



(12) 发明专利

(10) 授权公告号 CN 109983509 B

(45) 授权公告日 2021.10.15

(21) 申请号 201780029391.1
 (22) 申请日 2017.02.03
 (65) 同一申请的已公布的文献号
 申请公布号 CN 109983509 A
 (43) 申请公布日 2019.07.05
 (30) 优先权数据
 15/207,927 2016.07.12 US
 (85) PCT国际申请进入国家阶段日
 2018.11.12
 (86) PCT国际申请的申请数据
 PCT/CA2017/050121 2017.02.03
 (87) PCT国际申请的公布数据
 W02018/010010 EN 2018.01.18
 (73) 专利权人 曹尚稳
 地址 加拿大蒙特利尔市
 (72) 发明人 曹尚稳

(74) 专利代理机构 北京冠和权律师事务所
 11399
 代理人 朱健 张国香
 (51) Int.Cl.
 G06T 17/10 (2006.01)
 G06T 17/20 (2006.01)
 G06F 9/30 (2006.01)
 G06T 15/08 (2011.01)
 G06F 30/10 (2020.01)
 (56) 对比文件
 US 5825369 A, 1998.10.20
 CN 102682476 A, 2012.09.19
 US 2011295564 A1, 2011.12.01
 Sâm Landier. Boolean operations on
 arbitrary polyhedral meshes.《24th
 International Meshing Roundtable
 (IMR24)》. 2015, 200-212.
 审查员 董洪梅

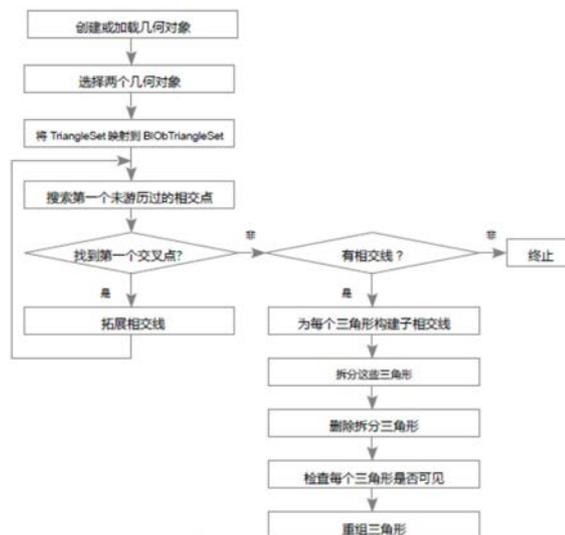
权利要求书4页 说明书9页 附图15页

(54) 发明名称

一种使用几何面的即时布尔运算方法

(57) 摘要

本发明提供了一种使用几何对象的及时布尔运算方法和系统,执行布尔运算,用基本的几何形体和扩展几何对象创建几何模型,包括:将几何对象的渲染绘图面片映射到含有相邻三角形的扩展三角形,计算面片间的相交线,拆分相交线通过的面片,确定每个面片是否可见,并重新组合面片以形成一个或多个几何对象。该方法不利用计算机辅助设计/计算机图形/实体建模系统中最流行的数据结构CSG和B-REP,但具有两者的优点:易于实现和灵活。此外,它是实体建模和曲面建模系统的一个统一方法,能够生成多变和精细的产品模型。



1. 一种执行即时布尔运算的方法,在计算机系统上实现并使用计算机进行操作,用基本几何对象的几何面创建几何产品模型,所述方法包括:

将几何对象的渲染绘图面映射到包含相邻三角形的扩展三角形,记每个几何对象产生的扩展三角形集为一个B1OpTriangleSet;

构造相交线,搜索第一对相交的三角形,它通过检测两个轴对齐的最小边界框是否重叠并执行边与三角形相交计算以定位相交点,然后跟踪包含最后一个相交点的一对三角形的相邻三角形以延长相交线,直到相交线的第一个相交点与最后一个相交点相同,确保相交线闭合或直到遍历完所有三角形;

用修改的Watson方法构建Delaunay二维网格拆分相交线通过的每个三角形,它构建一个轮廓点序列链接子相交线,在插入点时,检查网格中的每个三角形的外接圆是否包含该点或最后一个线段穿过网格中的该三角形,决定是否要进一步分割网格中的此三角形;

检查每个三角形是否可见;

分步骤重新组合面片,包括复制三角形,删除三角形,反转几何对象的每个三角形的法线,和合并保留的三角形以形成一个或多个新的扩展三角形集;以及将扩展三角形映射到渲染绘图面。

2. 根据权利要求1所述的方法,其中任何即时布尔运算,包括组合、相交、排除、差和分割,使用几何对象的渲染绘图面创建新几何对象,将几何对象的渲染绘图面映射到扩展三角形,构建相交线,拆分相交线通过的每个三角形,检查每个三角形是否可见,重新组合面片以形成新的扩展三角形集,和将扩展三角形映射到渲染绘图面而无需使用数据结构CSG和B-REP。

3. 根据权利要求1所述的方法,其中任何即时布尔运算,包括组合、相交、排除、差和分割,使用几何对象的渲染面创建新几何对象,将几何对象的渲染绘图面映射到扩展三角形,构建相交线,拆分相交线通过的每个三角形,检查每个三角形是否可见,重新组合面片以形成新的扩展三角形集,和将扩展三角形映射到渲染面,在造型系统中使用两类数据结构CSG和B-REP。

4. 根据权利要求1所述的方法,其中构造相交线这一过程重复多次,使用轴对齐最小边界框并展开边与三角形相交计算,确保相交点准确,相交线不是近似曲线,它包括:

建立公式 $P = P_i + t * (P_{(i+1) \% 3} - P_i)$,用于表示三角形 T_a 的第 i 条边, T_a 为三角形对中的一个三角形, P_i 和 $P_{(i+1) \% 3}$ 为三角形 T_a 的两顶点, t 是参数,为任意实数;

建立公式 $ax + by + cz + d = 0$,用于记录三角形对中 T_b 定义的平面;

和求两个线性公式的解。

5. 根据权利要求1所述的方法,其中搜索第一对相交三角形和跟踪相邻三角形,计算边与三角形相交点,通过验证一个点是否在三角形的边上,替换直接计算边与边相交。

6. 根据权利要求1所述的方法,其中拆分每个三角形用修改的Watson方法构建Delaunay二维网格,该方法定义一个三角形集、删除的三角形集和多边形,即使子相交线不是凸的,也能将三角形划分为不同的小三角形,包括:

构建一个轮廓点序列链接子相交线和三角形顶点;

根据三角形的法线将三维点序列映射到二维产生二维序列点;

添加四个点以形成一个大一些的边界框包围所有二维点;

假设边界框的一条对角线将框拆分为两个三角形,并将它们添加到三角形集中;

将除边界点以外的所有点插入三角形集中进一步拆分此网格中的三角形:对于每个点,检查三角形集中的每个三角形是否它的外接圆包含该点或最后一个线段穿过该三角形,当条件满足时,将其从三角形集中删除并添加到删除的三角形集中,使用删除的三角形集扩展多边形并立即清空删除的三角形集,使用多边形生成三角形并将其添加到三角形集;

从三角形集中删除边界三角形;

和将三角形集映射为三维三角形并将其添加到B1OpTriangleSet中。

7. 根据权利要求1所述的方法,其中检查每个三角形是否可见进一步包括:

计算属于几何对象A的三角形Ta的质心C;

构建线L, $P=C+t*N$ 穿过质心C并沿着Ta的法线N;

对于对象B的每个三角形Tb,检查线L与三角形是否交在三角形Tb内部,根据两点间的距离计算深度t并将t添加到深度缓冲区,如果缓冲区中负t的数目等于正t的数目,将Ta设置为不可见。

8. 根据权利要求1所述的方法,在曲面裁剪时检查三角形是否为不可见的步骤包括:

将所述曲面片的B1OpTriangleSet的一般点的位置序列号设为0,将相交线上的点的位置序列号按升序或降序设定值,这取决于相交线与裁剪轮廓线方向是否相同;

对于每个裁剪轮廓边界线上的三角形,根据其顶点的位置序列号的值确定它位于裁剪轮廓线的左侧或右侧,并将其为非边界的相邻三角形设置为左侧或右侧。

9. 根据权利要求1所述的方法,其中任何即时布尔运算,即组合、相交、排除、差和分割,使用一个或多个步骤重组构建操作结果:删除几何对象的不可见或可见三角形,将对象的不可见三角形复制到缓冲区或将三角形从缓冲区复制到对象,反转几何对象的每个三角形的法线,和合并对象的三角形以形成新的扩展三角形集。

10. 根据权利要求1所述的方法,其中将扩展三角形直接映射到渲染面以便显示并向下一个布尔运算提供数据。

11. 一种由硬件和软件组成的计算机系统,使用几何对象的渲染绘图面执行即时布尔运算,所述系统包括:

具有用于输入数据和命令的输入设备的计算机,显示用户界面、几何对象和附加数据的显示设备,具有存储构成软件系统的几何数据和指令的介质,或具有部分或全部嵌入指令的微芯片或集成电路,和执行以下步骤的处理器:

创建、修改或加载几何对象,包括旋转和拉伸的几何对象,并用计算机的输入设备将它们重新定位在适当的位置或方向上;

选择两个几何对象;

将几何对象渲染绘图面映射到包含相邻三角形的扩展三角形,记每个几何对象产生的扩展三角形集为一个B1OpTriangleSet;

构造相交线,搜索第一对相交三角形,它通过检测两个轴对齐的最小边界框是否重叠并执行边与三角形相交计算以定位相交点,然后跟踪包含最后一个相交点的一对三角形的相邻三角形以延长相交线,直到相交线第一个相交点与最后一个相交点相同,确保相交线闭合或直到遍历完所有三角形;

用修改Watson方法构建Delaunay二维网格拆分相交线通过的每个三角形,它构建一个轮廓点序列链接子相交线,在插入点时,检查网格中的每个三角形的外接圆是否包含该点或最后一个线段穿过网格中的该三角形,决定是否要进一步分割网格中的此三角形;

检查每个三角形是否可见;

分步骤重新组合面,包括复制三角形,删除三角形,反转几何对象的每个三角形的法向,和合并保留三角形以形成一个或多个新的扩展三角形集;以及

将扩展三角形映射到渲染绘图面。

12. 根据权利要求11所述的系统,其中任何即时布尔运算,包括组合、相交、排除、差和分割,使用几何对象的渲染绘图面创建新几何对象,将几何对象的渲染绘图面映射到扩展三角形,构建相交线,拆分相交线通过的每个三角形,检查每个三角形是否可见,重新组合面以形成新的扩展三角形集,和将扩展三角形映射到渲染绘图面而无需使用数据结构CSG和B-REP。

13. 根据权利要求11所述的系统,其中任何即时布尔运算,包括组合、相交、排除、差和分割,使用几何对象的渲染绘图面创建新几何对象,将几何对象的渲染绘图面映射到扩展三角形,构建相交线,拆分相交线通过的每个三角形,检查每个三角形是否可见,重新组合面片以形成新的扩展三角形集,和将扩展三角形映射到渲染面,在造型系统中使用两类数据结构CSG和B-REP。

14. 根据权利要求11所述的系统,其中构造相交线这一过程重复多次,使用轴对齐最小边界框并展开边与三角形相交计算,确保相交点准确,相交线不是近似曲线:

建立公式 $P = P_i + t * (P_{(i+1) \% 3} - P_i)$,用于表示三角形 T_a 的第 i 条边, T_a 为三角形对中的一个三角形, t 是参数,为任意实数;

建立公式 $ax + by + cz + d = 0$,用于记录三角形对中 T_b 定义的平面;

和求两个线性公式的解。

15. 根据权利要求11所述的系统,其中搜索第一对相交三角形和跟踪相邻三角形,计算边与三角形交点,通过验证点是否在三角形的边上,替换直接计算边与边相交。

16. 根据权利要求11所述的系统,其中拆分每个三角形用修改的Watson方法构建Delaunay二维网格,该方法定义一个三角形集、删除的三角形集和多边形,即使子相交线不是凸的,也能将三角形划分为不同的小三角形,包括:

构建一个轮廓点序列链接子相交线和三角形顶点;

根据三角形的法线将三维点序列映射到二维产生二维序列点;

添加四个点以形成一个大一些的边界框包围所有二维点;

假设边界框的一条对角线将框拆分为两个三角形,并将它们添加到三角形集中;

将除边界点以外的所有点插入三角形集中进一步拆分此网格中的三角形,对于每个点,检查三角形集中的每个三角形是否它的外接圆包含该点或最后一个线段穿过该三角形,当条件满足时,将其从三角形集中删除并添加到删除的三角形集中,使用删除的三角形集扩展多边形并立即清空删除的三角形集,使用多边形生成三角形并将其添加到三角形集;

从三角形集中删除边界三角形;

和将三角形集映射为三维三角形并将其添加到 $B1OpTriangleSet$ 中。

17. 根据权利要求11所述的系统,其中检查每个三角形是否可见进一步包括:

计算属于几何对象A的三角形Ta的质心C;

构建线L, $P=C+t*N$ 穿过质心C并沿着Ta的法线N;

对于对象B的每个三角形Tb,检查线L与三角形Tb是否交在三角形Tb内部点,根据两点间的距离计算深度t并将t添加到深度缓冲区,如果缓冲区中负t的数目等于正t的数目,将Ta设置为不可见。

18. 根据权利要求11所述的系统,在曲面裁剪时检查三角形是否为不可见的步骤包括:

将所述曲面片的B10pTriangleSet的一般点的位置序列号设为0,将相交线上的点的位置序列号按升序或降序设定值,这取决于相交线与裁剪轮廓线方向是否相同;

对于每个裁剪轮廓边界线上的三角形,根据其顶点的位置序列号的值确定它位于裁剪轮廓线的左侧或右侧,并将其为非边界的相邻三角形设置为左侧或右侧。

19. 根据权利要求11所述的系统,其中任何即时布尔运算,即组合、相交、排除、差和分割,使用的一个或多个步骤重组构建操作结果:删除几何对象的不可见或可见三角形,将对象的不可见三角形复制到缓冲区或将三角形从缓冲区复制到对象,反转几何对象的每个三角形的法线,和合并对象的三角形以形成新的扩展三角形集。

20. 根据权利要求11所述的系统,其中将扩展三角形直接映射到渲染面以便显示并向下一个布尔运算提供数据。

一种使用几何面的即时布尔运算方法

技术领域

[0001] 本发明为计算机辅助设计、计算机制图、实体造型系统和曲面造型系统提供一种即时布尔运算方法,用初级几何形体构建三维几何模型。机械工业、文化体育,有几何形体的领域,就可能有CAD/CG应用。

背景技术

[0002] 计算机硬件是如此高度发达,甚至普通的个人电脑都可以用来安装和运行商业CAD/CG系统,它们通常具有布尔操作功能,包括AND、OR和NOT。PC组件包括输入设备,如鼠标和键盘,主计算机,屏幕和打印机。软件系统包含几何和非几何功能。图1为PC机的主要组成部分,图2A到2D描绘了一个典型的CAD/CG软件系统构架。

[0003] 布尔运算为CAD/CG/实体造型系统提供了一个从不同的几何形体构建复杂实体几何模型的一般过程,造型的几何形体包括初级几何对象、旋转对象和拉伸对象。Lee应用布尔运算来划分曲面[Lee US Patent No.6,307,555]。

[0004] 布尔运算可以依赖构造实体几何(CSG)来记录几何对象和操作序列的层次结构,这在技术上很容易实现,然而边界表示(B-REP)被认为是一种支持更多几何形体的更加灵活的方法,如扩展的几何图形[Gursoz,1991]。

[0005] 本发明提出五个布尔运算命令:组合、交集、排除、差和分割,它们直接作用于绘制函数的几何面分解出来的三角形且不需要数据构造实体几何或边界表示。本发明中定义的数据结构是一些简单的类,包含在本发明中的方法简洁且易于实现,这五个命令允许用户通过选择几何对象类型并定义面片来创建几何模型。图3展示了一个有6个面片的长方体和一个有不同面片的球体布尔运算产生不同的结果。

[0006] 虽然这五个命令是为实体建模和曲面建模而设计的,本发明所包含的裁剪命令为曲面造型提供了一种替代方法,它在确定某一面片是否可见时使用不同的方法。

[0007] 本发明提出了一种不同于CSG和B-REP的数据结构和方法,包括三角形与三角形相交,用子相交线分割三角形,确定一个面是否可见,并重新分组三角形形成几何模型。

发明内容

[0008] 本发明提供了一组步骤,构成一个执行布尔运算的方法,来生成复杂的几何模型,并直接在由几何面分解的三角形上进行处理,该操作由计算机硬件和软件完成。软件包括图形渲染绘制系统,如OpenGL库函数。一个几何形体,例如球面、圆锥、圆柱、长方体、三角形网格体、旋转对象或拉伸或曲面,经过三角化,生成一个用于图形渲染绘制的集,记为TriangleSet。当为执行布尔运算选择了两个几何形体时,相邻的三角形被添加到TriangleSet中的每个三角形中,每个形体都产生了另一个集合,B1OpTriangleSet。

[0009] 本发明描述的布尔运算的第二步是在三角形集合之间搜索和创建相交线。它首先找到一对相交的三角形:这个系统为每个三角形建立一个轴对齐的最小边界框,并检查两个边界框是否重叠以决定是否需要进行边与三角形相交计算。当边与三角形相交点位于三

角形内时,该系统完成搜索任务,并将点数据存储到相交线集内。

[0010] 为了扩展当前相交线,此方法跟踪相邻三角形并计算边与三角形交点,直到相交线闭合。

[0011] 这个发明描述的布尔运算的第三步是拆分三角形。相交线的每一段引用两个三角形,其中的每个三角形具有至少一个子相交线,它包含一个或多个线段,线段将三角形划分为三个或更多的小三角形。拆分三角形后,删除原始三角形,并将这些较小的三角形添加到B1OpTriangleSet中。

[0012] 这个发明描述的布尔运算的第四步是确定每个三角形是否可见。如果一个三角形被其他三角形所包围,则它是不可见的。三角形是可见的,表示它位于另一个对象之外。

[0013] 本发明所描述的布尔运算的第五步是重新组合三角形:其中一些必须被删除,有些需要放在一起,重组共有五种情况。

[0014] 这个发明描述的布尔运算的最后一步是将B1OpTriangleSet映射到TriangleSet。

[0015] 所述的曲面裁剪命令也包含六步骤。最初,该系统将曲面映射到B1OpTriangleSet,并将其裁剪轮廓线之一映射成为拉伸形状以形成B1OpTriangleSet。步骤二、三和六与布尔运算相同。步骤四是检查三角形在裁剪轮廓线的左侧还是右侧,以决定是否需要保留。步骤五,重新组合三角形,在系统裁剪曲面时,仅删除左侧或右侧三角形。

附图说明

[0016] 图1显示个人电脑的主要部件,一般包括一台主机,输入设备,包括鼠标和键盘,显示器和打印机。一个复杂的CAD/CG系统可以在PC机上运行。

[0017] 图2A到2D描述了一个软件体系结构,其中一个CAD/CG/几何建模系统采用布尔运算和曲面裁剪来建立几何模型。

[0018] 图3表示不同面产生的各种结果,其原始几何对象类型和大小是相同的:左边的示例包含较少的面,右边的示例包含较多的面。在这些例子中,布尔相交操作在一个长方体和一个球体上。

[0019] 图4是使用几何面片的即时布尔运算的流程图。

[0020] 图5描述了一个三角形有三个相邻三角形,给定一个三角形及其两个顶点,在实体模型中只有一个相邻三角形。

[0021] 图6A和图6B显示两个最小边界框不重叠和两个边界框相互重叠。每个三角形都有一个最小边界框。如果两个边界框不重叠,包含在两个边界框中的三角形不相交。若边界框重叠,则边与三角形相交计算是需要的。

[0022] 图7A到7C描述三种三角形相交情况:交点落在三角形内部,交点位于三角形的边上,交点是三角形的一个顶点。

[0023] 图8A到8D显示搜索候选集,允许系统遍历下一个三角形,用于通过边与三角形计算来扩展交点线。填充颜色的三角形是相交线的最后一对三角形,未填充颜色的三角形由数据结构Triangle3dEx的成员m_NeigTri指引,它引导系统在生成相交线时搜索最小的三角形集。该集合包含一个三角形、两个三角形或零。

[0024] 图9A到9D显示四个相交线的例子。一个长方体与一个球体相交,球体有不同的面数。

[0025] 图10A到10C给出三个以暗色显示的子相交线的例子。图10A有一个子交点线,图10B有两个,图10C有一个。

[0026] 图11A到11D显示四个示例,子相交线把三角形分成三角形集。

[0027] 图12A到12H显示了一个Delaunay二维网格序列,其中每个相交点都是一步一步地插入到网格中的。

[0028] 图13是修改后的Delaunay二维网格Weston方法的流程图,它创建了12A到12H的二维网格序列。

[0029] 图14显示了三角形及其Delaunay二维网格。原始三角形被删除,只有Delaunay网格保留用于以后的计算。

[0030] 图15显示t缓冲区中t可能是正的和负的。如果在缓冲区中负t和正t的大小是平衡的,则有关的三角形被另一个对象包围,是不可见的。

[0031] 图16A到16E显示五个使用长方体和球体进行布尔运算的示例。图16F和图16G描述了两个布尔运算结果的内部网格:组合和排除。

[0032] 图17显示了一个轮廓线用于裁剪闭合曲面,一个变形球体,并产生两个孔。

[0033] 图18给出了一个实施例,其中轮廓线裁剪一个拉伸表面,一个管,产生了一个孔。

具体实施方式

[0034] 本发明定义了这些数据结构:Point3dEx、Triangle3dEx、B1OpTriangle3dSet分别继承Point3d、Triangle3d、Triangle3dSet。Triangle3dSet存储几何对象的渲染绘图面片。当执行布尔运算时,所述系统将渲染绘图面片映射到B1OpTriangle3dSet,后续的操作过程都将重点放在B1OpTriangle3dSet的成员和属性上。图4为描述本发明实施的布尔运算的主要过程的流程图。在布尔操作完成以后,所述系统将存储在B1OpTriangle3dSet中的结果映射到渲染绘制面。

[0035] 用于渲染绘图的几何面片

[0036] CAD系统常用面片表示几何对象,如球面、锥面、长方体、圆柱、拉伸对象或旋转对象。一个面片包括三个或更多点,面通常被分解成三角形,便于计算。一个长方体有六个面分解成十二个三角形。球体可能有十八个面,包含二十四个三角形。球体也可以使用超过1000个面和三角形来表示。图3显示用不同的面绘制的球面。在绘制几何对象时,该方法使用Triangle3dSet作为三角形集数据结构,它包含两个属性:三维点集和三角形集,其中Triangle3d引用Point3d。

```
class Triangle3dSet
```

```
{
```

```
    DataSet<Point3d> m_PointSet;
```

```
[0037]    DataSet<Triangle3d> m_TriangleSet;
```

```
};
```

```
class Triangle3d
```

```
{
```

```

        Point3d      *m_Points[3];
    };

    class Point3d
[0038]    {
        DataTypeI m_X, m_Y, m_Z;
    };
[0039] 布尔运算的三角形
[0040] 本发明描述的布尔运算方法定义了三个关键类:BIOpTriangleSet、Triangle3dEx
和Point3dEx。

```

```

    class BIoPTriangleSet
    {
        DataSet<Point3dEx> m_PointSet;
        DataSet<Triangle3dEx> m_TriangleSet;
    };

    class Point3dEx : Point3d
    {
        DataTypeII m_ID; // position and sequence index
        DataTypeIII m_X, m_Y, m_Z; // DataType III may be different from DataTypeI
[0041] };

    class Triangle3dEx : Triangle3d
    {
        Point3dEx *m_Points[3];
        DataTypeII m_ID;
        Plane m_Plane;
        DataTypeIV m_Normal[3];
        Triangle3dEx *m_NeigTri[3]; // neighboring triangles
    };

```

[0042] DataTypeII可能是整数、长整数、无符号长整数或其他整数类型。DataTypeIII是浮点数据类型,如浮动、双精度甚至长双精度。

[0043] 类Triangle3dEx说明每个三角形可能有三个相邻三角形。在BIOpTriangleSet中每个三角形仅存储一个副本。用长方体为示例,最简单的情况是它有十二个三角形,即使其

中的每个三角形都有三个相邻三角形,B1OpTriangleSet仍然总共存储十二个三角形。

[0044] 技术上Triangle3d可能具有属性m_Normal。如果DataTypeI和DataTypeIV是同一类型,例如,double,那么属性m_Normal可以继承。

[0045] 数据映射

[0046] 将Triangle3dSet映射到B1OpTriangleSet的过程包括从渲染绘图状态复制点集和三角形集,并填充默认属性。数据映射包含以下过程:

[0047] 1) 将点从Triangle3dSet中复制到B1OpTriangleSet,并确保没有相同的点。

[0048] 2) 将三角形从Triangle3dSet中复制到B1OpTriangleSet。

[0049] 3) 对于B1OpTriangleSet中的每个三角形,设置其相邻的三角形。

[0050] 4) 计算B1OpTriangleSet中每个三角形的法线并建立平面方程。

[0051] 备注1:给定两个点a和b,如果 $|x_a-x_b|<\epsilon$ 且 $|y_a-y_b|<\epsilon$ 且 $|z_a-z_b|<\epsilon$,其中 ϵ 为正值浮点数字,例如 $5.0e-16$,则b与a相同。

[0052] 备注2:当将点从渲染绘图映射到B1OpTriangleSet时,系统检查B1OpTriangleSet中是否存在相同的点。

[0053] 备注3:具有三点的三角形定义了一个平面,其数学公式为 $ax+by+cz+d=0$,在类平面内部,将其记录为四个数字的数组,如m_ABCD[4]。

[0054] 备注4:三角形,如果其三点不相同,则始终具有有效的法线。即使它与m_ABCD有关,一份单独的副本仍使事情变得更加清晰,便于以后处理。

[0055] 备注5:每个三角形有三条边,当没有重叠点时,它在实体模型中有三个相邻三角形。图5显示了一个例子:一个充满暗色的三角形和它的三个相邻三角形。在曲面裁剪中,三角形的一个或两个相邻三角形可能为空。

[0056] 第一交点

[0057] 每个三角形都有三个顶点,它们定义一个最小长方体。该方法采用了轴线对齐最小边界框的概念。给定一对三角形,如果它们的边界框不重叠,则两个三角形没有交点;否则,该方法进行边与三角形相交计算。

[0058] 如果三角形Ta的一条边与三角形Tb定义的平面相交,而相交点位于三角形Tb内,则相交点是第一个交点。如果相交点位于Tb之外,将对中的三角形位置(Ta、Tb)对调为(Tb、Ta),并再进行边与三角形相交计算。

[0059] 给定三角形Ta的第i条边, $i \in [0,2]$,它的公式是: $P=P_i+t*(P_{(i+1)\%3}-P_i)$;由三角形Tb定义的平面,其公式为: $ax+by+cz+d=0$ 。如果两个方程式有一个解,则边与平面相交。如果边与平面相交点落在三角形Tb内,则该点是边与三角形的交点。

[0060] 扩展相交线

[0061] 此方法将相交点的数据结构定义为PntEgTri:

```

classPntEgTri
{
    Triangle3dEx *m_Tri0, *m_Tri1;
    DataTypeII m_EdgeIndex;
[0062]    DataTypeII m_PointPosi;
    Point3dEx m_Point;
    Point3dEx *m_PntGlobalIndexA, *m_PntGlobalIndexB;
};

```

[0063] 根据在三角形上的位置,相交点可分成如图7A到7C所示的三类。

[0064] 1) 最普通的情况是边与三角形相交,相交点位于三角形Ta边上,并在三角形Tb内;

[0065] 2) 边与边相交,相交点位于三角形Ta的边上,并位于三角形Tb的边上;

[0066] 3) 边与顶点重叠,相交点位于三角形Ta的边上和三角形Tb的顶点。当扩展相交线时,系统跟踪下一个相邻的三角形,并检查边与三角形相交的情况,直到相交线闭合或遍历完所有三角形。

[0067] 子相交线

[0068] 相交线通过一组三角形,将其中的每个三角形都划分为多个分区。三角形内相交线的线段构成子相交线。图10A到10C显示三个的例子,其中暗线是子相交线。实际上,三角形可能有零条、一条、两条或三条子相交线。由下面的过程显示如何获取一个三角形具有多少子相交线:

[0069] 对于每个相交线,

[0070] 对于每个相交点,获取三角形的参考值:(m_Tri0,m_Tri1),

[0071] 对三角形对中的每一个三角形,如果它没有拆分,

[0072] 对每个相交线,搜索并生成子相交线

[0073] 给定一个有效的三角形和一条相交线,决定相交点是否属于三角形的子相交线,此方法检查是否:

[0074] 1) 相交点在三角形的边上,

[0075] 2) 或相交点在三角形内,

[0076] 3) 或相交点位于三角形的顶点。

[0077] 拆分三角形

[0078] 给定一组子相交线,拆分一个三角形,此方法

[0079] 1) 删除重复的交点。如果相邻的交点是相同的,这个方法只保留一个副本。

[0080] 2) 确定始末交点的位置:检查每个交点位于三角形的哪个边上。

[0081] 3) 拆分三角形的上部分区、下部分区和中间分区,如果这个分区存在。

[0082] 给定一个平面上的一组点,代表一个三角形的区域,将平面分解成一组三角形,本发明修改了狄劳尼(Delaunay)二维网格沃森(Watson)方法,原文是在1981年出版的[沃森,1981]。

[0083] 狄劳尼二维网格有三个数据集:三角形集,记录生成的三角形;已删除的三角形

集,只存储已删除的三角形;一个多边形,记录已删除三角形集的轮廓。

[0084] 修改后的狄劳尼二维网格沃森方法,包含以下步骤:

[0085] 1) 建立一个三维轮廓点序列,链接子相交线和三角形的顶点,如果存在的话。

[0086] 2) 根据三角形的法线,将三维点序列映射到二维平面上,产生二维点序列。

[0087] 3) 添加四点,形成一个大一些的边界框,包含所有的二维点。

[0088] 4) 假设边界框的一个对角线将该框拆分为两个三角形,并将其添加到三角形集中。

[0089] 5) 将除边界框之外的每个点逐步插入到三角形集合中。

[0090] a) 对于每个点,检查三角形集合中的每个三角形,它的外接圆是否包含此点或最后一个线段通过该三角形。如果条件满足,将其从三角形集合中删除,并将其添加到已删除的三角形集。

[0091] b) 使用已删除的三角形集来生成多边形并立即清空已删除的三角形集。

[0092] c) 使用多边形生成三角形集并将它们添加到三角形集合中。图12A至12H显示了狄劳尼二维网格序列。

[0093] 删除已拆分的三角形

[0094] 在上面的步骤中,一个已拆分的三角形得到了一个标记。遍历所有三角形后,此方法将删除有标记的三角形。图14显示了一个删除结果。

[0095] 不可见的面片

[0096] 给定两组三角形A和B,如果A包围B的三角形 T_b ,那么 T_b 就不可见;如果B包围A的三角形 T_a ,那么 T_a 是不可见的。

[0097] 要检查三角形T是否被对象O包围,本发明使用以下步骤。

[0098] 1) 计算三角形T的质心C;

[0099] 2) 构造一条直线 $L:P=C+t*N$,它穿过质心并沿三角形T的法线N;

[0100] 3) 对于对象O中的每个三角形 T_o ,计算直线与平面相交点。如果有一个有效的相交点,且交点落在三角形 T_o 内,然后计算t,它由质心C和相交点位置确定,并把t添加到深度缓冲区中。

[0101] 4) 检查t缓冲区中存储的负t和正t数量的大小。如果t缓冲区中负t的大小等于正t的大小,则三角形T是不可见的。

[0102] 当执行曲面裁剪时,此系统执行下面的过程以确定三角形是否不可见。

[0103] 1) 将有关曲面面片 $B10pTriangleSet$ 的每个 $Point3dEx$ 的成员 m_ID 设置为0。

[0104] 2) 按升序或降序标记该曲面裁剪中相交线的 $Point3dEx$ 的 m_ID ,这取决于所述相交线和裁剪轮廓线是否在同一个方向上,例如,两者都是逆时针的。

[0105] 3) 根据每个三角形的成员 m_Points 的 m_ID ,确定它是否为边界三角形。

[0106] 4) 对于每个边界三角形,确定它是在裁剪轮廓线的左侧还是右侧,并将其相邻的三角形设置为左或右。

[0107] 重组面

[0108] 本发明给定了五种布尔运算,每个都有不同的重组过程。

[0109] 组合操作,逻辑上是OR,合并两个实体几何对象,生成一个新对象,常丢弃不可见的区域并保留从外部查看可见的部分,具有以下过程。

- [0110] 1) 删除对象A不可见的三角形;
- [0111] 2) 删除对象B不可见的三角形;
- [0112] 3) 合并对象A和B的三角形。
- [0113] 相交操作在逻辑上是AND,它使用两个几何对象的公共区域创建一个实体几何对象,并丢弃共享公共区域之外的A和B的任何部分,具有以下过程。
- [0114] 1) 删除对象A的非不可见三角形;
- [0115] 2) 删除对象B的非不可见三角形;
- [0116] 3) 合并对象A和B的三角形。
- [0117] 排除操作通过剔除两个几何图形对象的公共区域并保留不共享分区,从而生成一个实体几何对象,其步骤如下:
- [0118] 1) 将对象A的不可见三角形复制到缓冲区中,bufferA;
- [0119] 2) 删除对象A中的不可见三角形;
- [0120] 3) 将对象B的不可见三角形复制到对象A;
- [0121] 4) 从对象B中删除不可见的三角形;
- [0122] 5) 将bufferA中的三角形复制到对象B;
- [0123] 6) 反转A和B每个不可见的三角形的法线;
- [0124] 7) 合并两个对象的三角形。
- [0125] 差操作,通过删除A内部含有B的任何部分,用几何对象B剪切几何对象A,具有以下过程。
- [0126] 1) 删除对象A不可见的三角形;
- [0127] 2) 删除对象B的非不可见三角形;
- [0128] 3) 反转对象B中每个三角形的法线;
- [0129] 4) 将对象A和B的三角形合并。
- [0130] 分割运算将两个实体几何对象A和B划分为三个对象,两个几何对象的公共部分,A和B的不共享部分,具有以下过程。
- [0131] 1) 将对象A的不可见三角形复制到缓冲区中,bufferA;
- [0132] 2) 将对象B的不可见三角形复制到bufferA;
- [0133] 3) 将对象A的不可见三角形复制到另一个缓冲区,bufferB;
- [0134] 4) 删除对象A的不可见三角形;
- [0135] 5) 将对象B的不可见三角形复制到对象A;
- [0136] 6) 删除对象B的不可见三角形;
- [0137] 7) 复制对象A存储在bufferA中的不可见三角形到对象B;
- [0138] 8) 反转A和B的每个不可见三角形的法线。
- [0139] 映射到渲染绘图面
- [0140] 布尔运算完成后,此方法将B1OpTriangleSet映射到用于渲染绘图的三角形。
- [0141] 1) B1OpTriangleSet中的每个Point3dEx都映射到TriangleSet中的Point3d;
- [0142] 2) B1OpTriangleSet中的每个Triangle3dEx都映射到TriangleSet中的一个Triangle3d。
- [0143] 美国专利文件

[0144] 6,307,555 10/2001Lee 345/423

[0145] 其他出版物

[0146] “Boolean Set Operations on Non-Manifold bounding Representation Objects”,E.Gursoz et al.,Computer-Aided Design 23(1991)Jan./Feb.No.1London, GB.

[0147] “Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopest”,D.F.Watson,The Computer Journal 24(2)1981.

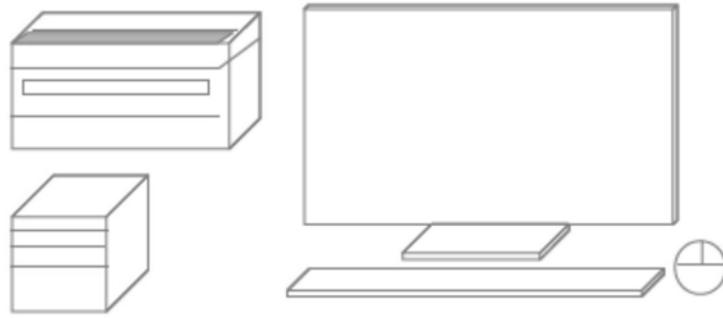


图1

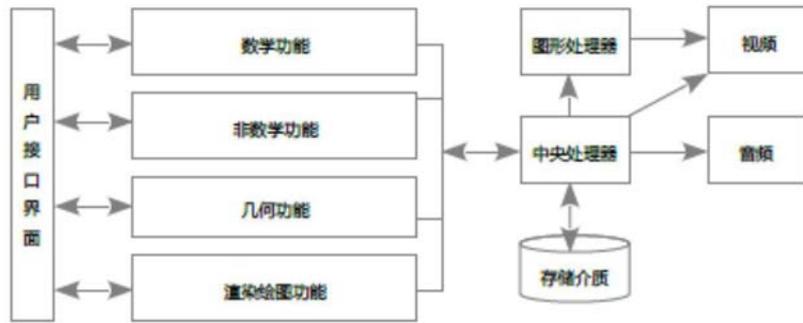


图2A

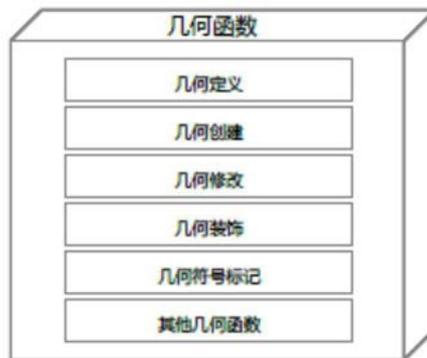


图2B



图2C

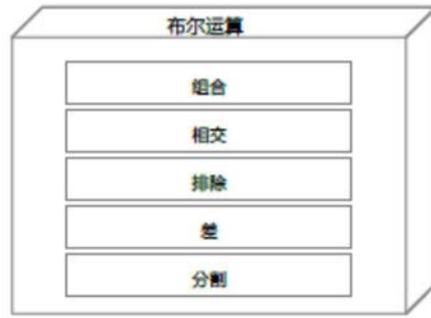


图2D

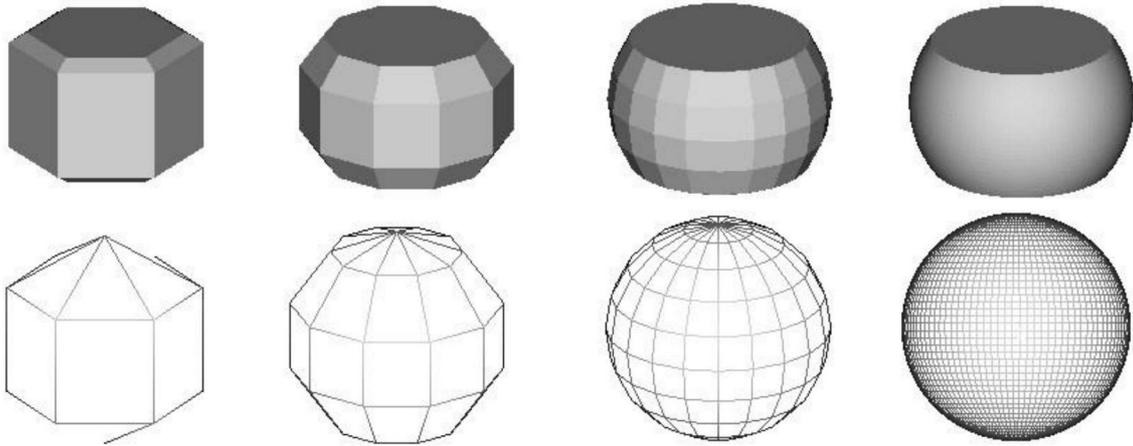


图3

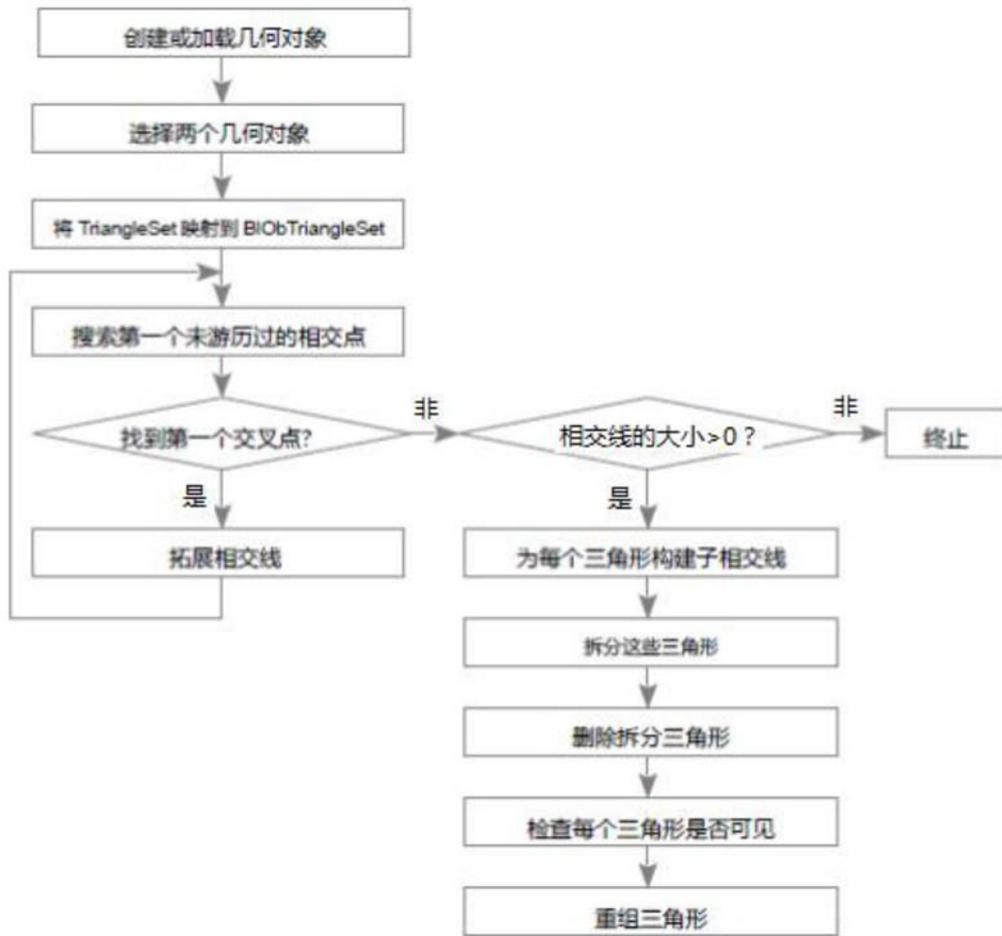


图4

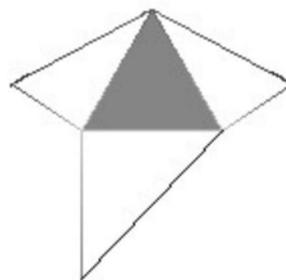


图5

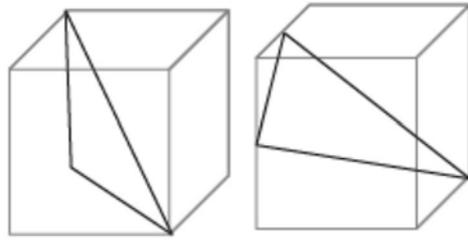


图6A

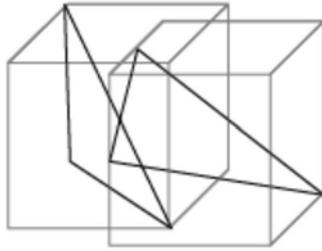


图6B

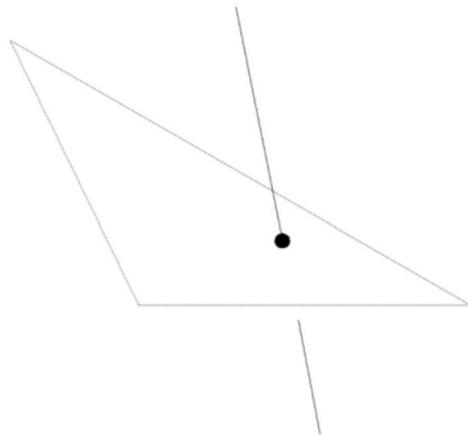


图7A

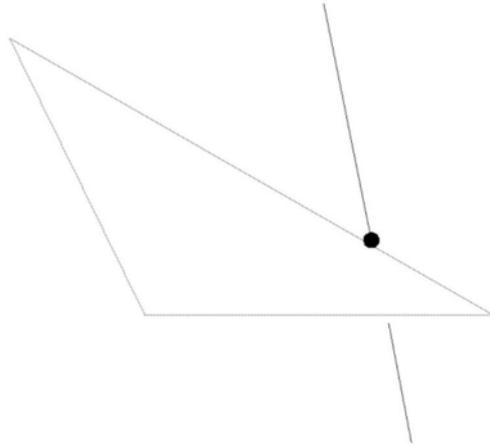


图7B

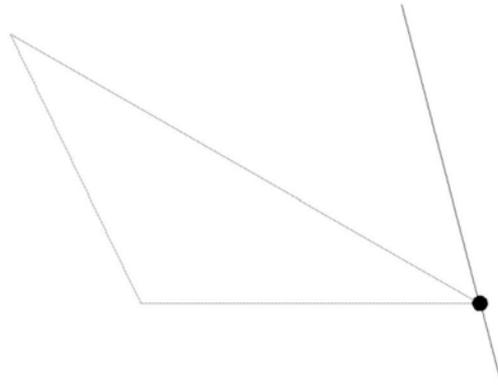


图7C

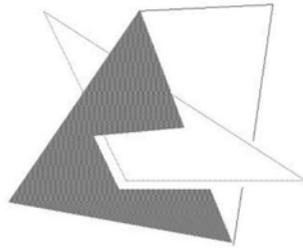


图8A

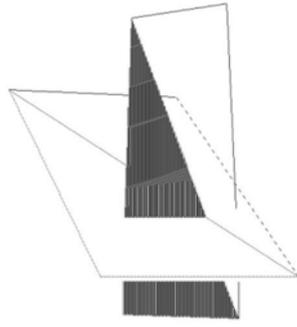


图8B

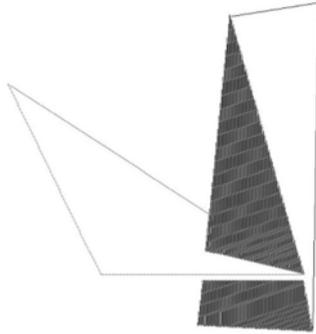


图8C

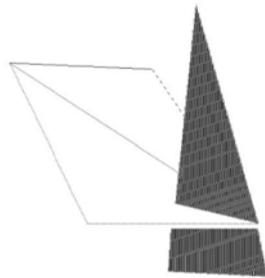


图8D

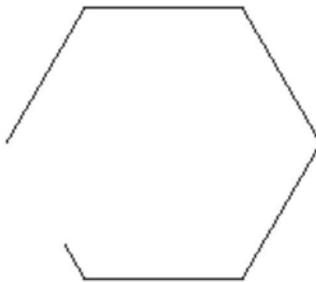


图9A

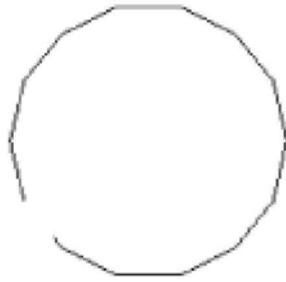


图9B

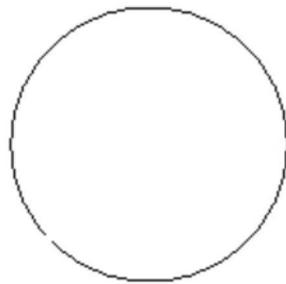


图9C

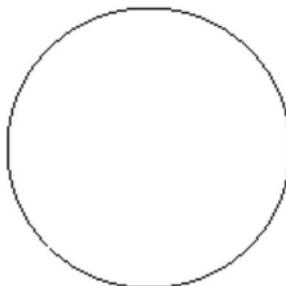


图9D

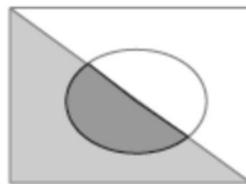


图10A

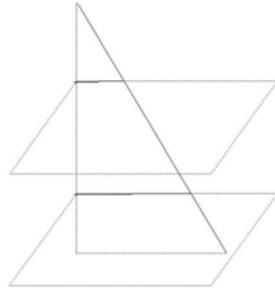


图10B

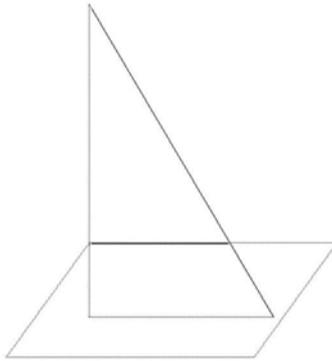


图10C

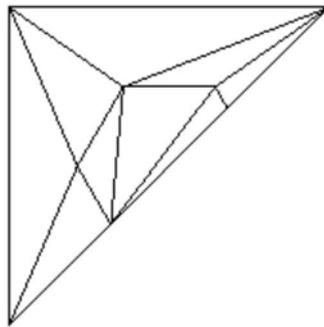


图11A

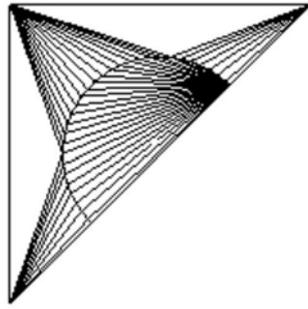


图11B

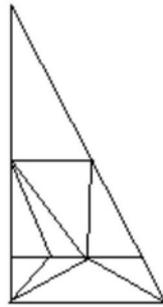


图11C

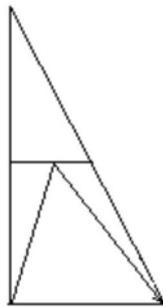


图11D

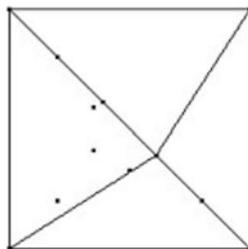


图12A

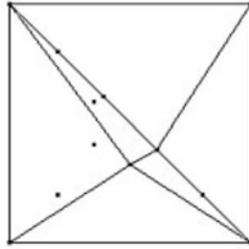


图12B

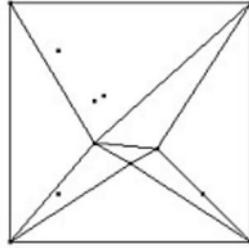


图12C

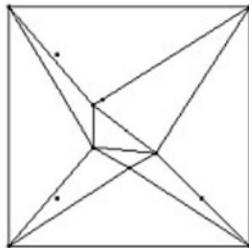


图12D

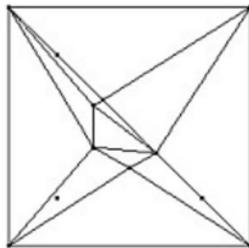


图12E

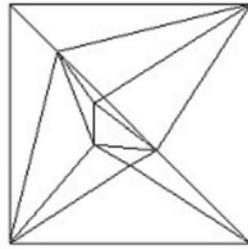


图12F

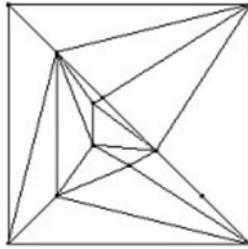


图12G

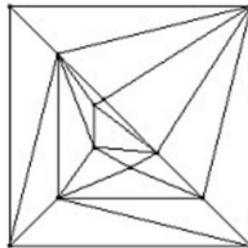


图12H

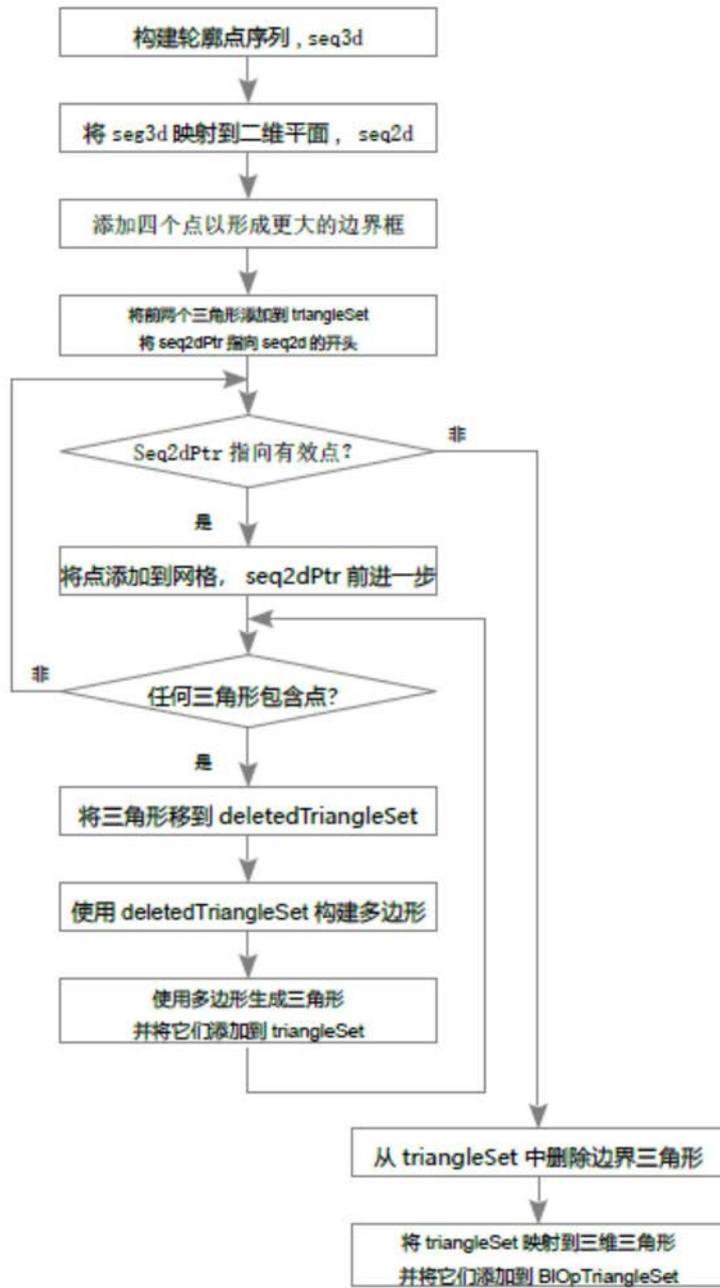


图13

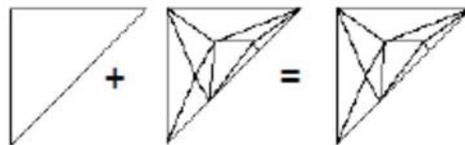


图14

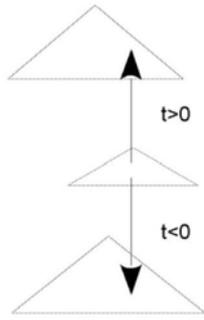


图15

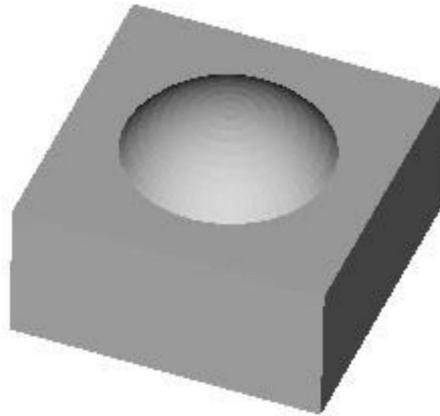


图16A

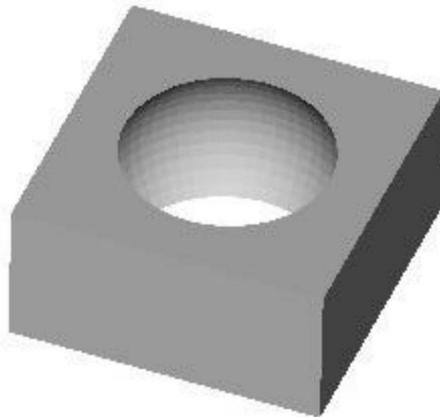


图16B

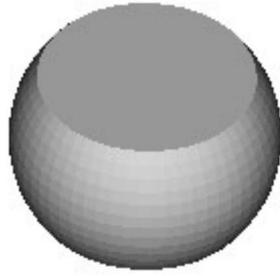


图16C

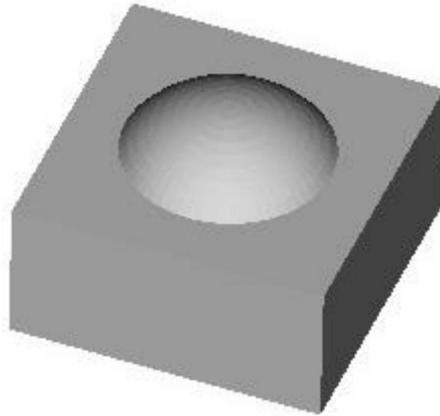


图16D

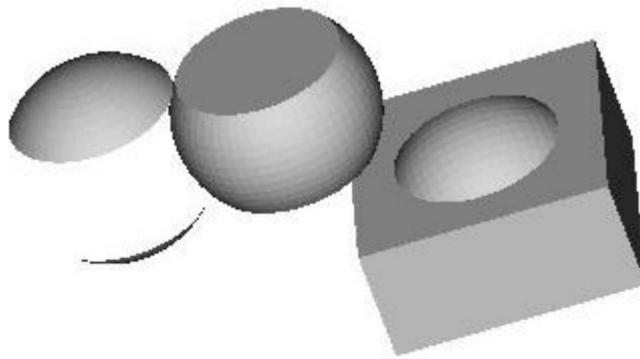


图16E

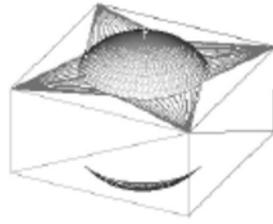


图16F

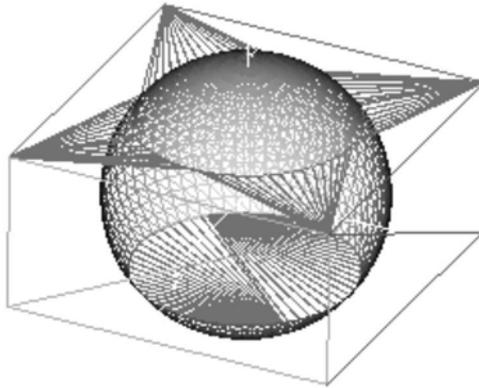


图16G



图17

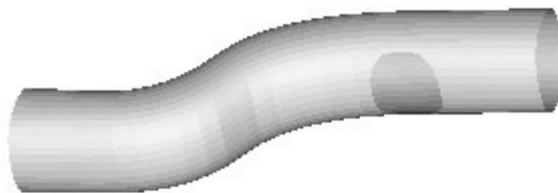


图18