

文章编号 1000-5013(2006)03-0284-04

# STL 文件格式在反求造型中的应用

肖 棋 林俊义

(华侨大学机电及自动化学院, 福建 泉州 362021)

**摘要** 分析 STL 文件格式在现实中的应用情况, 提出在反求造型中, 使用 STL 文件格式的优势. 在了解 STL 文件格式的基础上, 以 STL 文件格式在反求工程中的曲面拼接的应用为例, 阐述 STL 文件格式在反求造型中的优缺点. 针对 STL 文件格式存在冗余量大的问题, 提出去除冗余量, 以及在 Visual C++ 平台上使用 OpenGL 快速显示 STL 文件格式三维实体的方法, 给出使用 OpenGL 命令进行三维实体模型变换的具体方法. 实验证明, 它可以使拼接过程的计算量大大减少, 也提高可视化曲面拼接的效率.

**关键词** STL 文件格式, 曲面拼接, 反求工程, OpenGL  
**中图分类号** TP 317.4 **文献标识码** A

目前, CAD 技术已日趋成熟, 市场上出现了许多大型的造型软件, 这些软件为计算机辅助设计及制造提供了很大的方便. 然而, 在一些特殊的场合, 这些造型软件也完成不了全部的工作. 随着人们对造型技术的深入研究, 以及计算机图像处理、数字化和光电等技术的日臻成熟, 反求技术正在迅速兴起. 反求工程<sup>[1]</sup>是把实物进行三维数字化处理, 然后把得到的三维离散数据作为初始素材, 进行三维 CAD 模型重构. 它由三维测量技术、三维重构技术和快速制造三部分组成. 典型的三维测量技术, 如三坐标测量仪、光学三维测量技术、立体视角测量方法等, 都为反求技术的发展提供了便利条件和实现途径. 由测量的三坐标数据重构曲面模型的算法已经比较成熟, 但数据合成部分还有待研究. 由于曲面反求技术在许多领域具有巨大的应用潜力, 所以曲面反求中测量数据合成(拼接)技术已成为一个热门的研究领域. 本文着重阐述 STL 文件格式在拼接技术中的应用.

## 1 STL 格式的实际应用

市场上流行的中、大型 CAD 造型软件都有一个自己的固定文件格式, 如 Pro/E 的 PART 文件、AutoCAD 的 DWG 文件等, 它们彼此之间的文件一直存在兼容性问题. 然而, 几乎所有的大型造型软件都具备了支持 STL 文件的接口, 这为 STL 文件的实用起到了很大的作用. 快速成形技术(RP)是 20 世纪 80 年代中期发展起来的一种全新的制造技术<sup>[1-3]</sup>, 其工作原理是基于逐层制造的过程, 能快速制造出任意复杂形状的零件. STL 文件格式已经成为快速成形领域中描绘三维实体模型的准标准文件格式, STL 文件是通过对 CAD 实体模型或曲面模型进行表面三角形离散化后得到的, 是一种由小三角形面片构成的三维多面体模型. 它代表的是一个有序的、正确的、唯一的 CAD 实体数据模型, 其文件格式中包含了每个三角形面片的 3 个顶点及其法向. 其 BNF 格式定义如下:

$$\begin{aligned} \langle \text{STL 文件} \rangle &::= \langle \text{三角形 1} \rangle \langle \text{三角形 2} \rangle \langle \text{三角形 3} \rangle \dots \langle \text{三角形 } n \rangle \\ \langle \text{三角形} \rangle &::= \langle \text{法向量} \rangle \langle \text{顶点 1} \rangle \langle \text{顶点 2} \rangle \langle \text{顶点 3} \rangle \\ \langle \text{法向量} \rangle &::= \langle X_n \rangle \langle Y_n \rangle \langle Z_n \rangle \\ \langle \text{顶点} \rangle &::= \langle X \rangle \langle Y \rangle \langle Z \rangle \end{aligned}$$

虽然, 如前所述许多 CAD 软件系统都具备了支持 STL 文件格式的模块, 但在实际生产中, 往往需

收稿日期 2005-10-21

作者简介 肖 棋(1966-), 女, 副教授, 主要从事工程图学及快速成形的研究. E-mail: xqxt@hqu.edu.cn

基金项目 福建省重大科技基金资助项目(2002H095)

要对已生成的 STL 模型进行拼接和分割等再处理,现有的 CAD 软件系统不能有效的完成此工作。因为,在此过程中存在从 STL 模型到 CAD 模型,再到 STL 模型的两次转换,会使得处理后的 STL 模型精度大大降低,甚至出现缺陷。所以,基于 STL 格式的三维模型拼接已经成为研究的热点。

## 2 STL 格式三维模型拼接中的优缺点

三维模型拼接技术也即为曲面面片的拼接。在曲面拼接的几种方法中,有一种简单而直观的方法——人机交互法。即在一个窗口中同时显示两个曲面面片,再对曲面面片通过人机交互的方式进行平移或旋转,使两曲面处于相对准确的位置,以便实现两曲面面片的拼接。如此重复,以完成整个实体的拼接。最后,把拼接完的实体数据输出,即完成了实体的曲面拼接。

要实现交互式曲面面片的拼接,首先必须将三维曲面在窗口中显示出来,而 OpenGL 被认为是高性能图形和交互式视景处理的标准。它提供了大约 120 条不同命令,用来定义 3D 物体和交互式 3D 应用的各种操作。在构造三维实体时,从共面性能考虑,它特别推荐使用三角形<sup>[4]</sup>。为了能更加真实地显示物体,用 OpenGL 进行三维实体显示时,必须建立光照模型,而光照效果是根据构造单元的法向矢量来计算的。所以,在采用三角形构造三维实体时,必须先知道三角形的法向矢量。从上面的 STL 文件格式中可以看出,STL 文件记录了构造三维实体的每个三角形面片的法向矢量及 3 个顶点坐标,而这正符合了 OpenGL 的光照建模要求。采用 OpenGL 进行 STL 格式实体的显示,有其独特的优势。

在曲面拼接中,三维图形的显示只为拼接过程提供可视化的效果。要完成拼接工作,必须进行多次的平移、旋转变换。根据 OpenGL 的平移与旋转原理可知,对一个三角形进行平移(旋转)一次,必须将三角形的 3 个顶点与平移(旋转)矩阵相乘一次。然而,由于 STL 文件的冗余数据非常大,一般情况下,STL 文件中不重复的顶点数  $V_n$  约为面片数  $F_n$  的一半<sup>[5]</sup>,这就造成了  $2.5 F_n$  个冗余的顶点数。即在每次平移(旋转)中,要多进行  $2.5 F_n$  次重复的计算,而从反求数据重构的 STL 格式曲面中,三角形的数目一般都比较多。这就造成了计算量的大量增大,影响了拼接效率。因此,必须快速去除 STL 文件中的冗余点,建立无冗余顶点的点表。

## 3 STL 面片冗余点的去除

STL 文件是以三角形为单元进行存储的,然而,其存放的三角形是完全无序的三角形面片,不存在拓扑关系,对后续的曲面拼接工作非常不利。所以有必要在进行去除冗余点的同时,建立拓扑关系的工作。去除冗余点的实质是,从 STL 文件中依次取出三角形,通过比较三角形顶点的坐标值,归并重复的点,从而建立一个无重复点的点表。去除冗余点最直接的方法是直接比较法。用这种方法建立点表需要极大的计算量,两万个面片的 STL 文件仅建立点表就需几个小时,是根本无法接受的。为了提高效率,快速地去掉冗余点,提出用查找速度较快的坐标排序二分法。下面分析坐标排序二分法对排序的要求。

(1) 当新的顶点读进来时,首先对  $Z$  坐标进行比较,所以  $Z$  坐标必须全部排序。(2) 当  $Z$  坐标相同时,须比较  $Y$  坐标,此时只需比较  $Z$  坐标相同顶点的  $Y$  坐标。所以只需对具有相同  $Z$  坐标的顶点进行排序和查找插入位置,很明显这比在所有顶点中排序查找快得多。(3) 当  $Z$  坐标和  $Y$  坐标都相同时,才须比较  $X$  值。同样,只需对具有相同该  $Z$  坐标和  $Y$  坐标的顶点进行排序和查找。这样大大减少了查找范围,提高了排序和插入的效率。采用坐标排序二分法建立点表有 3 个步骤。(1) 按三角形为单元,从文件中读入该三角形的 3 个顶点,逐点进行比较。(2) 判断点表中是否已存在该顶点。(a) 用二分法进行  $Z$  坐标的比较。(i)  $Z$  坐标相同的顶点不存在,转入(3)。(ii)  $Z$  坐标相同的顶点存在,转入(b)。(b) 在所有  $Z$  坐标相同的顶点中,比较  $Y$  坐标。(i)  $Y$  坐标相同的顶点不存在,转入(3)。(ii)  $Y$  坐标相同的顶点存在,转入(c)。(c) 在所有  $Z, Y$  坐标相同的顶点中,比较  $X$  坐标。(i)  $X$  坐标相同的顶点不存在,转入(3)。(ii)  $X$  坐标相同的顶点存在,新的顶点不加入点表中。(3) 向点表中插入新点(完成排序)。(a)  $Z$  坐标不同。用二分法查找比较时,即可同时找到新点插入的位置,向点表中插入新点的同时,也完成了  $Z$  坐标的排序。(b)  $Z$  坐标相同,  $Y$  坐标不同。在具有该  $Z$  坐标的顶点中,找到新点插入的位置,同时完成  $Y$  坐标的排序。(c)  $Z, Y$  坐标相同,  $X$  坐标不同。在具有该  $Z, Y$  坐标的顶点中,找到新点插入的位置,同时完成  $Y$

坐标的排序. 经过坐标排序二分法建立的点表结构, 如图 1 所示. 用坐标排序二分法建立无冗余的点表效率很高, 而且在搜索比较的同时, 具有相同坐标值 ( $Z$  或  $Z, Y$ ) 的顶点自动排在一起, 实现了自动分段功能, 提高了后续搜索效率. 实现上面的步骤, 需要一定的数据结构. 在 Visual C++ 中使用的数据结构:

```

Class Vertex3f : public Object
{
double x, y, z;
// 顶点三维坐标
void Vertex3f(double TX, double TY, double TZ)
{
x = TX;
y = TY;
z = TZ;
}
};
CArray<Vertex3f, Vertex3f> m_pVers; // 点表数组

```

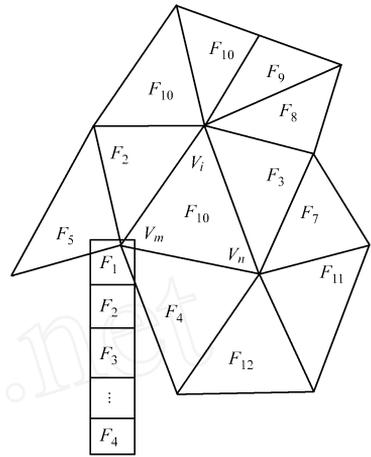


图 1  $V_m$  点的引用面

使用上述方法对所读入的 STL 文件进行处理后, 在去处冗余点的

同时也建立了点表. 表中坐标  $Z$  值按降序排列, 而每个  $Z$  值后有一个或几个  $Y$  值, 其中  $Y$  值也按降序排列; 同理, 每个  $Y$  值后有一个或几个  $X$  值, 其中  $X$  值也是按降序排列的. 这样  $Z$  值为主标志, 整个表按  $Z$  值的降序排列点的顺序,  $Y$  值为次标志, 表中的各段以  $Y$  值的顺序来排列.

### 4 OpenGL 图形显示与变换

基于 STL 文件格式的曲面拼接过程中, 利用 STL 文件中三角形面片之间的拓扑关系, 可以使拼接工作更为简单、方便. 在去除冗余顶点的同时, 需要建立起三角形面片间的拓扑关系. 在这里只建立两个基本的拓扑关系, 即点表和面表. 从点表中可以知道, 每个顶点所引用的所有面片; 从面表中可以知道, 每个面片所用到的具体顶点. 其他拓扑关系可以在这两个关系的基础上很容易建立起来. 面表中三角形面片的顺序就按读入的顺序, 还需要记录该面片的 3 个顶点在点表中的位置. 要建立点表, 需要在点的数据结构中记录该点所在的面片信息, 以及该点为所在面片中的第几个顶点. 在点的数据结构中增加变量, 即

```

CDWordArray m_FaceNum;          CDWordArray m_VertexInFace;
// 记录该点的引用面的数组      // 记录该点为引用面的第几个点的数组

```

这样所有的点都可以找到使用该点的面 (图 1), 而所有的面也都可以找到构成该面的点 (图 2). 建立起三角形面表后, 就可用 OpenGL 提供的三角形构造实体的方法, 来显示 STL 格式的三维实体图形, 如图 3 所示. 在 OpenGL 编程系统中, 用三角形构造实体所调用的命令格式<sup>[4]</sup>:

```

glBegin( GL_TRIANGLES );          glVertex3f( V2x, V2y, V2z );
glNormal3f( Nx, Ny, Nz );          glVertex3f( V3x, V3y, V3z );
glVertex3f( V1x, V1y, V1z );      glEnd();

```

要显示三维实体模型, 只要从面表中依次读入各个三角形的面片信息, 即法向量坐标和 3 个顶点坐标就可以了. 对于图形的变换操作 (平移、旋转等), OpenGL 采用矩阵变换, 即要变换的坐标值和矩阵相乘即可得到变换后的坐标值. 把所有变换集合于一个矩阵中, 作为视图矩阵, 如平移, 绕  $X$  轴旋转的矩阵

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos & \sin & 0 \\ 0 & -\sin & \cos & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

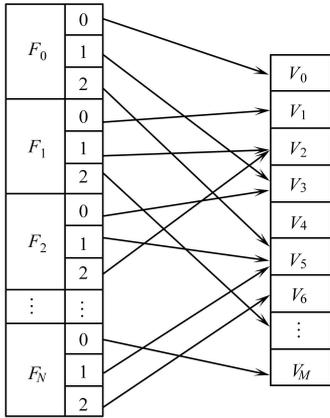


图 2 面表与点表的关系

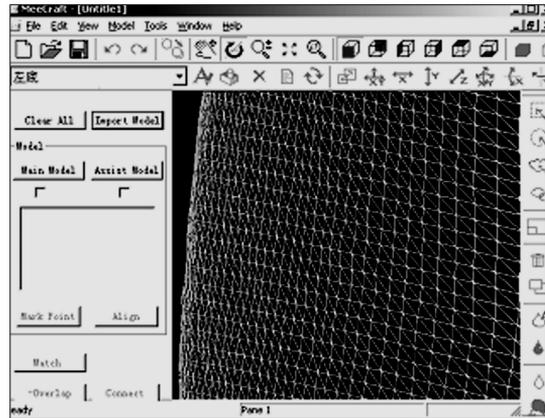


图 3 人脚的 STL 文件显示

式中,  $t_x, t_y, t_z$  分别为沿  $X, Y, Z$  轴的平移量,  $\theta$  为绕  $X$  轴的旋转角度. 变换实现过程中, 只需对点表中无冗余的点进行计算, 然后通过面表来提取所需点表中的点. 这样无需对每个三角形的 3 个顶点都进行计算, 大大减少了计算量, 也提高了效率.

### 5 结束语

本文讨论了人机交互法进行曲面拼接时, 基于 OpenGL 进行 STL 格式实体显示的优势, 分析了 STL 文件格式存在大量冗余数据对曲面拼接的影响, 提出采用分段二分法快速去除冗余数据. 整个思想方法应用于用 VC++ 6.0 开发的鞋楦拼接软件中, 取得了令人满意的效果.

### 参 考 文 献

- 1 张舜德, 魏正英, 卢秉恒. 逆向工程技术及其工业应用[J]. 甘肃工业大学学报(自然科学版), 2001, 27(2): 25 ~ 28
- 2 冯 伟. 快速成形工艺集成软件系统的研究与实现[D]. [学位论文]. 北京: 清华大学机械系, 1996. 12
- 3 肖 棋, 江开勇. 熔融沉积快速成形工艺控制研究[J]. 华侨大学学报(自然科学版), 1999, 20(1): 76 ~ 79
- 4 陈绪兵, 夏 华, 肖跃加, 等. 三维原型的 OpenGL 真彩光照方法[J]. 计算机辅助设计与制造, 2002, (3): 53 ~ 55
- 5 吴 斌, 段海波, 薛凤武. OpenGL 编程权威指南[M]. 北京: 中国电力出版社, 2001. 23 ~ 186

## The Application of STL File Format in Reverse Engineering

Xiao Qi Lin Junyi

(College of Mechanical Engineering and Automation, Huaqiao University, 362021, Quanzhou, China)

**Abstract** The application of STL file format in the actual circumstances was analyzed. The advantages of the application of STL file format in the reverse engineering were put forward. Based on the understanding of the STL file format and taking the case of the application of STL file format in the curve surface-splicing during the reverse engineering process, the advantages and disadvantages were set forth in detail. In accordance with the problem of the redundancy data in STL file format, methods to wipe off the redundancy data and to fast show three-dimensional (3-D) model of STL file format based on OpenGL in the Visual C++ were presented in this paper, and the usage of OpenGL order to bring about 3-D model entity model transformation was presented too. The experimental results revealed that theses methods can reduce the calculation and improve the efficiency of curve surface-splicing and provide a practical and fast path for 3-D reverse engineering.

**Keywords** STL file, curve surface-splicing, reverse engineering, OpenGL