

## 基于改进多规则区域生长的点云多要素分割

汪文琪, 李宗春\*, 付永健, 何华, 熊峰

中国人民解放军战略支援部队信息工程大学地理空间信息学院, 河南 郑州 450001

**摘要** 针对现有点云多要素分割算法分割面状点集时分割精度低、分割块合并效果差等问题, 提出了一种改进的多规则区域生长算法。一方面, 计算点云数据的平面拟合残差, 基于平面拟合残差设置种子点条件, 对面状点集分割进行优化, 以此提升面状要素分割的精度; 另一方面, 在距离条件的基础上, 结合相似性和体积变化条件对合并策略进行改进, 以实现分割块的有效合并; 此外, 利用中位数、Baarda 数据探测法和  $k$  均值聚类分别对算法中涉及的阈值参数进行自适应设置。采用三种不同类型的点云数据进行实验, 结果表明: 改进算法能够提升面状点集的分割精度, 提高了分割块合并的准确性; 与其他算法相比, 改进算法能够同时兼顾精度和效率, 分割结果更具优势。

**关键词** 遥感; 点云分割; 多要素分割; 平面拟合残差; 分割块合并

中图分类号 P237

文献标志码 A

doi: 10.3788/AOS202141.0528001

## Multi-Factor Segmentation of Point Cloud Based on Improved Multi-Rule Region Growing

Wang Wenqi, Li Zongchun\*, Fu Yongjian, He Hua, Xiong Feng

*Institute of Geospatial Information, PLA Strategic Support Force Information Engineering University, Zhengzhou, Henan 450001, China*

**Abstract** With regard to the low segmentation accuracy of planar point sets and poor merging effect of segments in the existing multi-factor segmentation algorithms of point clouds, an improved multi-rule region growing algorithm was proposed in this paper. On one hand, the plane fitting residuals of point clouds were calculated, based on which, the seed condition was set and the segmentation of planar point sets was optimized, so as to increase the segmentation accuracy of planar factors. On the other hand, on the basis of the distance condition, the merging strategy was improved in combination with similarity and volume changes to achieve effective merging of segments. In addition, the threshold parameters involved in this algorithm were set adaptively using the median clustering, Baarda data snooping, and  $k$ -means clustering. Furthermore, three different types of point clouds were tested, and the results show that the improved algorithm can boost the segmentation accuracy of planar point sets, and enhance the veracity of segments merging. Compared with other algorithms, the proposed algorithm can take into account both accuracy and efficiency and has better segmentation results.

**Key words** remote sensing; point cloud segmentation; multi-factor segmentation; plane fitting residuals; segments merging

**OCIS codes** 280.3420; 280.3640; 200.4560

## 1 引言

随着三维激光扫描测量技术的迅猛发展, 点云数据处理成为了研究热点<sup>[1]</sup>。点云分割是点云数据

处理的重要环节, 在建筑物提取<sup>[2]</sup>、道路标线识别<sup>[3]</sup>、森林资源调查<sup>[4]</sup>、农业生产<sup>[5]</sup>等方面应用广泛。常见的点云分割方法包括模型拟合法<sup>[6-7]</sup>、特征聚类法<sup>[8-9]</sup>和区域生长法<sup>[10-11]</sup>。模型拟合法常被用

收稿日期: 2020-07-13; 修回日期: 2020-09-09; 录用日期: 2020-10-21

\* E-mail: 13838092876@139.com

于提取点云数据中平面、柱面和球体等规则形状的要害,该方法需要模型先验信息,难以适用于复杂场景的点云分割。特征聚类法是依据特征的相似度进行聚类,其优点是能够自动进行点云分割,特征构建方式灵活多样,但该方法易受噪声影响,聚类过程较为耗时。区域生长法是利用点云数据之间的邻接关系与相关属性的相似性进行分割,该算法的核心在于种子点的选用以及点与点之间生长条件的确定。

目前点云分割的研究多集中于场景中单一类别物体的识别和提取,如车辆<sup>[12]</sup>、建筑物<sup>[13]</sup>的提取等。点云场景中通常包含植被、建筑、地面等多种要素,种类复杂多样,随着点云多目标识别提取需求的增长<sup>[14]</sup>以及基于对象基元的点云分类<sup>[15]</sup>、融合多点集特征<sup>[16]</sup>等方法的进步,相关分割方法需进一步发展,以实现点云场景的多要素分割。文献[17]将机载点云数据分为平面点和非平面点分别进行分割,场景中大部分地物要素能够取得较好的分割结果,但平面和非平面的划分不能完全描述点云数据的特点,一些线状地物分割效果较差。文献[18]提出了一种利用点云数据生成多尺度超级体素的方法,然后融合体素的颜色、反射强度、法向量、主方向等多种特征,利用图分割的方法进行点云分割;文献[19]以体素为节点构建无向图并利用连通分支对点云数据进行聚类分割,然后结合归一化分割方法对欠分割地物进行细分;文献[20]将点云数据表示为深度图,并将深度图和改进的 DBSCAN (density-based spatial clustering of applications with noise)算法结合,实现了点云数据的分割。上述三种方法在分割之前往往需要率先利用其他方法对地面点进行剔除,以避免影响地物的分割结果,破坏点云数据的完整性;另外,文献[18]算法需要点云

属性信息的辅助,限制了算法的应用场景。文献[21]扩展了区域生长算法的分割规则,首先利用维数特征将点云数据粗分类为杆状、面状和球状三类,然后根据类别的不同采取不同的生长准则进行分割,使得区域生长算法能够适应不同地物的特点,实现点云数据的多要素分割;文献[22]将其应用于建筑物平面的提取;文献[23]在文献[21]基础上增加了过分割块的合并操作,通过计算各分割块轮廓之间的距离对相邻的分割块进行合并,减小了过分割的程度。

本文在多规则区域生长算法<sup>[21-23]</sup>的基础上进一步改进,以期提升点云数据多要素分割的准确性,为点云多目标识别和基于分割块的点云分类提供技术支持。本文主要工作包括:1)增加了面状点集数据分割中种子点的选取准则,提高了面状地物分割的精度;2)改进了分割块的合并条件,减少了零散分割块的数量,提升了合并的准确性;3)研究了分割阈值参数的自适应设置方法,提升了分割算法的自适应性。

## 2 多规则区域生长算法

区域生长(RG)算法<sup>[10-11]</sup>是由二维图像扩展到三维点云的一种分割算法,已广泛应用于点云数据的分割处理。图 1 场景中主要包含建筑、地面和植被三种要素(本文所有分割结果图片中地面颜色唯一且确定,其他分割块的颜色随机且与地面颜色不同)。RG 算法对于点云数据中的面状点云(如地面、建筑等)可以获得较好的分割结果,但对于非面状点(如植被等)的分割,由于非面状点相互之间法向量夹角过大,容易产生过分割现象,非面状点难以生长。因此单纯利用 RG 算法难以有效分割多要素场景点云。



图 1 基于 RG 算法的点云多要素分割示例

Fig. 1 An example of point cloud multi-factor segmentation based on traditional RG algorithm

文献[21]提出了一种基于维数特征的多规则点云区域生长(MRG)算法。相比于 RG 算法,MRG 算法将点云数据分成三种不同的类别——杆状、面

状和球状,然后根据点云类别分别采用各自的生长准则进行分割,避免了非面状点难以有效分割的问题。如图 2 所示,场景中植被点的分割结果整体较

好,没有出现图 1 中点云过于零散的现象。但是,MRG 算法还存在以下两个问题:一方面,由于没有种子点条件的限制,容易将不同面状物体分割到同一区域中,如图 2 A、B 和 C 区域中的建筑屋顶面被误分割成了地面(建筑分割块与地面同一颜色);另一方面,点云初始分割结果中存在一些尺寸较小的零散分割块,这些分割块应该与相应的其他分割块

合并,否则会导致后续地物识别、提取等过程中出现错误结果。为解决这一问题,文献[23]利用各分割块轮廓构建邻接关系,通过设置一定的阈值对 MRG 算法分割结果进行合并,但仅仅依靠距离因素的合并策略容易导致不同类别的物体合并到同一分割块中,因此还需要进一步对合并条件进行优化。

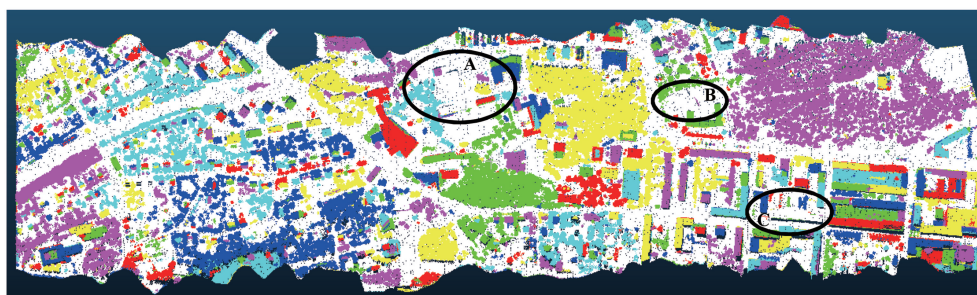


图 2 基于 MRG 算法的点云多要素分割示例

Fig. 2 An example of point cloud multi-factor segmentation based on MRG algorithm

### 3 改进的多规则区域生长算法

为了解决 MRG 算法中存在的问题,本研究提出一种改进的多规则区域生长(IMRG)算法。针对 MRG 算法在面状点集分割和分割块合并方面存在的问题,分别提出了相应的改进策略,改进算法有助于提升点云分割的准确性和完整性。

#### 3.1 面状点集分割优化

##### 3.1.1 误分割原因

点云面状点集中包含的要素主要有地面和建筑,面状点集是通过搜索邻域点、判断当前点与邻域点之间法向量夹角与阈值的关系来进行区域生长。从图 2 中可以看出,利用 MRG 算法进行点云分割

存在部分建筑平面被误分割为地面的情形。下面具体分析出现误分割现象的原因。以某建筑为例,如图 3 所示,屋顶面、建筑立面和地面各自内部点云,因法向量基本平行,在进行区域生长的过程中可以满足法向量夹角的阈值条件,能够得到有效生长;但是在建筑屋顶面和立面的边界处、建筑立面和地面边界处,由于点云数据变化剧烈,法向量计算不稳定,容易偏离正确方向,从图 3 A、B 区域中的点云法向量可以看出这一问题,因此,在进行区域生长时,不同要素边界处的法向量仍可以满足阈值条件,故生长点能够越过要素边界,导致不同要素被分割到同一分割块中。



图 3 法向量示例。(a)点云数据;(b)点云法向量

Fig. 3 An example of normal vector. (a) Point cloud; (b) normal vector of point cloud

##### 3.1.2 优化策略

面状点集误分割产生的原因主要在于不同要素边界处法向量存在偏差,如何控制不同要素边界处点云数据的生长是解决这一问题的关键,即当区域生长点到达要素边界处时停止生长。RG 算法在生

长过程中设置了种子点选取条件,只有新加入的生长点满足这一条件时才能继续搜索邻近点进行生长,故采用合适的种子点阈值能够取得较好的分割结果。下文借鉴 RG 算法中种子点选取方法,结合边界位置点云数据性质,对 MRG 算法面状点集分

割进行优化。

RG 算法将平面拟合残差用于区域生长种子点的选取<sup>[10]</sup>。面状点集中以面状地物要素为主,对于面状地物而言,其内部各点平面拟合残差通常较小,而边界点由于靠近其他物体,邻域点中包含有不同类别的点云数据,计算得到的平面拟合残差值会大一些。基于边界点这一特性,可以通过设置一定的平面拟合残差阈值  $v_t$  来实现对边界点的区分,当面状点平面拟合残差大于阈值  $v_t$  时,该点被认为是边界点,停止生长。

关于阈值  $v_t$  的选取,考虑从杆状点集的平面拟合残差入手进行设置。由于不同面状要素边界线具有杆状点云的特征,因此在进行维数特征分类时,部分要素边界线被分类到杆状点集中,从图 4 中可以看出,杆状点集生长结果中包含大量的建筑平面结

构线和边界线。如图 5 所示,阈值  $v_t$  的设置是为了区分出面状点集中的边界点  $B$  以避免面状点集的错误分割,而面状点集中的边界点  $B$  处于各物体要素的边缘位置,与杆状点集  $L$  中的结构线、边界线性质接近;换言之,面状点集中的边界点特征与杆状点集具有一定相似性,其平面拟合残差与杆状点集相近。因此,可以考虑利用杆状点集的平面拟合残差代表边界点残差,将其作为面状点集的阈值  $v_t$ 。由于前期利用维数特征进行点集分类时存在误差,杆状点集中存在少部分其他类型的误分点,导致残差数据中包含异常值。平均数容易受到异常值的影响,而中位数可以描述数据的集中趋势,具有稳健性,能够避免异常值的影响,是数据分析中重要的统计量<sup>[24]</sup>。因此,本文将面状点集平面拟合残差阈值  $v_t$  设置为杆状点集平面拟合残差的中位数。

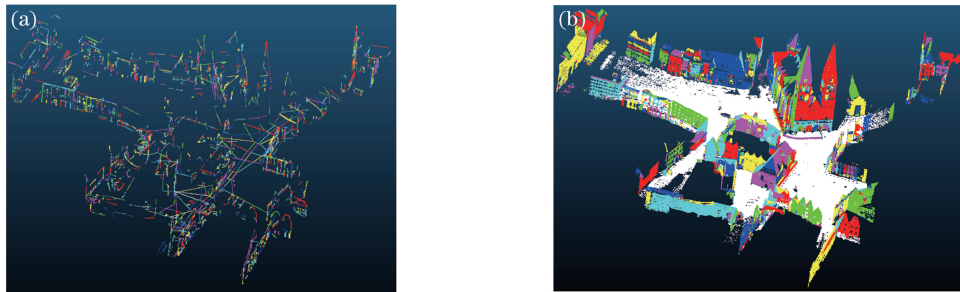


图 4 杆状、面状点集分割示例。(a)杆状点集;(b)面状点集

Fig. 4 An example of linear and planar point set segmentation. (a) Linear point set; (b) planar point set

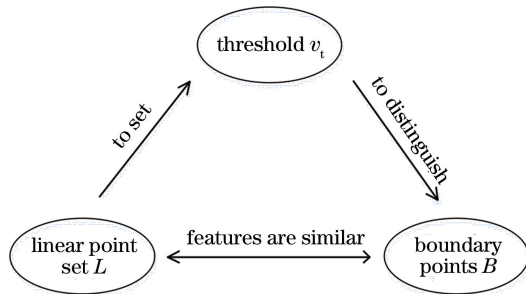


图 5 阈值  $v_t$  设置原理图

Fig. 5 Schematic diagram of threshold  $v_t$  setting

## 3.2 分割块合并优化

### 3.2.1 合并策略

点云数据利用维数特征分类时存在误差,且各点集生长结果由于阈值条件的限制会产生过分割现象,在这两种因素的作用下,点云数据的分割结果缺乏完整性,在此基础上进行的目标识别、提取等也将受到影响,所以有必要对 MRG 算法点云分割结果进行合并。文献[23]提出了一种合并方法,首先提取点云各分割块的轮廓,然后构建分割块轮廓之间

的邻接关系,最后通过设置一定的距离阈值对分割块进行合并,但仅仅依靠距离因素的合并策略容易导致不同类型的物体合并到同一分割块中。针对该方法存在的问题,本文提出了一种结合距离、分割块相似性和体积变化的合并策略,具体合并条件如下:

1) 距离条件。首先,利用凸包算法<sup>[25]</sup>得到各分割块的轮廓点;然后,计算分割块轮廓点之间的最小距离,并将其作为分割块之间的距离;最后,将分割块距离与阈值条件  $d_t$  进行比较,当分割块距离大于阈值  $d_t$  时,不需要进行合并操作。

2) 相似性条件。同一类型的分割块平面拟合残差一般接近,不同类型的分割块平面拟合残差往往相差较大,因此可以将分割块平面拟合残差作为分割块的相似性条件;计算分割块中各点平面拟合残差的平均值作为分割块的平面拟合残差,表达式为

$$R_i = \frac{1}{m} \sum_{j=1}^m r_j, \quad i=1,2,\dots,M, \quad (1)$$

式中: $R_i$  为分割块的平面拟合残差; $r_j$  为分割块中

点的平面拟合残差;  $m$  为分割块中点的个数;  $M$  为分割块的个数。若两分割块平面拟合残差的差值大于阈值  $s_i$ , 则不满足合并要求。

3) 体积变化条件。本文所涉及的分割块合并是将尺寸较小的分割块合并到同属于一个物体的较大分割块中。在将一小尺寸分割块  $S_i$  ( $i \in \{1, 2, \dots, M\}$ ) 合并到大尺寸分割块  $S_j$  ( $j \in \{1, 2, \dots, M\}, j \neq i$ ) 的过程中, 合并前后  $S_j$  对应的凸包体积  $V_j$  会有所变化, 当  $S_i$  和  $S_j$  属于同一物体时,  $S_j$  合并后增加的体积  $\delta V_j$  一般与  $S_i$  的体积  $V_i$  相差不大, 如图 6 所示; 但实际过程中还会存在一些误合并现象, 当于两个分割块属于不同的物体时,  $\delta V_j$  与

$V_i$  常有明显的出入, 如图 7 所示。因此需要设置一定的标准对  $S_j$  合并前后的体积变化进行衡量, 以提升分割块合并的准确性。本文设置体积变化统计参数  $\Delta V_{ij}$ , 表达式为

$$\Delta V_{ij} = \frac{V_j'' - V_j'}{V_i}, \quad i, j = 1, 2, \dots, M, \text{ and } i \neq j, \quad (2)$$

式中:  $\Delta V_{ij}$  是将分割块  $S_i$  合并到  $S_j$  体积变化参数;  $V_i$  为小分割块  $S_i$  的凸包体积;  $V_j'$  和  $V_j''$  分别是分割块  $S_j$  合并前后的凸包体积。同时设置体积变化阈值  $V_t$ , 当  $\Delta V_{ij} > V_t$  时, 说明  $S_j$  合并之后体积变化太大, 不满足合并要求。

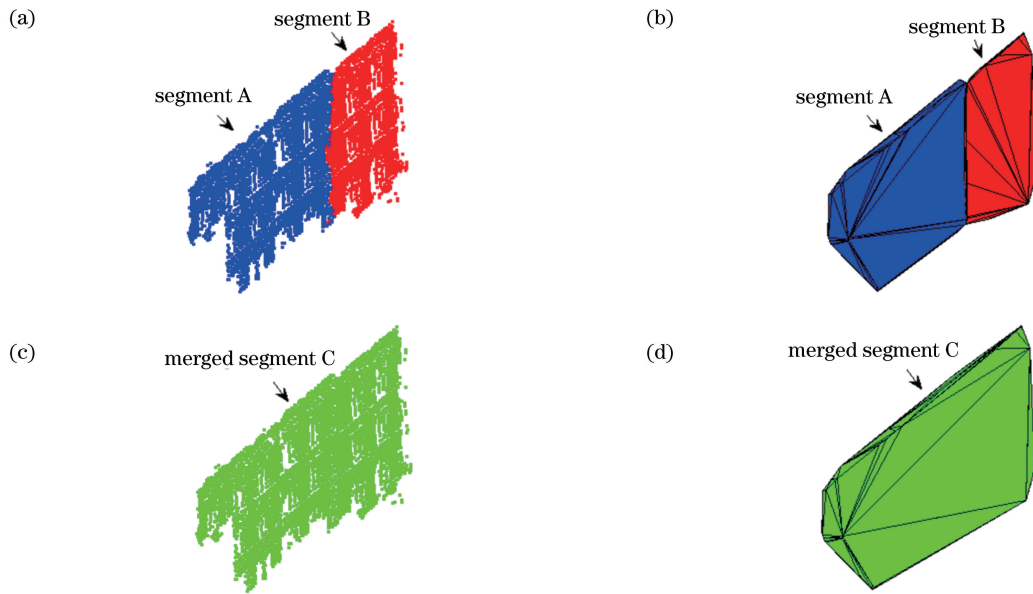


图 6 正确合并体积变化示例( $\Delta V=2.13$ )。(a)待合并分割块;(b)待合并分割块的凸包;  
(c)合并后的分割块;(d)合并后分割块的凸包

Fig. 6 An example of volume change based on correct merging ( $\Delta V=2.13$ ). (a) Unmerged segments; (b) convex hulls of unmerged segments; (c) merged segment; (d) convex hull of merged segments

分割块合并操作需要同时满足以上三个条件, 其中三个相关阈值——距离阈值  $d_i$ 、相似性阈值  $s_i$  和体积变化阈值  $V_t$  的自适应设置方法见后文, 具体合并步骤的伪代码如图 8 所示。

### 3.2.2 阈值自适应设置

合并策略中涉及三个参数——距离阈值  $d_i$ 、相似性阈值  $s_i$  和体积变化阈值  $V_t$ , 本文考虑对参数进行自适应设置来提升算法的自适应性, 具体设置方法如下。

#### 1) 距离阈值 $d_i$

对于每个分割块  $S_i$  ( $i \in \{1, 2, \dots, M\}$ ), 通过计算可以得到与其距离最近的分割块  $S_j$  ( $j \in \{1, 2, \dots, M\}, j \neq i$ ) 以及二者之间的邻近距离  $d_i$ , 而阈

值  $d_i$  是决定分割块与其距离最近分割块是否合并的条件, 因此可以从分割块之间的邻近距离集  $D = \{d_1, d_2, \dots, d_M\}$  出发进行  $d_i$  的设置。

邻近距离集合  $D$  可以分成两部分: 一部分是满足合并要求的邻近距离  $D_1$ ; 另一部分是不满足合并要求的邻近距离  $D_2$ 。由于初始分割结果中存在较多的零散分割块,  $D$  中包含的  $D_1$  类型数据多,  $D_2$  类型数据少, 最后要得到的数据是  $D_1$  类型数据, 而  $D_1$  类型数据数值通常较小, 与  $D_2$  类型数据有较为明显的区别。因此, 考虑将  $D_2$  类型数据作为粗差, 基于 Baarda 数据探测法<sup>[26]</sup> 采用向后选择的方式将  $D$  中粗差逐个剔除, 最后选择  $D$  中剩余数据的最大值作为距离阈值  $d_i$ 。

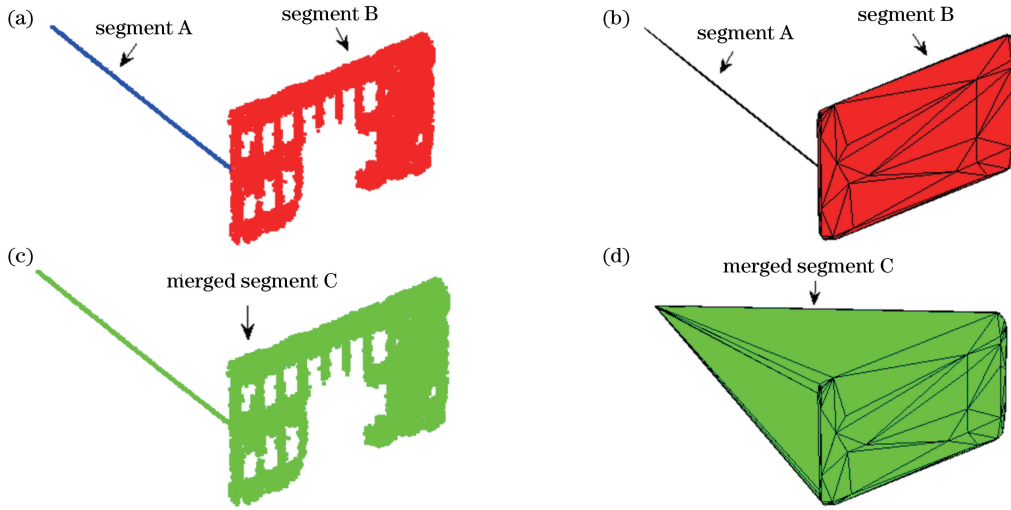


图 7 错误合并体积变化示例( $\Delta V = 17214.04$ )。(a)待合并分割块;(b)待合并分割块的凸包;  
(c)合并后的分割块;(d)合并后分割块的凸包

Fig. 7 An example of volume change based on incorrect merging ( $\Delta V = 17214.04$ ). (a) Unmerged segments; (b) convex hulls of unmerged segments; (c) merged segment; (d) convex hull of merged segments

---

**Algorithm:** merge strategy

---

**Input:** unmerged segments  $S_u$ , distance threshold  $d_t$ , similarity threshold  $s_t$ , threshold of volume change  $V_t$

**Output:** merged segments  $S_m$

initialize convex hull points set  $H = \emptyset$ ,  $S_m = \emptyset$

1.convex hull calculation

**for**  $i = 1$  to size $\{ S_u \}$  **do**

    calculate the convex hull points  $H_i$  of each segment  $S_i$

**end for**

  obtain the convex hull points set  $H$  of unmerged segments  $S_u$

2.segments merging

**while** size $\{ S_u \} > 1$  **do**

$S_1 = S_u \{1\}$

**for**  $i = 2$  to size $\{ S_u \}$  **do**

      calculate the distance  $d_{i-1}$  between  $S_1$  and  $S_i$  using  $H_1$  and  $H_i$

**end for**

    obtain minimum distance  $d_{\min} = \min_{1 \leq i \leq \text{size}\{S_u\}-1} \{d_i\}$  and the corresponding segment  $S_j$

    calculate the difference  $\Delta R$  of plane fitting residual between  $S_1$  and  $S_j$

    calculate the volume difference  $\Delta V$  before and after the merge

**if**  $d_{\min} > d_t$  or  $\Delta R > s_t$  or  $\Delta V > V_t$  **do**

$S_1 \xrightarrow{\text{insert}} S_m$ ,  $S_1 \xrightarrow{\text{remove}} S_u$ ,  $H_1 \xrightarrow{\text{remove}} H$

      continue

**else**

      merge  $S_1$  into  $S_j$

      recalculate and update the convex hull points  $H_j$  of  $S_j$

$S_1 \xrightarrow{\text{remove}} S_u$ ,  $H_1 \xrightarrow{\text{remove}} H$

**end if**

**end while**

$S_u \xrightarrow{\text{insert}} S_m$

---

图 8 合并策略的伪代码

Fig. 8 Pseudo-code of merge strategy

2) 相似性阈值  $s_t$ 

点云数据在进行初始分割之前被分类为杆状、面状和球状点集,由这些点集分割得到的点云分割块也可以分为杆状、面状和球状三类。不同类别分割块的平面拟合残差大小各不相同,相似性阈值  $s_t$  是用来判断不同分割块之间平面拟合残差的相似性程度,当点云分割块  $S_i$  和  $S_j$  的平面拟合残差之差  $\Delta R_{ij} = |R_i - R_j|$  超过不同类别分割块之间平面拟合残差的差值时,可以认为二者所属点云分割块的类别不同,不能进行合并。因此可以利用不同类别分割块之间平面拟合残差的差值设置相似性阈值  $s_t$ 。本文考虑利用  $k$  均值聚类方法对所有点云分割块残差进行聚类,聚类中心个数为 3,将聚类中心之间的最小差值作为相似性阈值  $s_t$ 。

3) 体积变化阈值  $V_t$ 

选取两个同时满足距离阈值  $d_t$  和相似性阈值  $s_t$  的点云分割块  $S_i$  和  $S_j$  ( $V_i < V_j$ ),计算  $S_j$  的体积变化参数  $\Delta V_{ij}$ ,把所有计算得到的分割块体积变化参数  $\Delta V_{ij}$  记为  $\Delta V$ 。 $\Delta V$  与  $D$  类似,其中同样存在两部分数据,因此可以参照距离阈值  $d_t$  的设置方法,采用 Baarda 数据探测法剔除其中的粗差,将最后剩余数据中的最大值作为体积变化阈值  $V_t$ 。

## 4 实 验

## 4.1 实验数据及评价指标

为验证本文算法的有效性,采用机载、站载和车

载三种不同场景(Scene I、Scene II、Scene III)的点云数据集进行实验,将三种点云数据集分别按高程赋色,如图 9 所示。其中,机载点云数据集来自 Terrasolid 软件官网 ([http://www.terrasolid.com/training/training\\_data.php](http://www.terrasolid.com/training/training_data.php)) 的训练数据,本文选取了其中部分数据进行分类实验;站载点云数据集选择了开源数据集 Robotic 3D Scan Repository 中场景 12 的数据 (<http://kos.informatik.uni-osnabrueck.de/3Dscans/>),该数据是在德国不莱梅市中心由 Riegl VZ400 站载扫描仪扫描得到。车载点云数据集采用了 Paris-Lille-3D 开源数据集<sup>[27]</sup>,该数据是由车载移动测量系统在法国巴黎和里尔两座城市扫描获得,本文只截取了其中部分数据。由于原始点云数据量太大,三种点云数据集均进行了降采样处理。另外,实验在 Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz RAM 16 GB 的计算机上实现。

为了更好地评价点云分割质量,采用了精度  $P$ 、召回率  $R$  和  $F_1$  得分三种评价指标<sup>[11]</sup>,具体计算表达式为

$$\begin{cases} P = \frac{T_P}{T_P + F_P} \\ R = \frac{T_P}{T_P + F_N} \\ F_1 = \frac{2 \times P \times R}{(P + R)} \end{cases}, \quad (3)$$

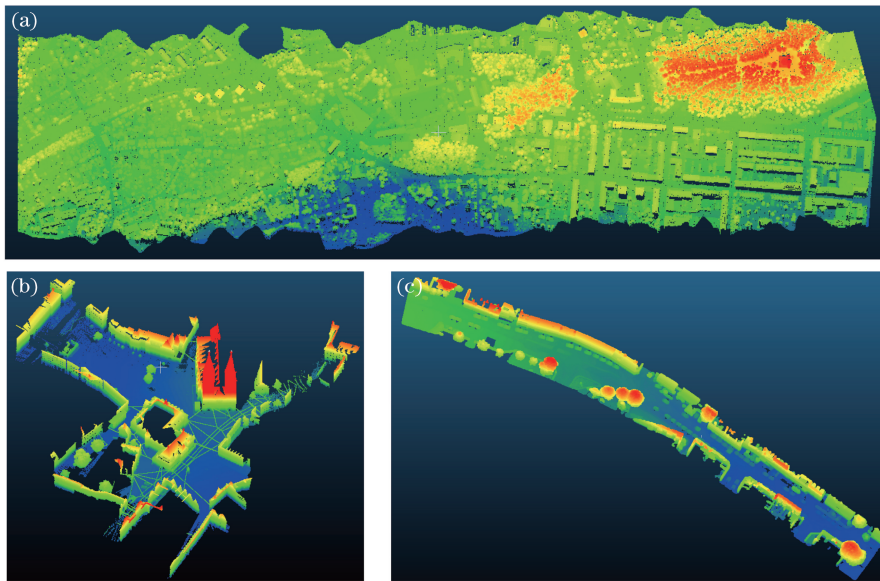


图 9 点云数据集。(a)机载点云(场景 I);(b)站载点云(场景 II);(c)车载点云(场景 III)

Fig. 9 Point cloud data set. (a) Airborne point cloud (scene I); (b) terrestrial point cloud (scene II); (c) vehicle-borne point cloud (scene III)

式中： $T_p$  表示正确点被归类为正确点的个数； $F_p$  表示错误点被归类为正确点的个数； $F_N$  表示正确点被归类为错误点的个数。另外，分别对三种场景点云数据进行了手工分割并将结果作为参考数据，见图 10。其中：Scene I 场景中包括建筑、植被和地面三类地物；Scene II 场景中包括建筑、植被、地面和电力线 4 类地物；Scene III 场景中包括建筑、植被、地面和车辆 4 类地物。

从图 10 中可以看出，参考分割数据都是完整的物体，在进行点云分割时，一个参考分割物体往往对

应多个分割块，因此，在进行评价时规定：1) 当分割块  $S_i (i \in \{1, 2, \dots, M\})$  与参考物体  $S_r (r \in \{1, 2, \dots, M_r\}, M_r$  为参考物体的个数) 中交叉点  $[p_c \in (S_i \cap S_r)]$  数目超过  $S_i$  中总点数的一半时，认为  $S_i$  为有效分割块；2) 当  $S_r$  中所有与其他分割块的交叉点总数超过  $S_r$  点数的一半时，认为  $S_r$  被有效识别。满足以上两个条件时，将交叉点  $p_c$  数目计入  $T_p$  中，同时把  $S_i$  中剩余点  $[p_r \in (S_i \setminus S_r)]$  数目计入  $F_p$  中；将  $S_r$  中交叉点扣除之后，剩余点的数目计入  $F_N$  中，同时其他未被识别的参考物体也计入  $F_N$  中。

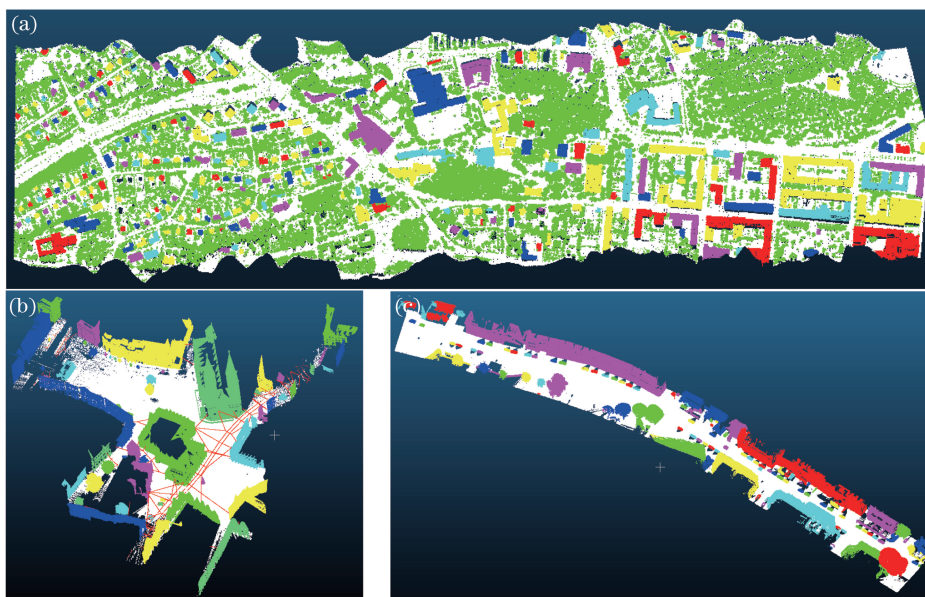


图 10 各场景点云参考数据。(a)场景 I;(b)场景 II;(c)场景 III

Fig. 10 Reference data of each scene. (a) Scene I; (b) scene II; (c) scene III

#### 4.2 面状点集分割对比

为验证本文算法面状点集分割优化的效果，分别采用 Scene I、Scene II 和 Scene III 三种场景的点云数据进行分割，未进行合并操作。分割过程中最优邻域计算选择了文献[28-29]方法；主向量

夹角阈值  $M_t$  设置为  $15^\circ$ ，法向量夹角阈值  $N_t$  分别取  $5^\circ$ 、 $10^\circ$  和  $15^\circ$ ；IMRG 算法在三种场景下的种子点阈值  $v_t$  按照自适应设置方法分别设置为 0.70 m、0.09 m 和 0.06 m。具体分割结果见表 1 和表 2。

表 1 各场景建筑平面分割评价

Table 1 Segmentation evaluation of building plane in each scene

Method	$N_t / (^\circ)$	$P / \%$			$R / \%$			$F_1 / \%$		
		Scene I	Scene II	Scene III	Scene I	Scene II	Scene III	Scene I	Scene II	Scene III
MRG	5	97.38	99.23	99.40	84.96	99.06	99.14	90.75	99.15	99.27
	10	97.32	99.40	99.34	77.80	97.41	75.80	86.47	98.40	85.99
	15	93.80	72.38	99.41	68.09	99.90	34.56	78.91	83.94	51.29
IMRG	5	97.56	99.33	99.45	87.68	99.00	99.25	92.35	99.16	99.35
	10	97.50	99.41	99.52	90.94	98.96	88.12	94.11	99.19	93.48
	15	97.23	99.43	99.38	89.91	96.91	64.50	93.43	98.16	78.23

结合表 1 和表 2 点云面状点数据分割结果进行分析，可以发现：

1) 总体来看，增加了种子点条件的多规则区域生长 IMRG 算法在面状点集方面分割效果良好，在



表 2 各场景地面分割评价

Table 2 Segmentation evaluation of ground in each scene

Method	$N_i / (^\circ)$	$P / \%$			$R / \%$			$F_1 / \%$		
		Scene I	Scene II	Scene III	Scene I	Scene II	Scene III	Scene I	Scene II	Scene III
MRG	5	88.41	97.64	98.61	87.87	98.09	99.03	88.14	97.87	98.82
	10	85.78	93.77	84.42	90.65	98.55	99.31	88.15	96.10	91.26
	15	83.09	93.65	67.05	90.55	3.66	99.54	86.66	7.05	80.13
IMRG	5	90.11	97.51	98.75	86.23	98.34	99.07	88.13	97.92	98.91
	10	89.85	97.39	91.50	89.81	98.56	99.31	89.83	97.97	95.25
	15	89.27	92.66	78.99	90.58	98.63	99.45	89.92	95.55	88.05

不同法向量夹角阈值  $N_i$ 、不同场景条件下分割精度均优于 MRG 算法,而且随着  $N_i$  的增大,优势更加明显,这证实了 IMRG 算法的有效性。

2) Scene I 场景中,IMRG 算法建筑平面召回率  $R$  高于 MRG 算法,这表明增加了种子点条件之后,建筑平面被误分割成地面的现象减少;与此同时,地面分割的精度  $P$  也有所提高,这说明地面分割块中非地面点减少,从侧面印证建筑物平面有效提取的点数得到了增加。由此可以看出 IMRG 算法的面状点集分割优化策略能够取得较好的效果。

3) Scene II 和 Scene III 场景中,在阈值  $N_i$  取值较小时,IMRG 算法分割精度和 MRG 算法相差不大;随着  $N_i$  取值的增加,MRG 算法的平面分割精度大幅下降,这是因为大量建筑平面和地面被误分割到同一分割块中;相比而言,IMRG 算法总体分割精度显著优于 MRG 算法,这说明 IMRG 算法能够在一定程度上抵御  $N_i$  设置不良的影响,体现了算法的稳定性和可靠性。同时,从图 11 中也可以看出,MRG 算法的分割结果中出现明显的误分割现象,大量建筑与地面位于同一分割块中(颜色相同);

而 IMRG 算法分割效果则明显优于 MRG 算法,可有效地将建筑平面与地面分割开来。

### 4.3 分割块合并对比

为了对比两种合并方法的效果,本文设置了两种合并策略,合并操作针对 4.2 节中 IMRG 算法的初始分割结果进行。策略①:仅根据分割块距离进行合并;策略②:结合距离、分割块相似性和体积变化进行合并。两种策略生长过程中主向量夹角阈值  $M_i$  设置为  $15^\circ$ ,法向量夹角阈值  $N_i$  分别设置为  $10^\circ$ (Scene I 和 Scene II)和  $5^\circ$ (Scene III);策略②中涉及的合并参数按照自适应阈值设置方法得到,结果见表 3;策略①涉及的距离阈值  $d_i$  取值参考策略②。两种策略合并结果分别见表 4、表 5 和表 6。

表 3 合并阈值参数

Table 3 Merge threshold parameters

Scene	$d_i / m$	$s_i / m$	$V_i$
Scene I	1.54	0.40	18.71
Scene II	0.25	0.05	16.33
Scene III	0.13	0.03	14.61

表 4 Scene I 场景合并结果

Table 4 Merge results in Scene I

Type of ground object	Building plan	Ground	Vegetation
Strategy① ( $M=1150$ )	$P / \%$	89.70	87.53
	$R / \%$	80.50	87.92
	$F_1 / \%$	84.85	87.72
Strategy② ( $M=1515$ )	$P / \%$	96.44	89.74
	$R / \%$	90.43	89.19
	$F_1 / \%$	93.34	89.46

表 5 Scene II 场景合并结果

Table 5 Merge results in Scene II

Type of ground object	Building plan	Ground	Vegetation	Power line
Strategy① ( $M=1550$ )	$P / \%$	97.67	91.15	99.78
	$R / \%$	96.22	97.25	96.47
	$F_1 / \%$	96.94	94.10	98.10
Strategy② ( $M=2730$ )	$P / \%$	99.38	97.42	99.78
	$R / \%$	98.97	98.45	99.79
	$F_1 / \%$	99.17	97.94	99.79

表 6 Scene III 场景合并结果  
Table 6 Merge results in Scene III

Type of ground object		Building plan	Ground	Vegetation	Car
Strategy①( $M=1673$ )	$P / \%$	97.09	98.30	98.75	94.67
	$R / \%$	98.65	98.18	96.40	94.14
	$F_1 / \%$	97.86	98.24	97.56	94.41
Strategy②( $M=2386$ )	$P / \%$	98.96	98.72	99.01	95.35
	$R / \%$	99.07	98.71	98.79	95.44
	$F_1 / \%$	99.01	98.71	98.90	95.39

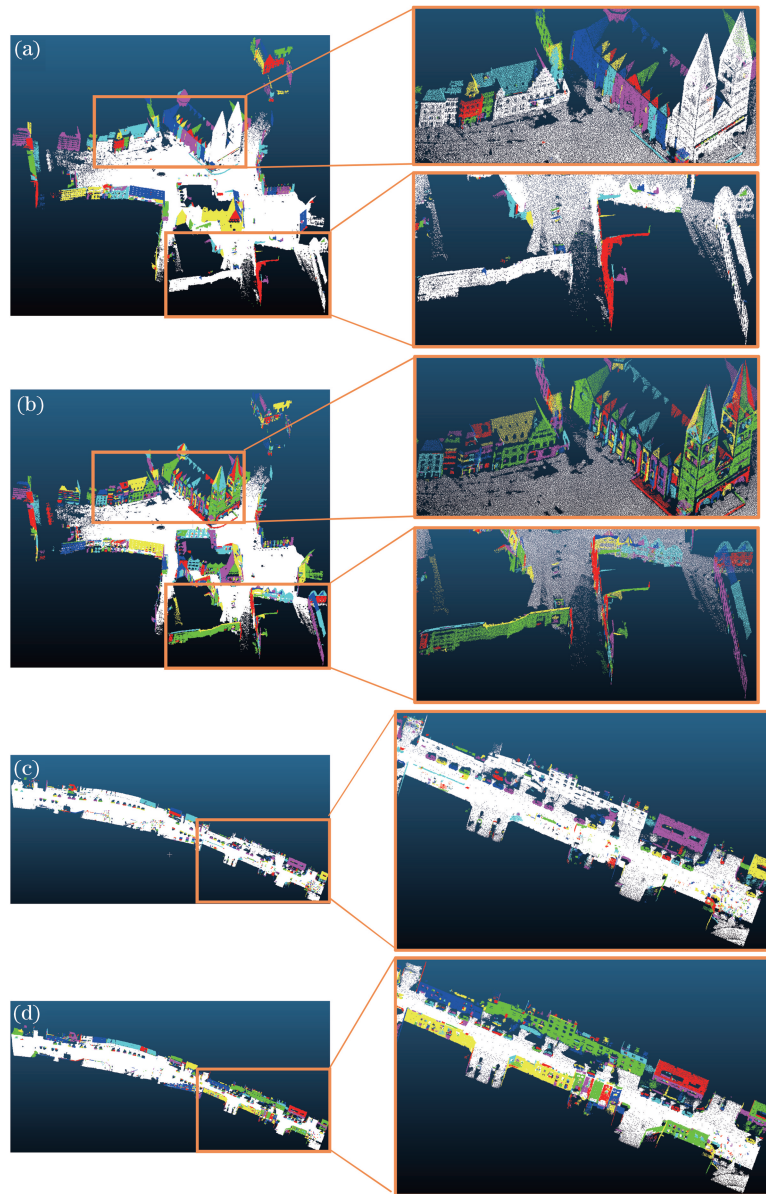


图 11 Scene II 和 Scene III 场景面状点集分割结果( $N_t=15^\circ$ )。(a) Scene II 中 MRG 结果;  
(b) Scene II 中 IMRG 结果;(c) Scene III 中 MRG 结果;(d) Scene III 中 IMRG 结果  
Fig. 11 Segmentation results of planar point set in Scene II and Scene III ( $N_t=15^\circ$ ). (a) MRG in Scene II;  
(b) IMRG in Scene II; (c) MRG in Scene III; (d) IMRG in Scene III

对比两种策略的合并结果,可以发现:

1) 结合距离、分割块相似性和体积变化的合并

策略②的整体合并效果优于仅依靠距离条件的合并策略①。利用策略①进行合并时,由于 Scene I 场景

中植被密集,建筑平面容易与相邻的植被分割块合并到一起,致使建筑平面分割精度下降最为明显; Scene II 场景中大部分电力线紧挨着建筑平面,致使合并之后电力线召回率  $R$  显著降低; Scene III 场景中各类别分割块合并后的精度均低于策略②。相比而言,本文提出的合并策略②考虑了分割块之间的相似性和分割前后体积的变化,避免了策略①中存在的问题,使分割块在合并过程中能够基本保持原有的分割精度。

2) 三种场景点云初始分割块数目分别为 2028、3787 和 3376,对比两种策略合并后剩余的分割块数量,利用策略①合并之后的分割块数量更少,这说明分割块的合并程度更深,但是一味地追求合并程度、忽略分割块的精度是没有意义的;与之相对,策略②能够在保持分割精度的情况下有选择的进行合并,同时兼顾了合并程度和分割块精度,合并思路更为合理。

#### 4.4 不同分割方法对比

目前针对点云多要素分割方面的算法较少,为了更好地评估验证本文提出 IMRG 算法的分割效果,选择了四种典型的方法与本文方法进行对比。Method 1: MRG 算法; Method 2: RG + mean shift 算法,该算法参考文献[17]思路,先采用 RG 算法进行面状点分割,然后利用 mean shift 算法对非面状点进行聚类,最后利用本文合并策略对分割结果进行合并; Method 3: 采用布料模拟滤波(CSF)<sup>[30]</sup>与 DBSCAN 聚类组合的方法——CSF + DBSCAN 算法,该算法参考文献[16,20]思路,先利用 CSF 算法去除地面点,然后利用 DBSCAN 算法对非地面点进行聚类,最后利用本文合并策略对分割结果进行合并; Method 4: 本文提出的 IMRG 算法; Method 5: 高斯映射迭代聚类方法(GMIC),该算法借鉴了文献[31]的分割思路,先将点云数据进行高斯映射得到高斯球,然后对高斯球上的点迭代进行 DBSCAN 聚类(由于机载点云中建筑物和地面在高斯球上的

映射点相近,聚类过程中难以区分,因此不进行机载点云的分割),最后将迭代剩余点映射回原始空间进行聚类处理,并将分割结果进行合并。其中 IMRG 算法参数设置参照 4.3 节,MRG 算法生长阈值  $M_i$  和  $N_i$  以及合并阈值  $d_i$  均参考 IMRG 算法;在 RG + mean shift 算法中, RG 算法生长阈值  $M_i$  和  $v_i$  均参考 IMRG 算法, mean shift 算法的带宽参数分别设置为 8 m、4 m 和 2 m; CSF + DBSCAN 算法中涉及的参数有网格尺寸  $l_c$ 、分类阈值  $h_c$ 、搜索半径  $r_d$  和最小点数  $m_d$ ,具体设置见表 7。各种算法在不同场景下点云分割的  $F_1$  得分分别见表 8~10。

表 7 Method 4 方法参数

Table 7 Parameters for Method 4

Scene	CSF		DBSCAN	
	$l_c/m$	$h_c/m$	$r_d/m$	$m_d$
Scene I	0.8	0.5	2	5
Scene II	2	0.4	0.5	10
Scene III	2	0.5	0.5	30

表 8 Scene I 场景分割  $F_1$  得分

Table 8  $F_1$  score of segmentation in Scene I

Method	$F_1/\%$			Time /s
	Building plan	Ground	Vegetation	
Method 1	76.84	85.03	90.62	294.03
Method 2	92.99	92.17	94.71	4697.64
Method 3	83.70	93.35	94.42	3230.18
Method 4	93.34	89.46	93.56	372.53

分析表 8~10 中不同算法得到的点云分割结果,可以发现:

1) 本文提出的 IMRG 算法整体分割效果良好,在分割精度和效率方面都具有一定的优势。IMRG 算法在 MRG 算法基础上进行了优化,分割精度显著提升;同时 IMRG 算法继承了多点集多规则生长的特色,能够根据不同要素的特点分别进行分割。相比而言, RG + mean shift 算法和 CSF + DBSCAN 算法无法概括点云整体场景的特点,特别是缺乏对

表 9 Scene II 场景分割  $F_1$  得分

Table 9  $F_1$  score of segmentation in Scene II

Method	$F_1/\%$				Time /s
	Building plan	Ground	Vegetation	Power line	
Method 1	97.00	94.56	97.53	54.95	499.06
Method 2	98.94	97.57	98.74	93.61	943.72
Method 3	98.64	97.51	99.19	79.06	1523.06
Method 4	99.17	97.94	99.79	98.88	830.65
Method 5	97.90	95.71	96.36	78.59	4627.44

表 10 Scene III 场景分割  $F_1$  得分  
Table 10  $F_1$  score of segmentation in Scene III

Method	$F_1/\%$				Time /s
	Building plan	Ground	Vegetation	Car	
Method 1	98.56	98.19	98.41	93.89	954.01
Method 2	97.52	98.69	97.01	92.72	1750.89
Method 3	93.08	96.18	89.58	82.65	1218.09
Method 4	99.01	98.71	98.90	95.39	987.68
Method 5	91.92	96.10	89.47	79.11	5922.04

杆状、线状地物的描述,影响分割精度;GMIC 算法可以较好地提取面性点,但对于法向量杂乱的非面性点难以有效分割;如图 12 所示,部分电力线被错误地分割到了建筑物中,GMIC 算法建筑分割结果

中还包括了部分植被点。

2) 在分割精度方面,IMRG 算法整体优于其他算法,但在 Scene I 场景的地面和植被分割中精度低于 RG+mean shift 算法和 CSF+DBSCAN 算法。

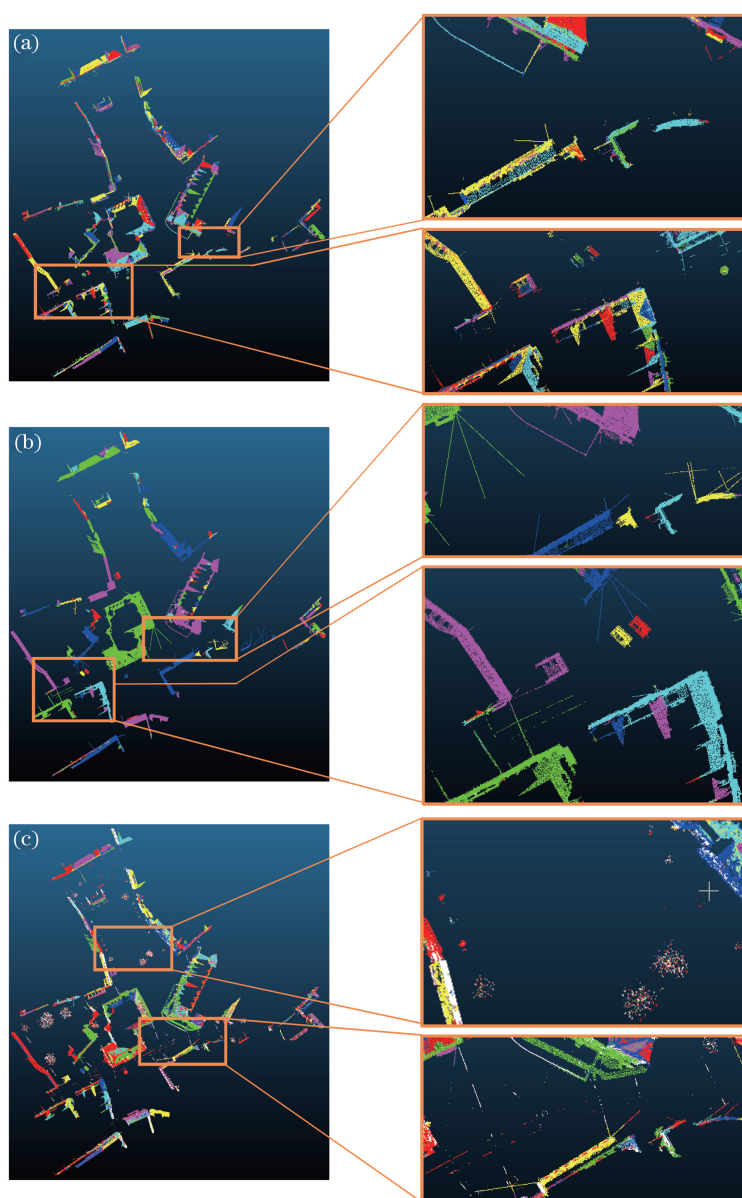


图 12 错误分割。(a) RG+mean shift;(b) CSF+DBSCAN;(c) GMIC

Fig. 12 Incorrect segmentation. (a) RG+mean shift; (b) CSF+DBSCAN; (c) GMIC

分析其原因, Scene I 场景覆盖范围大、地形起伏变化明显, 利用维数特征进行点集分类的误分率增大, 部分地面点没能正确归入面状点集, 如表 11 所示, 与其他算法相比, IMRG 算法中得到的地面分割块有效点数偏少; 同时, 点集分类误差也使得地面点中包含植被点云的可能性增大, 最终影响了地面分割块的精度, 从表 11 中也可以看出 IMRG 算法得到的地面分割块中错误点数偏多。

表 11 Scene I 场景地面分割结果

Table 11 Segmentation results of ground in Scene I

Method	$T_p$	$F_p$
RG+mean shift	351708	29669
CSF+DBSCAN	364594	34771
IMRG	340502	38946

3) 在分割耗时方面, IMRG 算法时间略多于 MRG 算法, 这是由于 IMRG 算法增加了种子点判断条件和合并条件, 延长了面状点集生长迭代和分割块合并的时间。RG+mean shift 算法、CSF+DBSCAN 算法和 GMIC 算法耗时长, 这与聚类算法复杂度高、计算效率低有关。

## 5 结 论

本文立足于点云场景多要素分割需求, 在多规则区域生长算法的基础上, 提出了一种改进的多规则区域生长算法。该方法保持了原有算法的优势, 对面状点集分割和分割块合并两方面内容进行了优化, 同时探索了相关阈值参数自适应设置的方法。在面状点集分割方面, 分析了 MRG 算法误分割现象产生的原因, 然后采取相应的措施——利用平面拟合残差设置种子点条件, 提升了面状点集分割的精度; 在分割块合并方面, 针对单一距离因素合并效果差的问题, 增加了分割块相似性条件和体积变化条件, 保证了分割块合并的准确性; 在阈值自适应设置方面, 结合各参数的特点, 分别采用中位数、Baarda 数据探测法、 $k$  均值聚类等方法实现阈值参数的自适应设置, 提升了算法的自适应性。利用三组数据进行实验, 结果表明本文算法所具有的优势——面状地物分割的准确性得到了提升, 同时能够兼顾合并程度和分割块精度。

本文算法可以有效实现点云场景的多要素分割, 为点云地物识别和提取、点云分类等应用提供了有效支持。但同时本文算法还存在一定的优化空间, 包括点云最优邻域的快速计算以及点云杆状、面状和体状数据维数特征分类方法的改善等, 这些问

题的解决将有助于算法进一步获取更优的点云多要素分割结果。

## 参 考 文 献

- [1] Li M L, Zong W P, Li G Y, et al. Extraction of structure line segments from point clouds using voxel-based region growing[J]. Acta Optica Sinica, 2018, 38(1): 0112001.  
李明磊, 宗文鹏, 李广云, 等. 基于体素生长的点云结构直线段提取[J]. 光学学报, 2018, 38(1): 0112001.
- [2] Wang Y, Cheng L, Chen Y M, et al. Building point detection from vehicle-borne LiDAR data based on voxel group and horizontal hollow analysis [J]. Remote Sensing, 2016, 8(5): 419.
- [3] Fang L N, Huang Z W, Luo H F, et al. Solid lanes extraction from mobile laser scanning point clouds [J]. Acta Geodaetica et Cartographica Sinica, 2019, 48(8): 960-974.  
方莉娜, 黄志文, 罗海峰, 等. 车载激光扫描数据中实线型交通标线提取[J]. 测绘学报, 2019, 48(8): 960-974.
- [4] Ma Z Y, Pang Y, Li Z Y, et al. Fine classification of near-ground point cloud based on terrestrial laser scanning and detection of forest fallen wood [J]. Journal of Remote Sensing, 2019, 23(4): 743-755.  
马振宇, 庞勇, 李增元, 等. 地基激光雷达森林近地面点云精细分类与倒木提取[J]. 遥感学报, 2019, 23(4): 743-755.
- [5] Feng J R, Ma X D, Guan H O, et al. Calculation method of soybean plant height based on depth information[J]. Acta Optica Sinica, 2019, 39(5): 0515003.  
冯佳睿, 马晓丹, 关海鸥, 等. 基于深度信息的大豆株高计算方法 [J]. 光学学报, 2019, 39(5): 0515003.
- [6] Ballard D H. Generalizing the Hough transform to detect arbitrary shapes [J]. Pattern Recognition, 1981, 13(2): 111-122.
- [7] Fischler M A, Bolles R C. Random sample consensus [J]. Communications of the ACM, 1981, 24(6): 381-395.
- [8] Lari Z, Habib A. An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2014, 93: 192-212.
- [9] Wang X H, Wu L S, Chen H W, et al. Feature line extraction from a point cloud based on region clustering segmentation [J]. Acta Optica Sinica,

- 2018, 38(11): 1110001.
- 王晓辉, 吴禄慎, 陈华伟, 等. 基于区域聚类分割的点云特征线提取[J]. 光学学报, 2018, 38(11): 1110001.
- [10] Rabbani T, Heuvel F A, Vosselman G. Segmentation of point clouds using smoothness constraint [J]. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2006, 36(5): 248-253.
- [11] Vo A V, Truong-Hong L, Laefer D F, et al. Octree-based region growing for point cloud segmentation [J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2015, 104: 88-100.
- [12] Zhang J X, Duan M Y, Yan Q, et al. Automatic vehicle extraction from airborne LiDAR data using an object-based point cloud analysis method[J]. *Remote Sensing*, 2014, 6(9): 8405-8423.
- [13] Zhao Z Z, Zhang Y J. Building extraction from airborne laser point cloud using NDVI constrained watershed algorithm[J]. *Acta Optica Sinica*, 2016, 36(10): 1028002.
- 赵宗泽, 张永军. 基于植被指数限制分水岭算法的机载激光点云建筑物提取[J]. 光学学报, 2016, 36(10): 1028002.
- [14] Chen M Y, Tang Y C, Zou X J, et al. Multi-shaped targets recognition and point clouds acquisition algorithm in complex environment [J]. *Laser & Optoelectronics Progress*, 2018, 55(11): 111505.
- 陈明猷, 唐昀超, 邹湘军, 等. 复杂环境下异形多目标识别与点云获取算法[J]. 激光与光电子学进展, 2018, 55(11): 111505.
- [15] Vosselman G, Coenen M, Rottensteiner F. Contextual segment-based classification of airborne laser scanner data [J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017, 128: 354-371.
- [16] Tong G F, Li Y, Zhang W L, et al. Point set multi-level aggregation feature extraction based on multi-scale max pooling and LDA for point cloud classification[J]. *Remote Sensing*, 2019, 11(23): 2846.
- [17] Wu T. Object oriented classification and building extraction using airborne LiDAR point clouds [D]. Wuhan: Wuhan University, 2016: 11-43.
- 吴腾. 机载 LiDAR 点云面向对象的分类与建筑物提取[D]. 武汉: 武汉大学, 2016: 11-43.
- [18] Dong Z, Yang B S. Hierarchical extraction of multiple objects from mobile laser scanning data[J]. *Acta Geodaetica et Cartographica Sinica*, 2015, 44(9): 980-987.
- 董震, 杨必胜. 车载激光扫描数据中多类目标的层次化提取方法[J]. 测绘学报, 2015, 44(9): 980-987.
- [19] Luo H F, Fang L N, Chen C C, et al. Road side multiple objects extraction from mobile laser scanning point cloud based on DBN [J]. *Acta Geodaetica et Cartographica Sinica*, 2018, 47(2): 234-246.
- 罗海峰, 方莉娜, 陈崇成, 等. 基于 DBN 的车载激光点云路侧多目标提取[J]. 测绘学报, 2018, 47(2): 234-246.
- [20] 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.
- [21] Yang B S, Dong Z, Wei Z, et al. Extracting complex building facades from mobile laser scanning data[J]. *Acta Geodaetica et Cartographica Sinica*, 2013, 42(3): 411-417.
- 杨必胜, 董震, 魏征, 等. 从车载激光扫描数据中提取复杂建筑物立面的方法[J]. 测绘学报, 2013, 42(3): 411-417.
- [22] Lu W X, Wan Y C, He P P, et al. Extracting and plane segmenting buildings from large scene point cloud[J]. *Chinese Journal of Lasers*, 2015, 42(9): 0914004.
- 卢维欣, 万幼川, 何培培, 等. 大场景内建筑物点云提取及平面分割算法[J]. 中国激光, 2015, 42(9): 0914004.
- [23] Yang B S, Dong Z, Liu Y, et al. Computing multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data [J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017, 126: 180-194.
- [24] Fan J C, Mei C L. Data analysis [M]. 2nd ed. Beijing: Science Press, 2010: 1-11.
- 范金城, 梅长林. 数据分析[M]. 2 版. 北京: 科学出版社, 2010: 1-11.
- [25] Barber C B, Dobkin D P, Huhdanpaa H. The quickhull algorithm for convex hulls [J]. *ACM Transactions on Mathematical Software*, 1996, 22(4): 469-483.
- [26] Sui L F, Song L J, Chai H Z. Error theory and foundation of surveying adjustment [M]. Beijing: Surveying and Mapping Press, 2010: 194-199.
- 隋立芬, 宋力杰, 柴洪洲. 误差理论与测量平差基础[M]. 北京: 测绘出版社, 2010: 194-199.
- [27] Roynard X, Deschaud J E, Goulette F. Paris-Lille-3D: a large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification [J]. *The International Journal of*

- Robotics Research, 2018, 37(6): 545-557.
- [28] Weinmann M, Jutzi B, Mallet C. Semantic 3D scene interpretation: a framework combining optimal neighborhood size selection with relevant features[J]. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2014, 2(3): 181-188.
- [29] Weinmann M, Jutzi B, Hinz S, et al. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2015, 105: 286-304.
- [30] Zhang W M, Qi J B, Wan P, et al. An easy-to-use airborne LiDAR data filtering method based on cloth simulation[J]. Remote Sensing, 2016, 8(6): 501.
- [31] Zhao B F, Hua X H, Yu K G, et al. Indoor point cloud segmentation using iterative Gaussian mapping and improved model fitting [J]. IEEE Transactions on Geoscience and Remote Sensing, 2020, 58(11): 7890-7907.