

点云体素细化生成树木骨架的方法

栗荣豪¹, 陈益楠², 甘小正¹, 张青¹, 汪沛^{1*}

¹北京林业大学理学院, 北京 100083;

²西安电子科技大学物理与光电工程学院, 陕西 西安 710126

摘要 基于地面三维激光扫描仪获取树木的点云数据,提出了一种通过细化点云数据体素生成树木骨架的方法。基于树木点云数据构建体素空间,计算点云体素坐标;根据各体素中包含点云数据的统计信息进行体素噪点滤除;利用细化模板对滤除噪声的体素进行细化处理,基于细化后的体素拟合出骨架节点;根据树木在自然空间上的连通性和深度优先搜索算法连接骨架节点,生成树木骨架。利用一棵银杏树和一棵重瓣榆叶梅树对算法进行验证。采用地面三维激光扫描仪分别对这两棵树进行扫描,基于不同扫描精度的树木点云,分析了不同参数对树木骨架生成的影响。在生成银杏树骨架时,本文方法运行时间约缩短至 GSA 方法的 1/30。而处理数据量更大的重瓣榆叶梅树点云数据时,树木骨架生成时间更是缩短至 GSA 方法的 1/67。实验结果表明,所提算法生成的两棵树骨架形态与树木原始点云所表现的形态结构相对一致,并且具有较好的运算效率,该算法具有一定的可行性和有效性。

关键词 遥感; 树木骨架; 点云; 体素; 细化; 地面三维激光扫描仪

中图分类号 TN958.58; TP391.9

文献标识码 A

doi: 10.3788/LOP56.192802

Tree-Skeleton Generation Method by Thinning Voxels of Point Cloud

Li Ronghao¹, Chen Yanan², Gan Xiaozheng¹, Zhang Qing¹, Wang Pei^{1*}

¹ School of Science, Beijing Forestry University, Beijing 100083, China;

² School of Physics and Optoelectronic Engineering, Xidian University, Xi'an, Shaanxi 710126, China

Abstract A method for generating tree skeletons by thinning the voxels of point cloud data has been proposed based on point cloud data acquired by a terrestrial three-dimensional laser scanner. First, the voxel space is constructed based on the point cloud data of a tree, and the voxel coordinates of point clouds are calculated simultaneously. Second, the noise points in the voxels are filtered according to the statistical information of point cloud data in each voxel. Third, the voxels without noise are thinned using thinning templates. The skeleton nodes are then fitted considering the thinned voxels. Finally, based on the connectivity of natural trees, a depth-first search algorithm is employed to connect nodes and generate tree skeletons. The proposed method is tested with a ginkgo tree and an *Amygdalus triloba* f. multiplex tree. The two trees are scanned by a terrestrial three-dimensional laser scanner. The effects of different parameters on the tree skeleton are analyzed by using the tree point clouds with different scanning accuracy. In comparison with the GSA method, the proposed method can reduce time consumption for the tree skeleton generation of the ginkgo tree and the *Amygdalus triloba* f. multiplex tree to 1/30 and 1/67, respectively. Experimental results show that the skeletons of the two trees generated by the proposed method are consistent with the original point clouds, and the proposed method is feasible and efficient.

Key words remote sensor; tree skeleton; point cloud; voxels; thinning; terrestrial three-dimensional laser scanner

OCIS codes 280.3640; 120.5800; 100.6890; 110.6880

1 引言

近年来,地面三维激光扫描作为一种新的测量技

术已被广泛应用于城市建模和森林调查等领域^[1-5]。地面三维激光扫描仪采用非接触的方式获取物体表面采样点的距离信息,通过计算距离与角度信息获得点

收稿日期: 2019-03-08; 修回日期: 2019-04-06; 录用日期: 2019-04-15

基金项目: 北京林业大学青年教师科学研究中长期项目(2015ZCQ-LY-02)、北京市自然科学基金(6164037)

* E-mail: wangpei@bjfu.edu.cn

云数据,并进行数据处理与分析^[6]。地面三维激光扫描仪因数据精度高和分辨率高等特点也常被用于林业测量,可以辅助人工测量,提高树木信息(如胸径)的采集效率^[7]。树木骨架是对树木三维结构的精简描述,其可表示树木的形状和拓扑结构,根据生成的树木骨架,可进一步计算出树木的几何结构、分枝模式和分枝角度等信息,从而体现树木的结构和生长特点。

目前,已有不少学者在基于三维点云生成骨架方面进行了一些研究工作^[8-16],其中包括一些生成树木骨架的研究成果。树木骨架的生成方法大致可以分为两种。一种是通过分割空间,得到体素、元胞等新的计量单位,进行骨架节点的定位与计算,最后连接节点生成树木骨架。Bucksch等^[17-18]提出了一种基于八叉树的骨架生成方法,生成八叉树之后,从八叉树元细胞中提取一个由点组成的连接图,然后消除图中的闭环,形成一条直线段,最后得到树木骨架。Ramamurthy等^[19]提出了一种基于体素的植物骨架生成算法。首先设定体素的体积大小,并计算点云数据所属的体素,通过细化算法删除冗余的体素后,利用近邻搜索连接得到植物骨架。另一种是基于点云局部分割,计算骨架节点,然后根据图论或法向量等方法连接骨架节点,生成树木骨架。Yan等^[20]将提取的树木骨架分成分割与重建两个部分,该方法基于K-均值聚类算法实现区域识别和定位,然后建立邻域图提取骨架节点和树木分支,最后通过B样条模型计算生成树木骨架。Xu等^[21]提出了一种利用点云生成数骨架的基于最短路径和聚类的算法,该方法首先根据每个点的邻接关系构造邻域图,然后用最短路径算法计算各点与根点之间的距离,并基于距离聚类生成聚类块,最后利用聚类块内的点云拟合骨架节点并连接,从而得到树骨架。Delagrance等^[22]扫描获得没有叶子的树木点云数据,并沿着Z轴提取210层宽度为1cm的切片,从每个切片中计算出质心,再根据最小生成树算法连接质心生成树木骨架。唐丽玉等^[23]基于水平采样和最小二乘法获取枝干骨架点,根据骨架节点与特征点的拓扑结构重构树木三维模型。赵春江等^[24]基于立体视觉原理,通过两幅特定角度的图像先提取二维骨架,然后通过极线约束算法和合理的匹配准则,最终生成玉米植株的三维骨架信息。李巍岳等^[25]提出了一种通过稀疏点云生成树木骨架的方法。该方法主要通过模糊C-均值分割点云,然后使用B样条拟合聚类团质心,根据法向量方向变化率确定节点位置,从而生成树木骨架。Li等^[26]基于图论的概念对点云数据进

行建图,采用近邻搜索算法对每个点进行标号,再进行非线性量化处理,使数据分层形成聚类块,然后根据聚类块中的点云拟合骨架节点,最后根据深度搜索算法连接骨架节点,生成树木骨架。

通过对上述算法的研究分析,结合树木点云的特点,本文基于体素空间细化处理的思想,提出了一种体素细化改进的树木骨架生成方法:1)通过树木点云的范围和体素的数目构建体素空间;2)根据滤除阈值对噪点体素进行滤除;3)对目标体素进行细化处理,并利用细化后的体素拟合骨架节点;4)基于树木的空间结构和深度优先搜索算法连接骨架节点,生成骨架。该方法提出了一种体积形状可变的体素,用于构建体素空间,同时增加了对体素孤立分布程度的分析,然后利用体素细化算法提取树木的骨架,大大提高了树木骨架生成的运行效率。

2 材料与方法

本文方法处理流程如图1所示,主要包括体素空间构建、噪点体素滤除、细化算法处理和树木骨架生成4部分。

1) 体素空间构建。计算树木点云所占空间大小,记为体素空间。按照输入参数将体素空间(X、Y、Z三个维度上的体素个数)分成若干个体素。算法中约定包含树木点云的体素为目标体素,包含非树木点云的体素为噪点体素。

2) 噪点体素滤除。根据体素中点云的密集程度和体素的分布情况,通过设定滤除点数阈值和孤立体素滤除阈值,对噪点体素进行判断与滤除。

3) 细化算法处理^[27-32]。噪点体素滤除完成之后,对目标体素进行细化处理,通过迭代删除边缘的目标体素,最后只保留中间呈线状连接的目标体素。

4) 树木骨架生成。根据细化处理后的体素内的点云数据拟合骨架节点,并根据树木结构与近邻搜索算法连接骨架节点,得到最后的树木骨架。

2.1 数据获取

本次实验使用的地面三维激光扫描仪器为RIEGL VZ-400型地面三维激光扫描仪,扫描距离最远可以达到400m,每秒最多可以获取 4.2×10^4 个点,垂直扫描角度范围为 $-40^\circ \sim +60^\circ$,水平扫描角度范围为 360° 。数据采集于北京林业大学校内的一棵银杏树(Ginkgo)和一棵重瓣榆叶梅树(*Amygdalus triloba* f. *multiplex* tree),采集时间为2018年1月23日。实验过程中,对银杏树和重瓣榆叶梅树分别进行了多次扫描,并通过配套的软

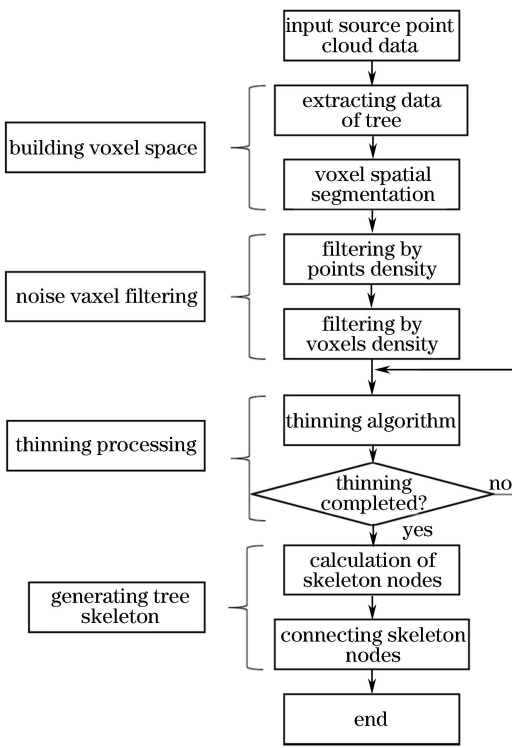


图1 方法流程图

Fig. 1 Flow chart of proposed method

件 RISCAN PRO 手动截取样本树木的点云数据, 删除周围噪点。数据对应的具体角分辨率和点数如表 1 所示。

表 1 点云数据采集信息

Table 1 Acquisition information of point cloud data

Tree	Scanning accuracy	Point number
Ginkgo	0.02	420492
Ginkgo	0.05	65994
Ginkgo	0.10	17002
<i>Amygdalus triloba</i> f. <i>multiplex</i>	0.02	686113
<i>Amygdalus triloba</i> f. <i>multiplex</i>	0.10	27950

2.2 构建体素空间

通过遍历点云, 获取数据在 X 、 Y 、 Z 三个维度上的最大值 (X_{\max} , Y_{\max} , Z_{\max}) 和最小值 (X_{\min} , Y_{\min} , Z_{\min}), 从而构建体素空间。输入参数 M 、 N 、 K , M 、 N 、 K 分别代表体素划分的行数、列数和层数, 即将体素空间划分为 $M \times N \times K$ 个体素。体素大小 (V_{x_size} , V_{y_size} , V_{z_size}) 计算公式为

$$\begin{cases} V_{x_size} = (X_{\max} - X_{\min}) / M \\ V_{y_size} = (Y_{\max} - Y_{\min}) / N \\ V_{z_size} = (Z_{\max} - Z_{\min}) / K \end{cases} \quad (1)$$

显然, 树木点云的位置在空间三个维度上不同, 故对于可调整的输入参数 M 、 N 、 K , V_{x_size} 、 V_{y_size} 、

V_{z_size} 可以设置为相同, 也可设置为不同, 从而保证体素划分的灵活性。

确定体素大小后, 设定体素所在的行数、列数和层数作为体素的坐标。则点云所属体素计算公式为

$$\begin{cases} V_x = \text{int}(x / V_{x_size}) \\ V_y = \text{int}(y / V_{y_size}) \\ V_z = \text{int}(z / V_{z_size}) \end{cases}, \quad (2)$$

式中: (V_x , V_y , V_z) 为点云所在体素的位置坐标; (x , y , z) 为点云的空间坐标; $\text{int}()$ 函数为取整函数。

2.3 噪点体素滤除

扫描设备、外部环境、树木特征和枝叶遮挡等会导致噪点的产生, 这可能造成骨架的冗余与缺失, 影响树木骨架的生成。本研究从两个方面对包含噪点的体素进行判断与滤除。

1) 基于体素内点云数量滤除体素。体素内点云数量过少, 说明这些点可能是噪点。在算法中设定滤除阈值 ϵ_1 , 当体素中点数 S_{num} 小于 ϵ_1 , 则认为该体素是噪点体素并删除。根据角分辨率和点到原点的距离可以计算出体素内充满点时点的个数, 选取一个小于 1 的比例系数, 即得滤除阈值 ϵ_1 , 可表示为

$$\epsilon_1 = a \times \frac{V_{x_size}}{R_x} \times \frac{V_{z_size}}{R_z}, \quad (3)$$

$$\begin{cases} R_x = 2L \times \sin \frac{\theta}{2} \\ R_z = 2L \times \sin \frac{\phi}{2} \end{cases}, \quad (4)$$

式中: R_x 和 R_z 分别为点云中 X 、 Z 方向上两点间距; L 为点到原点的距离; θ 和 ϕ 分别为水平方位角和俯仰角的扫描角分辨率, a ($a < 1$) 为滤除比例。

比较阈值和体素内的点数, 将体素区分为目标体素和噪点体素, 并进行标记。 l_{abel} 为 1, 表示该体素为目标体素; l_{abel} 为 0, 表示该体素为噪点体素:

$$l_{\text{abel}} = \begin{cases} 0, & S_{\text{num}} < \epsilon_1 \\ 1, & S_{\text{num}} > \epsilon_1 \end{cases}. \quad (5)$$

2) 根据目标体素分布的孤立程度滤除体素。若一个目标体素周围 26 邻域中的目标体素过少, 则该目标体素有较大可能是噪点体素。计算目标体素周围 26 邻域中目标体素数量 v_{num} 。设定孤立程度判断阈值为 ϵ_2 , 如果 v_{num} 小于 ϵ_2 , 视为噪点体素并删除。 l_{abel} 为 1, 表示该体素为目标体素; l_{abel} 为 0, 表示该体素为噪点体素:

$$l_{\text{abel}} = \begin{cases} 0, & v_{\text{num}} < \epsilon_2 \\ 1, & v_{\text{num}} > \epsilon_2 \end{cases}. \quad (6)$$

2.4 细化算法处理

本文算法采用文献[32]中提到的细化策略,对三维空间中的目标体素进行细化处理。设定一组体

素空间中的三维模板,用于判定边缘体素,删除模板需要考虑上下左右前后6个方向,“上”方向的6个删除模板如图2所示。

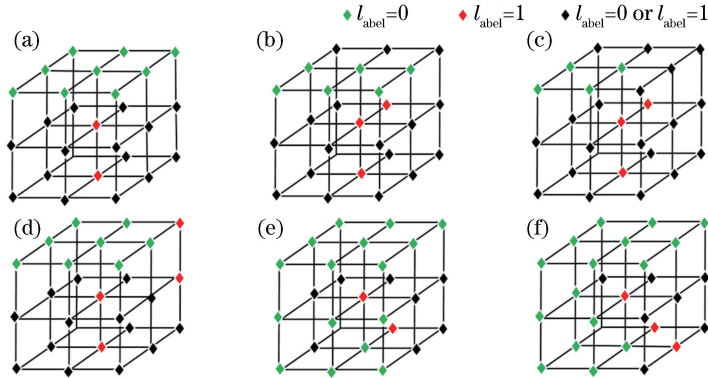


图2 “上”方向的6个删除模板

Fig. 2 Six delete templates in upper direction

删除模板是以当前目标体素 P 为中心的 $3 \times 3 \times 3$ 的体素集合,噪点滤除处理之后,每一个体素都有一个 l_{abel} 值,通过判断体素的 l_{abel} 值,可以将模板中的体素分为 A 类和 B 类两种, A 类体素指该体素的 l_{abel} 为固定值,标号为 0 或 1; B 类体素指 l_{abel} 不固定,标号也为 0 或 1。由于体素是三维立体的,每个模板可以顺时针旋转 90° 、 180° 和 270° ,得到的模板仍属于删除模板。为保证各向同性,需考虑上下左右前后 6 个方向(U、D、W、E、N、S),不同方向的模板可以通过相互翻转得到。若目标体素符合 6 个方向中的任何一个删除模板,即可进行删除, l_{abel} 值改为 0。图 2 为“上”方向的 6 个删除模板(不包含旋转之后的模板),其中中心点为需要判别的体素。

根据图 2 给出的删除模板,可以计算得到其余 5 个方向上,以及顺时针旋转 90° 、 180° 和 270° 后的所有模板。每个目标体素均与这些三维模板进行空间匹配,若与任意一个模板匹配,则删除该目标体素。当该处理多次循环迭代直至目标体素数量不再减少,即认为体素细化完毕。细化算法流程图如图 3 所示。

2.5 生成骨架

细化处理后的目标体素保留了树木的骨架结构,但相互连接关系尚未建立。骨架生成包括三步:

1) 通过计算体素内的所有点的坐标平均值,拟合出每一个体素内所有点云的几何中心点 S_p ,将其作为树木骨架中的骨架节点,可表示为

$$S_p(x, y, z) = \left(\frac{\sum_0^n x_i}{n}, \frac{\sum_0^n y_i}{n}, \frac{\sum_0^n z_i}{n} \right), \quad (7)$$

式中: $S_p(x, y, z)$ 为体素内部拟合的点的坐标;

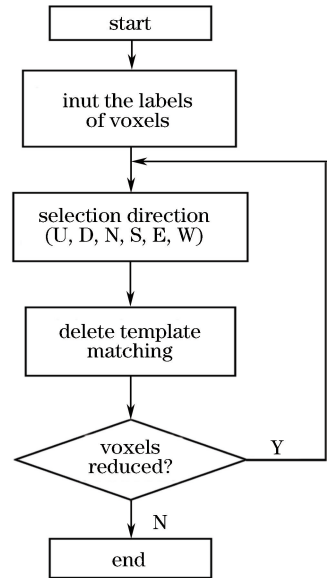


图3 细化算法流程图

Fig. 3 Flow chart of thinning algorithm

(x_i, y_i, z_i) 为体素内第 i 个点的坐标; n 为体素内所有点云的总个数。

2) 对体素进行 26 邻域搜索,判断哪些体素相互连接。

3) 确定体素之间的连接关系后,检索连接相邻体素内的骨架节点,构建树木骨架。图 4 为骨架节点连接示意图。

3 分析与讨论

本研究使用获取的银杏树和重瓣榆叶梅树的点云数据对算法进行验证。图 5 为银杏树与重瓣榆叶梅树的照片及骨架生成结果。图 5(a)左侧为银杏

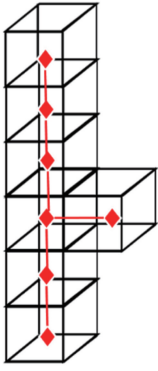


图4 骨架节点连接

Fig. 4 Connection of skeleton nodes

树照片,右侧为重瓣榆叶梅照片;图5(b)左侧为由地面三维激光扫描仪获取的银杏树的原始点云,右侧为由地面三维激光扫描仪获取的重瓣榆叶梅树的原始点云;图5(c)左侧为本文方法生成的银杏树骨架,右侧为本文方法生成的重瓣榆叶梅树骨架。树木骨架与点云数据在形态结构上相对一致,说明所提算法有一定的可行性和稳健性。

3.1 数据扫描精度对骨架生成的影响

本研究分别采用扫描角分辨率为 0.02° 、 0.05° 和 0.10° 时得到的银杏树木点云进行树木骨架生成。骨架生成结果如图6所示,角分辨率设置如表2所示。随着扫描精度的降低,得到的点云数据量减少,相应的运算时间也随之减少。采用本文方法处理3个角分辨率下采集到的数据,所得到主枝干的骨架无较大差别,但细小枝干处存在差异。显然,扫描角分辨率会影响点云数据量,从而影响骨架生成中的节点数量和骨架的精细程度,但对树干和主要枝条的骨架生成影响不大。

3.2 体素划分规格对骨架生成的影响

以重瓣榆叶梅树的数据进行参数分析。不同的 (M, N, K) 参数对骨架生成有一定影响。图7为不同划分规格下生成的重瓣榆叶梅树骨架,参数设置如表3所示。可以看出,划分的体素数量越多,得到的骨架节点越多,在骨架结构上表现有更多的枝干细节。同时,因为原始点云数据量一定,划分规格会出现饱和现象,即随着划分规格的增加,树木形态的保留程度不会出现明显的提高,图7(d)的划分程度比图7(c)的高,但生成骨架的形态无较大差别。

3.3 滤除参数对骨架生成的影响

算法中的滤除参数对树木骨架的生成也有一定影响。本研究根据体素内的点数与体素孤立程度进

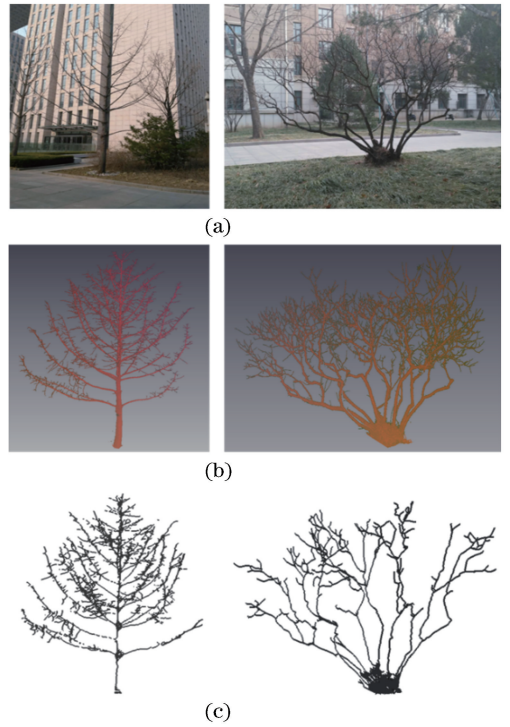


图5 银杏树和重瓣榆叶梅树的骨架生成结果。(a)左侧为银杏树实物图,右侧为重瓣榆叶梅树实物图;(b)左侧为银杏树的点云数据,右侧为重瓣榆叶梅树的点云数据;(c)左侧为银杏树生成骨架,右侧为重瓣榆叶梅树生成骨架

Fig. 5 Generated skeletons of Ginkgo tree and *Amygdalus triloba* f. *multiplex* tree. (a) Picture of Ginkgo on left and picture of *Amygdalus triloba* f. *multiplex* on right; (b) point cloud data of Ginkgo on left and point cloud data of *Amygdalus triloba* f. *multiplex* on right; (c) generated skeleton of Ginkgo on left and generated skeleton of *Amygdalus triloba* f. *multiplex* on right

行判定,设定不同的阈值时,相应保留的目标体素数目也不同,进而生成的树木骨架效果也不尽相同。本研究中 ϵ_1 分别取10和20,分析其对骨架生成的影响;树木骨架中应至少存在两个相连体素,所以体素邻域内目标体素小于2的一定是噪点体素,本研究中 ϵ_2 分别取2和4,分析其对骨架生成的影响。图8为不同 (ϵ_1, ϵ_2) 参数设置生成的骨架,参数设置如表4所示。可以看出,滤除参数越大,得到的骨架节点越少,运算时间越短。图8(a)和图8(b)为骨架节点孤立程度分析,可以看出, ϵ_2 阈值越大,滤除的点越多,但会在一定程度上破坏树骨架本身的结构;阈值较小时,在滤除孤立点的同时能够较好地保留骨架完整性。图8(c)和图8(d)为骨架节点点数阈值

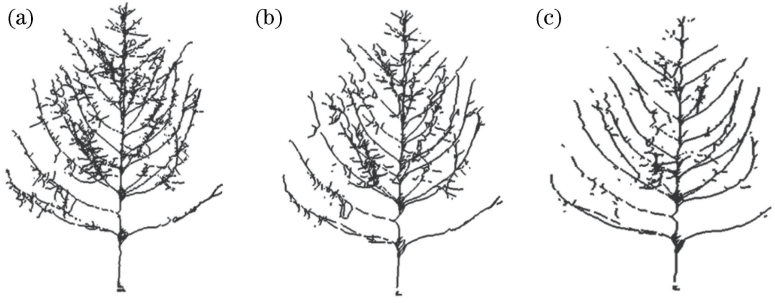


图 6 不同角分辨率下点云生成的银杏树骨架。(a) 0.02°; (b) 0.05°; (c) 0.10°

Fig. 6 Skeletons of Ginkgo tree generated by point clouds at various angular resolutions. (a) 0.02°; (b) 0.05°; (c) 0.10°

表 2 不同角分辨率下的银杏树骨架信息

Table 2 Skeleton information of Ginkgo tree with different angular resolutions

Angular resolution / (°)	Point number	Node number	Running time / ms
0.02	420492	2839	1183
0.05	65994	2138	647
0.10	17002	1393	392

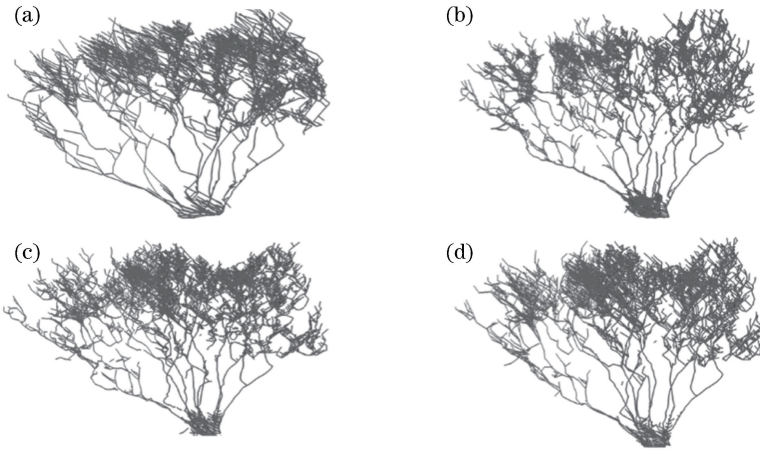


图 7 不同划分规格的重瓣榆叶梅树骨架生成结果。(a) (40, 40, 40); (b) (60, 60, 60); (c) (80, 80, 80); (d) (100, 100, 100)

Fig. 7 Generated skeletons of *Amygdalus triloba* f. multiplex tree with different partition specifications.

(a) (40, 40, 40); (b) (60, 60, 60); (c) (80, 80, 80); (d) (100, 100, 100)

表 3 不同划分规格的重瓣榆叶梅树骨架信息

Table 3 Skeleton information of *Amygdalus triloba* f. multiplex tree with different partition specifications

Point number	(M, N, K)	(ϵ_1, ϵ_2)	Node number	Running time / ms
686113	(40, 40, 40)	(0, 0)	3591	1193
686113	(60, 60, 60)	(0, 0)	6248	1755
686113	(80, 80, 80)	(0, 0)	8614	2810
686113	(100, 100, 100)	(0, 0)	11308	3730

表 4 不同滤除阈值的重瓣榆叶梅树骨架信息

Table 4 Skeleton information of *Amygdalus triloba* f. multiplex tree with different filtering parameters

Point number	(M, N, K)	(ϵ_1, ϵ_2)	Node number	Running time / ms
686113	(80, 80, 80)	(0, 4)	7188	2418
686113	(80, 80, 80)	(0, 2)	8233	2735
686113	(80, 80, 80)	(10, 0)	5331	1942
686113	(80, 80, 80)	(20, 0)	4358	1587

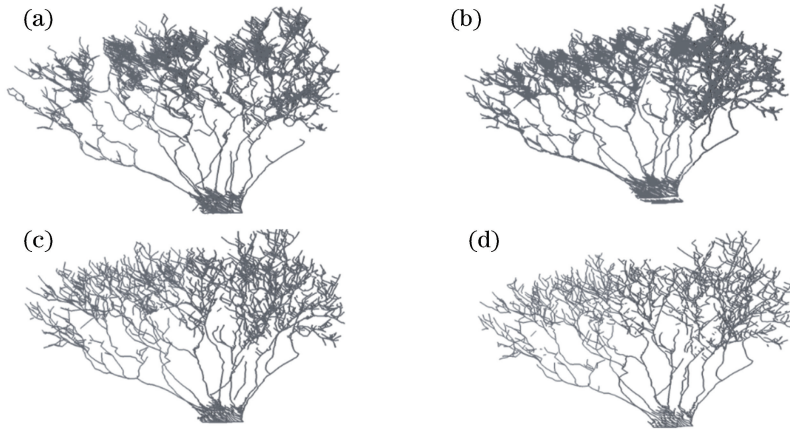


图 8 不同滤除参数的重瓣榆叶梅树骨架生成结果。(a) (0,4);(b) (0,2);(c) (10,0);(d) (20,0)
 Fig. 8 Generated skeletons of *Amygdalus triloba* f. multiplex tree with different filtering parameters.
 (a) (0,4); (b) (0,2); (c) (10,0); (d) (20,0)

判定,可以看出, ϵ 阈值越大,保留的骨架节点越少,滤除效果越明显。

3.4 算法效率比较

对本文方法(简称 IVTA 方法)与文献[26]中的方法(简称为 GSA 方法)进行比较,从生成的骨架节点数量与方法运行时间两个方面进行分析。图 9 所示为两种方法分别处理角分辨率为 0.02° 的银杏树和重瓣榆叶梅树的点云数据时生成的骨架结果。图 9(a)为 GSA 方法生成的银杏树骨架,图 9(b)为 GSA 方法生成的重瓣榆叶梅树骨架,图 9(c)为 IVTA 方法生成的银杏树骨架,图 9(d)为 IVTA 方

法生成的重瓣榆叶梅树骨架。表 5 中列出了 IVTA 方法和 GSA 方法处理点云数据生成树木骨架获得的骨架节点数和运行时间。显然,IVTA 方法相比 GSA 方法,在完整描述骨架的同时,能够生成更多的骨架节点,保留更多的枝干细节。而且,从表 5 可知,IVTA 方法的运行时间远低于 GSA 方法。对于实验中的银杏树,GSA 方法生成骨架的运行时间约为 IVTA 方法的 31 倍;而对于数据量更大的重瓣榆叶梅树,GSA 方法的运行时间约为 IVTA 方法的 67 倍。显然,相比 GSA 方法,IVTA 方法在运行效率上有很大提高。

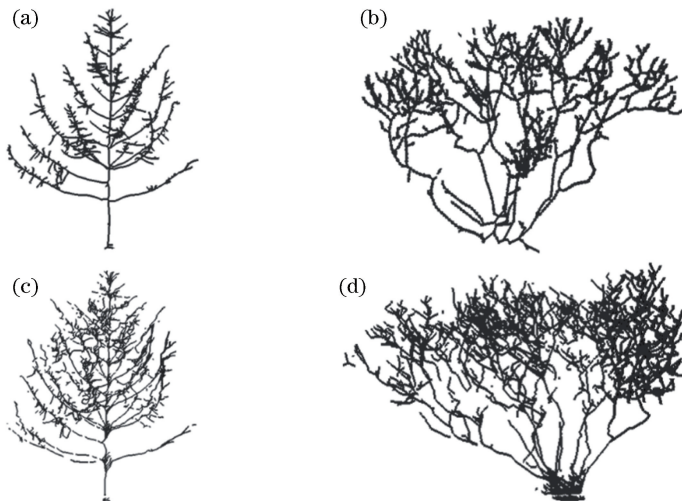


图 9 GSA 和 IVTA 方法下银杏树与重瓣榆叶梅树骨架生成结果。(a) GSA 方法生成的银杏树骨架;(b) GSA 方法生成的重瓣榆叶梅树骨架;(c) IVTA 方法生成的银杏树骨架;(d) IVTA 方法生成的重瓣榆叶梅树骨架
 Fig. 9 Tree skeletons of Ginkgo tree and *Amygdalus triloba* f. multiplex tree generated by GSA and IVTA methods.
 (a) Skeleton of Ginkgo tree generated by GSA method; (b) skeleton of *Amygdalus triloba* f. multiplex tree generated by GSA method; (c) skeleton of Ginkgo tree generated by IVTA method; (d) skeleton of *Amygdalus triloba* f. multiplex tree generated by IVTA method

表5 GSA 和 IVTA 生成的树骨架信息

Table 5 Tree skeleton informations generated by GSA and IVTA methods

Method	Tree	Point number	Node number	Running time /ms
GSA	Ginkgo	420492	1735	37165
GSA	<i>Amygdalus triloba</i> f. <i>multiplex</i>	686113	1953	100254
IVTA	Ginkgo	420492	2839	1183
IVTA	<i>Amygdalus triloba</i> f. <i>multiplex</i>	686113	4289	1493

4 结 论

提出了一种通过点云体素细化生成树木骨架的方法。通过构建体素空间,判断包含树木点云的目标体素和包含非树木点云的噪点体素;根据体素内的点数与体素邻域内目标体素的数量,对噪点体素进行滤除;利用细化算法对体素进行处理,得到树木骨架的单体素连接结构;最后通过现存的目标体素拟合骨架节点并连接,生成树木骨架。

基于获取的银杏树和重瓣榆叶梅树的点云数据对该方法进行验证。实验设置不同参数验证其对骨架生成的影响。实验结果表明,划分规格越大,单个体素的数量越多,保留的树木细节越多。体素内的点数与体素的孤立分布程度对噪点的滤除都有一定影响。点数滤除阈值比孤立分布阈值的滤除效果略好,对树木的枝干骨架生成效果的帮助更大。实验成功生成了树木骨架,且生成时间都在 4 s 以内。与 GSA 方法相比,本文方法能够大大减少生成树木骨架的运算时间,说明本文方法具有一定的有效性和可行性。

参 考 文 献

- [1] Berger M, Tagliasacchi A, Seversky L M, *et al.* A survey of surface reconstruction from point clouds [J]. *Computer Graphics Forum*, 2017, 36(1): 301-329.
- [2] Liu L X, Pang Y, Li Z Y, *et al.* Retrieving structural parameters of individual tree through terrestrial laser scanning data[J]. *Journal of Remote Sensing*, 2014, 18(2): 365-377.
刘鲁霞, 庞勇, 李增元, 等. 用地基激光雷达提取单木结构参数: 以白皮松为例[J]. *遥感学报*, 2014, 18(2): 365-377.
- [3] Li D, Pang Y, Yue C R. A review of TLS application in forest parameters retrieving [J]. *World Forestry Research*, 2012, 25(6): 34-39.
李丹, 庞勇, 岳彩荣. 地基激光雷达在森林参数反演中的应用[J]. *世界林业研究*, 2012, 25(6): 34-39.
- [4] Wang G, Sha C S, Wang J. Study on segmentation

of building facade considering local point cloud density [J]. *Laser & Optoelectronics Progress*, 2015, 52(6): 061001.

王果, 沙从术, 王健. 考虑局部点云密度的建筑立面自适应分割方法[J]. *激光与光电子学进展*, 2015, 52(6): 061001.

- [5] 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.
- [6] Chen T. Data processing methodology for laser measurement point cloud [J]. *Laser & Optoelectronics Progress*, 2011, 48(9): 091202.
陈田. 激光测量点云的数据处理方法研究[J]. *激光与光电子学进展*, 2011, 48(9): 091202.
- [7] Bu G C, Wang P. Adaptive estimation method for diameter at breast height based on terrestrial laser scanning [J]. *Laser & Optoelectronics Progress*, 2016, 53(8): 082803.
步国超, 汪沛. 基于单站地面激光雷达数据的自适应胸径估计方法[J]. *激光与光电子学进展*, 2016, 53(8): 082803.
- [8] Paulus S, Dupuis J, Mahlein A K, *et al.* Surface feature based classification of plant organs from 3D laserscanned point clouds for plant phenotyping [J]. *BMC Bioinformatics*, 2013, 14: 238.
- [9] Wu B, Yu B L, Yue W H, *et al.* A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data [J]. *Remote Sensing*, 2013, 5(2): 584-611.
- [10] Wu F C, Ma W C, Liang R H, *et al.* Domain connected graph: the skeleton of a closed 3D shape for animation [J]. *The Visual Computer*, 2006, 22(2): 117-135.
- [11] Saha P K, Borgefors G, di Baja G S. A survey on skeletonization algorithms and their applications [J]. *Pattern Recognition Letters*, 2016, 76: 3-12.

- [12] Li W W, Liu F, Gao L. A method of hand-shape recognition based on extraction of finger skeleton[J]. Journal of Central South University (Science and Technology), 2016, 47(3): 777-783.
李温温, 刘富, 高雷. 基于手指骨架提取的手形识别方法[J]. 中南大学学报(自然科学版), 2016, 47(3): 777-783.
- [13] Wang X H, Huang H Y, Chen C C, *et al.* Design and implementation of 3D geometrical tree modeling system based on terrestrial laser scanned point cloud data [J]. Journal of Fuzhou University (Natural Science Edition), 2014, 42(5): 705-712.
王晓辉, 黄洪宇, 陈崇成, 等. 基于激光点云的树木三维几何建模系统的设计与实现[J]. 福州大学学报(自然科学版), 2014, 42(5): 705-712.
- [14] Sun Z H, Lu S L, Guo X Y, *et al.* Surfaces reconstruction of plant leaves based on point cloud data [J]. Transactions of the Chinese Society of Agricultural Engineering, 2012, 28(3): 184-190.
孙智慧, 陆声链, 郭新宇, 等. 基于点云数据的植物叶片曲面重构方法[J]. 农业工程学报, 2012, 28(3): 184-190.
- [15] Wang C, Wu X H, Li Z W. Recognition of maize and weed based on multi-scale hierarchical features extracted by convolutional neural network [J]. Transactions of the Chinese Society of Agricultural Engineering, 2018, 34(5): 144-151.
王璨, 武新慧, 李志伟. 基于卷积神经网络提取多尺度分层特征识别玉米杂草[J]. 农业工程学报, 2018, 34(5): 144-151.
- [16] Zou W H, Chen Z Y, Ye X Z, *et al.* A new method for extracting feature skeleton from point cloud [J]. Journal of Zhejiang University(Engineering Science), 2008, 42(12): 2103-2107.
邹万红, 陈志杨, 叶修梓, 等. 一种新的点云数据特征骨架提取方法[J]. 浙江大学学报(工学版), 2008, 42(12): 2103-2107.
- [17] Bucksch A, Lindenbergh R, Menenti M. SkelTre: robust skeleton extraction from imperfect point clouds [J]. The Visual Computer, 2010, 26(10): 1283-1300.
- [18] Bucksch A, Lindenbergh R. CAMPINO: a skeletonization method for point cloud processing[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2008, 63(1): 115-127.
- [19] Ramamurthy B, Doonan J, Zhou J, *et al.* Skeletonization of 3D plant point cloud using a voxel based thinning algorithm [C] // 23rd European Signal Processing Conference (EUSIPCO), August 31-September 4, 2015, Nice, France. [S. l.: s. n.], 2015: 2686-2690.
- [20] Yan D M, Wintz J, Mourrain B, *et al.* Efficient and robust reconstruction of botanical branching structure from laser scanned points [C] // 2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, August 19-21, 2009, Huangshan, China. New York: IEEE, 2009: 572-575.
- [21] Xu H, Gossett N, Chen B Q. Knowledge and heuristic-based modeling of laser-scanned trees [J]. ACM Transactions on Graphics, 2007, 26(4): 19.
- [22] Delagrangé S, Rochon P. Reconstruction and analysis of a deciduous sapling using digital photographs or terrestrial-LiDAR technology [J]. Annals of Botany, 2011, 108(6): 991-1000.
- [23] Tang L Y, Zhang H, Huang H Y, *et al.* Improved method for 3D reconstruction of tree model based on point cloud data [J]. Transactions of the Chinese Society for Agricultural Machinery, 2017, 48(2): 186-194.
唐丽玉, 张浩, 黄洪宇, 等. 基于点云数据的树木三维重建方法改进[J]. 农业机械学报, 2017, 48(2): 186-194.
- [24] Zhao C J, Yang L, Guo X Y, *et al.* Corn skeleton reconstruction based on stereo vision [J]. Transactions of the Chinese Society for Agricultural Machinery, 2010, 41(4): 157-162.
赵春江, 杨亮, 郭新宇, 等. 基于立体视觉的玉米植株三维骨架重建[J]. 农业机械学报, 2010, 41(4): 157-162.
- [25] Li W Y, Liu C, Wu H B, *et al.* Single tree skeleton extraction with sparse point cloud data [J]. Remote Sensing Information, 2014, 29(6): 84-89.
李巍岳, 刘春, 吴杭彬, 等. 一种稀疏点云环境下的单株树骨架提取算法[J]. 遥感信息, 2014, 29(6): 84-89.
- [26] Li R H, Bu G C, Wang P. An automatic tree skeleton extracting method based on point cloud of terrestrial laser scanner [J]. International Journal of Optics, 2017, 2017: 5408503.
- [27] Palágyi K. A 3D fully parallel surface-thinning algorithm [J]. Theoretical Computer Science, 2008, 406(1/2): 119-135.
- [28] Palágyi K, Balogh E, Kuba A, *et al.* A sequential 3D thinning algorithm and its medical applications [M] // Insana M F, Leahy R M. Lecture notes in

- computer science. Berlin, Heidelberg: Springer, 2001, 2082: 409-415.
- [29] Zhang C F, Yang G W, Yue M M. Improving of Zhang parallel thinning algorithm [J]. Information Technology and Informatization, 2016(6): 69-71.
张翠芳, 杨国为, 岳明明. Zhang 并行细化算法的改进[J]. 信息技术与信息化, 2016(6): 69-71.
- [30] Bao J J, Fan J. Robust parallel thinning algorithm for binary images [J]. Computer Aided Engineering, 2006, 15(4): 43-46.
包建军, 樊菁. 鲁棒的二值图像并行细化算法[J]. 计算机辅助工程, 2006, 15(4): 43-46.
- [31] Palágyi K, Kuba A. A parallel 3D 12-subiteration thinning algorithm [J]. Graphical Models and Image Processing, 1999, 61(4): 199-221.
- [32] Palágyi K, Kuba A. A 3D 6-subiteration thinning algorithm for extracting medial lines [J]. Pattern Recognition Letters, 1998, 19(7): 613-627.