软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(10):2642−2653 [doi: 10.13328/j.cnki.jos.005078]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

带洞点云多层同步表面重建方法*

王筱婷[1,2], 王 璐[1,2], 孟祥旭[1,2]

[1](山东大学 计算机科学与技术学院,山东 济南 250101)

[2](数字媒体技术教育部工程研究中心,山东 济南 250101)

通信作者: 王璐, E-mail: luwang_hcivr@sdu.edu.cn

摘 要: 针对法向信息缺失和采样点缺失的带有洞的散乱点云数据,提出了一种高效、高质量的多层同步表面重建方法.首先利用动态等高线检测出含有洞的八叉树节点,并且基于HPR(hidden point removal)映射计算出八叉树顶点的内外状态,建立带有顶点内外标识的空间有向状态八叉树,然后基于八叉树节点内法向测试方法保证基于 k 近邻表面重建过程中采样点的法向的正确性,且该空间有向状态八叉树可以支持不同层次的点云同步重建,在保证重建结果正确性的前提下,提高重建效率.

关键词: 表面重建;点云;状态八叉树;k 近邻;HPR

中图法分类号: TP391

中文引用格式: 王筱婷,王璐,孟祥旭.带洞点云多层同步表面重建方法.软件学报,2016,27(10):2642−2653. http://www.jos.org.cn/1000-9825/5078.htm

英文引用格式: Wang XT, Wang L, Meng XX. Multi-Layers surface reconstruction method for point set with holes. Ruan Jian Xue Bao/Journal of Software, 2016,27(10):2642−2653 (in Chinese). http://www.jos.org.cn/1000-9825/5078.htm

Multi-Layers Surface Reconstruction Method for Point Set with Holes

WANG Xiao-Ting[1,2], WANG Lu[1,2], MENG Xiang-Xu[1,2]

[1](School of Computer Science and Technology, Shandong University, Ji'nan 250101, China)

[2](Engineering Research Center of Digital Media Technology, Ministry of Education, Ji'nan 250101, China)

Abstract: This paper proposes a multi-layer surface reconstruction method based on a special oriented status octree. The method is designed to handle point sets with missing normal information and with holes. First, the octree cells distributed on holes are detected by active contours. By using hidden point removal (HPR) operator, the inside or outside status of each corner of octree cells are calculated, and the mono-oriented status octree is constructed. Then the normal direction of the initial points set inside the status octree is determined by in-cell normal detection method, and the parallel multi-layer surface reconstruction from k-nearest neighbors is carried out by using the status octree. The proposed method improves the construction efficiency while guaranteeing the construction quality.

Key words: surface reconstruction; point set; status octree; k-nearest neighbors; HPR

1 引 言

基于三维扫描仪得到的点云数据进行三维建模的过程称为表面重建.在利用三维激光扫描仪获得点云模

* 基金项目: 国家自然科学基金(61472224, 61472225); 山东省自主创新及成果转化专项(2014ZZCX08201); 山东大学青年学者未来计划(2015WLJH41); 泰山学者建设工程专项经费

Foundation item: National Natural Science Foundation of China (61472224, 61472225); Special Funding of Independent Innovation and Transformation of Achievements in Shandong Province (2014ZZCX08201); Young Scholars Program of Shandong University (2015WLJH41); Special Funds of Taishan Scholar Construction Project

收稿时间: 2016-01-19; 修改时间: 2016-03-25; 采用时间: 2016-05-09; jos 在线出版时间: 2016-08-08

CNKI 网络优先出版: 2016-08-09 15:37:40, http://www.cnki.net/kcms/detail/11.2560.TP.20160809.1537.002.html

© 中国科学院软件研究所 http://www.jos.org.cn

型的数据时,点云的法向信息往往无法获取.此外,在扫描的过程中,由于有些地方无法扫描到,导致数据在拓扑上形成洞(hole)或者数据分布不均匀.这些信息缺失的噪声数据,严重影响了重建结果的正确性.所以,如何高效建模重建复杂对象和场景,成为目前学界研究的难点和热点问题.本文重点研究带有洞的、法向缺失的、采样点缺失的散乱点云数据的高效表面重建方法,提出基于空间有向状态八叉树的多层同步表面重建方法.针对点云数据带有洞的问题,本文方法通过利用动态等高线,可有效检测出含有洞的八叉树节点,并针对空间八叉树进行补洞.针对点云法向信息缺失的问题,本文通过 HPR(hidden point removal)映射可计算出八叉树节点每个顶点的内外状态,该带有顶点内外标识的有向状态八叉树可保证基于 k 近邻的表面重建过程的表面法向正确性.具体地,本文采用基于八叉树节点内法向测试的方法来保证表面重建过程中采样点的法向的正确性,并且该方法支持空间有向状态八叉树不同节点内的点云数据进行同步重建,打破了层剥法的局限,可以支持不同层次的点云并行重建,在保证重建结果正确性的前提下,提高了重建效率.

2 点云表面重建相关算法

Boissonnat[1]等人于 19 世纪 80 年代中期提出第一个表面重建算法.依据重建方法不同,目前的表面重建算法[2]主要分为两类:显式重建方法和隐式重建方法.显示重建方法又可以分为局部算法和全局算法.局部算法使用 k 近邻构建表面[3];全局算法使用 Voronoi 图和 Delaunay 三角化或者隐式曲面抽取表面,如 Cocone 算法[4]和 Powercrust[5]算法等.显示重建方法结果精确过初始采样点,但是计算复杂,且对噪声敏感.隐式重建方法用一个隐式函数对点云进行拟合,如基于径向基函数(RBF)的曲面重建[6]、基于泊松方程的曲面重建[7,8]、多层次剖分方法(MPU)[9]、基于 B-Spline 的重建方法[10]、插值面扩展的方法[11]等,隐式重建方法计算代价低,但是重建结果是对物体真实表面的逼近估计,难以保持表面细节特征.

上述表面重建算法是否成功和采样点的质量,例如采样精度、采样点封闭、采样点法向正确性等信息密切相关.目前针对信息缺失的噪声数据研究普适性的表面重建方法是一个热点问题.依据输入数据的不同,目前表面重建算法主要关注如下 5 种输入情况:具有噪声点的点云数据、法向缺失的点云数据、非均匀采样或低精度采样的点云数据、带有洞的点云数据、大规模的点云数据.

针对具有噪声的点云数据,基于泊松方程的表面重建方法[7]把有向点集的表面重建可以转化为一个空间泊松问题,该方法是全局性方法,同时考虑了所有的点,而不是借助于启发式的空间分割或合并,对数据的噪声有很大的帮助.为了克服过平滑问题,HOPPE 等人[8]提出采用点约束可以更好地保持表面细节,并可加速重建过程.泊松方法可以生成一个封闭的模型表面,但仅适用于空间有向点集.Giraudot 等人[12]提出一种自适应噪声的表面重建方法,通过计算鲁棒的距离函数,基于最小二乘法去除噪声,但是该方法需要假设物体是平滑的封闭流形.

针对法向缺失的点云数据,Cocone 表面重建算法[4],基于 Delaunay 三角化进行全局表面重建方法,重建结果具有优越性,但是难以对带有洞的点云数据进行处理,并且对于非均匀采样的点云模型,易产生错误结果.Chen 等人提出了一种基于二向树 BOT(binary orientation trees)的离散点集表面重建方法[13],该方法可以处理无法向信息的点云模型,并且执行的过程简单,通过对单元格顶点内外状态的确定可以有效地采用隐式表面重建算法,并过滤噪声产生的影响,但当输入的点云模型包含洞时,则会使重建的结果发生错误.

为处理带有洞的点云数据,Carr 等人[6]利用径向基函数(RBF)拟合三维数据,提取其中的等值面得到重建曲面,该算法利用 RBF 强大的非线性逼近能力,可以修补大量的缺失数据,但是该方法易出现过平滑现象,且内存消耗高,难以处理大规模数据.MPU 方法[9]改进了该思想,通过八叉树将点云数据划分成若干个子区域,并利用分段二次曲面拟合每个子区域的点云数据,最后拼接出全局光滑的重建曲面,该方法不仅可以完成点云孔洞的修补,还可有效减少重建过程的时间和空间复杂度.上述方法可有效处理带有洞的点云数据,但均需初始点云具有法向信息.Xie 等人对 MPU 方法进行了改进,提出了单朝向区域(mono-oriented regions)的理论,利用动态等高线碰撞的原理来检测出洞的区域[14].该方法可保持等高线扩展的速度与到洞的距离相等,那么当两条不同的等高线在某个区域发生碰撞时,则可认为该区域是洞的区域.该方法采用局部隐式表面重建的方法适用于法向缺

© 中国科学院软件研究所 http://www.jos.org.cn

失的点云数据.Yoshihara 等人通过 level set 方法确定点云的拓扑结构,用迭代的方法生成新的表面[10].Lin 等人用种子三角形向周围点扩散建立三角形网格的方法进行补洞,但是需要一个后续过程来修正拓扑结构[11].上述点云空洞修补方法均是隐式重建方法,重建结果和模型真实表面存在偏差,而目前针对显式重建方法的补洞方法尚比较缺乏.

针对法向信息缺失,且非均匀采样、低精度采样或者欠采样的点云数据,Lim 和 Tan[3]提出一种表面重建层剥法(layer peeling),该方法认为一个模型的表面是由多层表面片构成的,其中最外层(即第 1 层)的表面片如果沿着法向方向向外无限延伸,则永远不会和该模型的其他表面片相交.如果剥除前(i–1)层,第 i 层的表面片如果沿着法向或者法向的反方向,向外无限延伸,亦不会与剩余的表面片产生相交.在层剥法的执行过程中,使用全局法向映射测试可以保证表面重建总是从最外层开始,并逐层向内部扩展.通过这样一层一层的处理,该算法可以保证各层表面法向的正确性.并且,层剥法可以处理低精度采样或者欠采样的点云数据,避免非均匀采样引起的不同层次的点云连接错误,但是层剥法表面重建的过程较为复杂,需要一层层地对模型进行重建,对于小洞可以修补,但是对于带较大孔洞的模型,沿着法向正反两个方向延伸,均有可能不会碰到模型表面,故内层点云容易被误判为最外层点云数据,从而产生错误的重建结果.还有一些针对带洞网格模型的补洞方法,如使用泊松公式优化新网格的洞填充算法[15].

针对大规模的点云数据,目前通常采用内外存调度或者流处理的方法进行大规模的空间点云数据的表面重建,这些算法大多都需要局部算法,对输入数据有严格的限制.在已知采样点法向信息的前提下,Dey 等人[16]拓展了 Cocone 算法,可以对几百万个点的输入数据进行表面重建,该方法通过空间八叉树将点云数据组织在空间单元中,且对每一个单元进行 Cocone 计算.Allègre 等人[17]过改动 Chaine 的几何传输方法对一条条的输入数据流进行处理.算法通过加入一些额外的辅助点,使得通过 Delaunay 三角化得到四面体的包围球足够小,以适用于局部处理的需求.Bolitho 等人[18]在泊松表面重建的基础上使用自适应八叉树,并在局部区域应用泊松方程,但是该方法要求输入点云数据自带法向信息.此外,Manson 等人[19]提出基于小波的流处理表面重建方法,结合隐式表面重建和基于小波/细分的多层隐式方程对具有法向信息的点云数据进行表面重建,该方法需要输入数据按照某欧几里德方向预排序.上述方法均对输入数据具有严格要求,或者要求有法向信息,或者需要对输入数据进行特殊组织或者预排序,重建过程较为复杂.

本文重点研究带有洞的、法向缺失的、非均匀采样或者欠采样点云数据的高效表面重建方法,且可以处理大数据量的点云数据.

3 基于有向状态八叉树的多层同步表面重建方法

本文提出一种可以处理法向信息缺失和采样点缺失的带有洞的点云数据的方法.该方法可以高效地检测出含有采样点丢失的八叉树节点,同时准确确定采样点的法向信息,并且不同层次间的点云数据可以进行同步重建.

该算法主要包括 5 个步骤.

(1) 读取点云数据流,根据点的数目确定八叉树深度(不超过 10),并构造出能存储于单机内存的空间八叉树,结果如图 1(b)所示.

(2) 根据已经构建的八叉树的每个单元格的信息,检测出哪些空的单元格是洞,检测出的结果如图 1(c)所示(黄色表示检测出来的带洞的单元格).

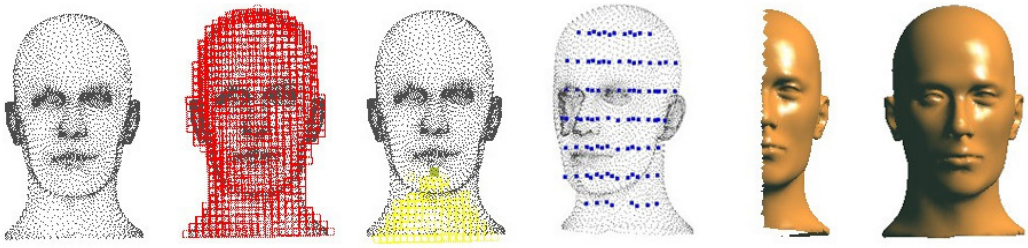
(3) 确定局部空间朝向,根据 HPR 操作计算出每个单元格顶点的内外状态,计算出的结果如图 1(d)所示(其中,在模型内部的点标记为蓝色(部分)),完成空间有向状态八叉树的构建.

(4) 根据单元格的每个顶点的内外状态,对单元格内采样点进行格内法向测试,通过测试的采样点作为种子点构建三角面扇,并向外扩展创建新的面扇,新增加的三角形进一步做格内法向测试,以保证重建结果的正确性.图 1(e)是在网格重建时的中间过程.

(5) 将已经构建的局部区域合并到一起,即为最终的重建结果,如图 1(f)所示.

© 中国科学院软件研究所 http://www.jos.org.cn

下面将详细介绍空间有向状态八叉树的构建和局部表面重建过程.



(a) 点云数据 (b) 空间八叉树 (c) 带洞的单元格(黄色) (d) 模型内部的点(蓝色) (e) 重建中间过程 (f) 重建结果

Fig.1 The work flow of our algorithm

图 1 本文算法流程

3.1 空间有向状态八叉树的建立

3.1.1 八叉树创建

假设输入点云数据有 n 个采样点,本文算法将八叉树的深度设置为 $h=\log_4(2.5n/c)+1$,其中,c 是叶节点期望包含点的平均数目,该算法中取 48.该八叉树深度计算公式是一个启发式公式,通过实例观察,对于深度为 h 的八叉树,非空叶节点的数目的比率在 $0.2/2^{h-2}$ 到 $0.6/2^{h-2}$ 之间.若输入数据过大,本文限定最大层数为 10,以保证八叉树可存储于内存.

本文采用自顶向下的方式构建空间八叉树.八叉树的初始高度设为 0,只有一个根节点,即输入数据的包围盒.当一个点 p 读入内存后,逐层向下寻找或者新建立包含 p 点的节点,直至叶节点.在新节点的创建过程中,父节点被分裂成 8 个新的子节点,并且分裂后的八叉树深度不可超过预先设置的八叉树的最大深度.当 p 点找到包含它的叶节点后,将 p 经过所有节点中包含点的数目均增加 1.

3.1.2 检测包含洞的节点

三维点云数据获取过程中往往存在洞(hole),如图 1 所示,对于 face 模型,由于颈部被截取,所以点云模型的下方有一个洞.洞的存在给空间八叉树的顶点状态计算带来困难.故本文首先检测八叉树中包含洞的单元格,生成一个封闭的空间八叉树.

本文采用文献[11]中检测洞的方法,在模型的内部和外部分别找到两个动态等高线,当两个动态等高线发生碰撞时,发生碰撞的区域即为洞.在检测洞时,为了找到合理的洞的区域,需要在模型的内外找两个点,并且保证这两个点到洞的距离相等且速度相等,因此根据这两个点的动态等高线会保证在合理的区域发生碰撞,而不会偏差过大.对于一个含有洞的模型,很难很快地确定这个模型的一个内点和一个外点,并且也无法保证它们到模型的距离等同且速度也相等.所以本文在算法实现上采用如下的方法.

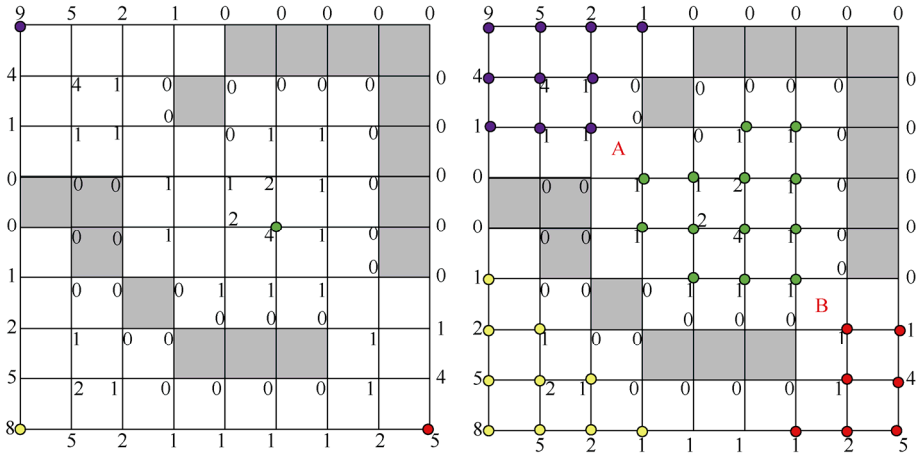
空间八叉树划分中所有叶子节点形成了一个空间网格.每个空间网格点记录其最近的非空的单元格的距离(此处使用距离的平方)(如图 2(a)所示).对于整个空间网格点,必然存在一些空间网格点,这些点的距离值比其相邻的空间网格点的距离值都大,那么这些点就叫做极值点.每个极值点都是一个动态等高线的起始点(图 2(a)中红色、黄色、紫色、绿色的点都为极值点).

基于图 2(a)所示的 4 个极值点,逐层传播构成 4 个动态等高线(即为 4 个组)(如图 2(b)所示),红色、黄色、紫色和绿色分别代表不同的等高线.在不断的传播过程中,其中有两个不同的组在单元格 A、B 中分别发生了碰撞,就认为 A、B 两个单元格为洞的单元格.

在动态等高线的传播过程中,使用堆存储单元格顶点,该堆中按照单元格顶点的距离值排序,堆顶存储距离值最大的单元格顶点.首先将初始极值点存储于一个堆中,且每个极值点标记一个组号,每次从堆顶的单元格顶点进行传播,检测该顶点周围的单元格顶点.若该顶点组号尚未标记,且距离值小于堆顶的距离值,则将该顶点

© 中国科学院软件研究所 http://www.jos.org.cn

的组号标记成堆顶的组号,并将该顶点加入堆中.对于每个单元格,如果 8 个顶点都属于同一个动态等高线(组号相同),则它就不属于洞,否则就属于洞.



(a) 极值点(红色、黄色、紫色、绿色的点) (b) 洞的单元格(A,B)

Fig.2 Distance value of corners of cells and active contours

图 2 单元格顶点的距离值和动态等高线

将所有计算出来的洞的单元格标记为“Hole”,若这些单元格填充后可以很好地把模型的洞的区域包裹住,构成一个封闭的空间八叉树,则可保证之后的求单元格顶点内外状态检测时算法的正确性.

3.1.3 有向状态八叉树的创建

有向状态八叉树即每个单元格的 8 个顶点均标记“内”、“外”两种状态.“内”表示位于模型内部,用“–”表示;“外”表示位于模型外部,用“+”表示.该状态为局部表面重建提供支撑.在计算非空单元格的顶点的内外状态时,本文采用 HPR 的思想[20]:以模型外部的一个单元格顶点作为视点,单元格内点云和其他单元格顶点作为观察对象,这个视点所能看到的点都处于模型外部,而看不到点的都处于模型内部.

1.HPR 操作

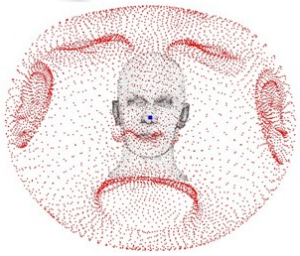


Fig.3 HPR operator for face model

图 3 Face 模型 HPR 映射

假设视点是 v,首先需要把坐标系的原点移到视点 v 的位置,坐标变换后的点集合假设是 P.构造映射函数,使得 $p_i(p_i\in P)$映射后的点在 vp_i 的射线上,并且是以 vp_i 向量的模单调递减的.采用的映射函数如下:

$$\hat{p}_i = p_i + 2(R-\|p_i\|)p_i/\|p_i\|,$$

其中,R 是 v 到达点集 P 的最远距离.由于 v 是坐标原点,所以用 p_i 表示原点到 p_i 的向量,$\|p_i\|$是向量的模.

经过 HPR 映射后,点集 P 中 v 可见的点均分布于一个凸包上.如图 3 所示,蓝色的点表示视点,红色的点是 face 模型采样点映射后的结果,所有落在凸包上的点相对于视点都是可见的.具体地,计算可见点时,不需要完全把凸包计算出来,只需要求出变换后采样点的 3 近邻点,分别与其中的任意两个点组成一个平面,如果新的点集合都处在组成的平面的一侧,则该点相对于视点是可见的.

2.单元格顶点内外状态计算

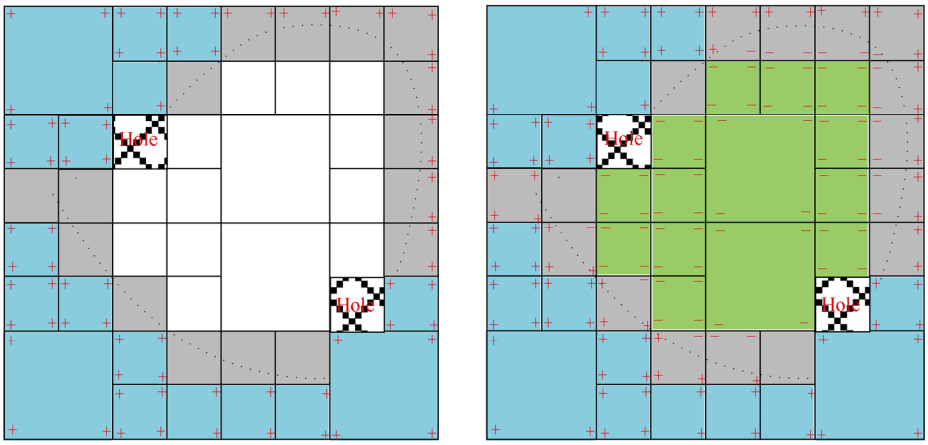
对于单元格中的每个顶点要么是在模型的内部,要么是在模型的外部.如果一个单元格顶点在模型的外部,则这个点通过 HPR 计算可见性时是可见的.所以算法开始需要找一个处在模型外部的单元格的顶点.假定处在一个模型外包围盒上单元格顶点都是处在模型外部的.所以可以将这些顶点的状态标记为外状态,并作为初始视点,进行基于 HPR 映射的可见性计算,从而计算出所有的单元格顶点的内外状态.

© 中国科学院软件研究所 http://www.jos.org.cn

单元格顶点的内外状态的计算具体可分 4 个步骤.

步骤 1. 计算空单元格的内外状态.

从包围盒 8 个角上的空单元格开始,将这些单元格标记为状态“外”,并入堆栈,取栈首单元格,对和其共享面的紧密临近的空的且非洞区域的单元格进行 flooding,标记新的空单元格状态为外部,并入堆栈,直至所有空单元格标记完成.标记为外部的空单元格的 8 个顶点状态均为外部,即标记为“+”.以二维空间示意图 4(a)为例.其中,涂蓝色的单元格是已经确定的外部的空的单元格,并且将这些单元格对应的 4 个顶点标记为外状态.



(a) 步骤 1 结果,涂蓝色的单元都是处在模型外部的空的单元格

(b) 步骤 2、步骤 3 结果,图中的绿色的单元格表示处于模型内部的空的单元格

Fig.4 2D schematic diagram

图 4 二维平面的示意图

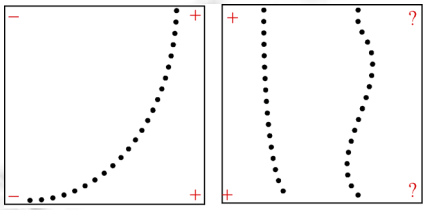
步骤 2. 计算非空单元格顶点的内外状态.

如果一个单元格非空,并且非标记为含有洞“hole”的单元格的邻居,则可以通过 HPR 映射计算该非空的单元格顶点内外状态.选取该单元格上已标记为处于外部的顶点 p 作为视点,单元格内所有的点云数据和单元格其他未标记内外状态的顶点 q 作为被观察对象,根据 HPR 映射计算被观察对象的可见性.

1) 如果网格顶点 q 针对 p 可见,则该网格顶点 q 也处于外部(“+”).

2) 如果存在单元格顶点 q,对于该单元格的其他所有已标记的外部顶点 p 均不可见.

a. 如果存在一个单元格顶点,从该顶点作为视点,单元格内的点云数据均可见,则该顶点 q 标记为内部(“–”),且该单元格标记为“单调单元格”.如图 5(a)所示.



(a) 单调单元格 (b) 非单调单元格

Fig.5 Status of non-empty cell

图 5 非空单元格顶点的内外状态

b. 否则,顶点 q 不能标记状态,并且该单元格标记为“非单调单元格”.如图 5(b)所示.

如图 4(b)所示.通过步骤 2,可以将部分非空的单元格的顶点的内外状态计算出来.

步骤 3. 计算内部单元格顶点的内外状态.

如果一个单元格有一个顶点已经标记为内部顶点“–”,并且该单元格为空,则将该单元格 8 个顶点状态均设置为“–”,并采用步骤 1 所示的 flooding 方法,对所有空的内部单元格顶点状态进行赋值.

步骤 4. 计算洞附近单元格和“非单调单元格”未确定状态顶点的内外状态.这些单元格的顶点状态依据临近单元格顶点的内外状态来确定.

© 中国科学院软件研究所 http://www.jos.org.cn

通过上述 4 步计算即可构建带有顶点内外状态的空间状态八叉树,以保证基于层剥法的表面重建过程中法向的正确性.

3.2 多层同步表面重建

在构建模型表面的过程中,算法根据 k 近邻的局部特性来构建三角网格模型.本文方法和 Lim 及 Tan[7]提出的全局法向测试方法不同,在他们的方法中,对于第 i 层上的任意点 p,必须等待第(i–1)层上的所有点都构建好表面,点 p 才有可能通过全局法向测试.而本文方法可以通过格内法向测试首先确定位于任意层次的满足测试条件的点的法向,并作为种子点开始重建过程.本文方法可以通过多线程编程或者多核程序,选取多个通过格内法向测试的种子点,这些种子点可能位于不同层次上,并从多个种子点开始可进行不同层次上点云的同步表面重建.图 6 选取 4 个种子点对 Heptoroid 模型进行多层同步表面重建(共 26 层),过程如图 6 所示.

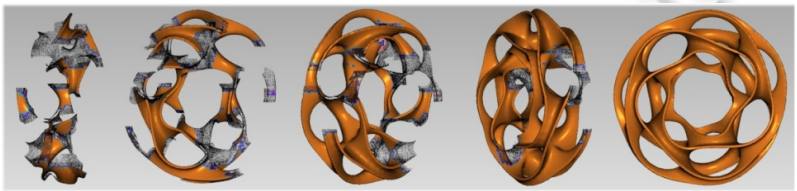


Fig.6 Multi-Layers surface reconstruction process for Heptoroid model

图 6 Heptoroid 模型多层同步表面重建过程

3.2.1 基于有向状态八叉树的格内法向映射

为了保证重建结果的正确性,重建过程依据状态八叉树进行格内法向映射是关键,以保证点云法向计算和重建后表面朝向的正确性.对于任一输入点 p,首先根据该点的 k 近邻预估计该点的法向 n,然后沿着 n 或者 n 的反方向($-n$)射出一条射线.依据该射线和所在单元格交点所在面的内外状态,确定该点法向的外朝向.

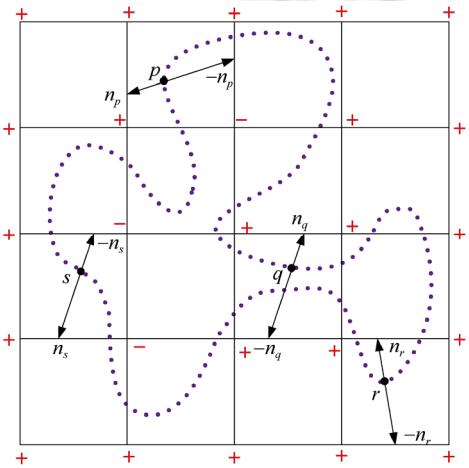


Fig.7 In-Cell normal projection test

图 7 格内法向测试

格内法向映射过程如图 7 所示.在格内法向映射过程中,对于一个采样点,

1) 如果采样点所在的单元格标记为“非单调单元格”,如图 7 所示中的点 q,沿着 q 的法向方向 nq 或者逆方向$-nq$ 与所在单元格内的面相交,则两个相交面可能均标记为外侧或者内侧,故非单调单元格内顶点的法线方向无法确定.本文通过进一步细分该单元格的形式,直到单元格内点云数据保持“单调”,再进行法向测试.

2) 如果采样点所在的单元格已经标记为“单调单元格”,如图 7 所示的 p、r、s 这 3 个采样点所在的单元格,则沿着采样点的法向方向和法线反方向与单元格的面相交.

a) 如果存在一个相交面的 4 个顶点状态相同,另一个相交面上存在一个顶点状态与之前相交的面的顶点状态互斥,则该采样点可确定法向方向,如图 7 中的点 p 所示.

b) 如果两个相交面标记状态相同,则法向无法确定,如图 7 中的点 r 所示,需要对单元格进行进一步细分.

c) 如果两个相交面中,每个相交面的 4 个顶点的状态均不一致,则该点法向无法确定,例如图 7 中点 s 所示.那么,s 不能通过格内法向测试.

3.2.2 构建模型表面

表面重建过程采用三角面扇来表达局部表面信息.假设点 p 的 k 近邻用 kNN_p 表示,p 点处的三角面扇用 T_p 表示.T_p 由点 p 依次与点 $p_0,p_1,\ldots,p_i$ 相连,其中,$p_0,p_1,\ldots,p_i\in N_p$.点 $p_0,p_1,\ldots,p_i$ 构成子集 Q_p,称为点 p 的表面邻居.p

© 中国科学院软件研究所 http://www.jos.org.cn

点处的法向量 n_p 可以通过 T_p 中三角形法向的角度加权平均得到.依据如下两个准则构建三角面扇.

局部最凸准则:kNN_p–Q_p 中的点不会位于三角面扇 T_p 的上面(沿着 p 的法向方向).换言之,T_p 位于邻近点的最外层.

法向连贯准则:对于三角面扇中的任一三角形 $t_j \in T_p$,t_j 的法向量为 n_{t_j}, 则有 $n_p \cdot n_{t_j} > 0$.

局部最凸准则可以避免三角面扇连接到其他表面层次上的点,通过该准则可有效处理低精度采样的点云数据,并有效避免位于不同表面层次上的点云数据产生互连.

法向连贯准则使得三角形扇可以用类似的磁盘的拓扑结构表示该局部表面信息,使得三角面扇关联到的点投影到 2D 平面上后,仍然保持原有连接次序的前提下,不会自相交.

具体地,构建模型表面过程如下:

1) 首先选择一个采样点,依据其 kNN 的法向方向进行格内法向测试,如果该点通过测试,则可作为种子点并构建一个三角面扇.对于种子点 p,将其 kNN 的包围球用分割线均匀分成 5 份,在每一份中寻找距离分割线最近的采样点,且该点与 p 点的距离需要满足某阈值设定,这些点依次连接构成三角面扇 T_p 相关联的点.如果三角面扇法满足局部最凸和法向连贯准则,则将 T_p 加入已构建表面 M 中.

2) 对于已构建网格 M 的边界点 p,使用 p 的某一关联点 q 作为依据点,开始构建 p 的三角面扇.使用 q 点通过贪心算法从 p 的 k 近邻 kNN_p 中寻找一个点作为下一个三角形的连接点,该新构建的三角形需要满足面积最小和二面角最小的原则.如果无法找到合乎条件的三角形,则算法回溯使用 p 的下一个关联点作为起始依据点.如果回溯到 q 点,则说明无法在 p 点构造合理的三角面扇.如果 p 点处可以顺利构造满足条件的面扇 T_p,则判断 T_p 是否满足局部最凸和法向连贯准则,如果满足,则将 T_p 合并入已构表面 M 中.在两个三角面扇 T_p 和 T_q 的合并过程中,需依据 p 和 q 两点正确的法线方向进行合并.

上述表面重建方法可以扩展为一种并行算法,通过将点云数据划分为几组在不同的处理器并行处理,每组数据又可以选择多个通过格内法向测试的种子点进行同步表面重建,可有效提升重建效率.同样,该方法是一种局部算法,适用于处理大规模的点云数据,通过对不同单元格内的数据读入内存后分别进行处理,再进行合并的形式,算法可处理数据量达到几百万个点的输入数据.

3.2.3 补洞处理

本文表面重建方法是一个局部显式重建方法,生成的三角网格模型过初始点云数据,故对于带有洞的点云数据重建出的网格模型也带有洞,若想得到封闭的表面模型,本文采用简单补洞算法来填补这些漏洞,但补洞结果并非能预估模型初始的形状.由于洞上的边界点可构成封闭的环,故补洞过程即对此环进行修补.补洞过程中,依据洞上每一点边界点 p_i 与其相邻的上一个洞上点 p_j 和下一个洞上点 p_k 之间的夹角作为修补顺序的标准,小的夹角具有高的优先级.每次检测具有最高优先级的 3 个连续边界点 p_i,p_j 和 p_k,并构建三角形,更新各相关点三角面扇.依据该贪心算法,可在漏洞处构造新的三角网格,进行漏洞填充.构建结果如图 8 所示.

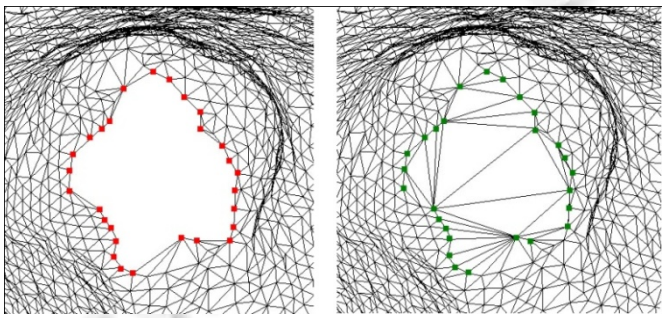


Fig.8 Hole filling process. Left: Boundary points located on a hole constitute a closed ring; Right: Hole filling

图 8 补洞处理过程.左:洞上边界点构成封闭环;右:补洞

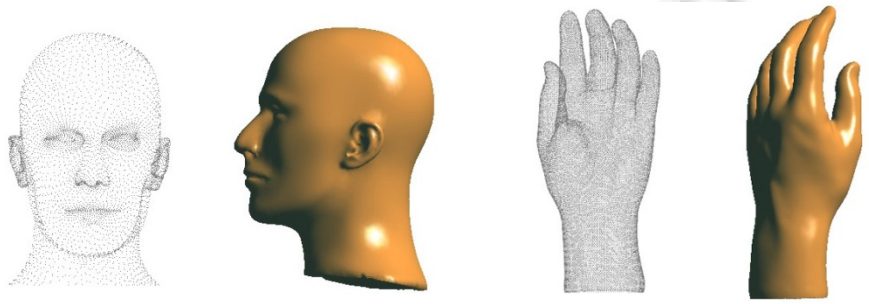
© 中国科学院软件研究所 http://www.jos.org.cn

4 实验结果与分析

本文对上述算法进行了大量验证,实验环境为 Visual C/C++.以下实验数据的实验平台为 Window Vista,Intel 双核 2.6GHz CPU,内存为 4GB.为了与其他算法的实验结果进行比较,目前算法只在单核 CPU 运行,没有实施多线程处理.

4.1 重建质量

本文适合对于法向缺失且带洞的点云数据进行表面重建,图 9 给出了两个带有单个洞的点云模型 Face 和 Hand 的重建结果,本文方法通过空间有向状态八叉树的建立以及格内局部法向测试,可有效保证重建过程中点云法向计算的正确性,从实验结果容易看出,本文方法可以处理带洞的点云数据.



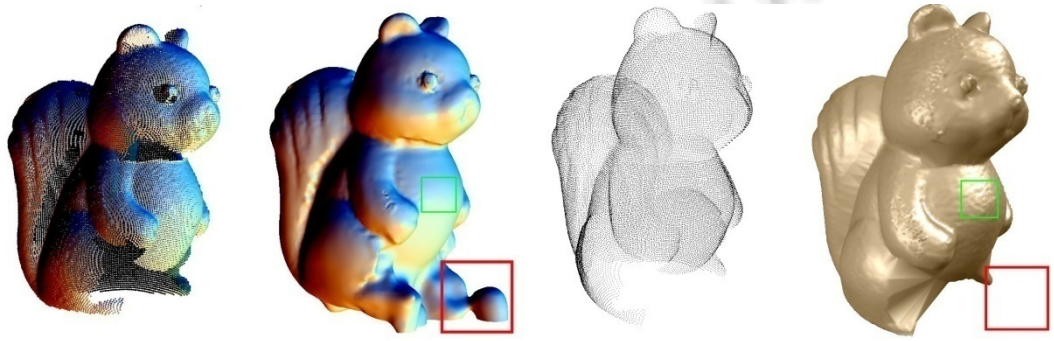
(a) Face 模型点云数据 (b) 本文方法重建结果 (c) Hand 模型点云数据 (d) 本文方法重建结果

Fig.9 Reconstruction results of point set with one hole

图 9 带单个洞的点云模型重建结果

为了比较重建质量,考虑到目前常用的补洞方法均是隐式重建方法,本文与适合补洞的 MPU 方法[9]的重建结果进行比较.考虑到 MPU 方法需要点云初始法向信息,首先经过预处理为 MPU 方法提供带有法向的点云模型作为输入(如图 10(a)、图 11(a)、图 12(a)和图 12(b)所示),而本文方法不需要输入点云模型即具有法向信息(如图 10(c)、图 11(d)、图 12(d)所示).

图 10 给出了带有多个复杂漏洞的均匀采样的点云数据的重建结果,MPU 方法和本文方法均可进行重建.MPU 方法得到全局光滑的封闭表面,故在洞附近的区域易产生多余的出格点,例如图 10(b)中 squirrel 模型的左脚(红色区域所示).本文方法是显式重建方法,可有效避免该问题,如图 10(d)中红色区域所示,但本文方法不能保证补洞结果能够体现物体初始形状.此外,本文方法与 MPU 方法相比,可以避免过平滑现象,可以更好地保持模型的表面细节,如图 10 中绿色区域所示.



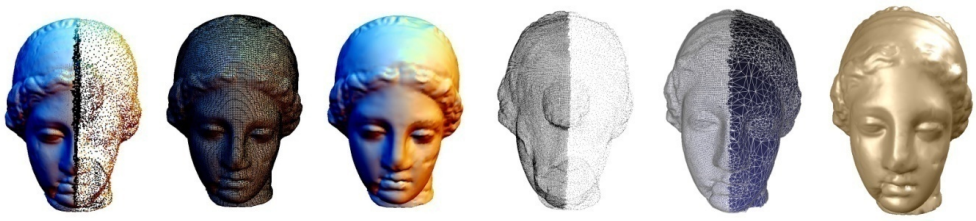
(a) Squirrel 带法向点云数据 (b) MPU 方法重建结果 (c) Squirrel 无法向点云数据 (d) 本文方法重建结果

Fig.10 Reconstruction result of uniform sampling point set with several holes

图 10 带有多个复杂漏洞的均匀采样的点云数据的重建结果

© 中国科学院软件研究所 http://www.jos.org.cn

图 11 给出了带有洞的非均匀采样的点云数据的重建结果.图 11 中的 Igea 模型颈部带有漏洞,且点云非均匀采样.MPU 方法和本文方法均可以对输入数据进行有效重建.相比于 MPU 方法的重建结果(如图 11(b)、图 11(c)所示),本文方法可生成非均匀采样网格模型(如图 11(e)、图 11(f)所示).

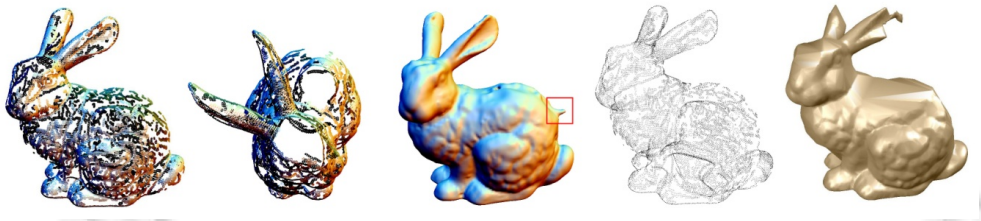


(a) Cyberware Igea 带法向点云数据 (b) MPU 方法重建结果线框图 (c) MPU 重建结果光照图 (d) Cyberware Igea 无法向点云数据 (e) 本文方法重建结果线框图 (f) 本文方法重建结果光照图

Fig.11 Reconstruction result of non-uniform sampling point set with holes

图 11 带有洞的非均匀采样的点云数据的重建结果

但当输入数据存在大量漏洞且采样点过于欠采样时,如图 12 所示,本文重建方法虽然可以保证法向的正确性,但受局部构造法的限制,在三角面扇构造过程中难以通过采样点生成满足局部最凸原则和法向连贯原则的三角面扇,重建结果难以逼近输入点云(如图 12(e)所示).MPU 方法则可通过全局隐式重建方法有效地处理此类数据,但是出格点却无法避免(如图 12(c)红色区域所示).



(a) Bunny 带法向点云数据 (b) Bunny 点云数据另外角度显示 (c) MPU 重建结果 (d) Bunny 无法向点云数据 (e) 本文方法重建结果

Fig.12 Reconstruction result of undersampled point set with holes

图 12 存在大量漏洞,且采样点过于欠采样的点云数据的重建结果

4.2 运行时间

由于本文方法是显式重建方法,运行时间和隐式重建方法 MPU 的重建无可比性.为验证本文多层同步表面重建方法的有效性,表 1 给出了一些点云数据采用本文方法和采用 LayerPeeling 层剥法的表面重构过程运行时间结果比较.层剥法首先通过手动补洞保证重建结果的正确,为了保证比较的公平性,如下算法均在单线程程序运行.由于本文算法点云只需计算一次 *kNN*,进行一次法向测试,并构建三角面扇,故算法具有线性时间复杂度.从表 1 中的实验数据容易发现,本文算法的运行时间和输入点集数目成正比.而 Layerpeeling 方法受点云层次制约,位于内层的点云需要等待外层点云均重建完毕后才能进行重建,故点云层次越多,相对越耗时.

Table 1 Performance comparisons of our method and the other methods

表 1 算法性能比较

Mesh	Points number	Number of layers	Runtime of other methods (s)[2]	Runtime of our method (s)
Face	6 737	9	14.5	0.9
Squirrel3	40 627	32	37.5	6.8
Bunny	56 120	150	87.1	9.2
Cyberware Igea	72 545	7	64.6	11.6
Squirrel6	76 268	17	100.0	13.2

© 中国科学院软件研究所 http://www.jos.org.cn

5 结 论

在获取点云数据时,点云的法向数据有时无法获取,获取的数据有时带有洞或数据分布不均匀,针对这些信息缺失的数据,本文基于空间有向状态八叉树提出高效的基于格内法向映射的多层同步表面重建方法.针对带有洞的点云数据,本文通过动态等高线,可有效检测出含有洞的八叉树节点,并针对空间八叉树进行补洞.针对点云法向信息缺失的问题,本文通过 HPR 映射计算出八叉树节点每个顶点的内外状态,带有顶点内外标识的有向状态八叉树可保证基于k近邻的表面重建过程的表面法向的正确性.通过使用基于状态八叉树空间数据结构,利用格内法向映射保证局部拓扑及几何结构的正确性,打破了层剥法的局限,可以支持不同层次的点云并行重建.实验结果对不同的层面上的点云数据进行同步表面重建,生成了较高质量的重建结果.下一步我们将研究在洞边界处进行处理,恢复洞内网格模型,延伸本文的工作.

References:

[1] Boissonnat JD. Geometric structures for three-dimensional shape representation. ACM Trans. on Graphics, 1984,3(4):266– 286. [doi: 10.1145/357346.357349]

[2] Lim SP, Haron H. Surface reconstruction techniques: A review. In: Artificial Intelligence Review. 2012. 1–20. [doi: 10.1007/s10462-012-9329-z]

[3] Lim CW, Tan TS. Surface reconstruction by layer peeling. Journal of the Visual Computer: Int'l Journal of Computer Graphics, 2006,22(9):593–603. [doi: 10.1007/s00371-006-0048-9]

[4] Tamal KD, Samrat G. Tight Cocone: A water-tight surface reconstructor. In: Proc. of the 8th ACM Symp. on Solid Modeling and Applications. Washington: ACM, 2003. 127–134. [doi: 10.1145/781606.781627]

[5] Ni TG, Ma ZH. A fast surface reconstruction algorithm for 3D unorganized points. In: Proc. of the 2nd Int'l Conf. on Computer Engineering and Technology. Chengdu, 2010. 7:15–18. [doi: 10.1109/ICCET.2010.5485908]

[6] Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC. Reconstruction and representation of 3D objects with radial basis functions. In: Proc. of the 28th Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH 2001). New York: ACM, 2001. 67–76. [doi: 10.1145/383259.383266]

[7] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In: Polthier K, Sheffer A, eds. Proc. of the 4th Eurographics Symp. on Geometry Processing (SGP 2006). Switzerland: Eurographics Association, 2006. 61–70.

[8] Kazhdan M, Hoppe H. Screened poisson surface reconstruction. ACM Trans. on Graphics (TOG), 2013,32(3):Article 29:13. [doi: 10.1145/2487228.2487237]

[9] Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel HP. Multi-Level partition of unity implicits. In: Fujii J, ed. Proc. of the ACM SIGGRAPH 2005 Courses (SIGGRAPH 2005). New York: ACM, 2005. Article 173. [doi: 10.1145/1198555.1198649]

[10] Yoshihara H, Yoshii T, Shibutani T, Maekawa T. Topologically robust B-spline surface reconstruction from point clouds using level set methods and iterative geometric fitting algorithms. Computer Aided Geometric Design, 2012,29(7):422–434. [doi: 10.1016/j.cagd.2012.03.007]

[11] Lin HW, Tai CL, Wang GJ. A mesh reconstruction algorithm driven by an intrinsic property of a point cloud. Computer-Aided Design, 2004,36(1):1–9. [doi: 10.1016/S0010-4485(03)00064-2]

[12] Giraudot S, Cohen-Steiner D, Alliez P. Noise-Adaptive shape reconstruction from raw point sets. Computer Graphics Forum, 2013,32(5):229–238. [doi: 10.1111/cgf.12189]

[13] Chen YL, Chen BY, Lai SH, Nishita T. Binary orientation trees for volume and surface reconstruction from unoriented point clouds. Computer Gaphics Forum, 2010,29(7):2011–2019. [doi: 10.1111/j.1467-8659.2010.01787.x]

[14] Xie H, McDonnell KT, Qin H. Surface reconstruction of noisy and defective data sets. In: Proc. of the Conf. on Visualization 2004. IEEE Computer Society, 2004. 259–266. [doi: 10.1109/VISUAL.2004.101]

[15] Zhao W, Gao S, Lin H. A robust hole-filling algorithm for triangular mesh. The Visual Computer, 2007,23(12):987–997. [doi: 10.1007/s00371-007-0167-y]

[16] Dey TK, Giesen J, Hudson J. Delaunay-Based shape reconstruction from large data. In: Proc. of the IEEE Symp. on Parallel and Large-Data Visualization and Graphics. 2001. 19–27. [doi: 10.1109/PVGS.2001.964399]

© 中国科学院软件研究所 http://www.jos.org.cn

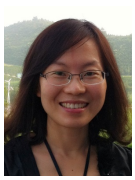
[17] Allègre R, Chaine R, Akkouche S. A streaming algorithm for surface reconstruction. In: Proc. of the Eurographics/ACM SIGGRAPH Symp. on Geometry Processing. 2007. 79–88. [doi: 10.2312/SGP/SGP07/079-088]

[18] Bolitho M, Kazhdan M, Burns R, Hoppe H. Multilevel streaming for out-of-core surface reconstruction. In: Belyaev A, Garland M, eds. Proc. of the 5th Eurographics Symp. on Geometry Processing (SGP 2007). Switzerland: Eurographics Association, 2007. 69–78.

[19] Manson J, Petrova G, Schaefer S. Streaming surface reconstruction using wavelets. In: Proc. of the Eurographics Symp. on Geometry Processing, 2008,27(5):1411–1420. [doi: 10.1111/j.1467-8659.2008.01281.x]

[20] Katz S, Tal A, Basri R. Direct visibility of point sets. ACM Trans. on Graphics (TOG), 2007,26(3):Article 24. [doi: 10.1145/1275808.1276407]

王筱婷(1976－),女,山东济南人,博士,工程师,主要研究领域为计算机图形学,计算机动画.



王璐(1981－),女,博士,副教授,CCF 会员,主要研究领域为计算机图形学,人机交互与虚拟现实.

孟祥旭(1962－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为人机交互与虚拟现实,制造业信息化.

www.jos.org.cn
www.jos.org.cn