

# 多面体的消隐绘图处理

张全伙

(计算机科学(电脑)系)

## 摘 要

隐藏线的消除是计算机绘图中的一个重要问题。本文介绍了在 PDP-11/34A 计算机驱动的 DP-3 绘图仪上实现的一个消隐绘图系统。本文首先给出多面体的定义,接着着重叙述了多边形的确定、预处理,隐藏线的寻找、消除的原理和方法,然后给出消隐绘图的实例。最后,对本系统作一简单的评价和说明。

## 一、问题的提出

我们看一个物体时,它的某些轮廓线常被前面的物体遮住而看不见,我们把这些看不见的轮廓线称为隐藏线,要画一个形象逼真的立体图形,应把隐藏线消去不画。作者在 PDP-11/34 计算机驱动的 DP-3 绘图仪上实现了一个用 FORTRAN 语言书写的消隐绘图系统,用户只要给出顶点坐标和棱(用顶点编号给出),该系统就能自动地画出多面体组合的消去了隐藏线的立体图形。所谓多面体,是由有限个多边形拼合而成的,这些多边形叫多面体的“面”,它们的边叫多面体的“棱”,而它们的顶点叫多面体的“顶点”。其拼合方式必须满足下面四个条件:

- (1) 它的每两个顶点可以由它的一些棱所组成的折线连接起来;
- (2) 它的每两个面或者没有公共点,或者恰有一个公共顶点,或者恰有一条公共的棱;
- (3) 它的每条棱恰是它的两个面的公共棱;
- (4) 它的每个顶点都是锥形的顶点,即每一个顶点处的棱和面可记为

$l_1, A_1, l_2, A_2, \dots, l_n, A_n, l_1$

这一串中的棱和面都以这顶点为一个顶点,这  $n$  条棱两两不同,这  $n$  个面两两不同,每个面都以左右邻的棱为它的两条边。例如,图 1 的顶点  $A_1$  处的这样的一串是:

$A_1A_2, A_1A_2A_3, A_1A_3, A_1A_2A_4A_4, A_1A_4, A_1A_4A_5A_2, A_1A_2.$

本文1987年1月22日收到。

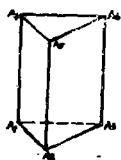


图 1

所谓多面体的组合,是指可以有一个以上的多面体,但不能有公共顶点——两个顶点的空间位置可以重合,但必须赋予不同的编号,作为两个顶点来处理。

## 二、消隐原理和方法

我们绘制这种多面体透视图的步骤是:先将多面体作透视变换,然后在投影轴垂直于投影平面的情况下计算隐藏线,检出隐藏垂线后最后把可见的线条画出。

为了检出隐藏线,就必须知道多面体的表面是由多少个多边形组成的,而每个多边形又是由哪几条边组成的。由于一个多边形可以是凸的也可以是凹的,凹的多边形处理起来比较麻烦,我们可以把凹的多边形分割成两个或多个相邻的凸多边形来表示。系统的流程简图如图2所示。

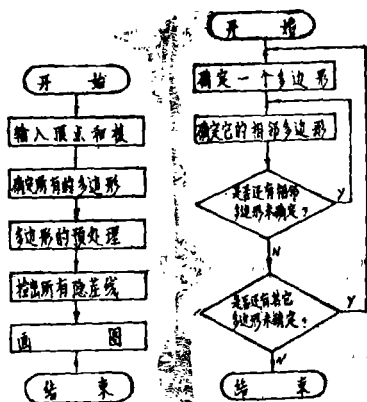


图 2 系统流程简图

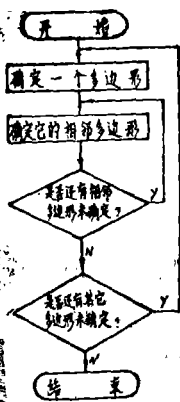


图 3 确定所有多边形流程图

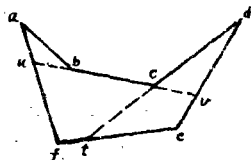


图 4 凹边和切割边示意图

### 1. 多边形的确定

我们所考虑的多面体可分解为一系列的顶点和棱。然后将所有顶点收集起来,按它们的编号顺序存放各顶点的坐标,这部分数据称顶点表。同时,也将所有棱收集起来,每条棱用它的两个顶点编号来表示,这部分数据称为边表。根据顶点表和边表,就可以确定一个完整的多边形,进而确定多面体的所有多边形。

一个多边形的确定:由于表面都是多边形,因此可由两条相邻的棱(我们称之为“原边”和“辅边”)确定一个平面的方程。然后,根据边表,从“原边”开始,依次取出与前一条棱相邻的且在此平面上的棱,直到又回到“原边”为止。这样,一个完整的多边形就确定了。

确定所有的多边形:我们用找相邻平面的方法来确定所有的多边形,其流程图如图3所示。

示。首先,任取两条相邻的棱确定第一个平面,为避免一个平面被重复确定,务必记住已确定的平面所含的边。如果一个多边形以某条棱作为它的一条边,我们就说这条棱被这个多边形“用过”了。在确定一个多边形的过程中对所有它“用过”的棱打上使用标记,同时,记下最后一条以前未打上标记的棱,以作为确定一个新多边形的“原边”。显然与选为新“原边”关联的两个多边形还有一个多边形尚未确定。我们找出一条与新“原边”相邻的且未打上标记的棱作“辅边”,这样,这个作为相邻新多边形就可以确定了。若它的所有相邻棱都已打上标记,则找出一条与它不同属于任一已确定的多边形的相邻棱来作“辅边”,然后重复上述一个多边形的确定过程。若这样的棱找不到,则说明多面体的所有多边形都已被确定出来了。

## 2. 多边形的预处理

在所有的多边形确定之后,如果其中某些多边形是凹的,则把它分割成两个或多个相邻的凸的多边形来表示。一个多边形称为是凸的,是当这个多边形位于它的每一条边所决定的直线的同一侧,否则,就称这个多边形是凹的。也就是说,一个凹多边形至少含有这样一条边,使得该多边形位于它所决定的直线的异侧,我们称这样的边为凹边。显然,凹边所决定的直线必与不和它相邻的边有交点,我们可以据此判定一条边的凹凸性,进而判定一个多边形的凹凸性。若一个多边形的所有边都是非凹的,则该多边形就是凸的了。

凹多边形的分割:选取这样一条凹边,它和不相邻边的交点只有一个,我们把它称为“切割边”,以它所决定的直线为界,把原多边形一分为二。对于和不相邻边的交点多于一个的凹边,不宜选取为“切割边”,因为它把原多边形分割成多于两个的多边形,处理起来较困难。如图4,边 $\overline{ab}$ 、 $\overline{bc}$ 和 $\overline{cd}$ 都是多边形 $abcde$ 的凹边。但 $\overline{bc}$ 所决定的直线对应的交点有 $u$ 、 $v$ 。它将多边形 $abcde$ 分成 $abu$ 、 $cdv$ 和 $uv$ 三个,因此不宜选为“切割边”。而 $\overline{ab}$ 、 $\overline{cd}$ 可任选一条作为“切割边”。这样分割出来的两个新多边形仍有可能是凹的,如图4中,以 $\overline{cd}$ 为“切割边”分割后的多边形 $abctf$ 仍是凹的,还要如此反复继续分割,直到最后将所有凹多边形都分成凸的为止,其流程如图5所示。

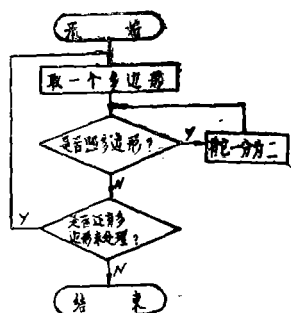


图5 多边形的预处理流程简图

## 3. 隐藏线的寻找、消除

多面体的所有表面多边形确定之后,我们按相邻顺序存放各多边形的顶点序号形成面表,然后根据边表和面表来计算隐藏线。方法是:对边表中的每条棱分别同面表中的所有多边形进行比较,看它是否被某些多边形遮住,遮住多少,遮住的部分就是隐藏线,把它从边表中消去。若棱整个被遮住则消去整条棱,若部分被遮住,则将该棱分成可见部分和隐藏部分(可能有若干段),然后用可见的部分来代替边表中的该棱。由此可见,消隐过程中边表是不断变化的,而面表却是一成不变的,其流程如图6所示。

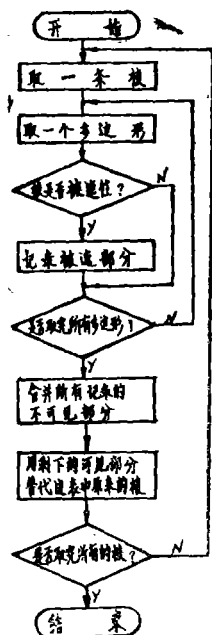


图 6 消隐流程图

如何判断一条棱是否被一个多边形遮住? 显然它取决于视点的位置。根据透视投影的原理, 我们从视点出发, 选取面表中的任一个多边形 (如图 7 的平面  $abc$ ), 经过平面  $abc$  的各顶点把视线投向屏幕, 形成一个以视点为顶点, 平面  $abc$  的像  $a'b'c'$  为底面的棱锥体, (见图 7,  $v-a'b'c'$ )。平面  $abc$  可看成锥体的一个截面, 锥体在平面  $abc$  和它的像  $a'b'c'$  (即底面) 之间的部分就是多边形  $abc$  的遮隐域, 记为  $\Omega$ 。显然, 落在  $\Omega$  中的线段都将被多边形  $abc$  遮住而成为隐藏线。一条棱是整个或部分成为隐藏线, 就看它是否整个或部分落在  $\Omega$  中。例如如图 7 中的线段  $de$  穿过  $\Omega$  域, 它与  $\Omega$  交于点  $f$  和  $g$ , 线段  $de$  的像  $d'e'$  被截成三段  $d'f'$ 、 $f'g'$  和  $g'e'$ , 其中  $f'g'$  是  $fg$  的像, 它是隐藏线, 而  $d'f'$  和  $g'e'$  是可见部分。显然, 每个面表中的多边形都有对应的  $\Omega$  域, 寻找隐藏线就是找出棱落在  $\Omega$  域中部分线段的像, 消去了隐藏线, 剩下的就是可见部分。

一条棱与  $\Omega$  域交点的求法: 我们注意到若棱整个落在  $\Omega$  内或整个落在  $\Omega$  外, 则它与  $\Omega$  无交点。因此, 我们通过求棱所决定的直线与  $\Omega$  的交点, 然后再确定交点与棱端点的关系。若两个交点 (必有两个) 同在棱任一端点的外侧, 则棱整个在  $\Omega$  外部; 若同在棱两端点的内侧, 则棱整个在  $\Omega$  内部; 若其中一个交点在棱两端点之间, 则棱的一部分在  $\Omega$  中。现在, 我们把棱所在直线和面表中的一个多边形的一条边投影到屏幕上, 然后求出它们的像的交点 (如图 7 中的  $f'$  点), 过交点和视点确定一视线, 求出视线与直线的交点  $P_1$  (如图 7 中的  $f$  点) 但是注意到直线可能位于多边形的前边, 就是离视点更近的位置, 因此还应求出视线与多边形的边的交点  $P_2$  (如图 7 中的  $f''$  点)。然后由视点到  $p_1$ 、 $p_2$  的距离来确定二者的前后关系。若  $vp_1 > vp_2$ , 则直线穿过  $\Omega$ , 否则直线位于多边形的前边。如果交点  $p_1$ 、 $p_2$  分别在视线上视点的两侧, 那就不能用它们与视点距离来判定前后关系, 此时, 我们总认为直线穿过  $\Omega$  域。

下面, 我们用数学方法来描述上面过程的处理思想。

一条棱及其所决定的直线可以表示为

$$P(s) = P_0(1-s) + P_1s$$

其中,  $0 \leq s \leq 1$ ,  $p_0$ 、 $p_1$  是棱的两个端点。对任一  $s$ ,  $-\infty < s < +\infty$ ,  $p(s)$  就是棱所在直线上的一点。下面来确定棱  $p(s)$  是否被面表中的多边形  $F$  遮住, 我们取多边形  $F$  的一条边  $Q_i(e)$ , ( $i = 1, 2, \dots, k$ ),  $k$  为多边形  $F$  的边数。求出  $p(s)$  的像  $p'(s)$ ,  $Q_i(e)$  的像  $Q_i'(e)$ , 然后

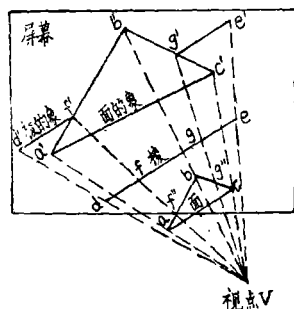


图 7 隐藏线示意图

再求出 $p'(s)$ 与 $Q'_i(e)$ 的交点。亦即,由方程

$$p'(s_i) = Q'_i(e_i)$$

求出 $s_i, e_i$ 。具体地说,设 $p'(s)$ 通过点 $(u_1, v_1), (u_2, v_2)$ ,  $Q'_i(e)$ 通过点 $(\zeta_1, \eta_1), (\zeta_2, \eta_2)$ , 则直线 $P'(s)$ 和 $Q'_i(e)$ 的交点方程为

$$\frac{X - u_1}{u_2 - u_1} = \frac{Y - v_1}{v_2 - v_1}$$

$$\frac{X - \zeta_1}{\zeta_2 - \zeta_1} = \frac{Y - \eta_1}{\eta_2 - \eta_1}$$

若矩阵

$$\Delta = \begin{bmatrix} u_2 - u_1 & v_2 - v_1 \\ \zeta_2 - \zeta_1 & \eta_2 - \eta_1 \end{bmatrix}$$

是奇异的,则或者棱 $p(s)$ 和面 $F$ 的边平行,或者它们之中某一个或两个都投影成一个点。不论哪种情况,棱都不必再和这条边比较了,因为当棱投影成一点时,此棱应从边表中消去。而对于另一种情况,它的可见性可由与其它边的比较而得到确定。

若 $\det \Delta \neq 0$ , 则交点存在,并由下式确定:

$$s_i = \frac{(\eta_2 - \eta_1)(\zeta_1 - u_1) - (\zeta_2 - \zeta_1)(\eta_1 - v_1)}{\det \Delta}$$

$$e_i = \frac{(v_2 - v_1)(\zeta_1 - u_1) - (u_2 - u_1)(\eta_1 - v_1)}{\det \Delta}$$

显然,若 $e_i < 0$ 或 $e_i > 1$ ,则棱与边的像无交点;若棱与多边形 $F$ 所有边的交点都呈 $e_i < 0$ 或 $e_i > 1$ 状态,则棱不会被多边形遮住,是可见的。

若 $0 \leq e_i \leq 1$ , 则说明由视点及 $Q'_i(e_i)$ 所确定的视线与 $Q_i(e)$ ,  $p(s)$ 都有交点( $0 \leq e \leq 1, -\infty < s < +\infty$ ), 就是 $p(s_i)$ ,  $Q_i(e_i)$  ( $0 \leq e_i \leq 1$ )。

下面就来确定 $p(s)$ 是否穿过 $\Omega$ 域,若是,则把交点 $p(s_i)$ 记录下来,为此,首先判断 $p(s_i)$ 与 $Q_i(e_i)$ 是否在视线上视点的两侧,若是,如前面约定,总认为 $p(s)$ 穿过 $\Omega$ ,记录 $p(s_i)$ 。(由于 $Q_i(e_i)$ 必在视点前方,因此只要判断 $p(s_i)$ 是否在视点后面就可以了。)否则,求视点 $v$ 与 $p(s_i)$ 及 $Q_i(e_i)$ 的距离差,它们由下面式子确定

$$d_i = P_x(s_i) - Q_{ix}(e_i)$$

其中下标 $x$ 表示取 $x$ 坐标。

若 $d_i > 0$ , 则 $p(s)$ 不穿过 $\Omega$ 域,  $p(s_i)$ 不记录;

若 $d_i = 0$ , 则应判断棱 $p(s)$ 是否在 $F$ 所决定的平面上。为此,取 $F$ 的三个顶点 $a, b, c$ ,

令

$$f_j = \begin{vmatrix} p_{jx} - a_x & p_{jy} - a_y & p_{jz} - a_z \\ b_x - a_x & b_y - a_y & b_z - a_z \\ c_x - a_x & c_y - a_y & c_z - a_z \end{vmatrix}$$

其中 $j = 0, 1$ ,  $p_0, p_1$ 为棱 $p(s)$ 的两个端点。若 $f_0, f_1$ 均为0, 则棱 $p(s)$ 与 $F$ 共面, 否则, 记录 $p(s_i)$ ; 若 $d_i < 0$ , 则 $p(s)$ 穿过 $\Omega$ 域, 记录 $p(s_i)$ 。

一旦对一物体的所有棱边计算出全部应予记录的 $s_i$ , 则棱 $p(s)$ 的隐藏情况就由下述的 $S_j$ ,  $S_k$ 完全确定, 此处

$$\left. \begin{array}{l} S_j = \min s_i \\ S_k = \max s_i \end{array} \right\}$$

通过考察 $S_j$ 和 $S_k$ 的值, 就可求出棱边的隐藏部分. 若 $S_k \leq 0$ 或 $S_j \geq 1$ , 则整个棱是可见的; 否则, 隐藏线部分存在, 并由下述的 $\bar{S}_j$ 和 $\bar{S}_k$ 来决定

$$\bar{S}_j = \begin{cases} S_j & 0 \leq S_j < 1 \\ 0 & S_j < 0 \end{cases}$$

$$\bar{S}_k = \begin{cases} S_k & 0 < S_k \leq 1 \\ 1 & 1 < S_k \end{cases}$$

我们注意到, 若 $\bar{S}_j = \bar{S}_k$ , 则整个棱是可见的, 当一个多边形经映象后退化成一条直线时就发生这种情况.

### 三、消隐绘图实例

以下图形都是由用户给出各顶点及棱, 系统按前述方法消去了隐藏线, 然后绘出在不同视点的立体图, 如图 8、9 所示.

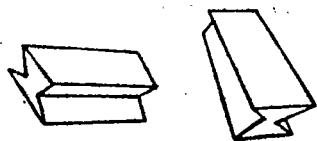


图 8 两边凹进去的长方体



图 9 房子

### 四、结 束 语

立体图形隐藏线的消去是计算机绘图中比较复杂的问题, 目前, 已有多种有效的消隐方法, 但都各有其特点, 难于说出哪一种方法最好, 也许还有更好的方法等待人们去寻找. 下面, 对本系统作一些简要的评价和说明.

作者认为本系统比较成功地实现了多面体在各种视点位置的透视图的消隐处理. 它为用户提供了两个入口, 用户既可直接输入物体的各顶点和棱, 也可直接输入多面体表面的多边形. 这样, 如果希望某个多边形不遮住其它棱, 是透明的, 只要不把它放入面表就可以了. 前面我们看到, 系统在确定一个多边形时, 只要在同一平面上的若干条棱首尾相连构成封闭图形, 就会被认为是一个多边形而被收集到面表中. 然而实际上物体可能并不存在这样一个表面多边形, 例如图 9 中的  $abcd$ , 就属于这种情况. 此时, 系统把  $ab$ 、 $bc$ 、 $cd$  和  $da$  作为两条棱的重合处理, 即  $a$ 、 $b$ 、 $c$  和  $d$  四个顶点都分别赋以两个编号, 作为重合的两个顶点输入. 由于

边表是变化的, 如果用户希望把隐藏线画成虚线, 只要把检出的隐藏线用虚线去替代, 并存放在边表中就可以了。而且, 每条边只用它的两个端点编号给出, 既灵活又节省许多存储单元。

### 参 考 文 献

- (1) 江泽涵, 多面形的欧拉定理和闭曲面的拓扑分类, 人民教育出版社, (1962)。
- (2) 杨学平, 计算机绘图, 电力工业出版社, (1980)。
- (3) 江涛, 计算机绘图, 湖南科技出版社, (1985)。
- (4) [法]R. 多尼著, 田宝华译, 微计算机BASIC语言科学绘图50例, 电子工业出版社, (1985)。
- (5) [美]S. Harrington著, 高福文等译, 计算机图形学, 北京师范大学出版社, (1985)。
- (6) Harry K. J., Microcomputer Graphics and Programming Techniques, published by Van Nostrand Reinhold Company Inc, (1982)。

### 摘 要

## Plotting a Closed polyhedron with Hidden Lines Eliminated

Zhang Quanhua

### Abstract

To eliminate hidden lines is an important part of computer plotting. This paper introduces a blanking plotting method by which hidden lines would be eliminated. It has been performed successfully on DP-3 plotter, with PDP-11/34A as its host computer.

The definition of a closed polyhedron is given at first; and then, the determination of its surface, the preprocess, the searching of hidden lines, and the principles and method of their elimination, are described; and finally, an example of blanking plotting is given for illustration.