

快速成形制造实体分层软件设计^{*}

江开勇 刘道远

(华侨大学机电工程系, 泉州 362011)

摘要 论述了快速成形制造技术中的 CAD 模型数据转换、实体分层、接口文件格式等一系列与 RP 技术有关的软件前置处理方法。着重介绍一种自行开发的基于 Auto CAD 软件平台与 C 语言结合的 CAD 实体模型分层算法与模拟显示的软件设计方法, 为自主开发 RP 软件技术作了有益的尝试。

关键词 快速成形, 接口文件, 实体分层, 模拟显示

分类号 TH 164

快速成形技术(Rapid Prototyping Technology)是近 10 年发展起来的一项高新技术, 与传统的成形理论相反, RP 技术提出了离散堆积成形的概念, 在人类制造史上开辟了一个崭新的天地。据预测, 它为制造业带来的影响, 不亚于国际范围内已广泛应用的数控技术^[1]。统计资料表明, 采用 RP+ RT (Rapid Tooling) 技术的模具制造时间和成本均为传统技术的 1/3。

快速成形的制造工艺与传统的加工方法截然不同, 传统的加工工艺是通过专用工具逐渐去除毛坯余量的“去材”过程加工出最终的零件, 快速成形则是通过把材料从无到有逐层堆积的“增材”过程得到原型。首先把三维 CAD 模型(电模型)分解成二维层片轮廓、加工轮廓边界以内的填充扫描轨迹信息, 形成 NC 代码后, 由计算机控制下, 通过扫描头的运动逐层堆积成形材料, 完成三维实物模型(物理模型)的建造。因此, 为了实现这样一种基于离散堆积概念的成形过程, 首先必须解决三维 CAD 模型的分层问题。

1 CAD 模型的数据转换格式

目前应用于 RP 技术的 CAD 模型的数据转换格式主要有: (1) RP 三维面片模型文件, 如 STL 文件, CFL 文件等; (2) CAD 三维数据文件格式, 如 IGES, DXF, VDA-FS, Universal Files 等; (3) 层片文件格式, 如 SLC, CLI, HPGL 等。下面仅介绍本文所采用的, 也是目前在 RP 设备上最常用的 STL 三维三角形表面面片文件格式和 CLI 通用层片文件格式。

1.1 STL 文件格式

STL 文件是美国 3D System 公司提出的一种用于 CAD 模型与 RP 设备之间的数据转换的文件格式, 目前已成为 RP 技术领域中的“准”工业标准。STL 文件是对 CAD 实体模型或曲面模型进行表面三角形面片离散化后, 得到的一种由许多小三角形面片拟合的三维多面体模

型. 它由文件名、每个三角形面片的法向量、内外环标识、每个三角形的 3 个顶点坐标逐行表示. 每个三角形面片信息以法向量(facet normal)开头, 以 endfacet 结束. 每个法向量以矢量端点的 3 个投影坐标表示, 矢量的模为 1. 顶点坐标以 Vertex 开头, 记录每个顶点的三维坐标值. STL 文件有 ASC 码和二进制码两种格式, 以下为该文件的 ASC 格式实例.

```
solid test.stl
facet normal - 1. 0000000e+ 000 0. 0000000e+ 000 0. 0000000e+ 000
outloop
vertex 3. 6000000e+ 001 1. 00000000e+ 000 5. 0000000e+ 000
vertex 3. 6000000e+ 001 1. 00000000e+ 000 1. 7000000e+ 001
vertex 3. 6000000e+ 001 3. 00000000e+ 000 1. 1000000e+ 001
endloop
endfacet
:
facet normal 8. 3146961e- 001 5. 5557023e- 001 - 2. 8473608e- 015
outloop
vertex 4. 8171573e+ 001 4. 1715729e+ 000 2. 1000000e+ 001
:
endsolid test.stl
```

STL 文件格式的最大特点是数据格式简单处理方便, 而且与具体的 CAD 造型系统无关. 任何具备 STL 文件接口的 CAD 系统, 均能与目前大多数的 RP 设备通讯, 所以已成为大家公认的 RP 数据转换标准. 目前有十几种商品化 CAD 软件具有 STL 文件接口, 如 Pro/Engineer, CADDs, I-DEAS, EUCLID, CADkey, CATIA, Strim 100, AutoCAD 13.0 等.

1.2 CLI 层片文件格式

CLI (Common Layer Interface) 通用层片接口文件格式是欧共体 BRITE-EURAM 快速原型技术项目提出并完善的一种层片接口文件, 与 STL 文件类似, CLI 文件也有 ASC 码和二进制两种格式, 但表示方法与 STL 文件完全不同. CLI 文件是用不同层的信息的叠加表示某个 CAD 实体模型. 每一层是介于两个平行截面之间的体, 具有一定的厚度和内外轮廓线. 内外轮廓线用多义线表示, 它定义了每一层内固态材料的边界. 正确的轮廓线应当是封闭的, 并且无自相交或与其它轮廓线相交的现象. CLI 文件的 ASC 码格式由头文件部分和几何文件两部分组成. 头文件部分由 \$\$HEADERSTART 开始, 以 \$\$HEADEREND 结束. 在头文件中可以记录所采用的计量单位, 文件创建日期, 总层数以及与用户数据有关的一些信息. 几何部分以 \$\$GEOMETRYSTART 开始, 以 \$\$GEOMETRYEND 结束, 记录所有层的信息, 其中每一层主要包括三项内容. (1) 层: \$\$LAYER/ Z . 其中 Z 为实数, 表示此层的高度值. 可以由两层的 Z 值之差确定每层的厚度. (2) 多义线: \$\$POLYLINE/ id , dir , n , p_{1x} , p_{1y} , ..., p_{nx} , p_{ny} . 其中 id 为整数, 用于标识同一个文件中包含的不止一个零件; dir 为整数, 表示多义线的方向, 即 0 为顺时针, 1 为逆时针, 2 为非闭合线(表示不是实体轮廓); n 为整数, 表示多义线中点的数量. (3) 填充线: \$\$HATCHS/ id , n , p_{1sx} , p_{1sy} , p_{1ex} , p_{1ey} , ..., $p_{n_{sx}}$, $p_{n_{sy}}$. 其中 id 意义同多义线, n 为整数, 记录填充线的数量. 每条填充线有 4 个参数, 即起点坐

标 p_{lx}, p_{ly} , 终点坐标 p_{lex}, p_{ley} . 以下为 CLI 文件的 ASC 码格式的一个实例.

```

$ $HEADERSTART                $ $HATCHS/ 0, 2, 10. 2, 10. 4, 12. 34, ...
$ $ASC                          $ $LAYER/ 5. 6
$ $Unit/ 0. 1                  $ $POLYLINE/ 0, 0, 200, 10. 23, 12. 34, ...
$ $DATE/ 062097                ...
$ $LAYERS/ 100                  $ $LAYER/ 15. 4
$ $HEADEREND                    $ $POLYLINE/ 0, 0, 200, 13. 23, 12. 34, ...
$ $GEOMETRYSTART                $ $HATCHS/ 0, 2, ...
$ $LAYER/ 5. 5                  $ $GEOMETRYEND
$ $POLYLINE/ 0, 0, 5, 1. 00, 2. 02 ...

```

2 实体分层算法与层面文件的生成

2.1 实体分层算法

对 STL 文件分层算法的核心是平面与三角形面片的求交, 下面介绍求交几个步骤: (1) 通过 CAD 软件的 STL 文件接口输入 STL 文件. (2) 给定实体在 Z 轴方向的最低点 Z_{\min} , 最高点 Z_{\max} . (3) 给定层厚 ΔZ . (4) 取分层高度 $Z = Z + \Delta Z$ (初始预置 $Z = Z_{\min}$). (5) 找到一个与 Z 平面相交的三角形 F_1 ; (6) 根据 F_1 的两个顶点 $V_1(x_1, y_1, z_1)$ 和 $V_2(x_2, y_2, z_2)$ 的坐标值, 按点斜式公式直接算出交点 $p_1(x_{p1}, y_{p1})$, 其中 $x_{p1} = x_1 + (z_{p1} - z_1)(x_2 - x_1)/(z_2 - z_1)$, $y_{p1} = y_1 + (z_{p1} - z_1)(y_2 - y_1)/(z_2 - z_1)$. (7) 类似(6)的算法, 求出 Z 平面与三角形 F_1 另一条边的交点 (x_{p2}, y_{p2}) . (8) 判断所有三角形面片是否全部完成求交运算, 是 结束; 否 返回步骤 (4). 在每层求交过程中, 都会碰到如何确定与 Z 平面相交的第一个三角形面片的问题. 这里采用全体搜索的方法^[1], 下面介绍其过程. 设三角形面片的顶点坐标为 $V_1(x_1, y_1, z_1)$, $V_2(x_2, y_2, z_2)$, $V_3(x_3, y_3, z_3)$, 截平面方程 $Z = h$ (h 为当前分层高度). 首先判断截平面与三角形面片是否相交. 若 $z_1, z_2, z_3 > h$, 或 $z_1, z_2, z_3 < h$, 则“三角形与截平面不相交”. 除此之外, 则必定是以下三种情况中的一种: (1) $z_1 = z_2 = z_3 = h$, 则三条边都在截平面上; (2) $z_1 = z_2 = h$, 或 $z_2 = z_3 = h$, 或 $z_1 = z_3 = h$, 则“一条边在截平面上”; (3) 若(1), (2) 都不满足, 则“三角形与截平面斜交”, 按点斜式公式求出交点坐标.

此算法适用于任何具备 STL 接口文件的 CAD 软件. 值得提出的是, 分层厚度的选择应根据成形精度要求和成形工艺确定. 另外, 三角面片的拟合数目直接决定成形表面的精度, 通过改变 AutoCAD 13.0 中的系统变量设置, 可以方便地改变三角面片的拟合数目.

2.2 通用层片接口(CLI)文件的生成

由求交算法得到的结果是许多杂乱无序的交点坐标的集合, 为了得到头尾相连的层片轮廓, 需要对某一层内的交点坐标重新排序, 按多义线形式写入 CLI 文件. 首先搜索所有交点中 x 值最小的交点作为多义线的起点坐标, 再判断共用此交点的相邻两条折线的斜率. 对于外圈, 取斜率小者为连线第一段, 多义线的走向为逆时针; 对于内圈, 取斜率大者为多义线第一段, 多义线走向为顺时针. 由此保证堆积材料总是处于边界轮廓线(即多义线)走向的左边. 层内所有后续的交点排序采用链表结构实现, 每个交点结构包含两个交点、一个指向下一个交

点结构的 NEXT 指针和一个取用情况标志 Used. 读入交点结构时, 标志置“0”, 排序时, 用过的交点标志置“1”. CLI 文件的形成过程: 先写入 CLI 文件的头文件, 层信息等, 然后与排序搜索过程同步从交点链表结构中依连线顺序写入交点的二维坐标值, 完成一条多义线的写入过程. 再用层结构链表进行循环, 直到每一层的多义线都写入 CLI 文件. 最后写入文件尾, 即得到一个完整的 CLI 文件.

3 STL 三维面片模型与 CLI 层片文件模拟显示

由于 Auto CAD13.0 只有 STL 文件输出接口, 没有输入接口. 为了显示经表面三角化后的三维多面体模型, 本文设计了 STL 面片模型与 CLI 层片轮廓的三维模拟显示软件. 其核心内容是解决对于任意的观察点位置, 重新定义位于世界坐标系中的原图形的坐标变换问题. 按照计算机图形学中的处理方法^[6], 这样的一种坐标变换过程, 可以通过映射、平移、旋转的组合变换完成. 这里假设视点坐标为 $P(a, b, c)$, 为了把观察图形转移到世界坐标系中, 需要经过五个方面的坐标变换过程. (1) 把观察面映射到世界坐标面上, 变换矩阵为 T . (2) 将视点平移到世界坐标系原点, 变换矩阵为 M_f . (3) 绕世界坐标系的 X 轴旋转, 将观察坐标系的 Z 轴转至世界坐标系 $X_w Z_w$ 平面共面. 记旋转角 α 则有 $\sin\alpha = -b / \sqrt{b^2 + c^2}$, $\cos\alpha = -c / \sqrt{b^2 + c^2}$, 旋转矩阵为 R_x . (4) 绕世界坐标系的 Y 轴旋转 β 角, 使视点坐标系的 Z 轴重合. 其中 $\sin\beta = a / \sqrt{a^2 + b^2 + c^2}$, $\cos\beta = \sqrt{b^2 + c^2} / \sqrt{a^2 + b^2 + c^2}$, 旋转矩阵为 R_y . (5) 再绕世界坐标系的 Z 轴旋转 ν 角, 使观察坐标系与世界坐标系的 Y 轴重合. 其中, $\cos\nu = bu / \sqrt{b^2 u^2 + a^2 c^2}$, $\sin\nu = -ac / \sqrt{b^2 u^2 + a^2 c^2}$, $u = \sqrt{a^2 + b^2 + c^2}$, 旋转矩阵为 R_z .

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad M_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a & -b & -c & 1 \end{pmatrix},$$

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_y = \begin{pmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_z = \begin{pmatrix} \cos\nu & \sin\nu & 0 & 0 \\ -\sin\nu & \cos\nu & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

坐标组合变换矩阵为

$$\text{Multi} = T \times M_f \times R_x \times R_y \times R_z.$$

4 结果与结论

图 1, 图 2 分别为表面三角形多面体模型显示和层片轮廓模拟显示结果. 证实了本文所设计的分层算法和模拟显示软件的正确性.

采用 STL 文件格式作为 CAD 显示造型系统与 RP 设备的接口文件, 具有算法简单、通用性强的特点. 任何具备 STL 文件输出接口的 CAD 软件都可以作为本文所设计的算法程序的造型平台. 由于 RP 技术发展至今仅 10 来年时间, 而且在技术开发和研究方面一直是 RP 设备开发商占据主导地位, 技术保密非常严格, 自主开发我国的 RP 技术, 包括软件技术是

发展我国 RP 技术的一项战略性任务. 本文所做的工作只是一个初步、同时又是非常有意义

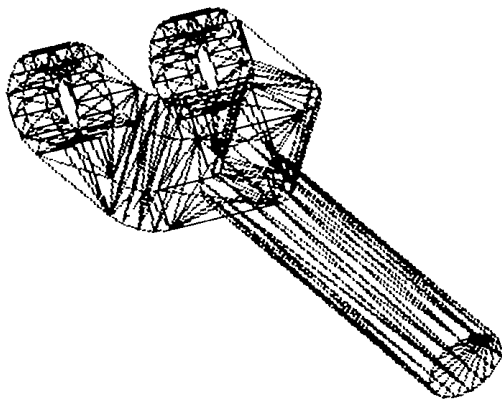


图 1 三角形多面体表面

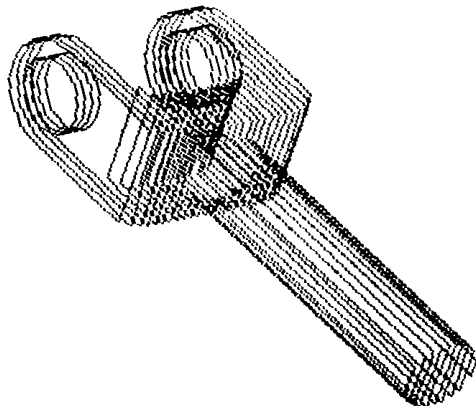


图 2 层片轮廓模拟

的尝试. 在此基础上, 如何建立 STL 文件格式的拓扑信息, 压缩数据冗余, 加快处理速度, 提高对 STL 文件的校验纠错能力将是本文今后进一步研究的内容. 此外, 直接分层算法, 如利用 Auto CAD 的三维图形的剖切功能^[6,5]等, 都将能成为今后 RP 软件技术的研究方向.

参 考 文 献

- 1 机械工业部科技信息研究院科技进步课题组. 先进制造技术(三). 机电信息, 1996, (4): 11 ~ 12
- 2 周雄辉, 陈普仁, 曹德祥. 快速原型制造系统与 CAD 系统接口软件开发. 模具技术, 1996, (6): 3 ~ 8
- 3 孔宪庶, 刘仁杰. 计算机绘图. 沈阳: 辽宁科学技术出版社, 1996. 1 ~ 112
- 4 李占才, 刘春燕. AutoCAD 三维图形剖切功能的实现. 计算机应用与软件, 1994, (4): 35 ~ 39
- 5 Sashidhar G, Richard H C. Direct generation of contour files from constructive solid geometry representations. Solid Freeform Fabrication Proceedings, 1993, (9): 291 ~ 302

Software design of Entity Layering in Rapid Prototype Modeling

Jiang Kaiyong Liu Daoyuan

(Dept. of Mech. & Electr. Eng., Huaqiao Univ., 362011, Quanzhou)

Abstract Concerning the rapid prototype modeling technique, a discussion is made on a series of software prefix processing methods, covering data conversion, entity layering, and interface file layout of CAD model. Emphasis is placed on the layering algorithm of CAD entity model based on the combination of Auto CAD software platform and C language; and the software design method displayed by simulation. This self-developed work is beneficial to the self-developing of quick forming software technique.

Keywords rapid prototyping, interface file, entity layering, display by simulation