

散乱数据点的增量快速曲面重建算法*

王青 王融清 鲍虎军 彭群生

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

E-mail: qwang@cad.zju.edu.cn

摘要 给出了一个新的散乱数据的曲面重建算法。算法充分利用邻近点集反映出的局部拓扑和几何信息,基于二维 Delaunay 三角剖分技术快速实现每个数据点的局部拓扑重建,然后通过自动纠正局部数据点的非法连接关系,以增量扩张的方式把局部三角网拼接成一张标准的整体二维流形网格。该算法在重建过程中能自动进行洞的检测,判断出散乱数据所蕴涵的开或闭的拓扑结构。实验结果表明,该算法高效、稳定,可以快速地直接重构出任意拓扑结构的二维流形三角形网格。

关键词 曲面重建,散乱数据点,三角剖分,二维流形,几何造型。

中图法分类号 TP391

散乱数据点的曲面重建一直是函数逼近论的一个重要研究内容。近几年来,随着计算机辅助设计与图形学的发展,散乱数据的曲面重建技术得到了广泛的研究和应用,如基于测距技术的几何模型自动生成、医学成像数据的可视化等。该技术的发展有力地促进了造型和可视化等技术的高速发展。

散乱数据的三维重构其难点在于,如何在数据点集中自动地得到邻近点间正确的拓扑连接关系。而正确的拓扑连接关系将有效地揭示散乱数据集所蕴涵的原始物体表面的形状和拓扑结构。根据数据点集组织形式的不同,可以把散乱数据点集分成 3 类,即点与点之间毫无内在联系的数据点集称为无组织数据集(unorganized data),此类数据点一般都蕴涵着一个隐含的假设:存在一个流形表面插值或拟合这些数据点;第 2 类主要来自于医学图像,称为轮廓线(contour)或体数据(volumetric data),数据点呈层状分布,每一层代表物体的一个剖面;第 3 类是深度散乱数据(range data),主要由三维激光扫描测距技术所获得,数据集往往由多幅深度图像所组成。事实上,这 3 种数据类型可以相互转化,如把深度图像数据转化成体数据。

目前主要有 3 种无组织散乱数据的三维重建方法。一种方法是构造点到物体表面的(有向)距离场,该距离场的零等值面即为重建曲面^[1]。但这种算法涉及到复杂的法向一致性检查和等值面抽取,重建非常耗时,且重建表面需经优化处理才能使用^[2]。另一种方法是直接采用隐函数曲面或参数曲面来逼近或拟合数据点集。Foley^[3]综述了从散乱点重构隐函数曲面和参数曲面的有关算法。而 Pratt^[4]则直接采用简单形状(如圆、球、直线、立方体)来拟合数据点,但这种方法难以重建复杂拓扑的曲面形状。最常用的是第 3 种方法,即应用 Voronoi 图对散乱点集进行 Delaunay 三角化。1972 年,Lawson^[5]提出了三角化的最大角最小化原则,符合这一原则的三角化称为局部均匀的。随后,Sibon^[6]证明了 Delaunay 三角化是符合这一原则的唯一形式。紧接着,Green and Sibon^[7]实现了二维空间的 Voronoi 图的计算及 Delaunay 三角化。Bowyer^[8]和 Watson^[9]把结果推广到任意维。随后出现了大量的文献,用各种不同的方法去实现 Voronoi 图和 Delaunay 三角化^[10~12]。由于 Delaunay 三角剖分后的结果是一个三角形(二维)或四面体(三维)的凸包,并不表示真正的原物体表面,其中包含许多冗余的三角形或四面

* 本文研究得到国家自然科学基金(No. 69925204)和高等学校骨干教师基金资助。作者王青,1976 年生,硕士,主要研究领域为计算机图形学,几何造型。王融清,女,1966 年生,助理工程师,主要研究领域为 CAD 设计,机械制造。鲍虎军,1966 年生,研究员,博士生导师,主要研究领域为计算机图形学,虚拟现实,几何造型。彭群生,1947 年生,教授,博士生导师,主要研究领域为真实感图形,虚拟现实,科学计算可视化,红外仿真。

本文通讯联系人:王青,杭州 310027,浙江大学 CAD&CG 国家重点实验室

本文 2000-02-28 收到原稿,2000-04-21 收到修改稿

体. Boissonnat^[3]对表面拓扑结构与球的物体介绍了一种算法,通过层层剥离冗余四面体,使物体表面的所有散乱数据点可见. 1994年, Edelsbrunner^[4]提出了 α -shape 的概念, α -shape 方法通过删除四面体凸包中其包围球或外接圆半径大于 α 的四面体、三角形和边得到重建表面. 后来, Baja^[15]基于 α -shape 生成了 C^1 连续的散乱数据点的插值曲面. 对于均匀一致的数据点集, α shape 方法很有效,但由于数据点集的不均匀性或表面的某种不连续性,有时很难自动选择合适的 α 值,以保留需要的三角化元素,删除不需要的所有元素.

本文借助于深度图像的重建思想来快速重建无组织数据集. 与其他无组织散乱数据重建算法相比,本文算法主要有 4 个优点: (1) 运行效率高,且思想简单易于实现; (2) 输出结果为最常用的三角网格表示,三角网符合 Delaunay 性质,且每个三角形拥有一致的法向; (3) 适用于任意拓扑结构的物体; (4) 允许数据集的分布具有一定的不均匀性.

1 相关概念

假设给定散乱数据集为 $P = \{V_i | i = 1, \dots, n\}$, 其中 V_i 的三维坐标为 (x_i, y_i, z_i) , n 为点的个数. 本文的目标是构造一张插值 P 的三角形流形网格. 为此,先介绍以下几个基本概念.

1.1 点的属性

在一个三角形网格中,对于顶点 V ,三角形 T 称为 V 的邻接三角形,若 V 为 T 的一个顶点. 顶点 A 称为 V 的邻接点,若存在边以 V, A 为端点,即 A 在 V 的一个邻接三角形中. 而 A 关于 V 的重数是指 V 的邻接三角形中含有 A 的个数. 对于封闭的二维流形拓扑网格来说, V 的每一邻接点关于 V 的重数均为 2.

这样,根据重数的定义,我们可将点分成 3 类:一类是孤立点,它没有邻接点;一类是内点,它的每一邻接点的重数均为 2;最后一类是边界点,它既拥有邻接点,又存在重数为 1 的邻接点. 点的属性就是指它是孤立点、边界点还是内点.

1.2 空间三角形的重叠

对于两个平面三角形,它们相互重叠则意味着它们除了共享顶点、共享边(若有的话)以外,还具有其他公共部分. 但对于两个空间三角形,它们的相互重叠则是一个很模糊的概念. 在本文中,两空间三角形 T_1, T_2 相互重叠是指相对于某一顶点而言的,其定义如下:设 $V \in P$ 是表面网格上的一个顶点,表面在 V 点处的切平面为 S ,如果 T_1, T_2 在 S 上的投影相互重叠,我们称 T_1, T_2 关于 V 相互重叠.

2 局部三角化及网格整合

如果已知物体表面上一点 V 的法向量,那么过此点与法向垂直的微平面(切平面)即是表面在该点邻域的一阶逼近. 又由于二维数据点的三角化已非常成熟,且远比三维数据点的三角化来得容易和简便,因此,我们可以先将 V 点及其附近的点投影到过 V 点的切平面上,然后进行二维三角化,得到点与点之间的连接关系,最后再将这些连接关系映射返回到三维空间,就形成了 V 点及其附近点的空间三角化(如图 1 所示). 这样生成的三角网称为物体表面在 V 点的局部三角网.



Fig. 1 2D local triangulation and its map to 3D space
图1 二维局部三角化及其到三维空间的映射

2.1 切平面的估计

对于点 V ,我们通过它的 k 个最近点来求 V 点的近似切平面(关于 k 的选取将在第 3.2 节讨论).

问题可以描述为:给定 V 的最近点集 $Q = \{Q_i | i = 1, \dots, k\} \subset P$,求 V 点的近似法向 N ,使得式(1)最小,即

$$error = \sum_{i=1}^k ((Q_i - V) \cdot N)^2. \quad (1)$$

应用最小二乘法,经过简单计算,可得到 3×3 矩阵 C ,即

$$C = \sum_{i=1}^k (Q_i - V)^T \cdot (Q_i - V). \quad (2)$$

容易证明,把 C 的最小特征值对应的特征向量单位化,即可作为 N 的近似值.

2.2 平面二维点的 Delaunay 三角化

求出 V 点的近似法向量以后,我们就可以把 V 点及其最近点集 Q 的空间三角化转化为在其切平面上投影点的二维三角化.目前已有很多方法实现了这一目标.本文选用的是 Sloan^[16]中的二维 Delaunay 三角化方法,其算法平均复杂度为 $O(m^{5/4})$ (m 为点数).算法简要描述如下:

(1) 增加 3 个附加点以构造一个足够大(大到能够包含所有参与三角化的点)的初始三角形.

(2) 对点集 Q 中每一未处理过的点 D ,从当前三角网中找到一个包含 D 点的三角形,将它与三角形的三个顶点相连,生成新的 3 个三角形,删除原三角形.

(3) 删除附加点及其相连的三角形.一个二维 Delaunay 三角网格便形成了.

2.3 三维网格的拼接

假设 $V_1, V_2 \in P, V_1$ 和 V_2 的最近点集分别为 $Q_1, Q_2 \subset P$.应用第 2.2 节的方法分别在 V_1 和 V_2 的邻域内进行局部三角化,生成两张局部三角网 M_1 和 M_2 .我们的目标是将它们整合成一张三角形网格,使得网格重建过程能以增量扩张的方式执行.

本方法先将 M_2 中的所有三角形无条件地加入到 M_1 中去,然后判别并删除非法三角形.设 A 为 Q_2 中的一点,记合并后 A 的邻接三角形集为 T_A ,我们希望找出并删除 T_A 中的非法三角形,使得 M_1 在 A 处仍符合二维流形的定义.具体地,我们可参照图 2 按以下方式进行:

• 如果 A 的某一个邻接点 B 关于 A 的重数为 3 或 4,则删去一个或两个拥有边 AB 的三角形,使得 B 关于 A 的重数为 2,且剩下的两个三角形不与 T_A 中的其他三角形关于 A 重叠(见图 2(a)和(b)).

• 当不存在重数为 3 或 4 的邻接点以后,如果 T_A 中还有两个三角形关于 A 重叠,则删除其中一个三角形,且使得另一个三角形不再与 T_A 中的其他三角形关于 A 重叠(如图 2(c),(d)和(e)所示).

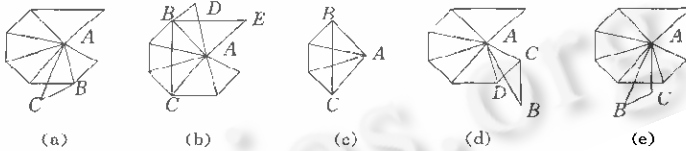


Fig. 2 Deleting abnormal triangle ABC

图2 删除非法三角形 ABC

用同样方法再对 Q_2 中的所有其他点进行检测,同时改变它们的属性.这样,两张局部三角网就合成一张三角形网格了.值得注意的是,每增加或删除一个三角形,三角形三顶点的属性都可能发生变化.

2.4 法向一致化

所谓法向一致化,即把整体三角形网格中的所有三角形的法向指向体外.对诸如布尔运算和绘制等应用,法向一致化是必须的.众所周知,三角形 ABC 法向为

$$N = (B - A) \times (C - A) / \|(B - A) \times (C - A)\|.$$

若 N 指向体内,我们需要交换三角形中某两个顶点的存储顺序,并把 N 置反.

如果三角形 T_1, T_2 具有公共边,不妨设 T_1 的 3 个顶点分别为 A, B 和 C ; T_2 的 3 个顶点分别为 B, C, D .再设两三角形的当前法向分别为 N_1, N_2 .如果已知 N_1 已指向体外,则我们有两种方法来快速判别 N_2 是否指向体外.

方法 1. 若 $N_1 \cdot N_2 < 0, N_2$ 指向体内,否则指向体外.

方法 2. 设 A, B, C 在 T_1 中的存储编号分别为 i_1, j_1, k_1 ; B, C, D 在 T_2 中的存储编号分别为 i_2, j_2, k_2 .记 $i_1 \dots$

$i_2, j_1 - j_2, k_1 - k_2$ 这 3 个数中 0 的个数为 flag, 若 flag 为 0 或 3, N_2 指向体内, 否则 flag 必为 1, N_2 指向体外. 如果 T_1 顶点的排列顺序为 B, C, A, T_2 顶点的排列顺序为 C, B, D , 则 $i_1 - i_2 = 2 - 1 = 1, j_1 - j_2 = 0 - 0 = 0, k_1 - k_2 = 1 - 2 = -1, \text{flag} = 1, N_2$ 指向体外.

方法 1 要求在物体表面曲率比较大的地方使用, 并且数据点应足够密, 同时使得两相邻三角形的法向夹角小于 90° . 而方法 2 则无任何限制.

于是, 我们可以采用下面的方法来实现三角形网格的法向一致化. 首先选择一个种子三角形 $\text{Seed}T$, 它的法向已指向体外, 然后应用以上方法, 将与 $\text{Seed}T$ 有共享边的三角形的法向指向体外, 并对它们进行标记, 然后选择一个已标记的三角形, 把与其有共享边且未标记的三角形的法向指向体外, 再进行标记, 不停地重复以上过程, 直至所有三角形都已标记过, 法向一致化就完成了. 这里, 种子三角形的选取可以通过交互, 在窗口图形中选取而得到, 当然, 亦可由算法自动确定得到.

3 算法的实现

3.1 算法描述

算法的输入为点集 $P = \{V_i\}$ 的空间坐标, 输出为一个 Delaunay 三角网 M 及每一个三角形的正确法向. 其基本过程如图 3 所示, 具体可分为以下 3 个阶段.

预处理.

- (1) 对每一点 V_i , 找出其 k_i 个最近点, 记为 $\text{Neigh}(V_i)$. 用第 2.1 节的方法拟合出 V_i 的近似法向量.
- (2) 置 M 为空集, 把每一点的属性均置为孤立点.

三角化. 顺序遍历点集中的每一点, 对边界点或孤立点 V_i , 从集合 $\text{Neigh}(V_i)$ 中删去内点, 得到点集 Q_i . 若 Q_i 中的点数大于 2, 则作以下处理:

- (1) 将 Q_i 投影在过点 V_i 的切平面上, 用第 2.2 节中的方法进行三角化, 得到一个二维三角网.
- (2) 将二维三角形网中点与点之间的连接关系返回到三维空间, 得到点 V_i 的局部三角网 M_i .
- (3) 将 M_i 加入到 M 中(见第 2.3 节), 并同时修改 V_i 及 Q_i 中各点的属性.

网格优化.

- (1) 使网格具有 Delaunay 性质. 对每一个有共享边的两个三角形所组成的空间四边形, 应用最大角最小化的原则, 对不符合要求的四边形, 若另一对角线还不存在, 则交换其对角线.
- (2) 先求出每一个三角形的法向量, 再把所有的法向量一致化(见第 2.4 节).

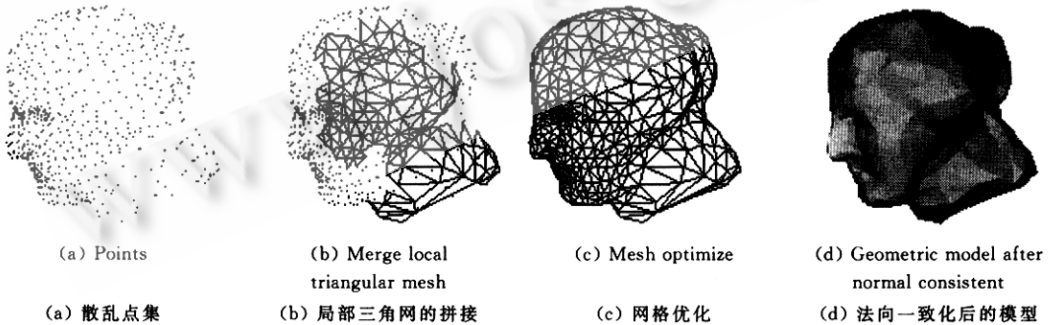


Fig. 3 Reconstruction of the head model

图 3 头模型的重构

3.2 算法分析

由于我们选用每一点的最近点集进行局部三角化, 所以可处理不规则分布的数据点集, 而且当算法结束时, 物体上的空洞能在结果三角形网格中自动保留下来. 如果用户想封闭三角形网格上的空洞, 在这里只要再增加一个非常简单的填补操作即可, 如图 4 所示. 图 4 为采用本算法生成的兔子模型的底部重建结果, 图 4(b)为正常

重建的开拓扑网格,图 4(c)则为采用填补操作后强行封闭的结果.因为此时洞边界上的点的属性恰好为边界点.所以,我们只要不停地进行以下填补操作,直到所有边界点的属性不再改变为止,然后再重新进行网格优化即可把洞封闭起来.

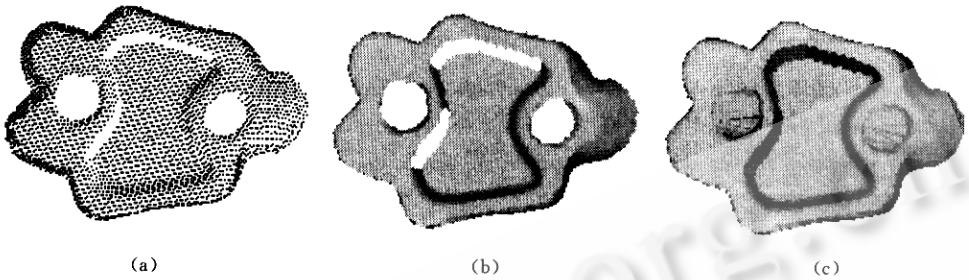


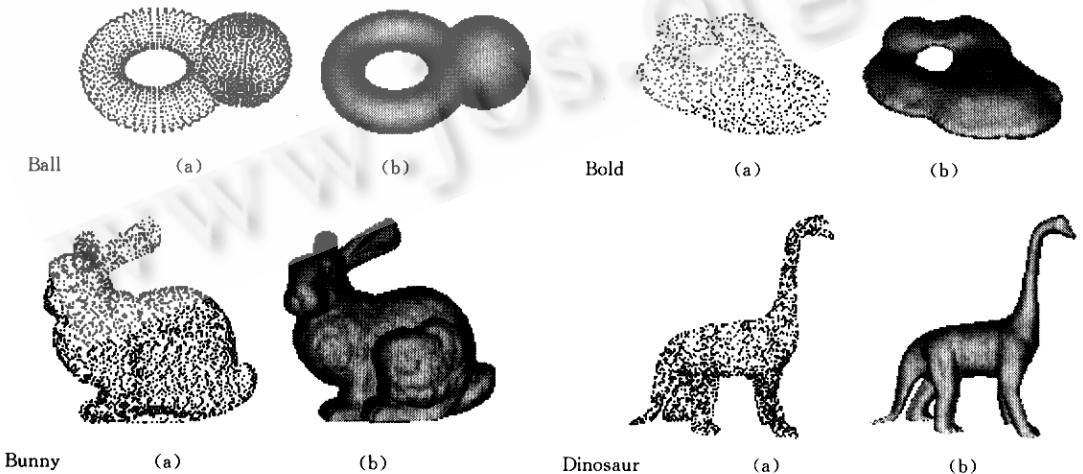
Fig. 4 Filling the hole at the bottom of bunny
图 4 兔子模型底部洞的填补

填补操作定义如下:对每一边界点 V ,如果它只有两个邻接点的重数是 1,设为 V_1, V_2 ,则连接 V_1V_2 ,生成三角形 V_1VV_2 ,如果此三角形不与 V, V_1 和 V_2 的任何邻接三角形重叠,则置 V 点属性为内点,否则,删除此三角形.

如何确定每个数据点 V_i 的最近点个数 k_i ,直接影响到算法的效率.为保证局部三角化的局部性, k_i 不宜取得太大. k_i 也不能取得太小,否则局部三角化就不完整,使得应该成为内点的点在局部三角化后只能成为边界点. k_i 一般在 5~15 之间,用户可以根据输入点集的分布情况为每个 k_i 设置同一个初始值,然后在重构过程中根据 V_i 处的曲率调节 k_i 值.在物体表面比较平坦的地方,可以适当增大 k_i 值;在物体表面曲率比较大的地方,可以适当减小 k_i 值.然而,在重构完成前,我们无法知道物体的表面,更无法知道它在某一点的曲率.不过,我们可以在求近似法向的时候,估计表面的平坦程度,当偏差和 $error$ (见第 2.1 节)比较小时,表面在此点比较平坦,可以适当放大此点的 k_i 值;反之,则缩小此点的 k_i 值.

4 执行结果

本算法已用 C++ 在奔腾 II-300 个人计算机上实现.我们已用多个例子进行了测试(如图 3 和图 5 所示),为简单起见,我们把 k 设为一个常数来处理.结果证明此算法高效、可靠.测试结果见表 1.



(a) Unorganized points, (b) Geometric model
(a) 无组织散乱点集, (b) 几何造型

Fig. 5 Results of the reconstruction

图 5 重建结果

Table 1 Test result

表 1 测试结果

Examples ^①	Number of points ^②	k	Number of triangles ^③	Runtime (s) ^④			
				Computer normal ^⑤	Mesh reconstruct ^⑥	Mesh optimize ^⑦	Normal consistent ^⑧
Head	679	10	1 354	0.03	0.45	0.06	0.03
Ball	2 934	12	5 868	0.12	2.32	0.32	0.11
Blob	9 478	12	18 956	0.33	8.76	0.99	0.26
Bunny	34 835	12	69 473	0.97	28.42	3.63	2.56
Dinosaur	23 931	9	47 858	0.28	18.34	2.41	3.24

①例子,②点数,③三角形数,④运行时间(s),⑤计算法向,⑥网格重构,⑦网格优化,⑧法向一致化。

5 结 论

本文介绍了一个快速、可靠的无组织散乱数据的曲面重建算法。当把局部三角网 M_i 加入到 M 中时,我们的拼接算法保证了 M 在每一点符合流形体的定义,所以,最后得到的三角网肯定是一个流形表面。算法适用于任意拓扑结构的物体,可根据给定的数据点分布情况自动地重建开或闭拓扑网格。今后,可将本算法应用于体数据和隐式场的等值面快速抽取及深度图像的曲面重建上。

参考文献

- Hoppe H, DeRose T, Duchamp T *et al.* Surface reconstruction from unorganized points. In: Cunningham S ed. Proceedings of the SIGGRAPH'92. Danvers: Addison-Wesley Publishing Company, 1992. 71~78
- Hoppe H, DeRose T, Duchamp T *et al.* Mesh optimization. In: Cunningham S ed. Proceeding of the SIGGRAPH'93, Danvers: Addison-Wesley Publishing Company, 1993
- Foley T A, Hagen H, Nielson G M. Visualizing and modeling unstructured data. The Visual Computer International Journal of Computer Graphics, 1993,9(8):439~449
- Pratt V. Direct least-squares fitting of algebraic surfaces. In: Stone M C ed. Proceedings of the SIGGRAPH'87. Danvers: Addison-Wesley Publishing Company, 1987. 145~152
- Lawson C L. Generation of a triangular grid with application to contour plotting. California Institute of Technology Jet Propulsion Laboratory, Technical Memorandum 299, 1972
- Sibson R. Locally equiangular triangulations. The Computer Journal, 1978,21(3):243~245
- Green P J, Sibson R. Computing dirichlet tessellations in the plane. The Computer Journal, 1978,21(2):168~173
- Bowyer A. Computing dirichlet tessellations. The Computer Journal, 1981,24(2):162~166
- Watson D F. Computing the n -dimensional Delaunay tessellation with application to Voronoi polytopes. The Computer Journal, 1981,24(2):167~172
- Dey T K, Sugihara K, Bajaj C L. Delaunay triangulations in three dimensions with finite precision arithmetic. Computer Aided Geometric Design, 1992,9(5):457~470
- Amenta N, Bern M, Kamvysselis M. A new voronoi-based surface reconstruction algorithm. In: Cohen M ed. Proceedings of the SIGGRAPH'98, Danvers: Addison-Wesley Publishing Company, 1992. 415~421
- Hoff K E, Culver T, Keyser J *et al.* Fast computation of generalized voronoi diagrams using graphics hardware. In: Rockwood A ed. Proceedings of the SIGGRAPH'99. Danvers, Addison-Wesley Publishing Company, 1992. 277~286
- Boissonnat J-D. Geometric structures for three-dimensional shape representation. ACM Transactions on Graphics, 1984. 3(4):266~286
- Edelsbrunner H, Mücke E P. Three-Dimensional alpha shapes. ACM Transactions on Graphics, 1994,13(1):43~72
- Bajaj C L, Bernardini F, Xu G. Automatic reconstruction of surfaces and scalar fields from 3D scans. In: Cook R ed. Proceedings of the SIGGRAPH'95. Danvers: Addison-Wesley Publishing Company, 1995. 109~118
- Sloan S W. A fast algorithm for constructing Delaunay triangulations in the plane. Advances in Engineering Software, 1987,9(1):34~55

A Fast Progressive Surface Reconstruction Algorithm for Unorganized Points

WANG Qing WANG Rong-qing BAO Hu-jun PENG Qun-sheng

(State Key Laboratory of CAD & CG Zhejiang University Hangzhou 310027)

Abstract A novel algorithm for fitting surface reconstruction of unorganized data points is presented in this paper. Since the local geometric and topological properties can be clearly described with its neighboring points for a given point, the algorithm takes full advantage of this fact to generate a local triangular mesh based on Delaunay triangulation technique in two dimensions. The main contribution of the proposed algorithm is to progressively integrate those local triangular meshes into a resulting mesh by deleting the illegal triangles. In addition, holes and gaps can be detected automatically during reconstruction, and the algorithm works well for models with arbitrary topology. Experimental results show that this algorithm is effective, robust and can be easily implemented.

Key words Surface reconstruction, unorganized points, triangulation, two dimensional manifold, geometric modeling.