

基于径向基函数的点云岛屿孔洞自动修复^①

张立国^② 王 静^③ 金 梅 康 乐

(燕山大学河北省测试计量技术与仪器重点实验室 秦皇岛 066004)

摘 要 研究了利用点云获得的模型的孔洞修复,针对目前主要通过人工修复带有岛屿面片的孔洞耗时较长的问题,提出了一种基于径向基函数(RBF)自动修复岛屿孔洞的方法。该方法首先利用最小权重三角化法修复模型主体上的孔洞,其次计算模型主体上孔洞与岛屿面片的相关性,利用模型主体上孔洞和与其相关岛屿面片周围点来计算径向基函数,最后将粗修复后细分的点调整到径向基函数描述的曲面上。实验表明,与其他方法相比,该方法能快速、准确地修复缺陷模型。

关键词 点云,孔洞,岛屿面片,径向基函数(RBF),最小权重三角化

0 引 言

随着计算机立体视觉的发展,点云(用仪器测得的产品外观表面的点数据集合)的处理技术在逆向工程、医学图像处理、文物模型保存中得以广泛应用。在现有的技术条件下,利用三维扫描仪或者计算机软件得到的点云大多都伴随着孔洞。由于物体表面的凹陷、数据获取时的角度、数据的丢失等原因,得到物体的模型还带有岛屿面片。要保证模型的完整性,点云孔洞修复是必不可少的一个环节。在模型使用之前,需要进行一系列的操作将孔洞修复。为此,本文提出了一种基于径向基函数(radial basis function, RBF)的点云岛屿孔洞自动修复方法。

1 相关工作

根据模型表示方式的不同,使用的孔洞修复方法分为基于体积元素的孔洞修复和基于三角网格的孔洞修复。

基于体积元素表示的孔洞修复方法,需要用具有简单、一致等特性的体积元素来表示模型。Davis 等人使用体积离散的方法实现孔洞修复,这种方法能够修复带有岛屿面片的模型,但是会改变原有模型的拓扑^[1]。Curless 等人将空间雕刻和等值面提取引入孔洞修复^[2],Podolak 等人利用最小切割算法确定空间中的各部分在模型的内部或外部,实现孔洞的修复^[3]。这两种方法能够修复复杂的模型,但是比较费时。

基于三角网格的孔洞修复方法需要将模型用具有存储、渲染效率高的三角网格表示。与直接将孔洞周围点连接实现孔洞修复不同,隐函数孔洞修复方法能够修复复杂的孔洞。Kazhdan 等人利用泊松方程重构出没有孔洞的模型^[4]。Zhao 等人提出利用波前推进法完成孔洞的粗修复,然后利用泊松方程将粗修复的点调整到曲面^[5]。Centin 等人利用泊松方程对孔洞进行逐层修复,每修复一层孔洞便更新一次泊松方程,直到孔洞修复完成^[6]。基于泊松方程的方法需要准确地计算模型中所包含点的法线,并且只能应用在封闭的模型。Xie 等人利用差分进化算法获得缺陷处的拓扑和形状,然后优化获

① 河北省自然科学基金(F2015203392)和秦皇岛市科技计划(201502A043)资助项目。
② 男,1978年生,博士,副教授;研究方向:惯性导航,三维立体重构及智能信号处理;E-mail: zlgtime@163.com
③ 通讯作者,E-mail: 307536367@qq.com
(收稿日期:2015-10-29)

得的三角形,这个方法对参数依赖较大^[7]。Carr 等人利用径向基函数(RBF)实现曲面重构,并且可以得到一个光滑模型^[8]。本研究在 Carr 提出的方法的基础上实现了孔洞自动修复。

理想的孔洞修复方法能够充分利用已有的信息,在给定的时间内给出满足要求的模型。本研究基于径向基函数进行孔洞修复的主要思想是:点云是表面上的采样点,通过这些点来计算整个曲面方程,然后通过等值面提取,实现模型的存储和可视化。本研究中,对于相互连接的总数小于 40 个三角形的面,便当作岛屿面片来处理,提出了岛屿面片与模型主体上孔洞的相关性检测方法。如果相关,则将模型主体上孔洞周围的点与岛屿面片上的点添加到一个点集中,利用集合中的点来计算径向基函数,将这个模型主体上的孔洞粗修复并且细分,新得到的三角形上的点调整到径向基函数表示的表面上,实现孔洞的自动修复。

2 径向基函数的孔洞修复算法

基于径向基函数方法的主要过程包括:孔洞检测、孔洞粗修复、将粗修复的点调整到由径向基函数所描述的表面上。

2.1 孔洞检测

进行孔洞修复,首先需要检测孔洞。模型是由图元组成的,通过图元特征分析实现孔洞的检测。在三角网格模型中,一个三角形在孔洞边界上,它至少有一条边没有被其它三角形共用,检查模型中所有的三角形,提取出全部没有被共用的边。那些可以首尾相接的边,就组成了一个边界孔洞。

对于孔洞,还有内孔洞和外孔洞的区分。通过边界点拟合一个平面,得到这个平面的方程。将所有的边界点和与它相邻的点投影到这个平面上,遍历领域点是顺时针还是逆时针来判定它是内孔洞还是外孔洞^[9]。集合 $S = \{p_i = (x_i, y_i, z_i) \mid i = 1, 2, \dots, k\}$ 构成的孔洞边界,利用最小二乘法估计得到这个边界的拟合平面,即解下面的方程:

$$AG = 0 \quad (1)$$

$$\text{其中 } A = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_k & y_k & z_k & 1 \end{bmatrix}, G = (a, b, c, d)。$$

将孔洞周围的点带入式(1)中,得到拟合平面的一般方程:

$$ax + by + cz + d = 0 \quad (2)$$

将边界上的点 $p_1 p_2 \dots p_k$ 和与边界相邻的点投影到这个拟合平面上,利用这些投影点来判断这个边界是内边界还是外边界。具体判断内外孔洞的过程如下:假设 $p_1 p_2 \dots p_k$ 是散乱点云的孔洞边界,应用最小二乘法计算该孔洞的拟合平面 N , 这个平面为该边界的特征平面^[10]。将 $p_1 p_2 \dots p_k$ 及其领域点投影到特征平面上,以特征面法矢方向为 z 轴正方向,规定沿投影边界遍历时领域点在它(边界)的右侧,以 z 轴正方向为观察方向,如果投影边界线的遍历方向是逆时针,则此边界线为内边界;反之边界线为外边界^[10]。

2.2 孔洞粗修复和细分

普通的隐函数孔洞修复方法都是将粗修复的点进行调整,得到最终的模型。最终修复新增点的个数与粗修复点的个数相等,并且新增点是通过粗修复点沿着某个方向移动得到的。新增点的相关点通过两个步骤得到:第一步是利用三角化算法将这个外孔洞修复;第二步是将这些新得到的三角形细分。

三角化孔洞修复的算法比较多,选择最小权重原则的三角化算法来修复孔洞。首先,计算出相邻边界边之间的权重,具有最小权重的两边组成一个三角形;更新孔洞边界和权重,再次将最小权重的两个边连接成三角形,直到孔洞被完全修复。

曲面细分能细化现有的模型而不改变现有模型的特性^[11],这里采用的曲面细分是将具有较长边的三角形裂变成几个三角形。这种细分方法速度非常快,而且不改变原来的曲面形状。细分的具体做法是首先计算孔洞周围三角形边长的平均值作为阈值,然后检测孔洞内新增三角形的三边长,如果某边长超过阈值,则将该边从中点处分裂,曲面细分示意图如下图 1 示。图 1 中,边长 bc 大于这个阈值,需

要将三角形 abc 分裂成三角形 abd 和三角形 acd 。

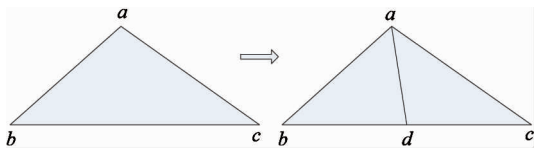


图 1 三角形细分

这一步后,就可以得到粗孔洞修复的点和面,将这些点调整到计算出来的曲面上,不改变点之间的连接关系,得到最后修复的模型。

2.3 求取径向基函数

2.3.1 利用径向基函数求曲面方程

隐函数重构就是由点云来计算曲面方程 f , 径向基函数重构就是隐函数重构中的一种,利用孔洞周围的点估计曲面函数,实现孔洞的修复。对于表面 M' 上的点 $d_i (i = 1, \cdots, n)$, 如果有一个函数 f 满足

$$f(d_i) = 0 \quad (i = 1, \cdots, n) \tag{3}$$

这个方程有一个平凡解,为了消除这种解,必须添加附加约束条件,为曲面加入附加约束点。我们选取一些不在平面上的点来添加约束条件。这样方程就转变为

$$\begin{cases} f(d_i) = 0, & (i = 1, \cdots, n) \\ f(d_i) = h_i, & (i = n + 1, \cdots, N) \end{cases} \tag{4}$$

基于径向基函数的隐式曲面可以表示为^[4]

$$F(r) = \sum_{j=1}^N \omega_j \phi(r - d_j) + P(r) = 0 \tag{5}$$

式中 r 为曲面上的任意一点, d_j 是建立曲面给定的点, ω_j 是每个给定点的权值, $P(r)$ 一般为 $P(r) = p_0 + p_1x + p_2y + p_3z$, $\phi(r - d_j)$ 是径向基函数,它是一个与距离相关的函数。这里选取常用的三维距离函数来表示欧氏距离,即 $\phi_{ij} = ((d_i^x - d_j^x)^2 + (d_i^y - d_j^y)^2 + (d_i^z - d_j^z)^2)^{\frac{3}{2}}$ 。为了解出这个方程,需要一个正交条件:

$$\sum_{j=1}^N \omega_j = \sum_{j=1}^N \omega_j d_j^x = \sum_{j=1}^N \omega_j d_j^y = \sum_{j=1}^N \omega_j d_j^z \tag{6}$$

通过这些条件建立起一个线性方程组,通过高斯消元法求出方程中的各个系数。

2.3.2 插值约束点和附加约束点

为了能够准确地估计出曲面方程,输入的插值约束点数量要足够。这里输入的插值约束点包括孔洞周围的点,使用 k 领域搜索的方法来添加。查找孔洞周围的点主要过程如下:

步骤 1: 读入提取得到的孔洞边界点,将其加入到插值约束点的集合 Q 中。

步骤 2: 以任意孔洞点为起点,查找其 k 邻域,删除与 Q 中元素重复的点,剩余的点添加到点集 Q 中。

步骤 3: 重复步骤 2,直到遍历完所有的孔洞点,得到所需的插值约束点的集合。

上面提到过,为了避免方程的平凡解,需要添加一些附加约束点。这些附加约束点是与此曲面相关的一些点,选择距离插值约束点 0.5 单位长度、法向量方向的点作为附加约束点,增加的附加约束点的数量与插值约束点的数量相同。虽然我们取得点是距离插值约束点 0.5 单位长度的一个点,但是这个点的函数值可以是 1 或者其它值,因为它们都是相等的,不会影响方程的解。

2.3.3 计算隐式曲面方程

通过插值约束点、附加约束点和正交条件,可以建立一个线性方程组。通过高斯消元法,求得方程中的参数,得到最终的曲面方程为

$$F(x, y, z) = \sum_{j=1}^N \omega_j ((x - d_j^x)^2 + (y - d_j^y)^2 + (z - d_j^z)^2)^{\frac{3}{2}} + p_0 + p_1x + p_2y + p_3z \tag{7}$$

2.4 调整点到曲面上

这里没有保留原本模型中的岛屿面片,只利用它们计算曲面方程。我们利用这些岛屿面片来计算曲面方程,然后把这些岛屿面片删除掉。将一个距离曲面不远的点调整到平面上,是这个过程的最后一步。将孔洞修复新增点的相关点调整到平面上,使用 Tasso Karkanis 的梯度下降法^[12]。这种方法能很快地逼近曲面,因为沿着负梯度方向的逼近最快,具体方程如下:

$$R_1 = R_0 - \gamma \times \nabla F \tag{8}$$

式中: ∇F 是曲面梯度, $\nabla F = (\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z})$; $F(R)$

为准则函数; γ 是学习速率, 是大于 0 的较小的常数; R_0 是迭代起点, R_1 是计算得到的新的迭代点。在迭代过程中, 如果 R_1 与 R_0 的差值满足某个阈值, 那么迭代就结束, 此时的 R_1 就是将相关点调整到表面上的点。

3 修复岛屿孔洞的方法

本文方法与普通的径向基函数孔洞修复方法不同的是, 需要利用岛屿面片上的点来计算径向基函数。主要步骤如下:

步骤 1: 检测出模型中的岛屿面片。相互连接小于 40 个三角形的面, 将它们从模型的主体上删除。得到岛屿面片的个数 n 。

步骤 2: 检测模型主体上的孔洞, 并且将它们粗修复和细分。得到模型主体上孔洞的个数 m 后, 将每个孔洞边界点和孔洞周围点添加到对应的集合 $S_i (i = 1, 2, \dots, m)$ 中。

步骤 3: 模型主体上的孔洞与岛屿面片相关性检测。对于模型主体上的孔洞, 与每个岛屿面片进行相关性检测。如果岛屿面片与孔洞相关, 将这个岛屿面片上的点添加到主体孔洞所对应的 S_i 中。

步骤 4: 利用点集合 S_i , 计算出得到径向基函数 $F_i (i = 1, 2, \dots, m)$, 将每个粗修复的三角形中的点, 调整到它对应的 F_i 上, 不改变它们之间的连接关系。

假设一个由点 $t_1 t_2 t_3 \dots t_n$ 组成的面, 计算它的拟合平面的形心 O , 即

$$O = \frac{1}{n} \sum_{i=1}^n t_i \quad (9)$$

判断岛屿面片与模型主体上孔洞的相关性。通过下面两个条件进行判断: (1) 设定一个阈值 $d_{\max}$, 并且岛屿面片的形心 O 距离模型主体上孔洞的拟合平面的距离小于 $d_{\max}$; (2) O 在这个内孔洞拟合平面的投影点 O' 在边界投影的内部。满足这两个条件, 这个岛屿面片和模型主体上孔洞是相关的。

如图 2 所示, 有两个由三个三角形构成的岛屿面片, 它们的顶点分别包括 $defgh$ 和 $ijklm$ 。 $nopqrstuv$ 是模型主体上一个孔洞在它拟合平面上的投影。 $defgh$ 、 $ijklm$ 的形心 a 、 b 到平面 $nopqrstuv$ 的距离 w^1 、

w^2 都小于给定的 $d_{\max}$, 并且它们的形心 a 、 b 在平面上的投影点 a' 、 b' 都在 $nopqrstuv$ 的内部, $defgh$ 和 $ijklm$ 与 $nopqrstuv$ 是相关的。将这两个岛屿孔洞上的顶点加入到与这个模型主体孔洞相联系的点集 S_i 中。

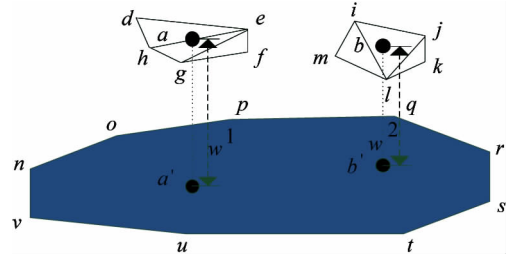


图 2 具有岛屿面片的缺陷模型

判断 a' 、 b' 都在 $nopqrstuv$ 的内部, 可以通过内外边界的判定方法来完成。如图 2 所示, 要判断 a' 在边界投影 $nopqrstuv$ 的内部, 可以假设 a' 是与边界投影 $nopqrstuv$ 相连的点, 由 a' 判定由点 $nopqrstuv$ 组成的边界是内边界还是外边界。如果判定 $nopqrstuv$ 为外边界, 这个点就在边界投影的内部。反之则在边界投影的外部。

通过点的集合 $S_i (i = 1, 2, \dots, m)$, 计算 m 个径向基函数, 并且将粗修复三角形的顶点调整到对应的曲面上, 最后将所有的岛屿孔洞删除, 实现孔洞的修复。如图 2 所示, 利用上述方法可以知道它们是相关联的孔洞, 利用 d 、 e 、 f 、 g 、 h 、 i 、 j 、 k 、 l 、 m 这些顶点与模型主体上孔洞周围的点来计算径向基函数。

4 实验及分析

本实验使用的计算机硬件配置是 AMD 羿龙 X4 和 8G 内存, 软件配置是 OpenGL 和 VC++。图 3 中所利用的模型, 是一个石头的模型。

图 3 中的 (a)、(b) 是模型中的缺陷和孔洞检测的结果。(c) 为利用最小权重三角化的方法将内孔洞填充后的效果, 这种孔洞修复方法速度虽然很快, 但是修复的模型不够光滑。(d) 中是利用一般的径向基函数孔洞修复技术修复这个孔洞的效果, 修复的模型较为光滑, 误差较最小权重三角化小。由于没有利用岛屿面片中顶点坐标来计算径向基函数, 与本文中提出的方法相比较, 误差较大。由于梯度

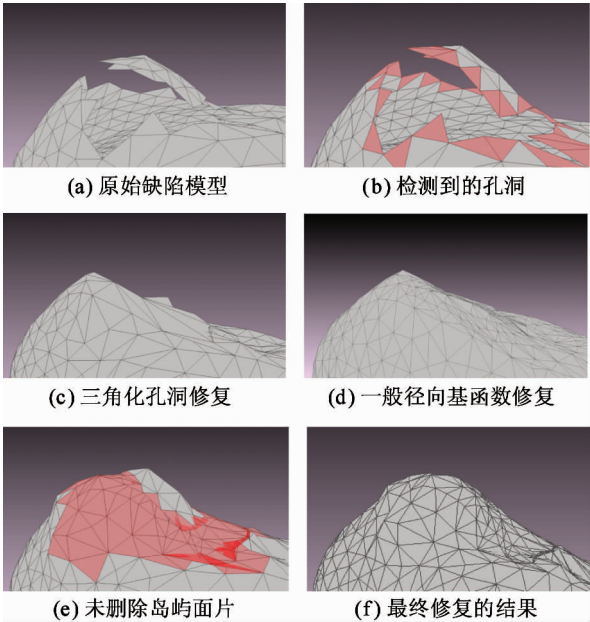


图 3 石头的缺陷模型和修复结果

下降法本身的一些缺陷,可能最终修复的模型的三角形大小不一,具有局限性。

图 3 中的(e)、(f)是利用岛屿孔洞修复方法,软件运行中的截图和最终结果的截图。由于石头的最大半径是 0.5m,所以我们设置相关孔洞中检测的参数 $d_{\max}$ 为 0.5m。逼近时 γ 取 0.1, R_1 与 R_0 的差值取 0.01。图 3 中(e)为相关点调整之后,但是没有删除岛屿面片的结果,新添加的红色面与岛屿面片有相交的部分,修复后的形状与实际更加接近;图 3 中(f)是删除岛屿面片后,最终修复后的模型。

在 Z 轴方向上取一个截面,获得二维的曲线。如图 4 为 Z 为 0.3 时的曲线图。利用岛屿孔洞修复

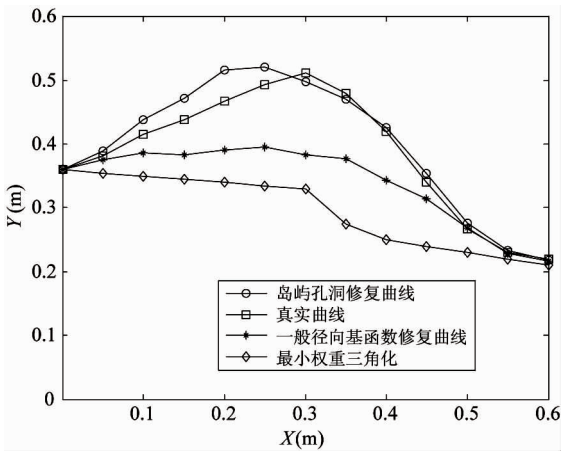


图 4 Z = 0.3 时各方法的曲线

后的曲线与真实的曲线最为接近,修复后的模型光滑并且准确。

从结果可以看出,本文的方法是有效的,修复以后的模型更加接近实际形状,并且平滑地与模型主体相接。表 1 给出了在这个实验中新增的点和面的信息。

表 1 实验中的数据

修复方法	修复用到的点	新增的点	新增的面	需要时间(s)	平均误差(m)
最小权重三角化	26	0	24	0.007	0.14
一般径向基函数	26	47	82	0.067	0.11
岛屿孔洞修复方法	40	51	82	0.092	0.08

本文方法还应用到实际项目中,下面介绍重构燕大电气工程学院石头的项目。图 5 中的 3 张照片是从 25 张石头照片中选出来的。利用 Structure from Motion,可以获得照片的位置信息和稀疏的点云,如图 6 中的左边照片所示。图 6 中右边照片是经过加密以后获得点云^[13]。



图 5 重构石头所用的照片



图 6 图片的位置信息与石头的点云

本文的目的是获得石头的模型,如图 7 中左边图片所示。图 7 中右边图片是实际中带有岛屿面片的孔洞,岛屿面片由于没有与模型主体相连接,拓扑结构也不一致。对图 7 中的孔洞依次经过孔洞检测、孔洞粗修复、新增点的位置调整和删除岛屿面

片后得到最终的结果,图8中四张照片与这个过程相对应。

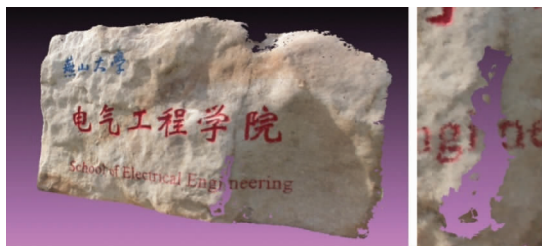


图7 删除无效区域的模型与带有岛屿面片的孔洞

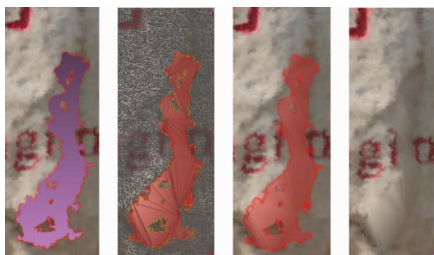


图8 修复石头孔洞过程中的效果图

5 结 论

本文提出的相关孔洞的检测方法,设置合理的参数后,能够检测到相关的孔洞;将岛屿面片上的点应用到径向基函数的计算中,增加了修复中所用点的数量,能够更好地用径向基函数来表示孔洞处的曲面,修复的曲面误差小。检测模型主体上孔洞与岛屿面片的相关性,基于投影和距离,具有局限性。综上所述,该方法对缺陷模型修复,新增加的面能与原始的面平滑过渡。添加岛屿面片上的顶点用于孔洞修复中径向基函数的计算,增加了孔洞修复中曲面的复杂度,有利于提高修复后模型的精度。可以应用到实际模型的孔洞修复中,并且修复的模型较为光滑、准确。

参考文献

[1] Davis J, Marschner S R, Garr M, et al. Filling holes in complex surfaces using volumetric diffusion. In: Proceedings of the IEEE International Symposium on 3D Data Processing Visualization & Transmission, Padova, Italy, 2002. 428-441

[2] Curless B, Levoy M. A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, New York, USA, 1996. 303-312

[3] Podolak J, Rusinkiewicz S. Atomic volumes for mesh completion. In: Proceedings of the Symposium on Geometry Processing, Vienna, Austria, 2005. 33-41

[4] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: Proceedings of the 4th Eurographics Symposium on Geometry Processing, Cagliari, Italy, 2006. 7-7

[5] Zhao W, Gao S, Lin H. A robust hole-filling algorithm for triangular mesh. *Visual Computer*, 2007, 23(12):22

[6] Centin M, Pezzotti N, Signoroni A. Poisson-driven seamless completion of triangular meshes. *Computer Aided Geometric Design*, 2015, 35: 42-55

[7] Xie W C, Zou X F. A triangulation-based hole patching method using differential evolution. *Computer-Aided Design*, 2013, 45(12):1651-1664

[8] Carr J C, Beatson R K, Cherrie J B, et al. Reconstruction and representation of 3D objects with radial basis functions. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, New York, USA, 2001. 67-76

[9] 晏海平,吴禄慎,陈华伟. 基于径向基函数的散乱点云孔洞修复算法. *计算机工程与设计*, 2014, 35(4): 1253-1257

[10] 顾园园. 散乱点云孔洞修补技术的研究与实现:[硕士学位论文]. 苏州大学计算机科学与技术学院, 2008. 33-36

[11] Deng C, Wang G. Interpolating triangular meshes by Loop subdivision scheme. *Science China Information Sciences*, 2010, 53(9):1765-1773

[12] Karkanis T, James Stewart A. Curvature-dependent triangulation of implicit surfaces. *IEEE Computer Graphics and Applications*, 2001, 21(2):60-69

[13] Xiao J, Owens A, Torralba A. SUN3D: A database of big spaces reconstructed using SfM and object labels. In: Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 2013. 1625-1632

Automatic repair of point cloud holes with isolated surfaces based on radial basis function

Zhang Liguo, Wang Jing, Jin Mei, Kang Le

(Measurement Technology and Instrumentation Key lab of Hebei Province, Yanshan University, Qinhuangdao 066004)

Abstract

The hole patching for the triangular meshes obtained by using point clouds under existing conditions was studied, and aiming at the serious time consumption problem of the manual repair method in common use at present, a method for automatic repair of the holes with isolated surfaces based on radial basis function (RBF) was proposed. This method uses the minimum weight triangulation method to fill the holes on the main body of the mesh; then calculates the correlation of the holes on the main body of the mesh and the isolated surfaces, and calculates the RBF by using the holes on the main body of the mesh and their related isolated surfaces' around points; and finally, adjusts the vertexes obtained by subdivision to the curved surface expressed by RBF. The experimental results show that the proposed method can achieve fast, accurate repair of the meshes with holes compared to other approaches.

Key words: point cloud, hole, isolated surface, radial basis function (RBF), minimum weight triangulation