DOI:10.3969/j.issn.1001 -4551.2016.04.007

基于自适应八叉树的三维点云快速拾取方法研究*

郑德炯,卢科青*
(杭州电子科技大学 机械工程学院,浙江 杭州 310018)

摘要:针对逆向工程中大规模点云数据快速拾取问题,对当前三维图形拾取基本方法进行了研究,对点云拾取的基本流程和点云快速拾取的关键问题进行了分析,提出了一种基于自适应八叉树的三维点云快速拾取方法。当用户在计算机屏幕上给出拾取多边形后,首先基于点云分布密度,对点云数据进行了自适应八叉树划分;然后对八叉树节点进行了投影,在屏幕上形成了八叉树节点的投影多边形,并对拾取多边形建立了矩形包围盒;接着对八叉树投影多边形和拾取多边形的矩形包围盒进行了相交检测,将不与矩形包围盒相交的八叉树节点包含的点云去除,从而缩小了点云拾取所需判断的范围,提升了拾取效率。最后对不同分布密度点云进行了定面积的拾取实验。实验结果表明,该点云拾取方法的点云分布密度越大,拾取时间相对越短,算法具有较高的拾取速度和准确度。

关键词:拾取;三维点云;八叉树;自适应
中图分类号:TH164;TP391　　**文献标志码**:A　　**文章编号**:1001 -4551(2016)04 -0417 -05

Quick picking method for 3D piont cloud based on adaptive octree

ZHENG De-jiong, LU Ke-qing
(School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: Aiming at the quick picking problem of massive point cloud data in reverse engineering, the basic picking method of 3D graphics currently was researched, the basic process of point cloud picking and the key problem of quick picking was analyzed, a quick picking algorithm for 3D point cloud based on adaptive octree was proposed. After the picking polygon was given by the user on screen, firstly, the adaptive octree division based on the distribution density of point cloud data was made, then the projection of the octree node was made and the octree projected polygon on the screen was formed, the rectangular bounding box of the picking polygon was established, then the intersection detection between octree projected polygon and rectangular bounding box of the picking polygon was executed, the point cloud of the octree node not intersected with the rectangular bounding box was removed, thereby the point cloud picking judgment was reduced and the picking efficiency was improved. Finally, a picking test under different point cloud distribution density was conducted. The results indicate that the greater the density of point cloud distribution, the picking time is relatively shorter, the algorithm has a high picking speed and accuracy.

Key words: picking; 3D point cloud; octree; adaptive

0 引 言

在三维交互式图形系统中,用户通过鼠标在系统中捕捉图形对象的操作叫拾取,以点云为捕捉对象的操作就称为点云拾取。逆向工程是利用先进测量技术对产品实物或模型进行测量,通过三维几何建模重构产品的三维模型,并在此基础上进行产品改造、创新设计、二次开发及生产的过程[1]。在逆向工程中大规模点云数据处理也都是以点云拾取操作为基础的。

目前,在计算机图形学领域,交互式拾取研究主要针对图形对象,针对三维点云拾取的研究较少。随着制造业产品生命周期的缩短、快速制造技术的发展,特

收稿日期:2015 -12 -07
基金项目:国家自然科学基金(青年)资助项目(51105332);浙江省科技计划资助项目(2014C31096)
作者简介:郑德炯(1991 -),男,浙江金华人,硕士研究生,主要从事逆向工程 CAD 建模方面的研究. E-mail:15067138810@163.com
通信联系人:卢科青,男,博士后,讲师. E-mail:lkq@hdu.edu.cn

别是逆向工程技术在大尺寸零件和复杂曲面反求测量方面的应用与发展,在产品设计领域大量点云数据的处理将会越来越普遍。逆向工程 CAD 建模的前期工作都是以点云为基础的数据预处理,因此研究快速有效的三维点云拾取方法具有十分重要的意义。

目前的三维图形拾取方法基本可分为两类:场景几何相关的拾取方法和场景几何无关的拾取方法。目前应用较多的是场景几何相关的拾取方法,基本思想是拾取点在三维空间作一条射线,然后用该射线与场景内的图元求交。针对场景几何相关的拾取方法,浙江大学王剑等[2]提出了一种向量法判断线段和包围盒相对位置的算法,并给出了实体和基本图元的拾取算法。中国测绘科学研究院徐胜攀等[3]针对 LiDAR(激光雷达)获得的大量点云处理,通过层次包围盒引入四叉树,提出了基于四叉树的三维点云快速拾取方法。场景几何无关的拾取方法通常是先对场景作特殊渲染,将物体信息按颜色编码并渲染,通过读取纹理屏幕坐标对应的颜色编码即可得到对应物体的信息[4-5]。张嘉华等[6]通过将坐标信息和对象面片指针绘制到一张 Render Target 型浮点纹理实现三维对象拾取。付昕乐等[7]利用 GPU 的并行计算能力,通过计算着色器实现点的空间变换和距离比较,提高拾取速度,特别对于点个数超过 4×10^6 点云,具有较大速度优势。点云作为一种没有大小的特殊图元,决定了场景几何无关的拾取方法不适用于点云拾取,而直接的场景几何相关的拾取方法在三维空间中建立射线求交,对大量点云来说计算量将非常大,从而会影响拾取速度。

针对现有点云拾取方法中存在的问题与不足,本研究提出一种基于自适应八叉树的三维点云快速拾取方法。笔者通过对点云的自适应八叉树划分,建立八叉树屏幕投影多边形,并且对用户拾取多边形建立矩形包围盒,通过八叉树投影多边形与矩形包围盒求交,判断点云是否被拾取,从而避免逐点判断,简化点云拾取计算过程。

1 三维点云快速拾取

1.1 点云拾取流程分析

逆向工程表面数据获取环节中得到的是产品表面的测点数据,大量测点数据的集合称为“点云”,要对点云数据进行操作必须先经过一系列的图形学处理,才能在计算机界面上显示点云数据。本研究以 Visual C ++ 为开发平台,利用 OpenGL 图形接口,自主开发了大规模点云显示软件,该软件可以有效地显示点云数据。

(1)三维显示

要在计算机屏幕上显示点云数据,就要将点云数据的坐标变换到屏幕坐标系。坐标变换常见的是下面几种形式:平移、旋转、缩放。在 OpenGL 图形接口中可以调用函数 glTranslatef( ),glRotatef( ),glScalef( )来实现平移,旋转和缩放。然后经过投影变换,调用 OpenGL 函数库中 glOrtho( )实现将坐标变换后的点云数据投影到计算机屏幕上。再调用 OpenGL 函数库中的 glViewport( )函数实现视口变换,使图形显示在计算机屏幕上指定窗口内。

(2)点云拾取

点云拾取的前端表现为:三维点云数据显示在屏幕上,用户在屏幕上用鼠标以交互形式建立一个拾取多边形,在多边形内的点即为拾取点,在多边形外的点为非拾取点。后台图形学运算的基本原理为:在屏幕坐标系下进行显示点是否在用户所确定的在多边形内的判断运算,若点在多边形内则该点被拾取,否则未被拾取。

1.2 点云快速拾取的关键问题

点云拾取问题的本质是点在多边形内外的判断。判断点在多边形内外的基本方法是射线法,假设从 P 点作一条水平向左的射线,若该射线与多边形的交点为奇数,则 P 在多边形内;若交点个数为偶数(包括 0 个点的情况),则 P 在多边形外部。射线法原理简单,但存在大量求交运算,并且包含一些临界情况[8],包括射线恰好通过多边形的顶点,点在多边形的边上,射线刚好与多边形的某条边重合等。此外判断点在多边形内外的方法还有弧长法/旋转角度法、叉乘判别法、面积判别法、有向回路法、网格法等。这些方法在一定程度上能简化点在多边形内外的判断过程。

在逆向工程 CAD 建模数据预处理过程中,系统一般面对的都是大规模的点云数据,如果进行点云拾取判断时对所有点进行点在多边形内外的检测,可能会产生很大的系统运算负荷,从而影响点云拾取速度。提高拾取速度的关键是在多点拾取过程中减少点与拾取多边形相交检测运算的次数,即减少需要与拾取多边形进行相交检测的点云数量。

2 基于自适应八叉树的处理技术

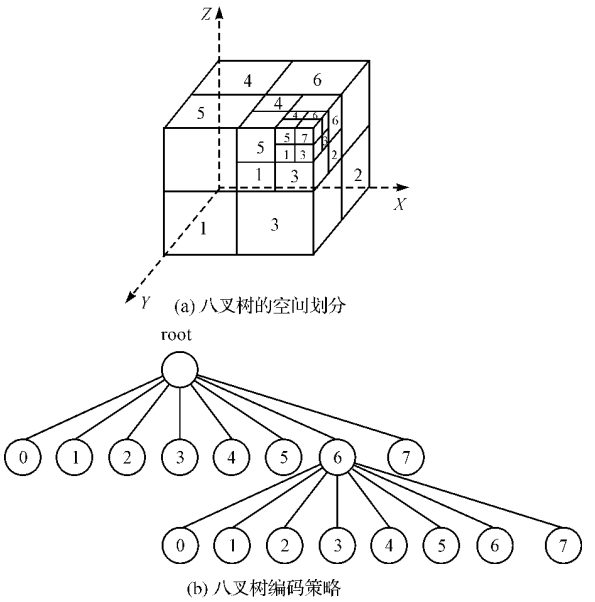
2.1 八叉树的引入

未经处理的离散点云只包含了点的坐标信息,点与点之间独立分布,为了方便后续的拾取运算,先要对点云建立一定的拓扑关系。目前常用的点云模型结构包括八叉树、K-D 树、R 树等。K-D 树和 R 树的缺点是结构复杂、构建速度较慢。相对而言八叉树的构造和集合运算比较容易,它具有显式的层次信息,且其各节点分布均匀。研究者通过八叉树划分,将点云分配到

各个相应的子节点中,通过搜索路径可实现快速地查询,所以八叉树结构适合大量点云数据处理。

八叉树的建立过程为:首先计算出包含所有点云的立方体包围盒作为八叉树的根节点,然后分别平行于 X 轴、Y 轴、Z 轴 3 方向对该立方体进行二等分割,获得对应根节点的 8 个子立方体(即子节点);接着对每个子节点依照同样的规则进行递归式分割,当子立方体边长小于某个预设值时,该节点不再进行分割。

八叉树的空间划分与编码策略如图 1 所示。



(a) 八叉树的空间划分　(b) 八叉树编码策略

图 1　八叉树空间划分及编码策略

2.2　八叉树的自适应划分

目前文献中提出的八叉树空间划分法一般是以边长为阈值,以最终划分的边长小于或等于预设值作为划分的终止条件[9]。均匀八叉树划分时,分割次数由递归深度决定,往往会有不必要的空间划分,因为有些工件的几何特征空间分布不均匀,从而点云空间分布也不均匀,可能对某个空子域进行不必要的分割,造成空间和时间的浪费。因此研究者应当对八叉树进行自适应划分,以减少内存损耗,增加计算速度。

八叉树自适应划分过程如下:

遍历点云数据,首先搜索出该点集中 X,Y,Z 坐标值最大和最小的点,设这些点坐标值分别为 $P_{X\max}$、$P_{X\min}$、$P_{Y\max}$、$P_{Y\min}$、$P_{Z\max}$、$P_{Z\min}$;然后根据这些点构造初始的包含所有点云数据的立方体包围盒,并将其作为八叉树分割的根节点;然后进行八叉树划分,分割规则如下[10]:如果某子节点内点云数目(包含在节点立方体上的点)大于自适应参数 K,则对该节点进一步划分,若该节点内点云数目小于或等于自适应参数 K,则不再对该节点作进一步划分。

自适应参数 K 根据点云密度选取不同的值,设 K 为节点内点数目的平均分布值,N 为点云总点数。自适应八叉树划分的时间复杂度约 $O(N^2/K)$,分割后拾取判断时间复杂度约为 $O(KN)$。为使整体消耗时间最小,定义时间函数为:

$$f(x)=N^2/K+K\cdot N \tag{1}$$

对时间函数求极值可以得到 $K=\sqrt{N}$。考虑到点云分布不均匀,在试验中加入影响系数。经试验影响系数取 0.5 时效率最高。节点划分终止条件为:点云数目小于或等于 K(K 值—$\sqrt{N}/2$,N—当前节点的父节点包含点云数目)。

2.3　高效多边形处理策略

因为点云数据具有大量性,本研究所建立的八叉树的节点数量还是相对较大,在与屏幕拾取多边形的相交检测判断过程中,往往由于屏幕拾取多边形边数较多导致相交检测的计算较多。为减少运算量,本研究首先对屏幕拾取多边形建立矩形包围盒,拾取时先将由节点表示的立方体投影到屏幕上,然后对节点投影多边形与矩形包围盒进行相交检测,若某子立方体的节点投影不与矩形包围盒相交,则表示该节点内的点云均不被拾取;若子立方体的节点投影多边形在矩形包围盒内或与矩形包围盒相交,则对该节点点云进行下一步判断。

屏幕拾取多边形的矩形包围盒的建立过程为:以屏幕拾取多边形顶点坐标最大值与最小值为顶点,边长分别平行于屏幕坐标系坐标轴而建立。

矩形包围盒的建立过程如图 2 所示。

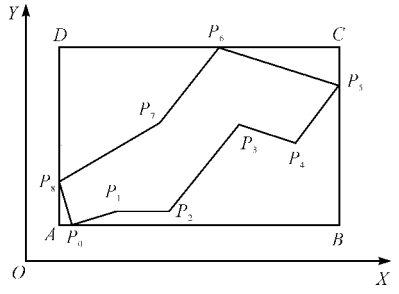


图 2　矩形包围盒的建立

图 2 中,多边形 $P_0P_1P_2P_3P_4P_5P_6P_7P_8P_9$ 的矩形包围盒为 $ABCD$。

3　基于自适应八叉树的点云拾取方法

算法的基本思路是:首先对八叉树节点与屏幕拾取多边形进行相交检测,搜索出与多边形相交的八叉树节点包含的点云数据块;然后再对节点执行多点拾取基本算法,即逐一判断点是否在拾取多边形内,若在拾取多边形内,则该点被拾取,反之未拾取。

八叉树节点与屏幕拾取多边形的相交检测过程为：

（1）先将八叉树节点的包围盒投影到屏幕形成一个凸多边形 Pi，然后对该多边形与用户鼠标所确定多边形的矩形包围盒 B 进行相交判断；

（2）若 Pi 在 B 内，则直接进行第 5 步判断；

（3）若 Pi 在 B 外，则该节点所包含的点云都未被拾取，不需要进行下一步点在多边形内外的判断；

（4）若 Pi 在 B 有相交（包括 Pi 有顶点在 B 上），则将该节点包含的点云加入查找集合，进行下一步判断；

（5）当前节点内包含的点云逐一判断是否在拾取多边形矩形包围盒内，若不在则表示未被拾取，若点在矩形包围盒内，则进行下一步判断；

（6）对点与拾取多边形进行相交检测运算，利用射线法判断点是否在拾取多边形内，若在拾取多边形内，则表示该点被拾取；反之未被拾取。

算法流程图如图 3 所示（算法中初始节点为八叉树根节点）。

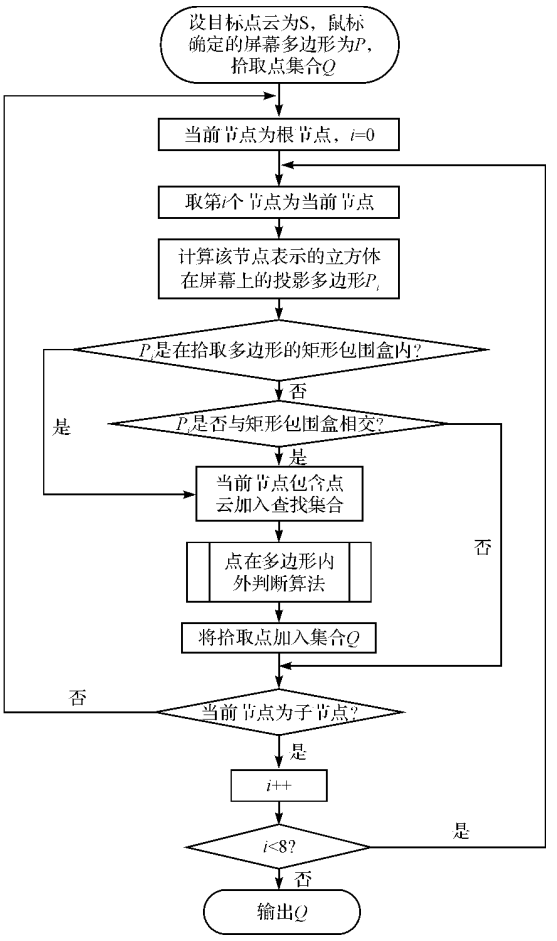


图 3 基于自适应八叉树的拾取算法

4 实验与评价

为验证基于自适应八叉树的点云拾取算法的有效性和准确性，本研究以大小为 25.5 M（330 945 个点）的格式为 IGS 的艺术头像三维扫描点云数据为对象，设计一组实验分别对点云数据密集、较密和稀疏的情况进行拾取试验。

点云分布状态如图 4 所示。

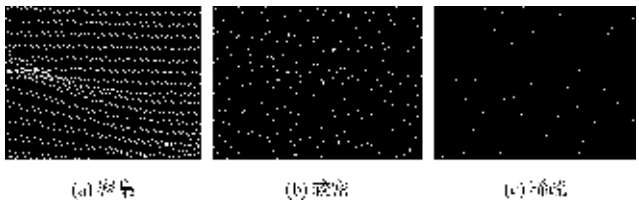


图 4 点云分布状态

分别采用不同大小的拾取多边形（随机）进行拾取操作，统计 20 次拾取时间并取平均值。

拾取时间统计表如表 1 所示。

表 1 拾取时间统计表

点云分布情况	拾取时间/ms		
	S_1 = 657.5 mm^2	S_2 = 1 352.6 mm^2	S_3 = 3 511.2 mm^2
点云分布密集	57.3	93.5	112.1
点云分布较密	109.6	156.3	273.8
点云分布稀疏	765.7	1 351.2	1 637.4

由表 1 可见，该算法能有效完成不同密度点云的多点拾取，在点云分布密集和较密集时，对于面积分别为 657.5 mm^2，1 352.6 mm^2，3 511.2 mm^2 的任意多边形都能进行较为快速的拾取；当点云分布稀疏时拾取速度有所下降。总体上点云分布越密，拾取时间相对越短。从拾取效果上看，该算法能准确完成多点拾取，有较高的拾取准确度。

多点拾取效果图如图 5 所示。

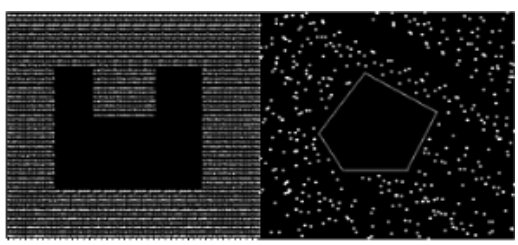


图 5 拾取效果图

（下转第 425 页）

本文引用格式：

郑德炯，卢科青. 基于自适应八叉树的三维点云快速拾取方法研究[J]. 机电工程，2016，33(4)：417－420，425.

ZHENG De-jiong, LU Ke-qing. Quick picking method for 3D piont cloud based on adaptive octree[J]. Journal of Mechanical & Electrical Engineering, 2016, 33(4):417－420,425.

《机电工程》杂志：http://www.meem.com.cn