体效果; (d) ①号框框选部分; (e) ②号框框选部分

Fig. 7 Ground segmentation results. (a) Before the ground division; (b) After the ground division; (c) Division of the overall effect; (d) frame selection part; (e) frame selection part

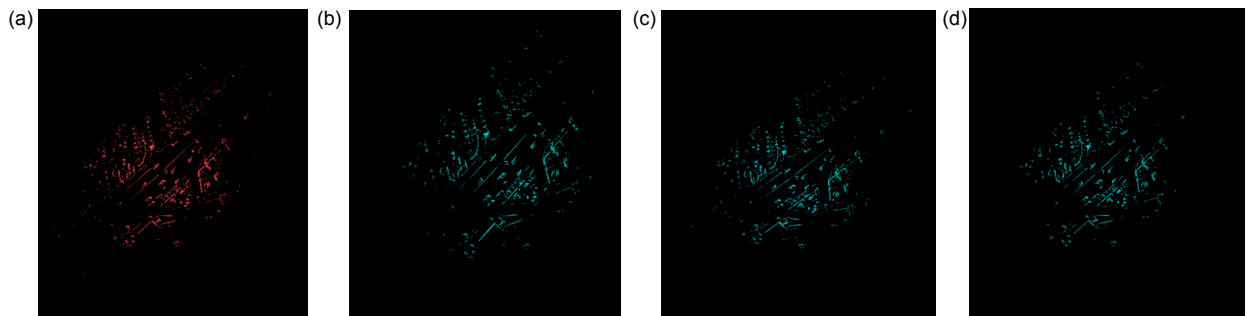


图 8 点云去除离群噪点结果。(a) 去噪前点云; (b) 统计滤波去噪; (c) 半径滤波去噪; (d) VG-DBSCAN 聚类滤波

Fig. 8 Point clouds remove outlier noise results. (a) Point cloud before denoising; (b) Statistical filter denoising; (c) Radius filtering denoising; (d) VG-DBSCAN clustering filter

表 1 三种滤波方法结果对比

Table 1 Comparison of the results of the three filtering methods

Method	Point size			Consuming time/ms
	Original	After segmentation	After denoised	
Statistical outlier removal			36529	180.23
Radius outlier removal	42618	40546	37487	274.37
VG-DBSCAN			27953	124.69

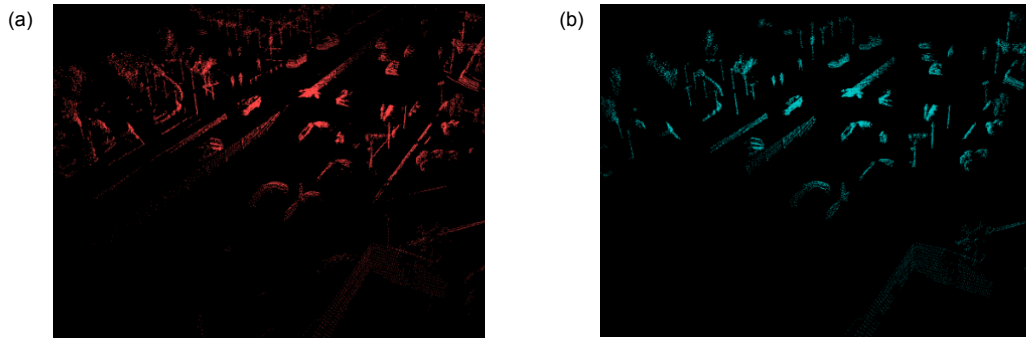


图 9 VG-DBSCAN 滤波去噪局部效果。(a) 去噪前; (b) 去噪后
Fig. 9 VG-DBSCAN filter denoising local effect. (a) Before denoising; (b) After denoising

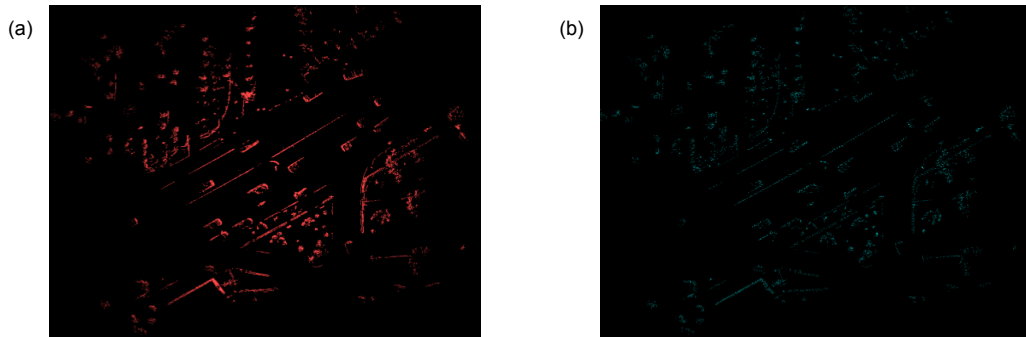


图 10 降采样前后对比
Fig. 10 Comparison before and after downsampling

所示。可以看出，基于 Voxel Grid 滤波器的降采样可以完好地保留点云的局部特征信息，并且降低了点云的密度。

表 2 总结了连续 100 帧点云经过地面分割、离群噪点滤除、降采样的预处理后点云数量、耗时以及简化率的变化情况。其中，简化率定义为在本次操作中点云数量的减少量占点云原始规模数量的百分比。从表中可以看出预处理过程能将点云的规模降低到原来的 8.17%，而且这一过程的耗时仅为 132.1 ms，可以达到实时处理的要求。

为了验证点云预处理操作对点云帧间匹配的改善

效果，使用开源库 PCL-1.8.1dev 中的匹配算法 GICP(generalized iterative closest point)对 100 帧连续点云预处理前后的帧间匹配进行了对比实验。实验结果如图 11 和表 3 所示。

将欧氏适应度^[12](Euclidean fitness score)作为点云匹配的准确度衡量标准，欧氏适应度为参与匹配的两帧点云中各对应点对之间欧氏距离的平方和。表 3 中的结果表明，经过预处理的点云在规模上得到大幅度简化，而且基本滤除了点云中包含的所有离群噪点，这使得预处理之后点云帧间匹配的精确度提高了 2 倍，平均耗时也仅为预处理前的 1/6。

表 2 点云预处理过程
Table 2 Point cloud pretreatment process

Preprocessing	Remaining point size (original size: 42618)	Consuming time/ms	Simplification rate/%
Segmentation	40546	5.27	4.86
Denoising	27953	124.69	29.55
Downsampling	3484	2.14	57.41

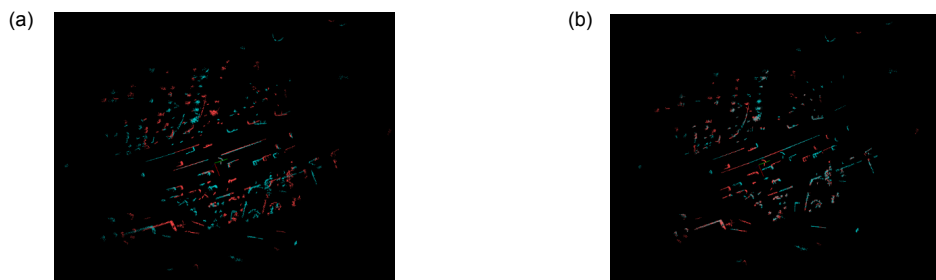


图 11 帧间匹配前后对比

Fig. 11 Comparison before and after interframe matching

表 3 预处理前后点云匹配结果

Table 3 Point cloud matching results before and after pre-processing

Preprocessing (with or without)	Point size		Euclidean fitness score	Consuming time/s
	Current frame	Previous frame		
Without	42618	44104	5.66952	18.752
With	3484	3509	2.41527	3.215

6 结 论

提出了一种应用于城市复杂环境三维 LiDAR 点云匹配的预处理方法。该方法包含地面分割、离群噪声滤除以及降采样三部分。路面分割方法将点云转化为均值高程图，利用网格之间的梯度差异对网格进行划分，进而准确地分离地面点与非地面点。然后采用三维体素栅格划分的形式对 DBSCAN 算法进行改进，改进后的方法用于滤除点云中的离群噪声点，通过对比实验证明该算法在点云去噪、简化以及耗时方面优于现有的典型方法。基于 Voxel Grid 滤波器的降采样方法通过采用体素栅格内所有点的重心替代该栅格的方法，大幅度降低了点云的规模数量。上述三个处理过程，在降低点云规模的同时又能完好地保留非地面点的几何特征，保证特征信息不会随着点云数量的减少而缺失，而且预处理过程耗时短，可以实现实时处理。将预处理方法应用于点云帧间匹配，不仅可以滤除两帧点云中大量的离群噪声点，而且大幅度降低了两帧点云的规模。实验证明，本文所提的预处理方法可以显著提高匹配的准确度、减少匹配耗时。

参考文献

[1] Kim J U, Kang H B. LiDAR Based 3D object detection using CCD information[C]//IEEE Third International Conference on Multimedia Big Data, 2017: 303-309.
 [2] Han D B, Xu Y C, Li H, et al. Calibration of extrinsic parameters for three - dimensional lidar based on hand - eye model[J]. *Opto-Electronic Engineering*, 2017, **44**(8): 798-804.
 韩栋斌, 徐友春, 李华, 等. 基于手眼模型的三维激光雷达外参数标定[J]. *光电工程*, 2017, **44**(8): 798-804.

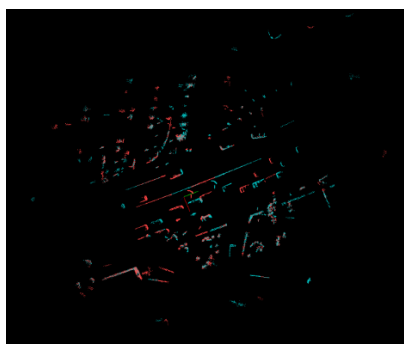
[3] Biosca J M, Lerma J L. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods[J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2008, **63**(1): 84-98.
 [4] Himmelsbach M, Hundelshausen F V, Wuensche H J. Fast segmentation of 3D point clouds for ground vehicles[C]// *Proceedings of 2010 IEEE Intelligent Vehicles Symposium*, 2010: 560-565.
 [5] Moosmann F, Pink O, Stiller C. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion[C]// *Proceedings of 2009 IEEE Intelligent Vehicles Symposium*, 2009: 215-220.
 [6] Zhang M F, Fu R, Guo Y S, et al. Road segmentation method based on irregular three dimensional point cloud[J]. *Journal of Jilin University (Engineering and Technology Edition)*, 2017, **47**(5): 1387-1394.
 张名芳, 付锐, 郭应时, 等. 基于三维不规则点云的地面分割算法[J]. *吉林大学学报(工学版)*, 2017, **47**(5): 1387-1394.
 [7] Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising[J]. *ACM Transactions on Graphics*, 2003, **22**(3): 950-953.
 [8] Li R Z, Yang M, Ran Y, et al. Point cloud denoising and simplification algorithm based on method library[J]. *Laser & Optoelectronics Progress*, 2018, **55**(1): 011008.
 李仁忠, 杨曼, 冉媛, 等. 基于方法库的点云去噪与精简算法[J]. *激光与光电子学进展*, 2018, **55**(1): 011008.
 [9] Su B Y, Ma J Y, Peng Y S, et al. Algorithm for RGBD point cloud denoising and simplification based on K-means clustering[J]. *Journal of System Simulation*, 2016, **28**(10): 2329-2334, 2341.
 苏本跃, 马金宇, 彭玉升, 等. 基于 K-means 聚类的 RGBD 点云去噪和精简算法[J]. *系统仿真学报*, 2016, **28**(10): 2329-2334, 2341.
 [10] Siciliano B, Khatib O. *Springer Handbook of Robotics*[M]. Berlin, Heidelberg: Springer-Verlag, 2007.
 [11] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise[C]//*Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996: 226-231.
 [12] Rusu R B, Cousins S. 3D is here: Point Cloud Library (PCL)[C]// *Proceedings of IEEE International Conference on Robotics and Automation*, 2011: 1-4.

A preprocessing method of 3D point clouds registration in urban environments

Zhao Kai¹, Xu Youchun^{2*}, Wang Rendong¹

¹Army Military Transportation University, Tianjin 300161, China;

²Institute of Military Transportation, Tianjin 300161, China



Two-frame point cloud registration result

Overview: Aiming at the problem that LiDAR point cloud data density is high in urban 3D environment, there are many outlier noises, and the scattered distribution is not conducive to the matching of point clouds in the later period, a preprocessing method for 3D LiDAR point cloud matching in urban complex environments is proposed. The method includes three parts: ground segmentation, outlier noise filtering, and downsampling.

The road surface segmentation method converts the point cloud into a mean elevation map, uses the gradient difference between the grids to divide the grid, and then accurately separates ground points from non-ground points. Then, the DBSCAN algorithm is improved by using a three-dimensional voxel grid partitioning method. The improved algorithm divides the three-dimensional point cloud data into multiple adjacent segments with a voxel grid as a unit according to dimensions, and creates a grid cell. The set of components, which greatly reduces the search scope of each object in the data space neighborhood, as long as the current object's spatially adjacent grid cells can be scanned to achieve its neighbors, rapid discovery of each cluster. The comparison experiments show that the proposed algorithm is superior to the existing typical methods in point cloud denoising, simplification and time-consuming. After the ground segmentation and denoising, the number of three-dimensional LiDAR point clouds and the point cloud density are still quite large. A lot of data describe the environment more accurately, but at the same time it also imposes a huge burden on the computational efficiency of the algorithm. Therefore, the third step of the preprocessing of point cloud data is downsampling of the data frame. In this paper, point cloud data is mainly used for point cloud inter-frame matching, and the point cloud density can be appropriately reduced to improve the efficiency of the algorithm without affecting the representation of environmental features. The downsampling method based on the Voxel Grid filter greatly reduces the size of the point cloud by replacing the grid with the center of gravity of all points in the voxel grid.

The above three processes can preserve the geometric characteristics of the non-ground point cloud while reducing the size of the point cloud, ensuring that the feature information will not be lost as the number of point clouds decreases, and the preprocessing process will take a short time. Realize real-time processing. Applying the pre-processing method to point cloud inter-frame matching can not only filter out a large number of outlier noises in the two-frame point cloud, but also significantly reduce the size of the two-frame point cloud. Experimental results show that the pretreatment method proposed in this paper can significantly improve the accuracy of matching and reduce the time-consuming matching.

Citation: Zhao K, Xu Y C, Wang R D. A preprocessing method of 3D point clouds registration in urban environments[J]. *Opto-Electronic Engineering*, 2018, 45(12): 180266

Supported by National Key R&D Plan (2016YFB0101001-6)

* E-mail: xu56419@126.com