

# 一种 STL 中交错三角片的识别与删除算法

章雅娟<sup>1,2</sup>, 姜晓峰<sup>1</sup>

(1 苏州大学 计算机科学与技术学院, 江苏 苏州 215006; 2 南通大学 计算机科学与技术学院, 江苏 南通 226019)

**摘 要:** 介绍了一种基于 STL 表示的三角网格模型中交错三角片的识别与删除算法. 该方法首先对 STL 三角网格模型重新建立拓扑关系; 接着将网格模型包围盒划分成若干个小的立方体包围盒, 并记录和各立方体包围盒相贯的三角片 ID 号; 然后检测单个立方体包围盒中的三角片包围盒空间是否重叠, 在空间重叠的基础上进一步检测两个三角片是否交错; 最后将所有标志为交错的三角片及其拓扑关系删除. 算例结果证明了该算法的可行性和高效性, 并优化了逆向工程相关应用中三角网格模型的质量.

**关键词:** 逆向工程; STL; 交错三角片; 网格优化

**中图分类号:** TP317.4

**文献标识码:** A

**文章编号:** 1000-7180(2010)11-0143-04

## Recognition and Deletion Algorithm of Interlaced Triangular Meshes in STL Files

ZHANG Ya-juan<sup>1,2</sup>, JIANG Xiao-feng<sup>1</sup>

(1 School of Computer, Soochow University, Suzhou 215006, China;

2 Schoole of Computer, Nantong University, Nantong 226019, China)

**Abstract:** This paper introduces the method for recognizing and deleting interlaced triangular meshes in the triangular mesh model which is expressed based on STL files. Firstly, establishes the topology relations to the STL triangular mesh model; Secondly the surrounding box of mesh model is divided into certain small cube surrounding boxes, and records ID numbers of triangular meshes of this cube space that is pierced in the cube surrounding boxes; Then examines whether space of surrounding box of triangular meshes to be overlapped or not in each cube surrounding box, further determines whether two triangular meshes to be folded or not on base of spatial overlap; Finally deletes all triangular meshes and its topology relations which are symbolized interlaced triangular meshes. The example proves that this algorithm is feasible and highly effective and optimizes the quality of triangular mesh model in application of reverse engineering.

**Key words:** reverse engineering; STL; interlaced triangular meshes; mesh optimization

### 1 引言

逆向工程技术一般是以数字化测量设备的输出数据作为原始信息来源, 随着软件功能的加强, 目前很多测量设备可以在输出测量数据的同时输出三角网格数据(即经过三角化以后的数据)或者 STL (Stereolithography) 格式数据. STL 文件格式使用一系列小三角形平面来逼近实体模型, 已经成为

CAD/CAM 系统接口文件格式的工业标准之一<sup>[1]</sup>. 由于测量方法、设备指标和模型复杂性等局限, 一般需要在物体的不同位置 and 不同侧面经过多次测量后才能获得物体的整体表面数据. 因此, 直接从测量设备测量并生成的 STL 文件所描述的网格模型经常会存在一些错误面片, 如退化、多余、孤立、法矢反向、裂缝、孔洞和不共顶点等. 为了提高逆向工程后续处理环节的效率和质量, 有必要对这些错误的三角面片进行处理, 因此有关研究人员提出了针对这

些错误的探测和修补算法<sup>[1-4]</sup>,但针对交错三角片这一类较难识别的三角面片错误现象的检测和处理方面的研究仍然较少<sup>[5]</sup>.

文中在重构 STL 文件所表示的三角网格模型拓扑结构的基础上,给出了识别交错三角片并予以去除的高效算法,并在原型系统中进行了实现,验证了所设计算法的正确性.

## 2 拓扑结构的重建

STL 文件所表示的网格模型中缺少顶点、边和三角面片之间的拓扑关系,这对后续处理环节存在诸多不便,因此作者在处理交错三角片前首先构建三角网格模型的拓扑关系,建立网格模型的数据结构,如图 1 所示.考虑到后续算法要应用边的数据信息,因此建立了网格边的邻接表,如图 2 所示.

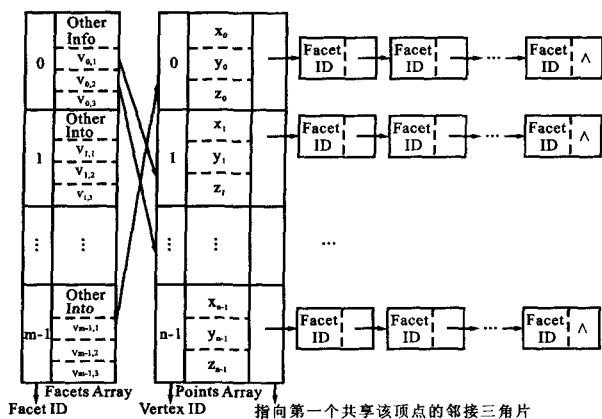


图 1 三角网格模型的数据结构

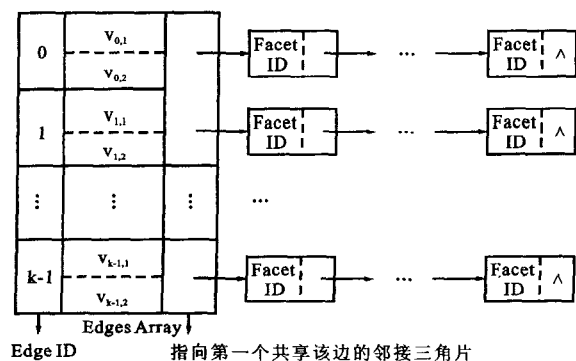


图 2 三角网格模型边的邻接表

从图 1 可以看出,网格模型的数据结构包含了三层,第一层是面片层,记录了每个三角面片的基本信息和每个顶点的 ID 值;第二层是顶点层,用来记录每个顶点的三个坐标值和共享该顶点的三角片链表指针,每个顶点的坐标只记录一次,这样完全解决了 STL 文件中重复记录顶点坐标的缺点,也很大程

度上减少了内存占有量;第三层是共享顶点的三角片链表,链表每个结点仅记录共享某顶点的三角片的 ID 值.面片索引 (FacetID) 和顶点索引 (VertexID) 是提取数据元素的两个重要索引.在面片数组中,每个面内不记录顶点的具体值,而是提供三个指针指向三个顶点在点表中的位置,进而每个顶点的三个坐标值也就确定了.同理,当一个 VertexID 确定了后,共用该顶点的面片也会被找到.因此,顶点和面片间的联系可以直接通过这样的数据结构获得.

从图 2 可以看出,边表的每条边记录两个顶点的 ID 值和共享该边三角片链表的指针,同理,链表中每个结点仅记录共享某边的三角片的 ID 值.因此,边与顶点和边与三角片之间的邻接关系可以通过这样的数据结构获得.

为了提高构造拓扑关系的速度和便于删除结点,作者采用哈希法创建顶点表,使用双链结构存储顶点的哈希表,边表的邻接关系根据三角面片表和顶点表创建,边表也以哈希法进行创建,以双链存储.

因为构造网格顶点表和网格边表求顶点结点和边结点的哈希地址时选择的哈希函数均为线性函数,所以对于不同的自变量数对不会发生冲突<sup>[6]</sup>,这里出现哈希值相等的现象是因为 STL 中有重复的点和边,用哈希法正好去除了这些冗余数据.

基于 STL 文件,采用上述数据结构和算法建立了三角网格模型的拓扑结构,为后续交错三角片识别和剔除算法建立了几何基础.

## 3 交错三角片的识别和删除

### 3.1 交错三角片的类型

相交三角片有四种类型:两三角片交于一共用顶点,两三角片共用一条边,三角片部分或全部重叠,一个三角片从其它三角片平面内穿过.前两种相交三角片都是正确的,不需要处理.第三种相交情况只要在三角化过程中适当控制误差或在造型后进行布尔运算就可以避免.第四种相交情况不能满足 STL 网格模型表面要求,属于错误的三角片相交关系,文中称其为交错三角片.交错三角片具体又可分为两种类型,即邻域三角片交错和非邻域三角片交错,如图 3 所示.

### 3.2 交错三角片的识别和删除

判断两个三角片是否相交的一般原则是:检查一个三角片的某条边是否与另一三角片交于一点,

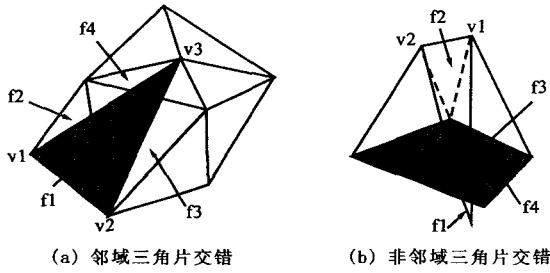


图3 交错三角片的类型

最坏的情况下要检查 5 次(每个网格的三条边分别与另一三角片比较,如果有交错,至少会有两条边与另一三角片相交).由于在实际应用中,三角片数目超过 10 万的 STL 文件是非常常见的,因此应该在保证所有交错情况都能识别的前提下尽可能减少比较的次数以提高算法的效率.

对于第一种交错三角片(如图 3(a)所示)只需要与其邻接三角片比较就容易发现这种错误,但对于第二种交错三角片(如图 3(b)所示)仅检测其邻接三角片并不能识别出错误,如果将检测范围扩大至网格模型中的所有三角片又造成时间的浪费,因此检查交错错误之前首先选择可能会发生交错的三角片范围.文中采用空间分块策略将三角网格模型所在的长方体包围盒空间分割成若干较小的立方体空间,如图 4 所示.三角网格模型所对应的整个长方体区域沿着  $x$ 、 $y$ 、 $z$  三个方向以  $\Delta d$  为边长分割成若干小立方体,每个方向小立方体的个数分别表示为:

$$n_x = \text{int}[R_x/\Delta d] + 1 \quad (1)$$

$$n_y = \text{int}[R_y/\Delta d] + 1 \quad (2)$$

$$n_z = \text{int}[R_z/\Delta d] + 1 \quad (3)$$

其中  $\text{int}[\ ]$  的功能是将一个实数转化成一个整数,  $R_x$ 、 $R_y$ 、 $R_z$  是顶点在每个坐标方向上的最大值.立方体包围盒的总数  $n_{\text{total}} = n_x \times n_y \times n_z$ .定义一个顶点的位置向量为  $\vec{P} = [p_x, p_y, p_z]^T$ ,则立方体的位置可以通过  $\vec{P}$  来定位,通过式(4)、(5)、(6)求其位置坐标( $\text{Index}(x)$ ,  $\text{Index}(y)$ ,  $\text{Index}(z)$ ):

$$\text{Index}(x) = \text{int}[(p_x - x_{\min})/\Delta d] \quad (4)$$

$$\text{Index}(y) = \text{int}[(p_y - y_{\min})/\Delta d] \quad (5)$$

$$\text{Index}(z) = \text{int}[(p_z - z_{\min})/\Delta d] \quad (6)$$

与某三角片可能会发生交错的三角片范围是与该三角片包围盒(将三角片包围在内部的最小长方体)相贯的所有立方体空间,将该三角片的 ID 号都记录到这些立方体空间中(一个三角片 ID 号被记录多次).这样一个三角片包围盒可能跨越了多个

立方体空间,被记录了多个位置.如何记录立方体并将三角片的 ID 号分配到立方体中是首先要解决的问题.文中将立方体以一个三维数组来记录,每一维的下标即它在每个坐标方向上的编号.用哈希法以三角片的顶点坐标为依据,将该三角片分配到立方体中,把分配到同一立方体的三角片串成链表.具体数据结构如图 5 所示.

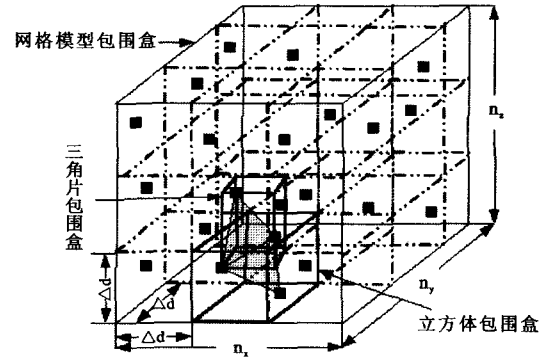


图4 包围盒细分

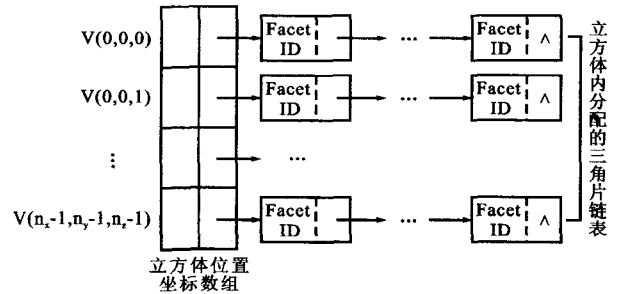


图5 包围盒内分配三角片的数据结构

基于上述数据结构,文中给出了 STL 中识别与删除交错三角片的算法流程如图 6 所示.具体描述如下:

Step1: 计算每个三角片  $\text{Facets}[i]$  沿着坐标系三个方向的最大值与最小值的差值绝对值,并用数组  $\text{Distance}[i]$  记录三个方向坐标差值绝对值最大的值;

Step2: 从  $\text{Distance}[i]$  选择出最大的值作为立方体的长度;

Step3: 在与每个三角片包围盒相贯的所有立方体中记录该三角片的 ID 号.使得每个立方体中分配了若干个 Facet ID;

Step4: 对每个立方体中记录的三角片,检测三角片包围盒空间是否相互重叠,若是转 Step5,否则转 Step8;

Step5: 以其中一个三角片为参照,检查其它三角片的三条边是否与参照三角片相交,即检查每条

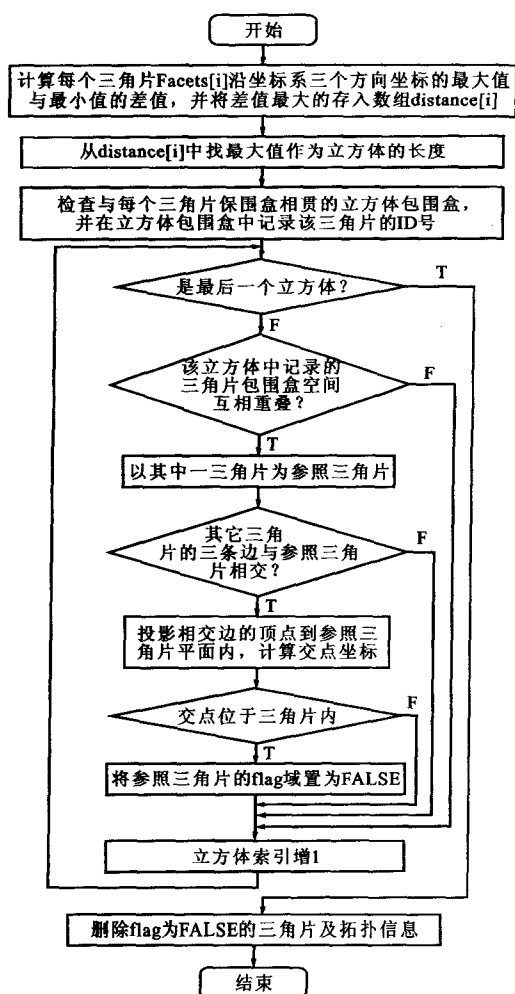


图6 交错三角片的识别与删除

边的两个顶点是否位于参照三角片所在平面的两侧,若是转 Step6;

Step6:将边的两个顶点分别投影在参照三角片所在的平面上,计算交点的坐标;

Step7:判断交点是否位于参照三角片内,若是,则将参照三角片的 Flag 置为 FALSE;

Step8:若是最后一个立方体,则转向 Step9,否则,转 Step4;

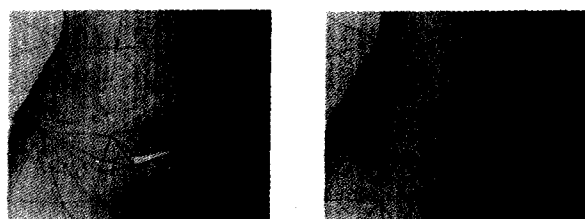
Step9:删除 Flag 为 FALSE 的三角片及拓扑关系。

### 3.3 实例验证

基于上述算法,文中在 Windows XP 操作系统环境中,以 OpenGL 图形库作为基础,采用 VC6.0 作为编程语言实现了一个原型系统以验证算法的正确性.文中给出了在 P4 1.5GHz CPU,1G 内存,显存 512MB 的硬件环境下相关算例的执行结果。

图7给出了某卡通头部三角网格模型的唇部附近区域删除第一种情况交错三角片的实例。

实验中采用的卡通头部网格模型,初始三角片数量为 5140,执行完本算法后三角片数量为 5119,识别出 21 个交错三角片(图7所代表的两种交错类型都能检测出来),执行时间为 1.71 秒.同等条件下执行文献[5]中的算法,执行时间为 1.96 秒,究其原因主要在于文中算法节约了三角片包围盒的排序时间;其次,排除了更多的不可能发生交错错误的三角片.因此,文中算法的执行效率较文献[5]中的算法有一定的优越性.但文中算法因为要建立更多的数据结构(如图5所示的包围盒内分配三角片的数据结构)而要消耗较多的内存空间。



(a) 初始网格

(b) 三角片删除后

图7 邻域三角片交错实例

## 4 结束语

文中针对 STL 文件中所存在的交错三角片问题,在重构网格模型拓扑结构的基础上,给出了识别交错三角片并予以剔除的有效算法.该算法将网格模型包围盒划分成若干小的立方体包围盒,并在立方体包围盒中记录相贯于该立方体空间的三角片的 ID 号;接着检测每个立方体包围盒中的三角片包围盒空间是否重叠,在空间重叠的基础上进一步检测两个三角片是否交错;最后将所有标志为交错三角片的三角片及其拓扑关系删除.该算法在原型系统中的实验结果证明了其正确性和高效性,在一定程度上提高了三角网格模型的质量.进一步工作的重点是研究孔洞填充算法以进一步优化三角网格模型,保证其完整性。

## 参考文献:

- [1] Gueziec A, Taubin G. Cutting and stitching: converting sets of polygons to manifold surfaces[J]. IEEE Transactions On Visualization And Computer Graphics, 2001, 7 (2):136-151.
- [2] Boston M, Kobbelt L P. A robust procedure to eliminate degenerate facets from trangle meshes[C]// Vision, Modeling, Uisualization 2001, VMV2001. Germany, Stuttgart, 2001:283-290.

(下转第 150 页)

图 6 为最差条件下的仿真波形, 仿真条件为:  $V_{dd}=1.08V$ ,  $T=150^{\circ}C$ , 使用低速晶体管模型, 读取时间的选取同上. 由仿真结果可得, 在最差条件下, 读取时间为约 17ns.

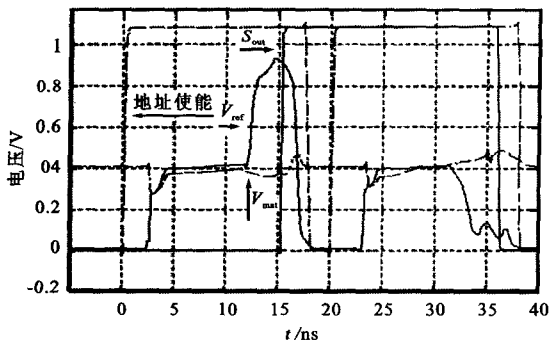


图 6 最差条件下仿真波形

由图 5 和图 6 的仿真结果可以得出, 在 1.2V 16MBit 的嵌入式 nor flash 存储器中, 新型灵敏放大器能够在较大范围的温差和电源电压变化中稳定工作. 表 2 为新型灵敏放大器在不同条件下的读取时间对比<sup>[8]</sup>.

表 2 读取时间对比

| 工作电压/V | 环境温度/ $^{\circ}C$ | 读取时间/ns |
|--------|-------------------|---------|
| 1.32   | -40               | 10      |
| 1.08   | 150               | 17      |
| 1.2    | 27                | 12.5    |

## 5 结束语

文中提出的基于折叠共射-共基放大电路面向 DSP 的新型高速低功耗灵敏放大器, 可以在  $-40^{\circ}C \sim 150^{\circ}C$  的温度范围内, 以及低至 1V 的电源电压条件下工作, 并且有快速的读取速度, 在最差工作环境

下读取时间为 17ns, 最佳工作环境下为 10ns, 常温 1.2V 条件下的读取时间为 12.5ns. 该结果证明, 新型灵敏放大器适用于 1.2V 16MBit 的嵌入式 nor flash 存储器.

## 参考文献:

- [1] Banba H, Shiga H, Urnezawa A, et al. A CMOS bandgap reference circuit with sub-1-V operation[J]. IEEE Solid - State Circuits, 1999, 34(5): 670 - 674.
- [2] Kuijk K E. A precision reference voltage source[J]. IEEE Solid - State circuits, 1973, 8(3): 222 - 226.
- [3] Atsushi Nozoe. A 256Mb multilevel flash memory with 2 MB/s program rate for mass storage applications[J]. IEEE Solid - State circuits, 1999, 34(11): 1544 - 1550.
- [4] Conte A, Giudice G L, Palumbo G. A high - performance very low - voltage current sense amplifier for nonvolatile memories[J]. IEEE Solid - State Circuits, 2005, 4(2): 507 - 514.
- [5] Chrisanthopoulos A, Moisiadis Y, Varagis A. A new flash memory sense amplifier in 0.18 $\mu m$  CMOS technology [C]// IEEE ICECS. Malta, 2001(2): 941 - 944.
- [6] 张帆, 邹雪城, 刘冬生, 等. RFID 中嵌入式 EEPROM 的超低功耗设计[J]. 微电子学与计算机, 2007, 24(7): 169 - 172.
- [7] 石乔林, 李天阳, 张树丹, 等. 一种 4Mb 高速低功耗 CMOS SRAM 的设计[J]. 微电子学与计算机, 2005, 22(11): 63 - 70.
- [8] 郑曰, 李斌, 黄裕泉. 一种快速、低压的电流灵敏放大器的设计[J]. 微电子学与计算机, 2006, 23(6): 130 - 137.

## 作者简介:

王艺燃 男, (1984 -), 硕士研究生. 研究方向为超大规模集成电路设计.

于宗光 男, (1964 -), 博士, 博士生导师.

(上接第 146 页)

- [3] Petik A. Some aspects of using STL file format in CAE Systems[J]. Int Workshop CA System Technol., 2000: 80 - 86.
- [4] 千学明, 莫才松, 林晓萍. STL 数据的远程显示方法研究[J]. 微电子学与计算机, 2007, 24(7): 149 - 152.
- [5] 李巍毅, 张云, 周华明, 等. STL 内插错误的高效检查算法研究[J]. 电加工与模具, 2007(4): 41 - 43.
- [6] 严蔚敏, 吴伟民. 数据结构: C 语言版[M]. 北京: 清华大

学出版社, 2009: 253 - 256.

## 作者简介:

章雅娟 女, (1977 -), 硕士研究生, 讲师. 研究方向为计算机辅助设计与制造技术.

姜晓峰 男, (1972 -), 博士, 副教授. 研究方向为计算机辅助设计与制造技术、多媒体技术.