

Catmull-Clark 细分网格数据点拾取

张湘玉*, 马希青

(河北工程大学 机电学院, 河北 邯郸 056038)

(* 通信作者电子邮箱 xiangyu1977@163.com)

摘要:针对将 OpenGL 选择拾取机制直接作用于 Catmull-Clark 细分网格数据点的拾取,可能会因细分网格数据量过大而导致名字堆栈溢出的问题,借鉴细分曲面求交的思想,提出一种新的细分网格数据点拾取方法。该方法通过提取拾取对象的邻域网格并进行局部细分,将对细分任意层次上网格数据点的拾取转化为对初始控制网格以及在达到细分层次要求以前每一次局部细分网格点、边、面的拾取和对最后一次局部细分网格数据点的拾取。采用多个拾取算例进行对比分析实验,当细分网格顶点数量较多时,所给拾取方法的拾取命名对象总量和拾取时间都远小于传统 OpenGL 选择拾取方法。实验结果表明,所给拾取方法能快速准确实现细分网格数据点的拾取,尤其适用于数据量较大的复杂细分模型,可有效避免因拾取名字堆栈溢出而导致的拾取错误。

关键词:拾取;细分网格;Catmull-Clark 细分;OpenGL;邻域网格;局部细分

中图分类号: TP391.72 **文献标志码:** A

Pickup algorithm for data points of Catmull-Clark subdivision mesh

ZHANG Xiangyu*, MA Xiqing

(College of Mechanical and Electrical Engineering, Hebei University of Engineering, Handan Hebei 056038, China)

Abstract: Focused on the issue that adopting the OpenGL selection mechanism to directly act on the data points of Catmull-Clark subdivision mesh may cause the name stack overflowing because of the too large data volume, referencing the intersection theory of subdivision surfaces, a new pickup method for subdivision models was proposed. Through extracting and subdividing the neighboring mesh of pickup objects, the method converted the pickup for data points of subdivision mesh into the pickup for points, edges and faces of the initial mesh and the local subdivision meshes at successive levels, and the pickup for points of the local subdivision mesh at last level. Comparison and analysis experiments of some pickup examples were conducted. The total number of naming objects and the pickup time consumption of the given method were both far less than those of the traditional OpenGL selection method when the subdivision mesh had plenty of data points. The experimental results show that the proposed method can quickly and accurately pick up the data points of subdivision mesh, and it is especially suitable for the complex subdivision models with a large number of data points, and can effectively avoid pickup errors caused by the name stack overflowing.

Key words: pickup; subdivision mesh; Catmull-Clark subdivision; OpenGL; neighboring mesh; local subdivision

0 引言

OpenGL 是目前用于开发 3D 图形应用程序的首选环境,也是现在最广泛采用的计算机图形标准,已经在 CAD/CAM、计算机仿真、虚拟现实等领域得到广泛应用。在 OpenGL 实际使用中,不仅要求能够在屏幕上绘制出 3D 图形,更多情况下还需要对这些图形进行交互编辑操作,而图形拾取是实现图形交互的基础^[1]。

伴随细分曲面造型技术的不断发展成熟,其在曲面建模方面的应用越来越广^[2-3]。在现有细分方法中,Catmull-Clark (以下简称 C-C)细分曲面,作为三次 B 样条曲面向任意拓扑网格的推广,在复杂形体的表示方面具有更强大的功能。有效实现对 C-C 细分网格数据点的精确拾取对于进行细分曲面分析、曲面编辑等都尤为重要。

针对物体拾取,目前主要有以下几种方法^[4-9]: OpenGL

选择拾取机制、射线拾取法、基于视口空间的拾取方法、对象缓冲区拾取方法以及 depth ray 方法等。其中,OpenGL 选择拾取机制需要在选择模式和正常模式之间进行转换,对于数据量较大的物体,会因为名字堆栈的溢出而无法成功实现拾取;射线拾取法拾取算法较为复杂,尤其是对于比较复杂的物体,会给整个系统实现带来一定难度;基于视口空间的拾取方法要进行多次空间变换,过于复杂;对象缓冲区拾取方法需要在内存中创建对象缓冲区,对内存进行大量操作,效率较低;depth ray 方法操作复杂,更加适合用于虚拟现实环境中。

结合 C-C 细分规则,借鉴细分曲面求交思想^[10-11],本文给出了一种 C-C 细分网格数据点拾取方法,并已将其应用于自行开发的软件系统中,取得了较好的拾取效果。

1 C-C 细分曲面

细分曲面是采用一组顶点加密规则对给定的初始控制网

收稿日期:2014-12-09;修回日期:2015-02-04。 基金项目:河北省自然科学基金资助项目(E2010001010);校博士基金资助项目。

作者简介:张湘玉(1977-),女,河北辛集人,讲师,博士,主要研究方向:计算机辅助设计、计算机辅助制造;马希青(1963-),男,河北故城人,教授,主要研究方向:计算机仿真、虚拟设计。

格逐层加密顶点所得到的一系列网格序列的极限,实际中往往采用一定细分层次的控制网格来表示细分曲面。目前常用的几种细分方法主要有 C-C 细分、Loop 细分、Doo-Sabin 细分及蝶形细分,其细分规则各不相同。本文拾取方法的研究基于 C-C 细分展开,以下给出 C-C 细分的几何和拓扑规则。

细分几何规则规定了细分过程中所产生新点几何位置的确定方式。针对 C-C 细分的几何规则如图 1 所示,对应每个网格面产生一个新面点,每条网格边产生一个新边点,每个网格顶点对应产生一个新顶点。图 1 中数字为新点计算时原网格各点所占的权重。

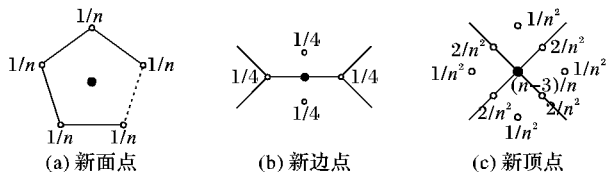


图1 C-C 细分几何规则

细分拓扑规则规定了细分过程中所产生的各新点之间的拓扑连接关系。如图 2 所示,C-C 细分的拓扑连接规则是:每个新面点与周围的新边点连接;每个新顶点与周围的新边点连接。

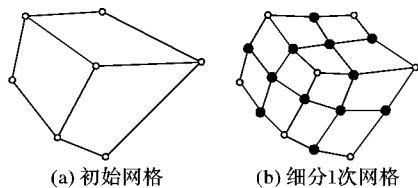


图2 C-C 细分拓扑规则

从 C-C 细分规则可以看出,对于初始网格细分 1 次后得到的控制网格,其顶点个数等于初始网格点、边、面的数量总和。那么,对于细分 3 次甚至更多次之后得到的控制网格,其顶点个数将远远大于初始网格点、边、面的数量之和,如果采用传统 OpenGL 选择拾取方法,则对于比较复杂、数据量较大的几何体,经过若干次细分后,分别再对每个细分控制网格顶点进行命名,及有可能会因为名字堆栈的溢出而无法成功实现拾取。

2 拾取方法

2.1 拾取中心思想

针对细分曲面的求交,文献[10-11]中将其转化为对细分初始控制网格的求交,然后提取初始相交面片的 1-邻域网格带,再对其进行细分操作,之后在此局部细分网格带上继续提取相交面片及其 1-邻域网格带,继续细分,这样通过不断地提取和细分直至达到曲面求交的精度。本文对于细分网格顶点数据的拾取是通过鼠标点击所要拾取的细分网格顶点交互实现的。过鼠标点击点做垂直于屏幕的射线,拾取过程可以看作为此射线与所要进行拾取的细分网格的求交过程,进一步也可看作为以此射线为轴线,以一无限小 R 值为半径的细分圆柱面与所要进行拾取的细分网格的求交过程,求解得到的交点或交线的极限值也即所要拾取的细分网格顶点。由此,借鉴细分曲面求交的思想,这里给出本文细分网格顶点拾取方法的主要中心思想:

对细分 1 次后网格数据点的拾取,可转化为对初始网格点、边、面的拾取以及对所拾取对象 1-环邻域网格的局部细分

网格顶点数据的拾取两次拾取过程。同样对于细分 n 次后的网格数据点的拾取,可以转化为对初始网格以及在达到细分次数要求以前每一次局部细分网格点、边、面的拾取和对最后一次局部细分网格顶点数据的拾取。

针对 1-环邻域网格的定义,如图 3 所示,点的 1-环邻域网格是指与此点相邻接的所有边和面所构成的局部网格;边的 1-环邻域网格指与此边两个点相邻接的所有边和面所构成的局部网格;面的 1-环邻域网格是指与此面所有顶点相邻接的所有边和面所构成的局部网格。

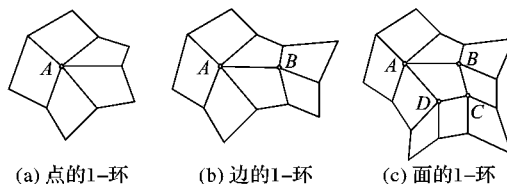


图3 1-环邻域网格

2.2 拾取分析

根据 C-C 细分的几何规则(如图 1 所示)可以看出,在一般情况下,一个网格面所产生的新面点在屏幕上的投影位于该网格面本身在屏幕上投影的内部;一条网格边产生的新边点投影可能落在与此边邻接的两个面投影内;而对应一个网格顶点产生的新顶点的投影位置则位于与此顶点相邻接的所有边、面所构成局部邻域网格的投影内部。根据这种投影关系,在对细分 1 次后控制网格顶点进行拾取时,若所要拾取的网格顶点在屏幕上的位置落在初始网格的某一点、边投影处,或某一面投影内,则提取这些点、边、面的 1-环邻域网格,进行 1 次细分,在得到的若干新点中必包含了所要拾取的网格顶点。因此,前述拾取方法在一般情况下能够实现对细分网格顶点数据的成功拾取。但在一些特殊情况下,当细分 1 次控制网格中所要拾取新点的屏幕投影位于初始网格相应点、边、面元素的 1-环邻域网格的屏幕投影之外,采用前述拾取方法就无法成功实现细分网格顶点的有效拾取。比如图 4 所示(图中粗线表示初始网格,细线表示细分 1 次后网格),初始网格面 $ABCD$ 存在 $> 180^\circ$ 的内角,是一个凹四边形,其新面点 F 的屏幕投影在三边面 MNQ 屏幕投影的内部,位于网格面 $ABCD$ 的 1-环邻域网格的屏幕投影之外,这时,因为提取三边面 MNQ 的 1-环邻域网格中不包含网格面 $ABCD$,自然在对所提取的局部网格进行细分操作后,得到的若干新点中也就不会包含所要拾取的网格面 $ABCD$ 的新面点 F ,故而会导致拾取失败。

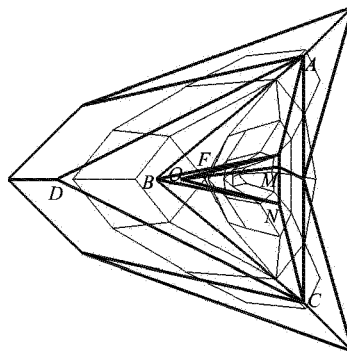


图4 拾取失败特殊情况

考虑拾取失败的原因,可以通过扩大初始网格拾取对象

对于局部细分网格,提取拾取对象的1-环邻域,考虑到局部细分网格点、边、面数量较少,搜索时间自然也较短,文中对其不再进行星形结构转换,因为转换过程也是搜索的过程,此转换存在于拾取过程中,其耗时比采取直接搜索时间还长。

下面对星形结构的建立进行说明。

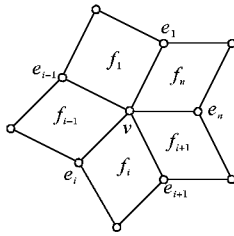


图7 与 v 邻接的边和面

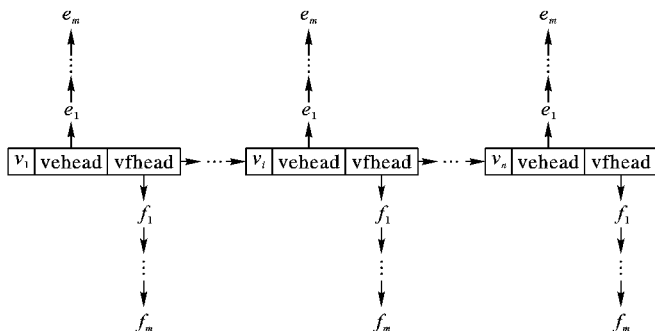


图8 星形结构链表

首先针对细分初始控制网格中每一个顶点,搜索与之相

邻接的网格边、面,并且按逆时针(或顺时针)方向进行排列。如图7所示,以顶点 v 为例,与其相邻接的边、面按逆时针排列为: $e_1, f_1, \dots, e_i, f_i, \dots, e_n, f_n$,其中 e_i 表示与顶点 v 相连接的边的另一顶点索引号, f_i 表示与顶点 v 相邻接的面的索引号。然后建立星形结构链表如图8所示,链表中的每个节点存储的实际就是相应顶点的1-环邻域网格信息。

针对细分初始控制网格,若拾取对象为顶点,则根据拾取前所建立的星形结构可以直接得到此顶点的1-环邻域网格;若拾取对象为边或面,可以直接提取边的两个顶点或面的所有顶点的1-环邻域网格,然后求并集得到拾取边、面的1-环邻域网格。针对2-环邻域网格的提取,可以由所提取到的1-环邻域网格,根据星形结构,继续提取1-环所有外围顶点的1-环邻域网格,然后求并集得到。依此类推可以提取到拾取对象任意 t -环邻域网格。

5 拾取算例对比分析

本文在 Borland C++ Builder 6.0 集成开发环境中实现了本文拾取算法。表1给出了4个拾取算例分别运用在不同拾取方案情况下,对同一细分网格数据点进行拾取,命名对象总个数以及拾取时间的对比数据。本文拾取方法作为方案I,将OpenGL选择拾取机制直接作用于细分网格数据点拾取定为方案II。表中时间为在一台配备有Intel酷睿2双核T5550 1.83 GHz CPU和1 GB内存的个人笔记本电脑上,实际运行获得的平均时间。

表1 拾取算例对比分析

模型	初始网格 顶点数	细分网格 顶点数	拾取命名对象总数		方案I中邻域网 格提取总时间/s	方案I中局部 细分总时间/s	拾取总时间/s	
			方案I	方案II			方案I	方案II
自行车坐垫	77	1090	540	1090	0.0041	0.012	0.024	0.014
飞机	72	6152	549	6152	0.0019	0.019	0.029	0.075
花瓶	289	18466	1349	18466	0.0023	0.025	0.046	0.191
马	187	71042	1428	71042	0.0029	0.028	0.051	0.527

方案I的拾取命名对象主要涉及了初始控制网格以及逐层拾取对象邻域网格的局部细分网格,这些网格数据量都相对较少,而方案II是将OpenGL选择机制直接作用于细分网格,对每个细分网格顶点都进行命名,因此方案I尤其对于细分次数多、数据量大的复杂细分网格模型,在缩减命名对象数量上具有更突出的优势,可有效避免因拾取名字堆栈溢出而导致的拾取错误。从表1可以看出,4个细分模型采用方案I的拾取命名对象总数均小于方案II,尤其细分网格顶点数最多的马模型,其采用方案I的拾取命名对象总数仅为方案II的2%。

在拾取时间方面,方案II只涉及OpenGL命名对象选择拾取时间,与命名对象的数量相关,数量越多,选择拾取的耗时也越多。而方案I的拾取总时间除包括OpenGL命名对象选择拾取时间,还包括了逐层拾取对象邻域网格的提取时间以及邻域网格的局部细分时间。经过前面的分析,尤其在细分网格数据量较大时,方案I的命名对象数量要远小于方案II,因此其OpenGL命名对象选择拾取耗时也将远小于方案II。另外,对于方案I中的邻域网格提取以及局部细分,其作用网格主要是初始网格、逐层局部细分网格以及逐层所提取拾取对象的邻域网格,这些网格的数据量都较小,且初始网格在拾取之前已转换为星形结构,因此针对这两部分的耗时方

案I也是较少的。从表1可以看出,飞机、花瓶及马模型采用方案I的拾取总时间都小于方案II,尤其是马模型,其采用方案I的拾取总时间相比方案II缩减了90%,但自行车坐垫模型因为细分网格顶点数较少,采用方案I,虽然OpenGL命名对象选择拾取时间要小于方案II,但加上邻域网格提取时间及局部细分时间,方案I的拾取总时间反而超过了方案II。从拾取时间的对比分析可以看出,针对数据量大的复杂细分网格模型,本文所给拾取方法在拾取速度上更具优势。

图9所示给出了四个细分模型的数据点拾取图例。可以看出,采用本文所给拾取方法,均没有出现拾取错误的情况,达到了对细分模型数据点的准确拾取。

6 结语

本文从C-C细分的几何和拓扑规则出发,借鉴细分曲面求交的思想,给出了一种C-C细分网格数据点拾取方法。相比将OpenGL选择拾取机制直接用于细分网格数据点的拾取,尤其对于数据量较大的复杂细分网格模型,此方法大大缩减了拾取时间和命名对象数量,避免了因名字堆栈溢出而导致的拾取错误。实际中已将该拾取方法应用于自行开发的软件系统中,证实了该方法的有效可行性。另外本文的拾取方法是基于C-C细分进行研究的,也可以将其推广到其他的细

分模式,为细分网格数据点的拾取提供一种有效方法。

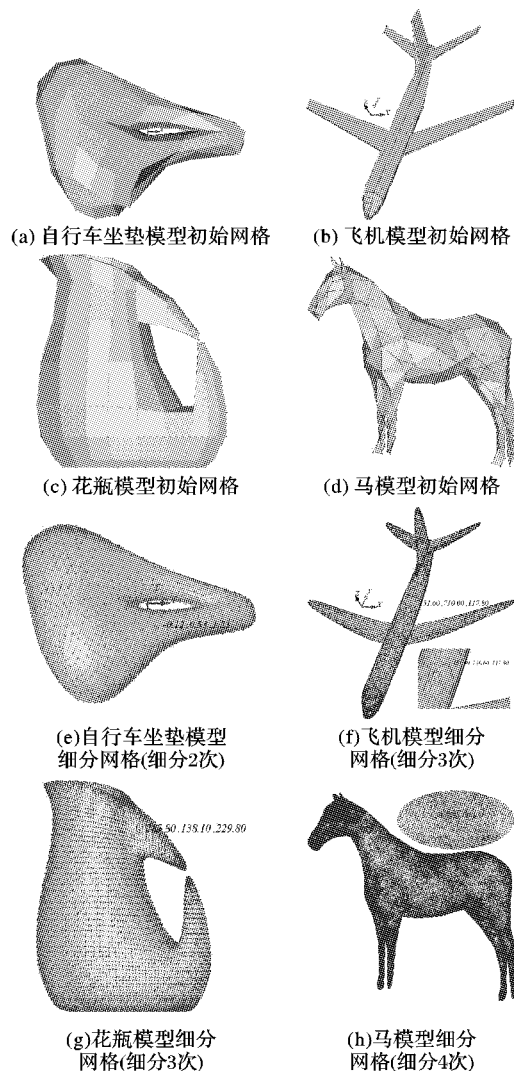


图9 细分模型数据点拾取图例

参考文献:

- [1] XU S, LIU Z, ZUO Z. Study on quick picking technology for massive 3D point cloud[J]. Computer Engineering and Design, 2013, 34(8): 2764-2768. (徐胜攀, 刘正军, 左志权. 大规模三维点云快速拾取技术研究[J]. 计算机工程与设计, 2013, 34(8): 2764-2768.)
- [2] MA W, WANG H. Interpolating an arbitrary number of joint B-spline curves by Loop surfaces[J]. Computers and Graphics, 2012, 36(5): 321-328.
- [3] ANTONELLI M, BECCARI C, CASCIOLA G, et al. Subdivision surfaces integrated in a CAD system[J]. Computer-Aided Design, 2013, 45(11): 1294-1305.
- [4] YAO J, LI X. Research on 3-dimension pick-up of human-computer interaction in computer graphics[J]. Journal of Engineering Design, 2006, 13(2): 116-120. (姚继权, 李晓豁. 计算机图形学人机交互中三维拾取方法的研究[J]. 工程设计学报, 2006, 13(2): 116-120.)
- [5] CHEN Y, YIN F. Research and implementation of a Web3D demonstration system based on the picking algorithm[J]. Applied Science and Technology, 2011, 38(12): 9-14. (陈煜, 殷凤华. 基于拾取算法的产品交互式 Web3D 模拟的研究与实现[J]. 应用科技, 2011, 38(12): 9-14.)
- [6] ZHANG J, LIANG C, LI G. 3 D primitive picking on GPU [J]. Journal of Engineering Graphics, 2009, 30(1): 46-52. (张嘉华, 梁成, 李桂清. GPU 三维图元拾取[J]. 工程图学学报, 2009, 30(1): 46-52.)
- [7] ZHU M, DONG B, WANG Y, et al. Algorithm for picking in 3D scenes based on viewport space[J]. Journal of Engineering Graphics, 2008, 29(2): 94-97. (朱明亮, 董冰, 王祎, 等. 三维场景中基于视口空间的拾取算法[J]. 工程图学学报, 2008, 29(2): 94-97.)
- [8] WANG X, TANG Z. Application of object buffer pick-up algorithm in 3D geology volume visualization[J]. Science Technology and Engineering, 2012, 12(20): 5066-5068. (王雪莉, 唐志辉. 基于对象缓冲区的拾取法在地质体可视化中的应用[J]. 科学技术与工程, 2012, 12(20): 5066-5068.)
- [9] VANACKEN L, GROSSMAN T, CONINX K. Multimodal selection techniques for dense and occluded 3D virtual environments[J]. International Journal of Human-Computer Studies, 2009, 67(3): 237-255.
- [10] ZHOU H. Research on modeling technology of subdivision surface [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2004: 48-51. (周海. 细分曲面造型技术研究[D]. 南京: 南京航空航天大学, 2004: 48-51.)
- [11] YUAN H, LIU H, LIAO W. Basic boolean operation research in Catmull-Clark subdivision surface[J]. Journal of Computer Research and Development, 2008, 45(5): 1259-1268. (袁鸿, 刘浩, 廖文和. 基于 Catmull-Clark 细分的曲面布尔运算基础研究[J]. 计算机研究与发展, 2008, 45(5): 1259-1268.)
- [12] LIU H. Fundamental technology research on subdivision surface modeling based on quad meshes [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2005: 24-33. (刘浩. 基于四边形网格的细分曲面造型基础技术研究[D]. 南京: 南京航空航天大学, 2005: 24-33.)
- [8] BARUAH S K, BONIFACI V, D'ANGELO G, et al. Scheduling real-time mixed-criticality jobs [J]. IEEE Transactions on Computers, 2012, 61(8): 1140-1152.
- [9] LU C. Feedback control real-time scheduling [D]. Charlottesville: University of Virginia, 2001.
- [10] JIN H, WANG H, FU Y, et al. A fuzzy feedback control real-time scheduling algorithm [J]. Journal of Software, 2004, 15(6): 791-798. (金宏, 王宏安, 傅勇, 等. 模糊反馈控制实时调度算法[J]. 软件学报, 2004, 15(6): 791-798.)
- [11] NIU Y, DAI G, MU D, et al. Design and implementation of self-adaptation scheduling algorithm for open real-time system [J]. Computer Science, 2008, 35(9): 59-61. (牛云, 戴冠中, 慕德俊, 等. 基于反馈控制的开放式实时系统自适应调度算法设计与实现[J]. 计算机科学, 2008, 35(9): 59-61.)
- [12] LU C, STANKOVIC J A, TAO G, et al. Feedback control real-time scheduling: framework, modeling, and algorithms [J]. Journal of Real-Time Systems, 2002, 23(1/2): 85-126.

(上接第1429页)