

华中科技大学计算机图形学课后习题指导

一、第五章

5.3 试用中点 Bresenham 算法画直线的原理推导斜率为负且大于 1 的直线段绘制过程（要求写清原理、误差函数、递推公式及最终画图过程）。

分析：这是典型的算法推导题，推导时需要首先确定最大位移方向，然后根据最大位移方向确定候选点，并利用候选点的中点构造判别式；根据判别式确定选择候选点的条件，然后确定判别式的递推公式以及初值。

解：

$$k \leq -1 \quad \frac{|\Delta y|}{|\Delta x|} \geq 1 \quad y \text{ 为最大位移方向}$$

$$\text{故有} \begin{cases} x_{i+1} = \begin{cases} x_i - 1 \\ x_i \end{cases} \\ y_{i+1} = y_i + 1 \end{cases}$$

$$\begin{aligned} \text{构造判别式: } d &= f(x_M, y_M) = f(x_i - 0.5, y_i + 1) \\ &= y_i + 1 - k(x_i - 0.5) - b \end{aligned}$$

推导 d 各种情况的方法(设理想直线与 $y=y_i+1$ 的交点为 Q):

所以有: $y_Q - kx_Q - b = 0$ 且 $y_M = y_Q$

$$d = f(x_M, y_M) = y_M - kx_M - b - (y_Q - kx_Q - b) = k(x_Q - x_M)$$

所以，当（注意这里的判断方法）

$d > 0$ 时, $k < 0$, $x_Q < x_M$ M 点在 Q 点右侧, 取 $P_l(x_i - 1, y_i + 1)$

$d < 0$ 时, $k < 0$, $x_Q > x_M$ M 点在 Q 点左侧, 取 $P_r(x_i, y_i + 1)$

$d = 0$ 时, M 点与 Q 点重合, 约定取 $P_r(x_i, y_i + 1)$

$$\text{所以有} \begin{cases} x_{i+1} = \begin{cases} x_i - 1 & (d > 0) \\ x_i & (d \leq 0) \end{cases} \\ y_{i+1} = y_i + 1 \end{cases}$$

递推公式的推导：

$$d_2 = f(x_i - 1.5, y_i + 2)$$

当 $d > 0$ 时, $= y_i + 2 - k(x_i - 1.5) - b$ 增量为 $1+k$
 $= d_1 + 1 + k$

$$d_2 = f(x_i - 0.5, y_i + 2)$$

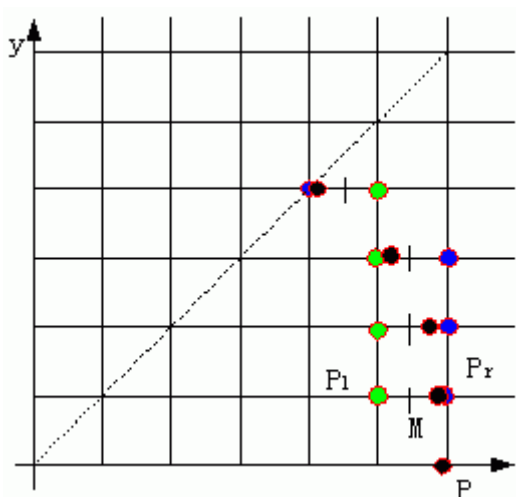
当 $d \leq 0$ 时, $= y_i + 2 - k(x_i - 0.5) - b$ 增量为 1
 $= d_1 + 1$

$$d_0 = f(x_0 - 0.5, y_0 + 1)$$

$$= y_0 + 1 - k(x_0 - 0.5) - b$$

$$= y_0 - kx_0 - b + 1 + 0.5k = 1 + 0.5k$$

5.7 利用中点 Bresenham 画圆算法的原理推导第一象限 $x=y$ 到 $y=0$ 圆弧段的扫描转换算法
 （要求写清原理、误差函数、递推公式及最终画图过程）。



分析：这是典型的算法推导题，推导时需要首先确定最大位移方向，然后根据最大位移方向确定候选点，并利用候选点的中点构造判别式；根据判别式确定选择候选点的条件，然后确定判别式的递推公式以及初值。圆算法应该注意的是算法是从理想圆与坐标轴交点开始的。
 解：在 $x=y$ 到 $y=0$ 的圆弧中， $(R, 0)$ 点比在圆弧上，算法从该点开始。最大位移方向为 y ，由 $(R, 0)$ 点开始， y 渐增， x 渐减，每次 y 方向加 1， x 方向减 1 或减 0。（注意算法的起始点）

设 P 点坐标 (x_i, y_i) ，下一个候选点为 $P_r(x_i, y_i+1)$ 和 $P_l(x_i-1, y_i+1)$ ，取 P_l 和 P_r 的中点 $M(x_i-0.5, y_i+1)$ ，设理想圆与 $y=y_i+1$ 的交点 Q ，构造判别式： $d=F(xM, yM)=(x-0.5)^2+(y+1)^2-R^2$

当 $d < 0$ 时， M 在 Q 点左方，取 $P_r(x_i, y_i+1)$ ；

当 $d > 0$ 时， M 在 Q 点右方，取 $P_l(x_i-1, y_i+1)$ ；

当 $d=0$ 时， M 与 Q 点重合，约定取 $P_l(x_i-1, y_i+1)$ 。

所以有：
$$\begin{cases} y_{i+1} = y_i + 1 \\ x_{i+1} = \begin{cases} x_i - 1 & (d \geq 0) \\ x_i & (d < 0) \end{cases} \end{cases}$$

推导判别式：

$d \geq 0$ 时，取 $P_1(x_i-1, y_i+1)$ ，下一点为 (x_i-1, y_i+2) 和 (x_i, y_i+2)

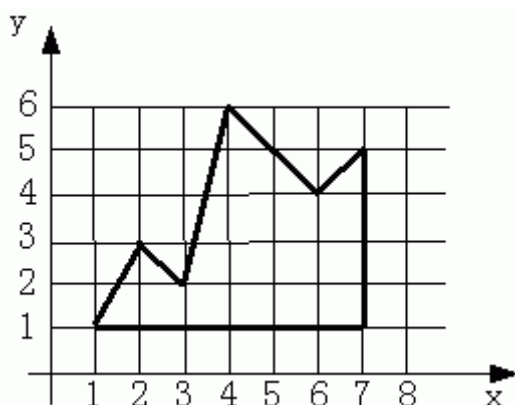
$$\begin{aligned} d_2 &= F(x_i-1.5, y_i+2) \\ &= (x_i-1.5)^2 + (y_i+2)^2 - R^2 \\ &= (x_i-0.5)^2 - 2(x_i-0.5) + 1 + (y_i+1)^2 + 2(y_i+1) + 1 \\ &= d_1 - 2x_i + 2y_i + 5 \end{aligned}$$

$d < 0$ 时，取 $P_2(x_i, y_i+1)$ ，下一点为 (x_i, y_i+2) 和 (x_i-1, y_i+2)

$$\begin{aligned} d_2 &= F(x_i-0.5, y_i+2) \\ &= (x_i-0.5)^2 + (y_i+2)^2 - R^2 \\ &= (x_i-0.5)^2 + (y_i+1)^2 + 2(y_i+1) + 1 \\ &= d_1 + 2y_i + 3 \end{aligned}$$

$$d_0 = F(R-0.5, 1) = R^2 - R + 0.25 + 1 - R^2 = 1.25 - R$$

5.11 如图所示多边形，若采用扫描转换算法（ET 边表算法）进行填充，试写出该多边形的 ET 表和当扫描线 $Y=4$ 时的有效边表（AET 表，活性边表）。



分析：改进的有效边表算法是用软件方法实现扫描转换效率较高的算法，它利用了边表来构造有效边表。需要注意的有以下几点：

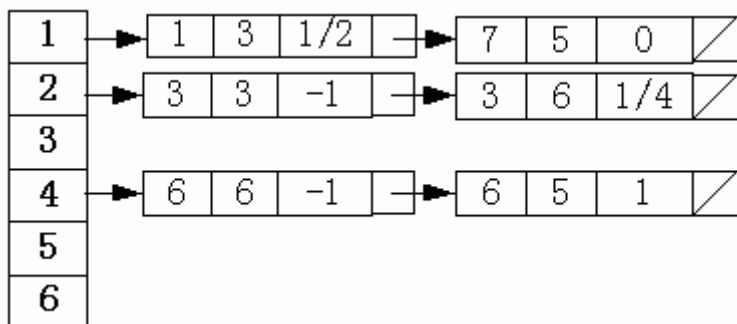
（1）构造边表时，水平边不需要构造，算法能够获取到水平边的两个端点，配对填充后水平边被填充，因此水平边的数据不参与计算。

（2）边表中纵向链表的长度等于多边形覆盖的扫描线数，并且是从多边形覆盖的最低扫描线处开始。

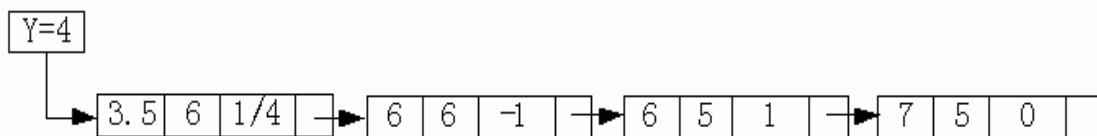
（3）边表与有效边表中每个结点的第三项为 $1/k$ ；

(4) 构造有效边表时，每个结点的第一项，即当前扫描线与多边形边交点处的 x 坐标不需要四舍五入，否则在计算下一条扫描线时可能会造成误差。

解：ET 表



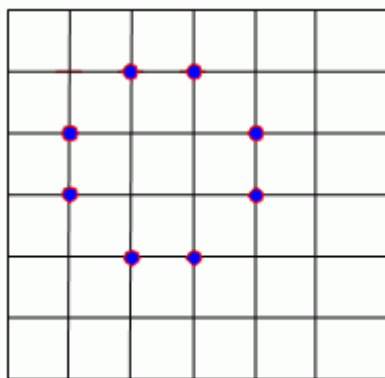
$y=4$ 时的 AET 表



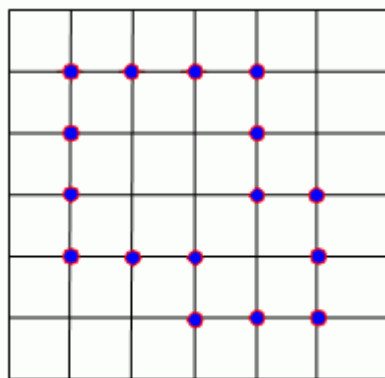
5.12 构造两个例子，一个是 4-连通图，其边界是 8-连通的，另一个是 8-连通图，其边界是 4-连通的。

分析：由于八邻接点中包含四邻接点，所以四连通区域也可以看作八连通区域，但是四连通区域与八连通区域的边界条件是不同的，通常在边界表示的区域中，四连通区域边界（内环和外环边界）的连通性是八连通，而八连通区域边界（内环和外环边界）的连通性是四连通。在一些特殊的情况下一个区域的连通性可能既是四连通也是八连通，例如内点表示的四连通区域（内点表示没有显示的边界），或者将边界表示的四连通区域的所有边界（内环和外环边界）都改为四连通性质。

解：



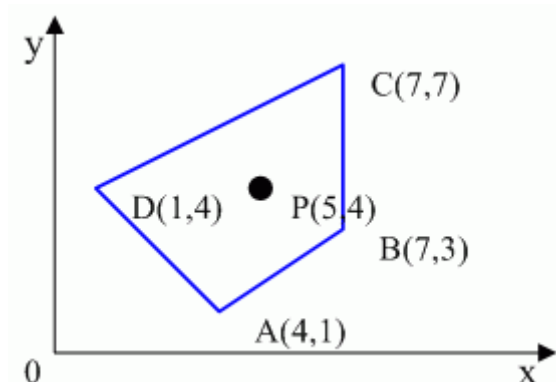
4-连通区域



8-连通区域

二、 第六章

6.7 如图所示四边形 ABCD，求绕 P(5, 4) 点分别旋转 45° 的变换矩阵，并求出各端点坐标，画出变换后的图形。



分析：对于变换的计算，重点在于分析变换的过程。一般来说，基于点的变换是将图形与点一起平移使点与原点重合，然后相对于原点变换，最后将点与图形一起平移使点回到原来的位置。

解：变换的过程包括：

(1) 平移：将 P 点平移至原点，变换矩阵为：

$$T_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -5 & -4 & 1 \end{bmatrix}$$

(2) 旋转：图形绕原点（P 点）旋转 45 度，变换矩阵为：

$$T_R = \begin{bmatrix} \cos 45^\circ & \sin 45^\circ & 0 \\ -\sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3) 反平移：将 P 点移回原处，变换矩阵为：

$$T_t^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 4 & 1 \end{bmatrix}$$

变换矩阵为：

$$T = T_t \bullet T_R \bullet T_t^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -5 & -4 & 1 \end{bmatrix} \begin{bmatrix} \cos 45^\circ & \sin 45^\circ & 0 \\ -\sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5 & 4 & 1 \end{bmatrix}$$

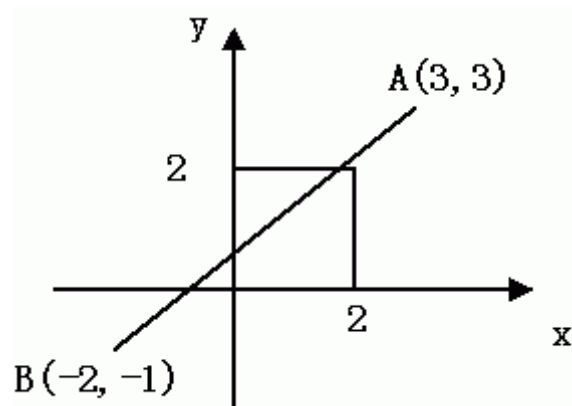
$$= \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 5 - \sqrt{2}/2 & 4 - 9\sqrt{2}/2 & 1 \end{bmatrix}$$

变换过程为：

$$P' = P \bullet T = \begin{bmatrix} 4 & 1 & 1 \\ 7 & 3 & 1 \\ 7 & 7 & 1 \\ 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 5 - \sqrt{2}/2 & 4 - 9\sqrt{2}/2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 5 + \sqrt{2} & 4 - 2\sqrt{2} & 1 \\ 5 + 3\sqrt{2}/2 & 4 + \sqrt{2}/2 & 1 \\ 5 - \sqrt{2}/2 & 4 + 5\sqrt{2}/2 & 1 \\ 5 - 2\sqrt{2} & 4 - 2\sqrt{2} & 1 \end{bmatrix}$$

6.11 试用编码裁剪算法裁剪如图所示线段 AB。



分析：编码裁剪法的特点是能够快速的接受和拒绝可见和不可见的直线段，但是其求交的过程相对复杂。

解：1) 首先对 AB 进行编码，A 的编码 code1 为 1010，B 的编码 code2 为 0101；由于 $\text{code1}|\text{code2} \neq 0$ ，且 $\text{code1} \& \text{code2} = 0$ ，因此对直线段 AB 既不能简取也不能简弃。

(2) 求交处理。A 的编码为 1010，右起第二位编码不为 0，故求出直线段 AB 与窗口右边界的交点 C(2, 11/5)；

C 的编码为 1000，由于该编码不为 0，所以求出的交点为虚交点，继续求交。

C 点编码的最高位不为 0，故求直线段 AB 与上边界的交点 D(7/4,2)，其编码为 0000，此时 AD 必在窗口外，可以用 D 点的坐标和编码代替 A 点的坐标和编码，code1=0000

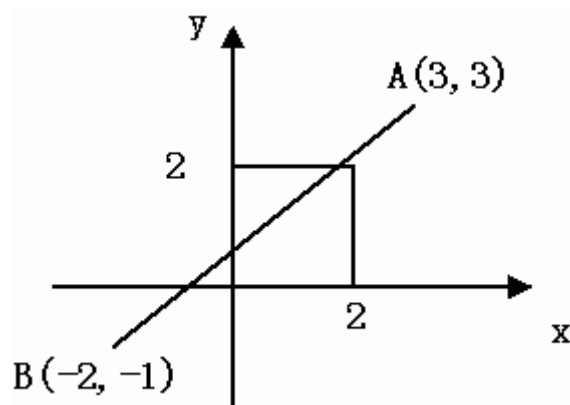
由于 code1|code2 ≠ 0，且 code1&code2 = 0，直线段 A (D) B 仍然不能简取也不能简弃。

(3) 求交处理。A (D 点) 的编码为 0000，交换 code1 与 code2 的值以及 AB 点的坐标值保证起点在窗口外，此时 code1=0101，code2=0000

由于 code1=0101，右起第一位编码不为 0，故求出直线段 A (B) B (D) 与窗口左边界的交点 E(0,3/5)；E 点的编码为 0000，此时 EA (B) 必在窗口外，可简弃。用 E 点坐标和编码代替 A (B) 点坐标和编码

由于 code1 | code2 = 0，直线段 A (E) B (C) 简取。

6.14 用梁友栋算法裁剪如图所示线段 AB。



分析：梁友栋算法是参数法裁剪在标准窗口下的特例，只要按照公式，先构造直线的参数方程，然后计算 p，q，u 值，最后求出 umax 和 umin 值，再代入到参数方程，求出直线段裁剪结果的边界坐标。

解：以 A (3, 3) 为起点，B(-2, -1) 为终点（这里起点和终点可以互换，但这样直线的参数方程将会改变）。

所以有 x1=3, y1=3, x2= -2, y2=-1, wxl=0, wxr=2, wyb=0, wyt=2

构造直线参数方程：

$$\begin{cases} x = x_1 + u(x_2 - x_1) \\ y = y_1 + u(y_2 - y_1) \end{cases} \quad (0 \leq u \leq 1)$$

$$\text{即} \begin{cases} x = 3 - 5u \\ y = 3 - 4u \end{cases} \quad (0 \leq u \leq 1)$$

计算各个 p 和 q 值有：

$$\begin{aligned} p_1 &= x_1 - x_2 = 5 & q_1 &= x_1 - wxl = 3 \\ p_2 &= x_2 - x_1 = -5 & q_2 &= wxr - x_1 = -1 \\ p_3 &= y_1 - y_2 = 4 & q_3 &= y_1 - wyb = 3 \\ p_4 &= y_2 - y_1 = -4 & q_4 &= wyt - y_1 = -1 \end{aligned}$$

根据 $u_k = q_k/p_k$ ，有

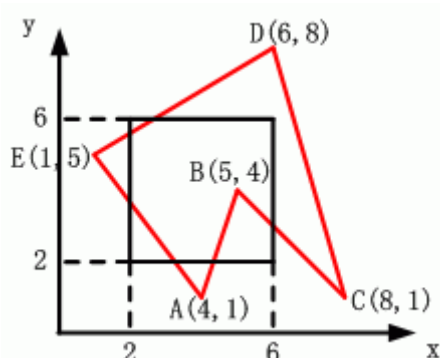
$$u_1 = 3/5 \quad u_2 = 1/5 \quad u_3 = 3/4 \quad u_4 = 1/4$$

$$\therefore u_{\max} = \text{MAX}(0, u_2, u_4) = \text{MAX}(0, 1/4, 1/5) = 1/4$$

$$u_{\min} = \text{MIN}(u_1, u_3, 1) = \text{MIN}(3/5, 3/4, 1) = 3/5$$

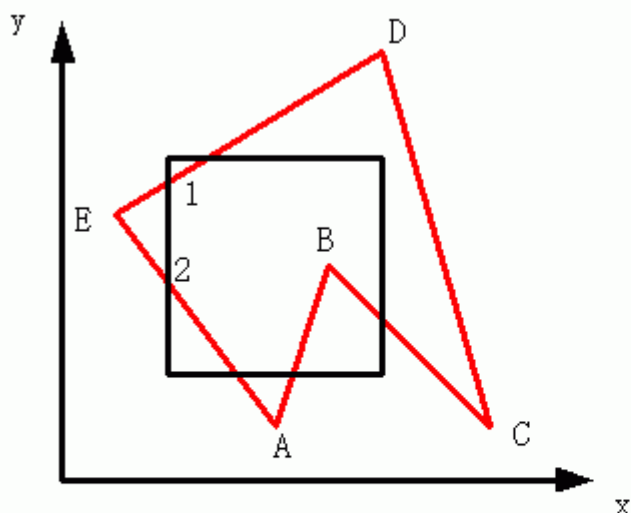
由于 $u_{\max} < u_{\min}$ ，故此直线 AB 有一部分在裁减窗口内，将 u_{\max} 和 u_{\min} 代入直线参数方程，得到直线在窗口内部分的端点 C (7/4, 2) 和 D(0, 3/5)。

6.15 试用 Sutherland-Hodgeman 算法对如下图所示多边形进行裁剪，要求画出每次裁剪对应的图形，并标明输入和输出的顶点。

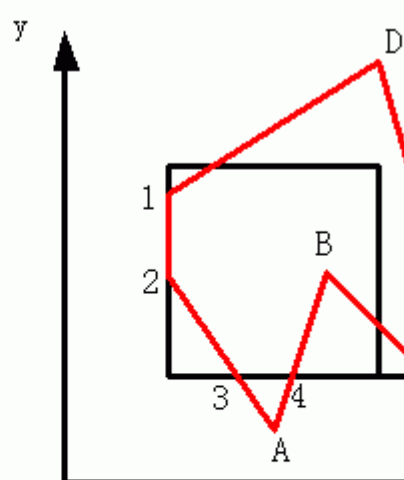


分析：Sutherland-Hodgeman 算法是按照规定的策略，在每条边界裁剪的时候，输出顶点序列。注意裁剪时使用的交点是窗口边界及其延长线与多边形边的交点。

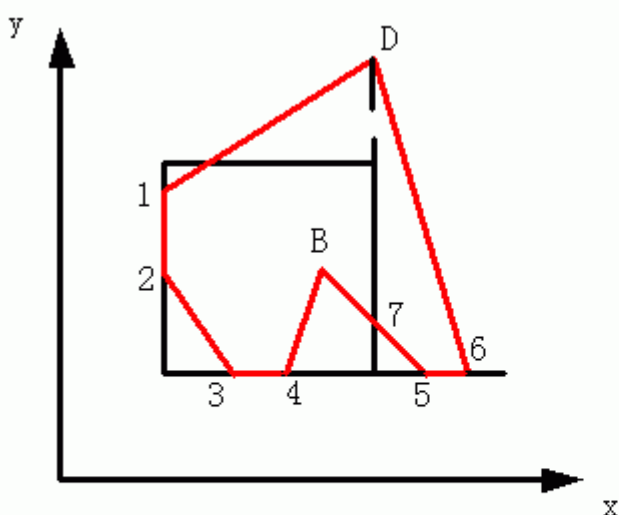
解：



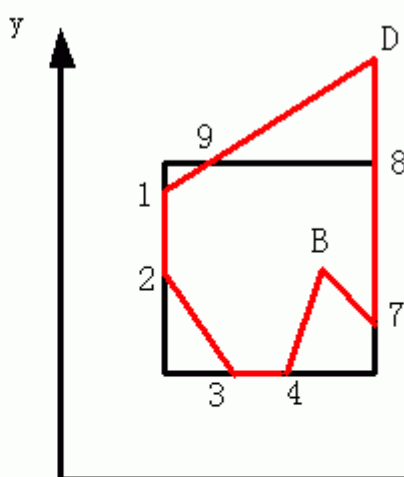
(1) 用左边界裁剪
输入: ABCDE
输出: BCD12A



(2) 用下边界裁剪
输入: BCD12A
输出: 56D1234B



(3) 用右边界裁剪
输入: 56D1234B
输出: D1234B7

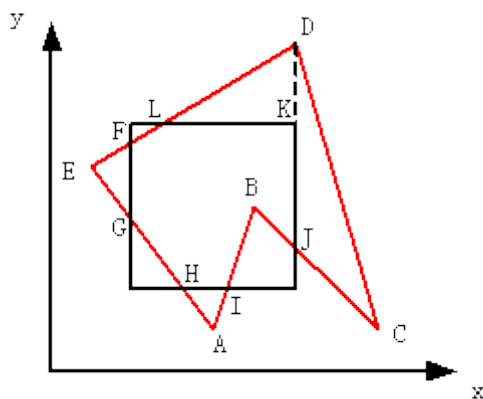


(4) 用下边界裁剪
输入: D1234B7
输出: 91234B7

6.16 试用 Weiler-Atherton 算法对如下图所示多边形进行裁剪, 要求画出每次裁剪对应的图形, 并标明输入和输出的顶点。

分析: Weiler-Atherton 算法是沿着窗口边界和多边形边界裁剪, 处理时可以先利用直线段裁剪算法用窗口边界裁剪多边形的边, 如果遇到由窗口内到窗口外的情况则需要用对应交点所在的窗口边界与多边形所有的边求交, 求出实交点后排序, 根据裁剪的方向求得所要的点, 如果没有则需要输出一段窗口边界, 然后用下一条边界继续求交, 直到找到所要的点为止。但多边形所有的边处理完以后, 则根据输出的直线段端点连接成一个或者多个多边形。

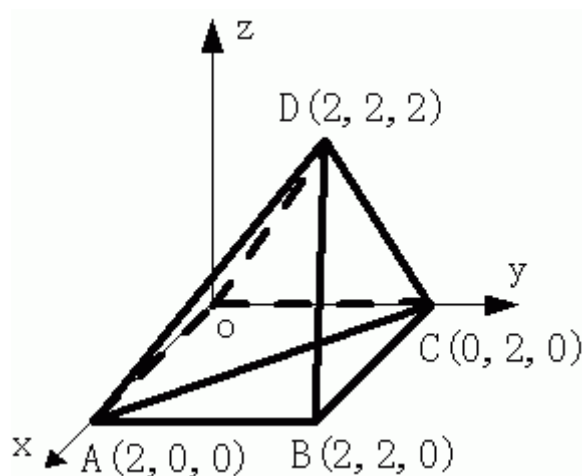
解: 如下图所示, 输入顶点序列 ABCDE。逆时针处理多边形, 此时沿窗口边界处理的顺序也是逆时针。



- (1) 输入 **AB**，由不可见侧进入可见侧，输出直线段 **IB**；
- (2) 输入 **BC**，由可见侧进入不可见侧，输出直线段 **BJ**，并从 **J** 点沿窗口逆时针找到距离最近的交点 **L**，输出直线段 **JK** 和 **KL**；
- (3) 输出 **CD**，由不可见侧进入不可见侧，不输出；
- (4) 输出 **DE**，由不可见侧进入可见侧再进入不可见侧，输出直线段 **LF**，并从 **F** 点沿窗口逆时针找到距离最近的交点 **G**，输出直线段 **FG**；
- (5) 输出 **EA**，由不可见侧进入可见侧再进入不可见侧，输出直线段 **GH**，并从 **H** 点沿窗口逆时针找到距离最近的交点 **I**，输出直线段 **HI**；

三、 第七章

7.5 求将下图中的空间四面体关于 $P(2, -2, 2)$ 点整体放大两倍的变换矩阵，写出复合变换后图形各顶点的规范化齐次坐标。



分析：三维变换的过程与二维变换的过程类似。

解：关于 P 点整体放大两倍，可以先将 P 点平移至原点，再进行比例变换，最后反平移。

变换矩阵为：

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -2 & 2 & -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & -2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 1 & 1 & 1 \end{bmatrix}$$

变换过程为：

$$P' = P \cdot T = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 2 & 2 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 1 & -1 & 1/2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & 1/2 \\ 1 & 3 & -1 & 1/2 \\ -1 & 3 & -1 & 1/2 \\ 1 & 3 & 1 & 1/2 \end{bmatrix}$$

所以变换后各点的规范化齐次坐标（注意要转换成规范化齐次坐标）分别为： $(2, 2, -2, 1)$ ， $(2, 6, -2, 1)$ ， $(-2, 6, -2, 1)$ ， $(2, 6, 2, 1)$

注意：本题中的比例变换矩阵也可写为

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

变换矩阵为：

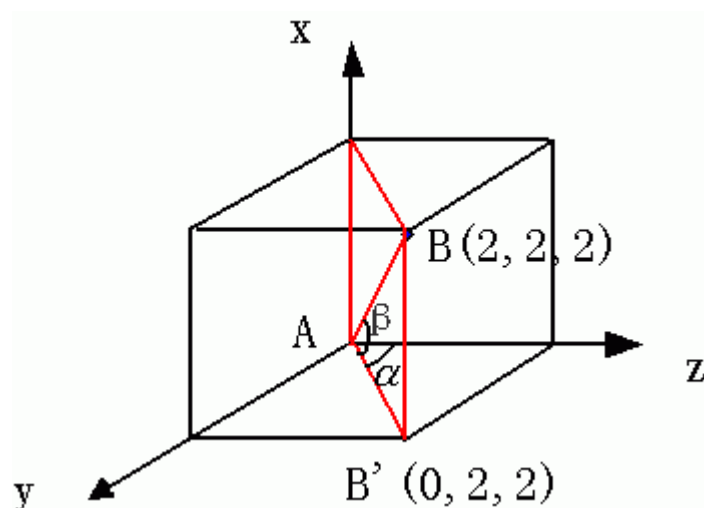
$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -2 & 2 & -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & -2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ 0 & 0 \\ -2 & 2 \end{bmatrix}$$

变换过程：

$$P' = P \cdot T = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 2 & 2 & 2 & 2 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ -2 & 2 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 & -2 & 1 \\ 2 & 6 & -2 & 1 \\ -2 & 6 & -2 & 1 \\ 2 & 6 & 2 & 1 \end{bmatrix}$$

所以变换后各点的规范化齐次坐标分别为：(2, 2, -2, 1)，(2, 6, -2, 1)，(-2, 6, -2, 1)，(2, 6, 2, 1)

7.6 假定一空间直线 AB 的两端点坐标为 A(0, 0, 0) 和 B(2, 2, 2), 试写出绕 AB 旋转 30° 的三维复合变换矩阵。



分析：三维空间中绕任意轴旋转是较为复杂的过程。在经过平移使轴通过原点之后，通常需要经过两次旋转才能使轴与某个坐标轴重合：第一次将轴与某个坐标轴构造成一个面，并将这个面绕这个坐标轴旋转到某个坐标平面内，然后在该坐标平面中通过旋转使轴与该平面中的任意一个坐标轴重合；在绕轴旋转之后需要反旋转使轴回到原来的方向，最后反平移使轴回到原始的位置。这里需要注意旋转角度的正负以及角度值的确定。

解：变换步骤如下

(1) 将 ABB' 绕 x 轴逆时针旋转 α 角，则 AB 旋转到 xoz 平面上。

$$T_{Rx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & -\sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2)将 OB 绕 y 轴顺时针旋转 β 角，则 OB 旋转到 z 轴上。

$$T_{Ry} = \begin{bmatrix} \cos(-\beta) & 0 & -\sin(-\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(-\beta) & 0 & \cos(-\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{6}/3 & 0 & \sqrt{3}/3 & 0 \\ 0 & 1 & 0 & 0 \\ -\sqrt{3}/3 & 0 & \sqrt{6}/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(3)此时 AB 轴与 z 轴重合，绕 AB 轴的旋转转换为绕 z 轴的旋转。

$$T_{Rz} = \begin{bmatrix} \cos 30^\circ & \sin 30^\circ & 0 & 0 \\ -\sin 30^\circ & \cos 30^\circ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{3}/2 & 1/2 & 0 & 0 \\ -1/2 & \sqrt{3}/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(4)最后，求 T_{Rx} , T_{Ry} 的逆变换，回到 AB 原来的位置。

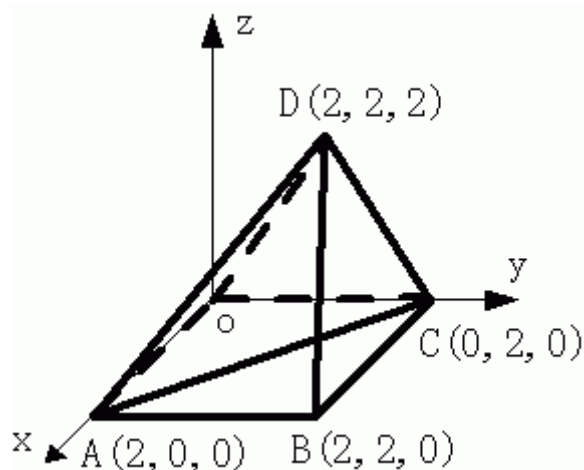
$$T_{Ry}^{-1} = \begin{bmatrix} \cos \beta & 0 & -\sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{6}/3 & 0 & -\sqrt{3}/3 & 0 \\ 0 & 1 & 0 & 0 \\ \sqrt{3}/3 & 0 & \sqrt{6}/3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{Rx}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ 0 & \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

最后的变换矩阵是这 5 个矩阵相乘的形式：

$$T = T_{R_x} \cdot T_{R_y} \cdot T_{R_z} \cdot T_{R_y}^{-1} \cdot T_{R_x}^{-1} = \begin{bmatrix} \frac{1+\sqrt{3}}{3} & \frac{1}{3} & \frac{1-\sqrt{3}}{3} & 0 \\ \frac{1-\sqrt{3}}{3} & \frac{1+\sqrt{3}}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1-\sqrt{3}}{3} & \frac{1+\sqrt{3}}{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

7.7 试作出下图中空间四面体的三视图，要求写清变换式（设平移矢量均为1）。



分析：三视图是最为简单的一种正投影，可以利用主视、俯视和侧视图的公式直接计算。

解：（1）主视图（主视图不需要平移）

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = P \cdot T = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 2 & 2 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 2 & 1 \end{bmatrix}$$

（2）俯视图

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

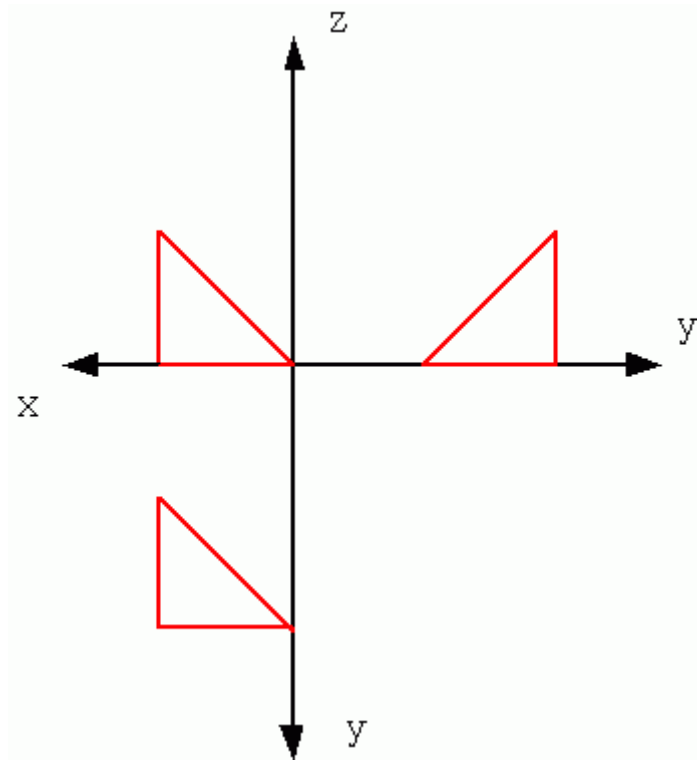
$$P' = P \cdot T = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 2 & 2 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & -1 & 1 \\ 2 & 0 & -3 & 1 \\ 0 & 0 & -3 & 1 \\ 2 & 0 & -3 & 1 \end{bmatrix}$$

(3) 侧视图

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = P \cdot T = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 2 & 2 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ -3 & 0 & 0 & 1 \\ -3 & 0 & 0 & 1 \\ -3 & 0 & 2 & 1 \end{bmatrix}$$

最后的图形为：



注意：三个图要画在一个坐标系中，以及点与点的连接关系以及直线的可见性问题。