计算机图形学

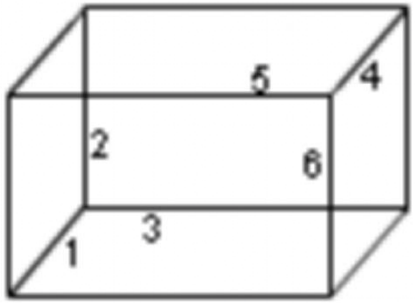
（第七讲）

信息学院　郭浩

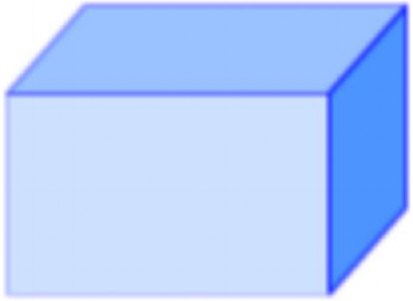
第七讲 可见面判别算法（消隐）

在许多计算机图形系统中，通常用多面体来表示三维物体。在计算机的显示器上输出物体的图形时，必须经过某种投影变换，将三维几何信息转换成为能够被显示器接收和输出的二维几何信息。由于投影变换失去了深度信息，这可能导致图形理解的二义性。

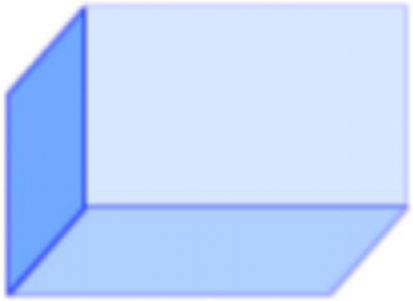
长方体线框图的平行投影的二义性



a【线框图】



A



B

要消除二义性，在显示时就必须消除物体被遮挡的（即不可见的）线或面，通常称作消除隐藏线和隐藏面，或简称为消隐。

消隐的结果与被观察的物体有关，也与视点的位置有关。

学习目标

——按照消隐算法的工作空间分类,消隐算法大致有哪几类？各有何特点？

——了解常见的可见面判别算法

内容

7.1 可见面判别算法的分类

通常按照消隐对象的不同，有两类消隐问题：

线消隐（Hidden-line removal）

面消隐（Hidden-surface removal）

如果按照消隐的工作空间进行分类，可以将消隐算法分为：

1）对象空间（Object Space）消隐算法（物空间算法）

2）图像空间（Image Space）消隐算法（像空间算法）

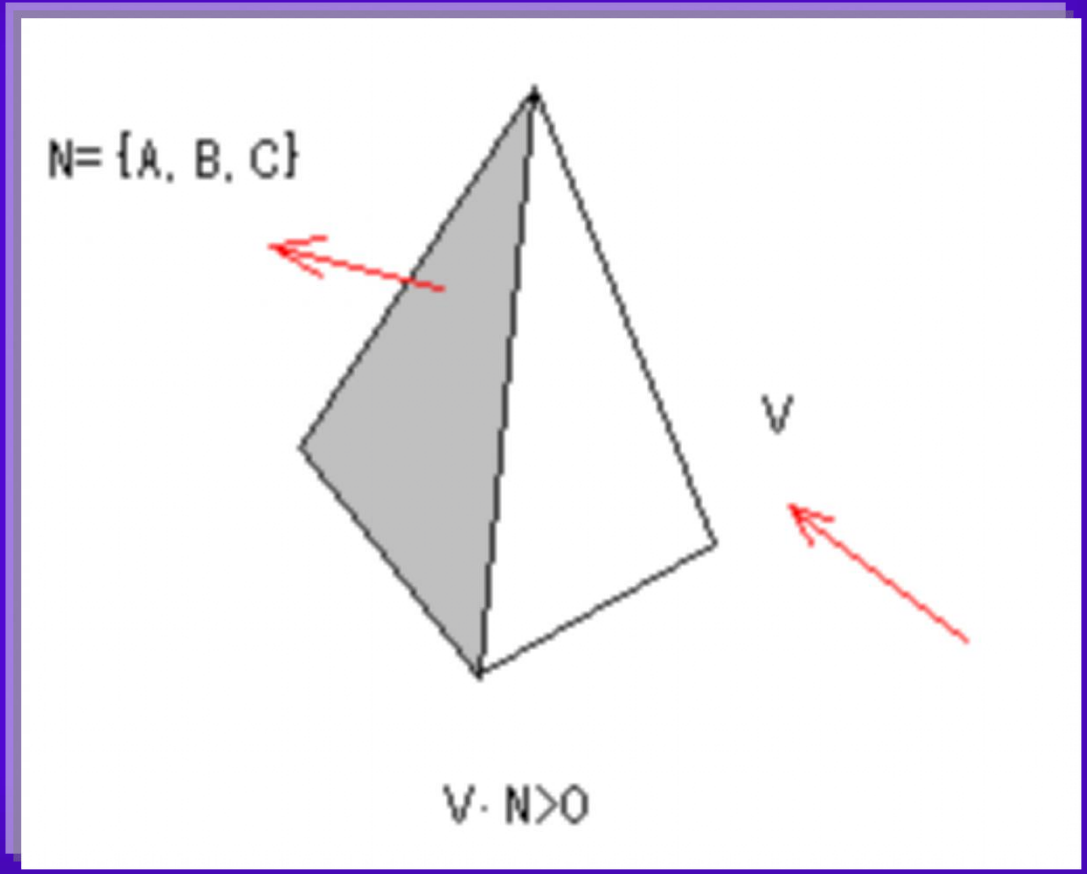
物空间算法

物空间算法将场景中的各个物体和物体的各个组成部件相互进行比较，从而最终判定出哪些面是可见的。

像空间算法

像空间算法则在投影平面上逐点判断各个像素所对应的可见面。

7.2 后向面判别



如果观察方向平行于观察坐标系中的 Z_v 轴，

则 $V=\{0,0,V_Z\}$，且

$$V \cdot N = V_Z C$$

这时只需要检查法向量 N 的 Z 分量 C 的符号就可以快速判别后向面

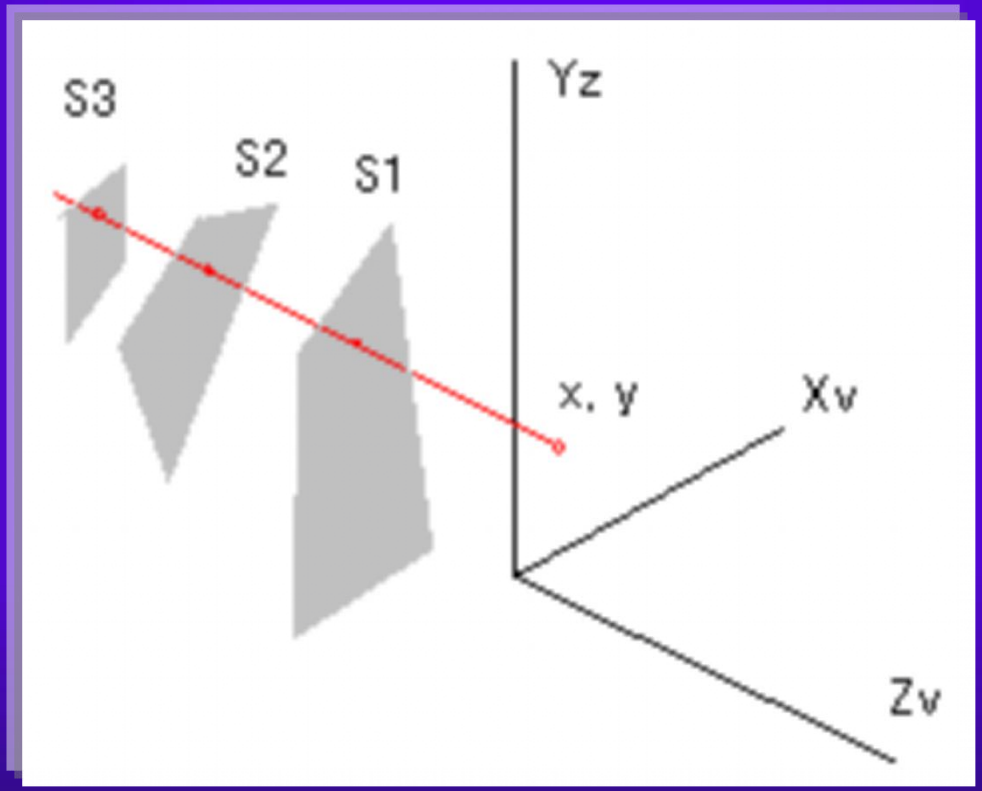
1）在一个沿着 Z_v 轴反向观察的右手观察系统中，

若 $C<0$，则该多边形为后向面；

2）$C=0$ 时，观察方向与该面向切，该面也是后向面。

7.3 深度缓存算法

对投影平面上每一个像素所对应的表面深度进行比较。



随着物体描述转化为投影坐标，多边形面上的每一个点(x,y,z)均对应于观察平面上的正交投影点(x,y)，物体深度比较可以通过比较 z 值来实现。

算法关键：

尽快计算出多边形内各点的深度值，这需要考虑并应用点和点之间位置相关性。

设某个多边形所在的平面方程为：

$ax+by+cz+d=0$，

若c不为0，则

$z=-(ax+by+d)/c$。

在点(x_i,y_i)处，

$z_i=-(d+ax_i+by_i)/c$。

而在点(x_{i+1},y_i)处

$z_{i+1}=-(d+ax_{i+1}+by_i)/c$

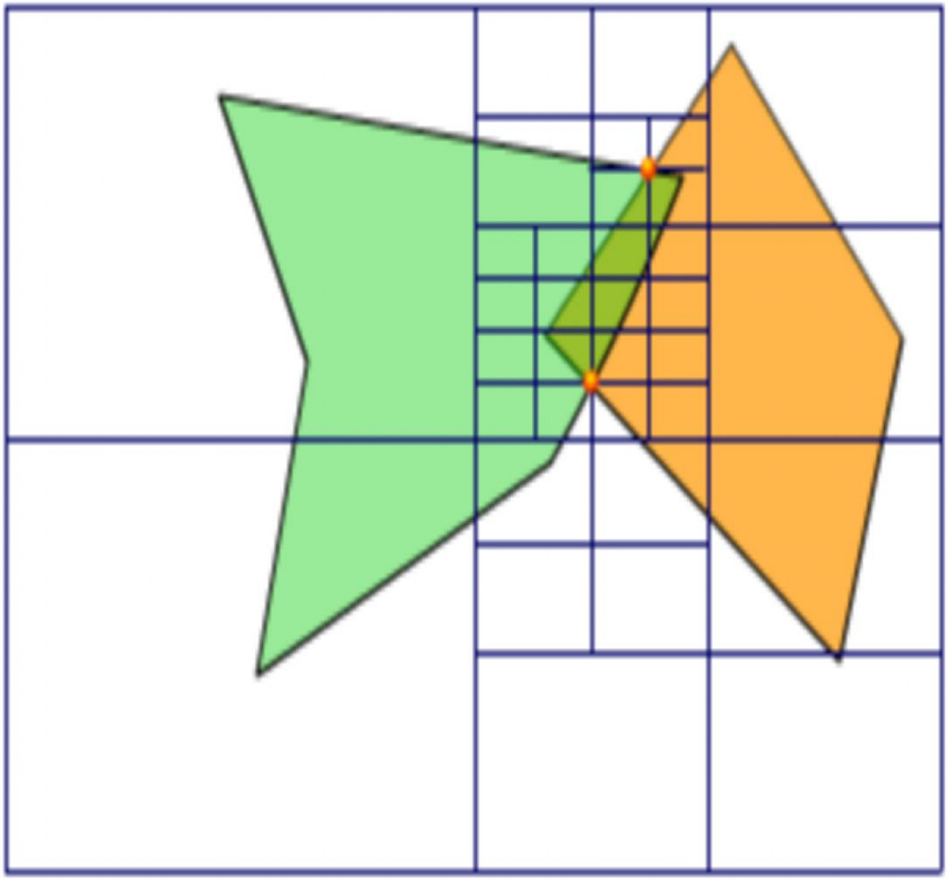
因为逐点扫描时，$x_{i+1}=x_i+1$，所以

$$\Delta z_x=z_{i+1}-z_i=-a/c,(c\neq 0)$$

7.4 区域细分算法

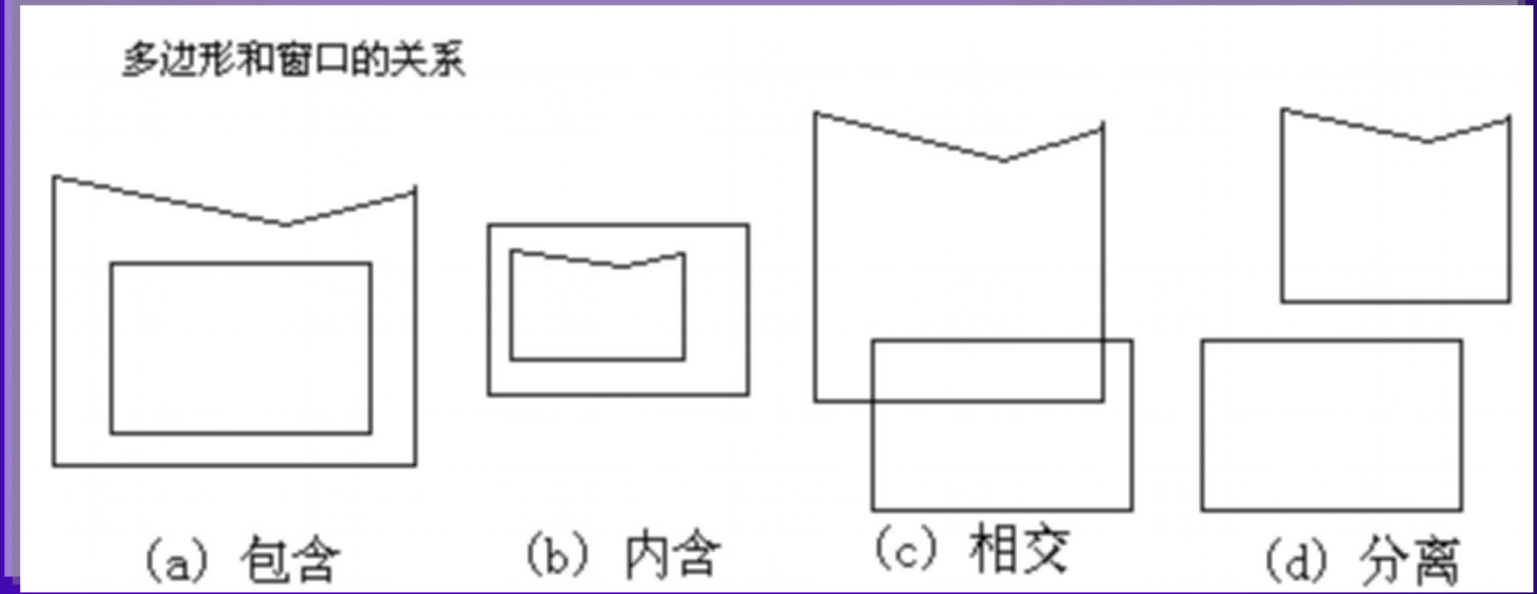
区域子分算法采用分而治之的消隐策略。其基本思想是：把物体投影到全屏幕窗口上，然后递归分割窗口，直到窗口内目标足够简单，可以显示为止。

区域子分过程



该算法最初把初始窗口取作整个屏幕，将场景中的多边形投影到窗口内。如果窗口内没有物体，则按背景色显示；若窗口内只有一个面，则把该面显示出来。否则，把窗口等分成四个子窗口，然后对每个子窗口重复上述过程。

具体来讲，每一次子分，都要把要显示的多边形和窗口的关系作一判断，多边形和窗口的关系有以下四种：



在多边形和窗口的关系确定后，满足下列条件的窗口可直接绘制出：

1） 所有多边形都和窗口分离，把窗口填上背景颜色；

2） 只有一个多边形和窗口相交，或只有一个多边形包含在窗口中，先把窗口填上背景色，然后再对窗口内的多边形部分用扫描线算法填上相应颜色；

3） 只有一个多边形和窗口相交，这个多边形把窗口包含在内，把整个窗口填上该多边形的颜色；

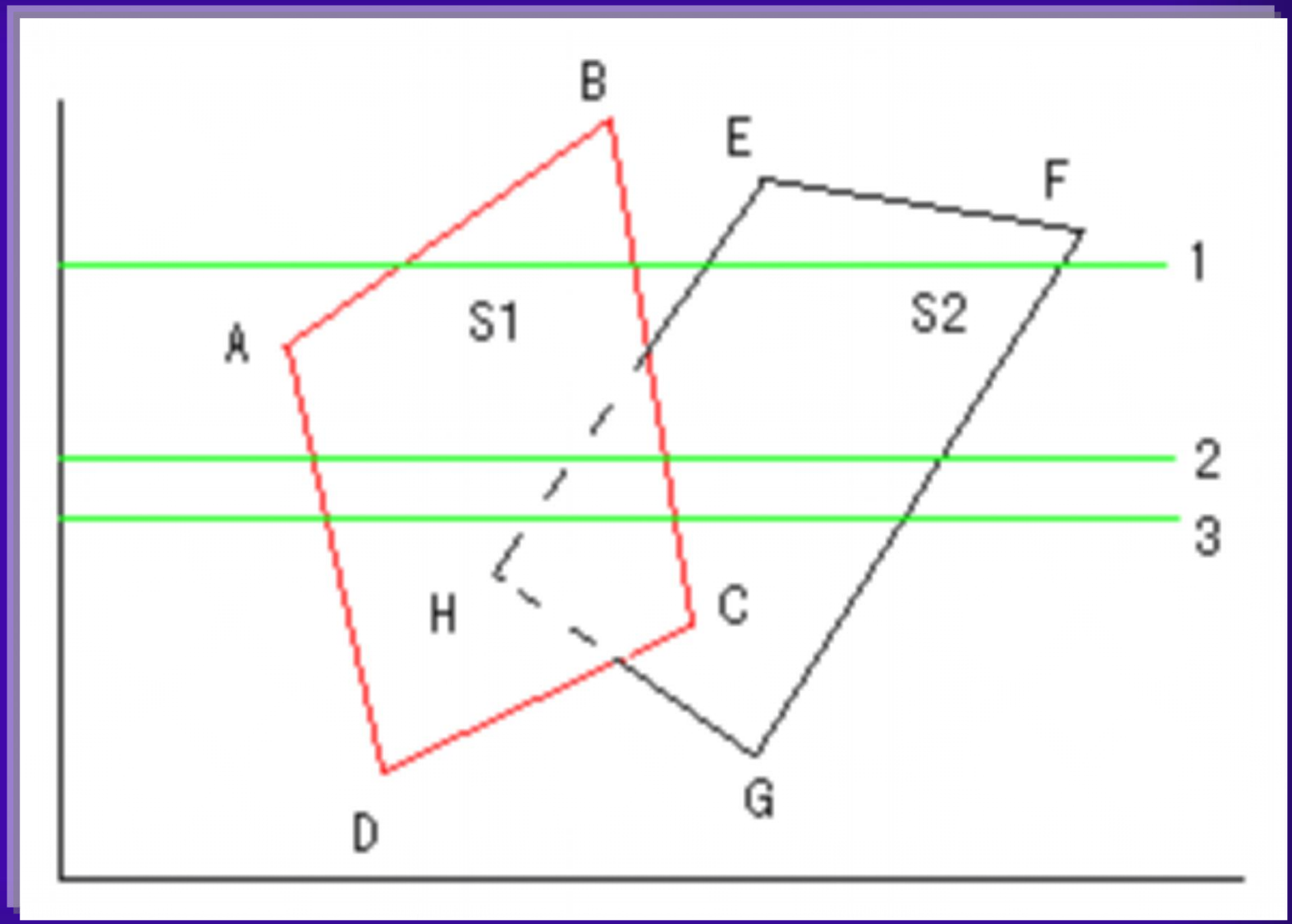
4） 虽有多个多边形与窗口相交，但离观察者最近的多边形包含了窗口，则把整个窗口填上离观察者最近的多边形的颜色。

5）不满足上述条件的窗口需进一步细分。如果到某个时刻，子窗口仅有一个象素大，而窗口内仍有两个以上的面，则该窗口内象素的颜色可取离观察者最近的多边形的颜色，或取和这个窗口相交的所有多边形的平均值。

7.5 扫描线算法

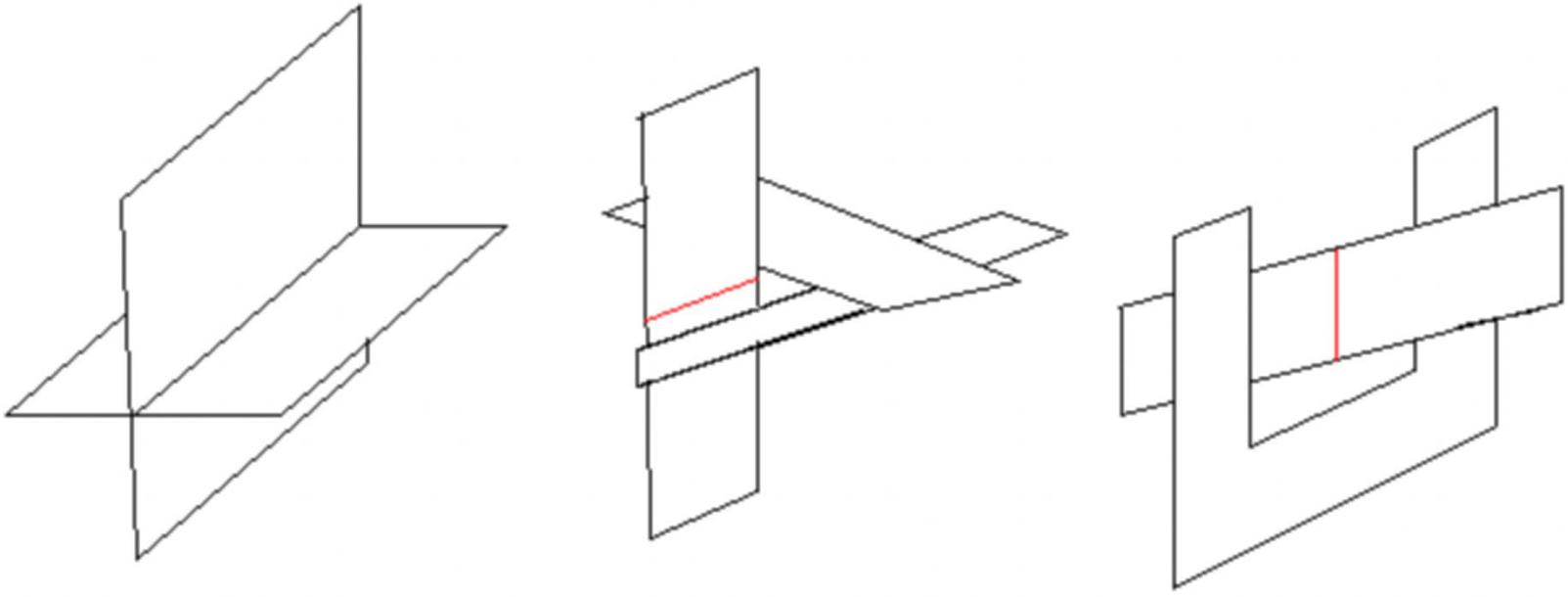
这是一个像空间的消隐算法。

逐个处理各条扫描线时，首先要判别与其相交的所有表面的可见性，然后计算各个重叠表面的深度值以找到离观察平面最近的表面。


B
E
F
1
A
S1
S2
2
3
H
C
D
G

过程：

- 1)针对各个表面，可以建立一张边界表和一张多边形表。需要考察面的边界表
- 2)为了加速查找与扫描线相交的表面，可以由边表中提取的信息建立一张活化边表，该表包含与当前扫描线相交的边，并按照x升序排列，
- 3)另外为多边形面建立一个标志位，便是扫描线上某像素点位于多边形内还是外。
- 4)逐条处理扫描线时，应利用线段的连惯性。比如扫描线3和扫描线2具有相同的活化边表，由于线段交点没有发生变化，因醋无需再计算EH和BC之间的深度。



表面相交并且循环遮挡的情况

7.6 深度排序算法（画家算法）

深度排序算法就是按多边形离观察者的距离进行排序，根据距离的的远近建立一张优先级表，距离观察者近的优先级高，远的优先级低。

正确地建立该表后，只要从优先级低的多边形开始，依次绘制相应的多边形，直到绘制出优先级最高的多边形为止，就生成了整个场景的绘制结果。

本章小结

本讲首先对可见面的判别算法进行了分类，然后介绍了目前用于三维消隐的几个主要算法：深度排序算法（画家算法）、深度缓存(Z-buffer)算法、扫描线(Scan line)算法和区域子分算法(Warnock算法)。