

复杂拓扑结构的树木枝干重建算法

张 冬, 云 挺, 薛联凤, 罗 毅

(南京林业大学信息科学技术学院, 江苏 南京 210037)

[摘要] 具有复杂拓扑结构的树木枝干重建问题是国内外研究的一个热点和难点. 本文提出了一种有效且鲁棒的树木枝干重建算法. 首先在原始树木点云上建立基于黎曼流形的 Delaunay 邻域关系, 然后将所有顶点当作位置约束加 Laplace 方程, 再迭代地解 Laplace 方程将点云收缩到我们预想的程度, 然后利用聚类 and 连接算法得到一个初步的树木枝干, 最后再通过修复得到最终的树木枝干. 本文的算法在对含笑树和樱花树上进行了验证, 实验结果表明该算法有很好的重建效果.

[关键词] Laplace 变换, 枝干重建, 激光点云处理, 黎曼流形

[中图分类号] TP391.9 [文献标志码] A [文章编号] 1001-4616(2015)01-0128-09

Reconstruction Algorithm with Complex Topology of Tree Branches

Zhang Dong, Yun Ting, Xue Lianfeng, Luo Yi

(Department of Information Science and Technology, Nanjing forestry University, Nanjing 210037, China)

Abstract: Currently, the problem of branches of trees with complex topology reconstruction is a hot and difficult domestic and international research. In this paper, we proposed an effective and robust algorithm for extraction curve-skeletons from point clouds. Firstly based on Riemannian manifolds Delaunay neighborhood relations, and constructed a Laplace matrix. We treated all points as positional constraints. We solved and updated the discrete Laplace system iteratively, until all points contracted to the positions we needed. Then we employed the Principle Component Analysis (PCA) to differentiate between joints and branches of the contracted points. We clustered the two kinds of regions separately to get the key nodes. Then we connected these key nodes by the connection surgery we proposed to get a raw curve-skeleton of the given point cloud. We constructed a graph on the curve-skeleton, and computed the Minimum Spanning Tree (MST). Finally, we refined the MST and gained the final curve-skeleton.

Key words: the Laplace transform, branches reconstruction, laser point cloud processing, Riemannian manifold

1 相关工作

模型的重建工作在计算机图形学、计算机可视化等领域的应用中是一个非常基本的问题. 目前, 具有简单拓扑结构的模型重建已有多种相对成熟的方法. 例如有基于 Voronoi 图的几何方法和 Level-set 方法. Aujay 等人^[1]在 2007 年提出的 Voronoi 图的几何方法能直接处理点云数据, 但其显著缺点是对噪声数据比较敏感. 由 Sethian 和 Osher^[2]于 1988 年提出的 Level-set 方法同样受噪声数据的影响, 而且由于其提取的骨架常丢失一些模型特征, 所以显得较为粗糙. 由于树木的拓扑结构非常地复杂, 如何对树木枝干进行有效的重建模拟是亟待解决的问题, 如果重建出来的枝干过于粗糙则会不足以表现树木的复杂特征, 枝干过于细节化过多则可能导致模型过于复杂, 从而造成实用性不大的劣势.

本文提出了一种新的可处理具有复杂拓扑结构的树木枝干重建算法, 大致可以分为以下几个步骤: 首先进行原始点云三角化. 先根据黎曼流形 Delaunay 三角化方法建立原始点云之间的邻域关系. 然后计算点云的拉普拉斯矩阵. 根据第一步建立的连接关系, 构建拉普拉斯加权矩阵. 接着解拉普拉斯方程. 将所有点看成是全局的位置约束加入矩阵, 解一个线性最小平方问题, 得到新的顶点的位置并更新拉普拉斯矩

收稿日期: 2014-06-19.

基金项目: 国家自然科学基金(31300472)、江苏省自然科学基金(BK2012418).

通讯联系人: 薛联凤, 副教授, 研究方向: 图像处理. E-mail: 285201972@qq.com

阵的收缩权和约束权. 然后计算前后两次点云的总面积比率, 决定继续迭代还是结束. 若小于 0.1 则结束迭代, 否则继续迭代. 接着顶点聚类 and 连接得到初步的树木枝干骨架. 最后计算初步树木枝干骨架的最小生成树并修复该树得到最终的树木枝干.

与现有的树木枝干重建算法的方法比较, 本文的算法对噪声数据不是很敏感, 能直接处理具有复杂拓扑结构的点云并且对于具有残缺的点云数据也有很好的重建效果, 所重建出来的树木枝干稳定性比较好.

2 黎曼流形 Delaunay 三角化

基于黎曼流形 Delaunay 三角化的一个重要研究领域是生成算法. 目前存在的文献主要集中解决以下这两类问题: (1) 在黎曼流形的研究区域内指定黎曼度量, 然后在参数空间中按照给定的黎曼度量生成网格, 其要解决的核心问题就是计算研究区域的测地距离. (2) 直接在黎曼流形上生成 Delaunay 三角化, 采用的方法多为波前推进法.

2.1 黎曼流形

定义 1 给定一个有 n 个顶点的点云数据, 设顶点集合 M 是该点云数据集合的一个三维拓扑流形, A 是 M 的一个 C 微分结构, 则称 (M, A) 是一个三维 C 微分流形. C 微分流形也称为光滑流形.

定义 2 给定一个有 n 个顶点的点云数据, 设顶点集合 M 是一个三维光滑流形, g 是顶点集合 M 上一个光滑的二阶协变张量场, 假如 g 是对称、正定的, 即对于每一个数据点 $p \in M$, $g(p)$ 是切空间 $T_p M$ 上的一个对称、正定的二阶协变张量, 则称该二阶协变张量场 g 是顶点集合 M 上的一个黎曼度量, 指定了黎曼度量的光滑流形称为黎曼流形, 记为 (M, g) , 黎曼流形中具有了度量. 因此, 在黎曼流形上存在有距离的概念, 而在其他流形上则没有.

2.1.1 测地线

黎曼流形中最值得研究的曲线是测地线, 测地线是黎曼流形上两点之间的局部最短距离, 它是黎曼流形的内蕴几何特征. 黎曼流形上测地线和测地距离的定义如下:

定义 3 仿射联络空间 (M, D) 中的一条光滑曲线 $\gamma: I \rightarrow M$ 称为测地线, 如果它的切向量 $\gamma'(t)$ 沿 γ 是平行的切向量场, 即

$$\frac{D}{dt} \left(\frac{d\gamma}{dt} \right) = D_{\gamma'} \gamma'(t) = 0. \quad (1)$$

定义 4 设 (M, g) 是一个三维黎曼流形, p 和 q 分别为 M 上的两点, 则 p 到 q 的测地距离为 M 上连接 p 和 q 的最短测地线的长度, 记为 $d(p, q)$. 若用 $\gamma: [a, b] \rightarrow M$ 表示 M 上连接 p 和 q 的最短测地线, 则

$$d(p, q) = \int_a^b |\gamma'(t)| dt = \int_a^b \sqrt{-\gamma'(t), \gamma'(t)} dt. \quad (2)$$

2.1.2 Delaunay 三角化

黎曼流形 Delaunay 三角化是欧氏空间 Delaunay 三角化在黎曼空间的自然推广. 欧氏空间 Delaunay 三角化的定义不再赘述. 黎曼空间中研究 Delaunay 三角化是以测地距离为度量标准. 下面给出黎曼流形的 Delaunay 三角化的定义.

定义 5 设 M 是一个三维黎曼流形, s 为 M 上的有限点集, $T(s)$ 是 s 的一个三角网格, 如果对于任意一个 $t, t \in T(s)$ 都存在外接测地线 D , 并且满足 $D \cap s = \emptyset$, 则 $T(s)$ 是点集的一个 Delaunay 三角化.

2.2 基于坐标卡的生成算法

本文采用黎曼流形的观点描述研究对象, 并基于坐标卡生成三维黎曼流形的网格. 该方法分为两个步骤: 采用黎曼流形的观点描述对象, 即: 黎曼流形的 Delaunay 三角化. 根据初始数据建立坐标卡, 坐标卡覆盖流形的整个区域. 以坐标卡为基础, 按照指定的度量要求, 采用弹性力学模型在流形上均匀布点, 并根据分布点生成 Delaunay 网格.

设研究对象为三维黎曼流形 M , 描述它的初始数据是三角网格 TN , 它具有任意亏格, 下面将建立该初始网络的坐标卡 (U, φ) , 并进一步生成 Delaunay 三角网格.

2.2.1 建立坐标卡

(1) 坐标卡的几何形态. 在初始网格 TN 上取一点 a , 找到其周围的点并与 o 组成点集 PS . 以 o 为原

点, PS 的最小二乘面为 XY 平面, 建立局部坐标系 LC . 在 LC 内采用距离加权法生成 PS 的最小二乘曲面 $F(u, v) = a_1 u^2 + a_2 v^2 + a_3 uv + a_4 u + a_5 v + a_6$, 将曲面与原始数据的拟合误差控制在 e 范围内, 则坐标卡的几何形态可由 $F(u, v)$ 描述, 局部坐标系上黎曼流形与欧氏空间的映射采用垂直映射.

(2) 坐标卡的范围. 欧氏空间中, 坐标卡的范围采用椭圆 E 描述, 设 o 点处的黎曼度量所确定的黎曼椭圆为 E_c , 则 E 的长轴(短轴)方向与 E_c 的短轴(长轴)方向一致, E 的长短轴比例与 E_c 的比例相同, 这样描述范围的优势是坐标卡能够更精确地拟合原始数据. 坐标卡在黎曼流形上的范围由函数 $F(u, v)$ 决定, 其定义域为 E . 此外, 要求坐标卡在黎曼流形上的范围落在原点 o 的测地凸邻域内, 以保证坐标卡内 Delaunay 三角化的存在性.

(3) 转换函数. 黎曼流形 M 的全局坐标系为 GC , 设全局坐标系 GC 到局部坐标系 LC 的映射为 T , 局部坐标系 LC 到全局坐标系 GC 的映射为 T^{-1} . 可以推出, 黎曼流形 M 上坐标卡覆盖的区域 U 到欧氏空间区域 E 的映射为 $\varphi = F^{-1} \circ T$, 欧氏空间区域 E 到黎曼流形上区域 U 的映射为 $\varphi^{-1} = T^{-1} \circ F$. 坐标卡 (U_i, φ_i) 与 (U_j, φ_j) 之间的转换函数定义为 $\psi_{ij} = \varphi_j \circ \varphi_i^{-1}$.

(4) 坐标卡集. 后继坐标卡的局部坐标原点选在已存在坐标卡范围边界上, 并记录坐标卡之间的连接关系. 算法主要步骤如下: ①在初始网格 TN 上取一点 o ; ②找到点集 PS , 并求得坐标卡的几何形态 F ; ③定义坐标卡的映射函数、混合函数, 并计算它的范围; ④求得坐标卡集合, 并定义黎曼流形的混合函数.

2.2.2 生成网格

(1) 布点

对于布点有 2 个要求: ①要满足指定的黎曼度量; ②最终点分布要均匀, 满足网格的存在条件. 为此, 本文采用弹性力学布点. 假设给定黎曼流形 M 定义域上的黎曼度量控制函数为 $g(U, V)$, 按以上要求, 布点步骤如下:

①在坐标卡 (U_i, U_i) 的局部坐标系中初始布点, 同时考虑了度量函数 $g(U, V)$;

②根据转换函数 ψ_{ik} , 将坐标卡 (U_i, U_i) 中的点映射到它的连接卡 (U_k, φ_k) 中, 其中 $k = 1, \dots, n$, 表示 n 个连接卡;

③对所有的坐标卡执行①和②步骤, 直到所有卡中初始布点结束;

④采用弹性力学模型, 以坐标卡为单位迭代计算黎曼流形上每个点的位置, 直到所有点均匀分布. 布点时采用测地距离计算, 测地距离可以采用数值的方法计算, 也可以采用简化的方法, 数值方法计算结果更准确些, 速度也会慢些.

(2) 生成网格

在布点基础上, 生成黎曼流形的 Delaunay 三角网格, 步骤如下:

①在坐标卡 (U_i, U_i) 的局部坐标系中, 采用 Delaunay 空洞算法生成三角网格, 记三角网格为 ξ_i ;

②根据转换函数 ψ_{ik} , 将坐标卡 (U_i, φ_i) 中的三角形映射到它的连接卡 (U_k, φ_k) 中, 其中 $k = 1, \dots, n$, 表示 n 个连接卡;

③对所有坐标卡执行步骤①和②, 直到生成黎曼流形的 Delaunay 三角网格, 记作 τ_i ;

④记 Delaunay 三角网格顶点 v_i 的环邻域内 k 个原始点为该点的邻域, 记作 $N_i = \{\text{proj}(v_{i,1}), \text{proj}(v_{i,2}), \dots, \text{proj}(v_{i,k})\}$.

3 曲面重建

曲面的表示和处理是计算机图形学中十分重要的课题. 简单地来说, 三维模型的表示方式基本就决定了它能用到的技术和应用领域. 例如比较流行的逐片线性曲面表示, 如何显示三角网格、如何研究其拓扑性质等都有直接的方法. 另一方面, 三角网格表示在许多造型需求上显得不够好. 而且, 通常扫描得到的曲面非常复杂, 经常会带有噪声, 这需要滤波、重采样和压缩等几何处理手段来处理, 使得其方便进行存储和传输.

3.1 拉普拉斯算子和曲面微分表示

3.1.1 拉普拉斯算子定义

假设 $M = (V, E, F)$ 是一个有 n 个顶点的三角网格, 其中 V 表示顶点集合, E 表示边集合, F 表示面集合, δ_i 是点 v_i 的微分表示形式:

$$\xi_i = (\xi_i^{(x)}, \xi_i^{(y)}, \xi_i^{(z)}) = \sum_{j \in N(i)} w_{ij} (v_i - v_j). \quad (3)$$

$$w_{ij} = \frac{\omega_{ij}}{\sum_{(i,k) \in E} \omega_{ik}}. \quad (4)$$

其中 $N(i) = \{j | (i, j) \in E\}$.

一般拉普拉斯矩阵中对于权重的选取有以下两种:

$$(1) \omega_{ij} = 1,$$

$$(2) \omega_{ij} = \cot \alpha + \cot \beta.$$

其中(1)表示的是均匀权,此时

$$\delta_i = \frac{1}{d_i} \sum_{j \in N(i)} (V_i - V_j). \quad (5)$$

称作 Uniform Laplace 算子,上面式子是积分 $\frac{1}{|\gamma|} \int_{v \in \delta} (V_i - V_j) dl(v)$ 的离散表示形式,其中 γ 是点云曲面上绕 v_i 的任意封闭的点云曲线. 同是 $|\gamma|$ 长度,利用微分几何学中的知识可知

$$\lim_{|\gamma| \rightarrow 0} \frac{1}{|\gamma|} \int_{v \in \delta} (V_i - V_j) dl(v) = -H(v_i) \mathbf{n}_i. \quad (6)$$

其中 $H(v_i)$ 是顶点 v 处的平均曲率, \mathbf{n} 是该点法向量. 因此,微分坐标向量的方向是局部法向方向的近似,大小和局部平均曲率的比例近似. 直观上,这意味着 δ 坐标包含局部点云曲面形状的性质. 该权值比较适合于均匀的点云数据,对于复杂且分布不规则的点重建效果不是很好. (2)表示的是余切权, Meyer 等人提出了用余切权重代替均匀权. 此时

$$\delta_i = \sum_{j \in N(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (v_i - v_j). \quad (7)$$

称作 Cotangent Laplace 算子,是对 Laplace 算子采用与几何信息相关的离散方法. 由于余切权可能是负数,并且当角度比较大时会带来一些问题.

3.1.2 点的势函数

在文中我们利用其他方法来计算点之间的权值,我们都知道虽然散乱数据点的分布没有规律,但是数据点的分布和相互作用决定了重构曲线的形状和走向,每个数据点对重构曲线的贡献受其他数据局点分布的影响,散乱 S 中的数据点 p' 对数据点 p 的影响用如下函数表示:

$$f(P) = \frac{e^{-\frac{|p-p'|^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}. \quad (8)$$

其中 δ 是调节变量. 在式中,数据点 p' 与点 p 距离越远,函数值越小,说明这两个数据点的相互影响越小. 数据点集 S 中所有数据点对点 p 影响的总和称为点 p 的势函数值. 定义数据点 p 的势函数如下:

$$F(P) = \sum_{p_i \in S} \frac{e^{-\frac{|p-p'|^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}. \quad (9)$$

在上述定义的函数关系下,数据点的势函数值越大,表明该点对重构曲线的贡献越大;反之,对重构曲线的贡献越小,作为干扰点的可能性越大. 局部势函数值较大的数据点形成的峰基本反映了数据点集的形状和走向. 式中 δ 是一个可变参数,势函数形成的曲面形状随 δ 的变化而变化. 实验发现 δ 值较大时,势函数图中峰比较平滑. 峰过于平滑或者尖锐对重构曲线都是不利的. δ 取值为采样密度时,重构曲线效果较好. 数据点集 S 的采样密度指:对数据点集 S 内任意数据点 P' ,都存在另外一个数据点 P_j ,使得 $P_j - P_i \leq \delta$ 成立. 采样密度 δ 由下列方法估计得到:在数据点集 S 中随机选取 m_0 个数据点 p ,得到每一个点的 k -邻域,计算每一个点到该点 p 的距离,记为 s_i ,则采样密度为:

$$\delta = \frac{1}{m_0} \sum_{i=1}^{m_0} S_i. \quad (10)$$

由以上的分析可知,势函数值大的数据点对原始点云重构曲线的贡献较大. 在连通图 Graph' 中,这些

点相连的边距离重构曲线较近,是要最终进行搜寻的目标. 根据数据点集的分布情况和 Delaunay 三角网格的分布性质相似,这些边的长度也较短. 利用图论中求解最短路径的方法求解这些边,要求距离重构曲线近的边权值小,因此定义 Graph' 中边的权值与该边两个端点的势函数值成反比,与该边的长度成正比:

$$\text{Weight}(\langle P_i, P_j \rangle) = \frac{\|P_i - P_j\|}{F(P_i) + F(P_j)}. \quad (11)$$

3.1.3 微分坐标重建曲面

我们用上面定义点云曲面的微分表示,就不同的问题,对微分坐标做不同的计算,最终进行点云曲面重建. 这是因为 Laplace 矩阵 L 是奇异的. $\text{rank}(L) = n - k$, 其中 k 是 M 的连通分支数. 假设 M 是连通的,即 $k=1$, 那么需要添加一个顶点的笛卡尔坐标来约束增加矩阵的秩. 给出 M 上顶点的位置作为约束条件,通常是多个顶点作为约束. 即 c 为空间位置已知的顶点的下标集合, 因此有 $|c|$ 个额外的方程

$$x_j = c_j, \quad j \in c \quad \delta = \frac{1}{m_0} \sum_{i=1}^{m_0} S_i.$$

c 是顶点 v 的分量的坐标不失一般性, 假设 $c = (1, 2, \dots, m)$, 则线性方程组的形式如下:

$$\left\{ \begin{array}{l} L \\ wI_{m \times n} \end{array} \right\} x = \left\{ \begin{array}{l} wc_{lm} \\ \delta^x \end{array} \right\}. \quad (12)$$

记上式中左边的系数矩阵为 L . 增加的约束使得线性方程组是超定的(方程数多于未知变量数), 一般来说方程组无解. 但是这个方程是满秩的, 解这个方程组得到的是最小平方意义下的解:

$$\tilde{x} = \text{argmin}(\|Lx - \delta\|^2 + \sum_{j \in c} \omega^2 |x_j - c_j|^2). \quad (13)$$

实验结果表明,通过这个方法得到的结果还是较为满意的.

3.2 基于拉普拉斯算子的曲面重建算法

目前在三维网格模型或者体素化模型上提取骨架的方法比较多,而直接在点云上提取骨架的方法较为罕见. 本章提出了一个有效并且鲁棒的点云曲线骨架提取方法. 该方法的特点是仅利用了点云的几何位置信息来提取骨架. 大致步骤如下: 首先根据黎曼流形的方法来建立散乱点之间的连接关系, 根据这个连接关系, 我们构建拉普拉斯加权矩阵, 并且将所有点看成全局的位置约束, 通过解一个线性最小平方问题, 将点云进行收缩, 收缩的结果是使得点云初步到达其自然骨架附近. 接着根据本文提出的一种聚类点云的方法, 对收缩后的点云进行聚类. 将聚类后的点看作关键点, 然后根据本文的连接手术将这些关键点连接成线, 得到初步的曲线骨架. 将初步的曲线骨架看成一个图, 计算该图的最小生成树, 最后修复该最小生成树从而得到最终的曲线骨架.

3.2.1 构建拉普拉斯矩阵

经过前几步得到了每个顶点的邻域后, 我们就可以根据这些邻域信息构建拉普拉斯加权矩阵. 通常拉普拉斯矩阵会有权值, 在这里我们取由点之间的势函数所计算的边权:

$$\omega_{ij} = (\langle p_i, p_j \rangle) = \frac{\|p_i - p_j\|}{F(p_i) + F(p_j)}. \quad (14)$$

3.2.2 点云收缩

当得到拉普拉斯矩阵, 继而得到一组点云顶点的微分坐标 $\delta = LV = [\delta_1^T, \delta_1^T, \dots, \delta_1^T]^T$ 后, 我们希望找到一组新的顶点 v' , 使得 $LV' = 0$. 然而 L 是奇异的. 为了防止得到零解我们将顶点的位置赋以权值作为软约束. 于是我们将解方程

$$LV' = 0. \quad (15)$$

转化为解方程

$$\begin{pmatrix} W_L L \\ W_H \end{pmatrix} V' = \begin{pmatrix} 0 \\ W_H L \end{pmatrix}. \quad (16)$$

其中 W_L 和 W_H 均为对角矩阵. W_L 控制收缩的力度, 控制保持原有位置的力度. $W_L (W_H)$ 的第 i 个对角线元素记为 $W_{L,i}$ 、 $W_{H,i}$, 注意该方程是超定的, 这样我们得出的解是在最小平方意义下的解, 等价于极小化下面的函数.

$$\| \mathbf{W}_L \mathbf{L} \mathbf{V} \|^2 + \sum_i \mathbf{W}_H^2 \| V_i' - V_i \|^2. \quad (17)$$

第一项对应收缩约束,第二项对应保形约束。

通过解拉普拉斯方程,将点云上的点沿着我们估计出来的法向方向移动一定的步长,这样做的结果是将点云的细节部分和噪声部分进行了光滑。如果我们不加任何顶点约束,那么解的结果是将所有点收敛到一个点上,这不是我们想要的结果。因此需要给拉普拉斯方程增加约束,具体我们将点云上所有点看成是位置约束点加入方程组。

解一次该最小化方程,通常不能一下子将点云收缩到我们满意的位置。我们需要用迭代的方法重复多次,直到点云上的点收缩到可以接受的程度。通过前几次解方程后,去掉了一些高频的细节和噪声,新的点云会有一个较明显的收缩。但是在每次解完方程后,下一步的迭代过程中如果保持权值不变,对点云的继续收缩将不再有效果,点云仍然会保持在原来的位置上。因此要能继续收缩点云,我们需要每次更新这两个权值。具体而言,每次迭代前,要能继续收缩,对每一个顶点增加其收缩权 \mathbf{W}_L 的值。但是为了防止过度地收缩,也要根据该顶点收缩的程度,确定其约束权 \mathbf{W}_H 的值。顶点收缩的程度我们用其邻域面积确定。计算出每个顶点在收缩前的邻域面积和收缩后邻域面积的差,对于差值越大的点,也就是收缩程度越大的顶点,给它赋以更大的位置约束权。相反,对于差值越小的点,给的位置约束权也相对较小。收缩算法的步骤如下:

(1) 对第 1 次迭代,解方程 $\begin{pmatrix} \mathbf{W}_L \mathbf{L} \\ \mathbf{W}_H \end{pmatrix} \mathbf{V}' = \begin{pmatrix} 0 \\ \mathbf{W}_H \mathbf{L} \end{pmatrix}$, 得至新的顶点位置 \mathbf{v}^{t+1} 。

(2) 更新权值 $\mathbf{W}_L^{t+1} = sl \mathbf{W}_L^t$, $\mathbf{W}_H^{t+1} = \mathbf{W}_H^0 \sqrt{A_i^0 / A_i^t}$ 。其中 A_i^0 和 A_i^t 分别是顶点域的面积和最初邻域的面积。

(3) 根据第 1 步解得的新的顶点的位置 \mathbf{v}^{t+1} , 构建新的拉普拉斯矩阵 \mathbf{L}^{t+1} 步, 返回第一步。

在我们的实验中,计算顶点的 k -邻域时选取 $k = 0.012 |v|$, 其中 $|v|$ 是点云的数量。初始的收缩权和约束权 $\mathbf{W}_L^0 = 1.0$, $\mathbf{W}_H^0 = 1.0$, 取 $S_L = 2.0$ 或者 $S_L = 3.0$ 。实验表明这样选取的权值通常不到 10 次的迭代, 就可以将点收缩到较好的位置, 如果 S_L 越大, 则每次收缩的幅度越大, \mathbf{W}_H^0 越大, 则迭代的次数会更多。

3.2.3 点云聚类

收缩后的点云在分支处和节点处的点的密集程度不同。一般在节点处点云分散地较大。我们需要区分分支和节点的点云, 并且将节点处的点云尽可能地再次收缩。

识别出节点, 将这些节点用其重心代替。整个点云基本上成线状分布, 在该点云上做聚类。算法简要步骤如下:

(1) 设定一个邻域值 K 。从任意一个点出发, 找出其 K -邻域, 将其放入一个聚类当中, 并打上标记。

(2) 判断是否所有点都在某个聚类中, 如果是则停止迭代, 转第 4 步; 否则转到第 3 步。

(3) 再接着找下一个点和它的 K -邻域, 如果这些点已经属于某个聚类, 则跳过, 继续找下一个; 否则标记为新的聚类。

(4) 将聚类里的点用其重心坐标代替, 从而得到一些关键点。

聚类的过程中我们将关键点和原始点云的对应关系做了存储。这样在后面的连接中根据原始点云的连接关系做粗连接。

3.2.4 连接关键点

(1) 粗连接

将收缩的点云聚类后得到了一些关键点。利用这些关键点, 我们构造一个无向图 G 。因为我们保存了聚类点和原始点云之间的对应关系, 所以关键点的连接关系可以通过计算其与原始点云对应的点的连接关系。当两个关键点对应的原始点云上的两组点集的 Delaunay 邻域有重合的部分, 则认为这两组点集是相邻关系, 继而将对应的两个关键点视为相邻。这样连接的一个问题是会出现 3 个或者更多个关键点相互连接的情况。这是因为这些点对应的原始点云的 Delaunay 邻域都有重合的部分。这样的方法不会有遗漏有效采样点的问题出现。

(2) 最小生成树简化连接

粗连接根据上一步建立得到的无向图 G , 为每一条边计算其长度作为它的权值。计算该图的最小生成树从而得到点云的初步曲线骨架。但是由于最小生成树不存在封闭的环, 这对于有亏格的点云, 计算得到

的曲线骨架的拓扑不能保持一致. 因此这样的曲线骨架需要进一步修正.

(3)修正曲线骨架

通过前两步的操作,有可能出现 3 个或者更多个关键点相互连接的情况. 对于这样的问题,我们很容易找出最小生成树的所有叶子节点,一一判断每两个叶子节点是否在原来的粗连接中. 如果在粗连接中,则将它们再次相连.

4 实验和比较

本节给出实验的结果和分析. 本节所有的实验均运行于 AMD E-350 Processor 1.06 GHz CPU、4 GB 内存的 PC 机上.

1)实验结果分析

图 1 是在樱花树上进行的实验验证图,从(a)到(c)收缩权值更新的力度取值分别为 0.5、1、2,更新次数为 2 次,从图中可以看出当更新的力度取值为 2 时,所获得的中间效果比较好,基本上也能反映出原始树木枝干的轮廓. 图 2 是在含笑树上的实验图,从(a)到(c)控制收缩的力度取值分别为 0.5、1、2,更新次数为 2 次,从图中可以看出当收缩的力度取值为 1 时,所获得的效果比较好. 图 3 是在樱花树上的实验图,从(a)到(c)控制保持原有位置的力度取值分别为 1、3、5,从图中可以看出当取值为 3 的时候重建的效果比较好.

(1)不同权值对实验结果的影响

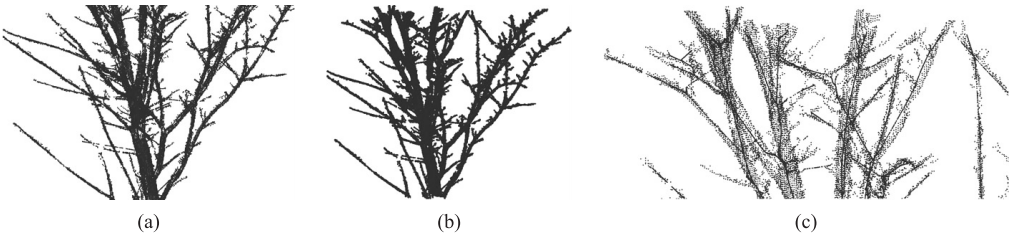


图 1 权值 S_L 的影响

实验中取 $W_L^0 = 1.0, W_H^0 = 1.0$,迭代次数为 2 次,从图(a)到(c) S_L 取值分别为 0.5、1、2

Fig. 1 The impact of S_L

$W_L^0 = 1.0, W_H^0 = 1.0$ and iterate 2 times. From figure(a) to(c), S_L is 0.5, 1, 2 separately

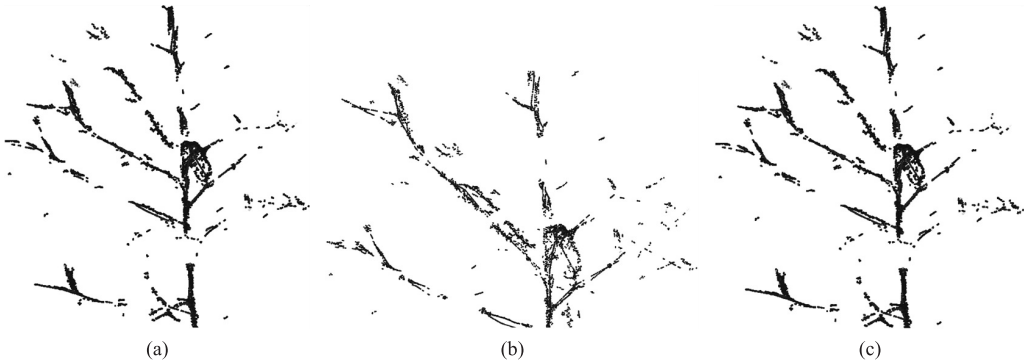


图 2 权值 W_H^0 的影响

实验中取 $W_L^0 = 1.0, W_H^0 = 3.0$,迭代次数为 2 次,从图(a)到(c), W_H^0 取值分别为 0.5、1、2

Fig. 2 The impact of W_H^0

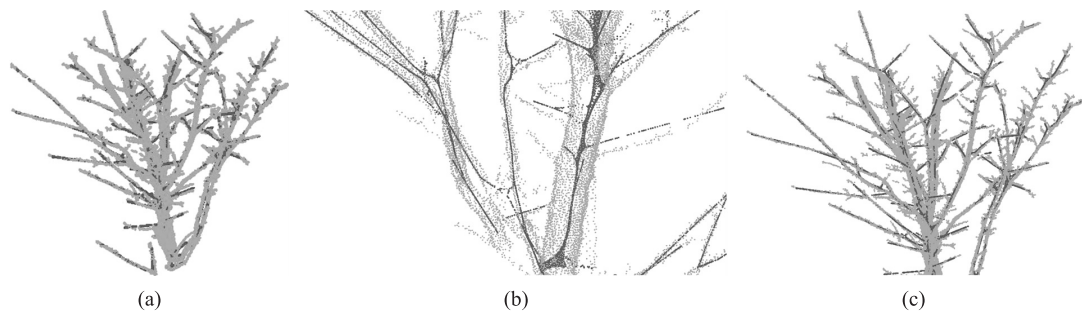
$W_L^0 = 1.0, W_H^0 = 3.0$ and iterate 2 times. From figure(a) to(c), W_H^0 is 0.5, 1, 2 separately

表 1 樱花树实验参数

Table 1 Experimental parameters of cherry tree			
试验次数	更新力度	收缩力度	控制力度
1	0.5	1	3
2	1	1	3
3	2	1	3

表 2 含笑树实验参数

Table 2 Experimental parameters of michelia figo			
试验次数	更新力度	收缩力度	控制力度
1	2	0.5	3
2	2	1	3
3	2	2	3

图3 权值 W_L^0 的影响

实验中取 $W_L^0 = 1.0$, $W_H^0 = 3.0$, 迭代次数为 2 次. 从图 (a) 到 (c), W_L^0 取值分别为 1、3、5

Fig. 3 The impact of W_L^0

$W_L^0 = 1.0$, $W_H^0 = 3.0$ and iterate 2 times. From figure (a) to (c), W_L^0 is 1, 3, 5 separately

(2) 最终重建结果

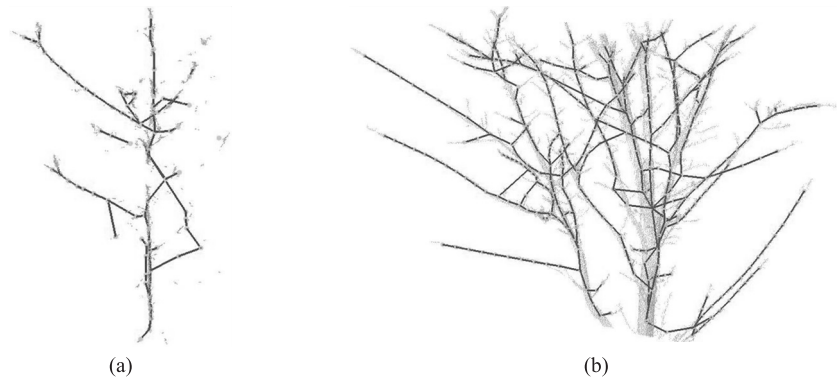


图4 最终的树木枝干重建效果

图 (a)、(b) 分别为最终重建出来的含笑树和樱花树的枝干

Fig. 4 The final reconstruction

From figure (a) to (b) is cherry tree and michelia figo separately

2) 实验复杂度分析

表3 不同的方法在含笑树上的效率分析

Table 3 The efficiency analysis of different methods on michelia figo

方法	时间/s	内存/KB
基于 Dijkstra 的方法	153	35
基于 L-参数系统的方法	165	26
基于离散曲率的方法	197	30
本文的方法	134	13

表4 不同的方法在樱花树上的效率分析

Table 4 The efficiency analysis of different methods on cherry tree

方法	时间/s	内存/KB
基于 Dijkstra 的方法	265	23.65
基于 L-参数系统的方法	281	27.20
基于离散曲率的方法	305	48.35
本文的方法	214	21.95

3) 结论

本文的算法所用的数据含笑树有 6 885 个样本点、樱花树有 36 619 个点,从图中可以看出实验结果很理想. 因为在建立拉普拉斯矩阵时需要建立点云的邻域关系,而三角网格本身就带有邻域关系,故而本文提出的点云曲线重建算法能够很容易地移植到网格的曲线重建上. 因为网格的邻域是本身带有的,不需要计算,可以预见在网格上进行曲线重建会更加鲁棒. 本算法对于有亏格和有残缺的点云也有很好的重建效果. 尽管本文的算法有以上诸多的优点,但是依旧还有一些问题有待进一步解决. 比如对于点云中出现的平面或接近平面形状的部分,用本文提出的算法效果则不是很理想. 这是因为在这些部分收缩的点云不能很好地呈线性分布,用 PCA 的方法会将该部分识别为节点,从而对点云曲线骨架的拓扑一致性产生一定的影响,这样实验的结果会产生较大的误差.

[参考文献]

- [1] Deok S K, Young S C, Dengue K. Euclidean Voronoi diagram of 3D balls and its computation via tracing edges[J]. Computer-Aided Design, 2005, 245(20): 3 713–3 721.
- [2] Cao A W, Yung T. Finding constrained and weighted Voronoi diagrams in the plane[J]. Computational Geometry: Theory and Applications, 1998, 283(16): 1 027–1 035.
- [3] Incur C, Deborah S, Xiao S, et al. Computing hierarchical curve-skeletons of 3D objects[J]. The Visual Computer, 2005, 89(11): 895–907.
- [4] Lawson W, Richard E P. Automated generation of control skeletons for use in animation[J]. The Visual Computer, 2002, 275(21): 2 175–2 183.
- [5] Taube G. A signal processing approach to fair surface design[C]//International Conference on Computer Graphics and Interactive Techniques. Los Angeles: The International Institute for Science, Technology and Education, 1995.
- [6] 王林峰. 加权 Laplace-Beltrami 算子及相关问题研究[D]. 上海: 华东师范大学数学学院, 2007.
- [7] 李义琛. 点云模型骨架提取算法的研究与实现[D]. 南京: 南京师范大学教育科学学院, 2012.
- [8] 张亶, 陈为, 单开佳, 等. 基于拉普拉斯算子的 Snakes 方法分析[J]. 计算机辅助设计与图形学学报, 2005, 6(20): 527–531.
- [9] Dieter M, Maria S, Rodrigo I S. On the number of higher order Delaunay triangulations[J]. Theoretical Computer Science, 2011, 281(45): 41 229–41 235.
- [10] Jonathan, Richard, Shewchuk. Reprint of: delaunay refinement algorithms for triangular mesh generation[J]. Computational Geometry: Theory and Applications, 2014, 365(78): 15 081–15 090.
- [11] Rodrigo I, Silveira, Marc van Kreveld. Towards a definition of higher order constrained Delaunay triangulations[J]. Computational Geometry: Theory and Applications, 2008, 424(21): 1 051–1 059.
- [12] Marian N. Delaunay configurations and multivariate splines: a generalization of a result of B N Delaunay[J]. Transactions of the American Mathematical Society, 2007, 207(20): 3 597–3 602.
- [13] 金龙存. 3D 点云复杂点云曲面重构关键算法研究[D]. 上海: 上海大学计算机学院, 2012.
- [14] 何学铭. 点云模型的孔洞修补技术研究[D]. 南京: 南京师范大学教育科学学院, 2013.
- [15] 丁帆. 点云数据三维网格构造方法研究[D]. 武汉: 华中科技大学计算机学院, 2007.
- [16] Zhou K, Huang J, Snyder J. Large mesh deformation using the volumetric graph Laplacian[J]. ACM Transactions on Graphics, 2005, 217(20): 1 207–1 213.
- [17] Lipman Y, Sorkine O, Cohen-or D, et al. Differential coordinates for interactive mesh editing[C]//Proceedings of the International Conference on Shape Modeling and Applications. San Francisco: Morgan Kaufmann, 2004.
- [18] Gong W, Bertrand G. A simple parallel 3D thinning algorithm[C]//10th International Conference on Pattern Recognition. Istanbul: Nova Science Publishers, 1990.
- [19] Cornea N D, Demirci M F, Silver D, et al. 3D object retrieval using many-to-many matching of curve skeletons[C]//Proceedings of the International Conference on Shape Modeling and Applications. New York: Los Andes, 2005.
- [20] Kobatake S, Kawakubo Y, Suzuki S. Laplace pressure measurement on laser textured thin-film disk[J]. Teratology International, 2003, 364(26): 10 631–10 642.
- [21] Adam M B. Finite difference methods for the infinity Laplace and p-Laplace equations[J]. Journal of Computational and Applied Mathematics, 2013, 254(419): 1 872–1 882.
- [22] Humid R, Soon-Mo J, Themistocles M R. Laplace transform and Hyers–Ulam stability of linear differential equations[J]. Journal of Mathematical Analysis and Applications, 2013, 381(29): 4 031–4 045.
- [23] Tomasz J K, Krzysztof P, Igor R. Multivariate generalized Laplace distribution and related random fields[J]. Journal of Multivariate Analysis, 2013, 113(25): 3 085–3 093.

[责任编辑: 黄 敏]