计算机图形学

（总复习）

考试相关：

1）闭卷；

2）卷面65分；平时（含实验）35分。

3）10周周4的3-4节 / 2021-11-04 10:00-11:35

4）学汇108

计算机图形学

计算机图形学（**Computer Graphics**，简称**CG**）是一种使用数学算法将二维或三维图形转化为计算机显示器的栅格形式的科学。

计算机图形图形

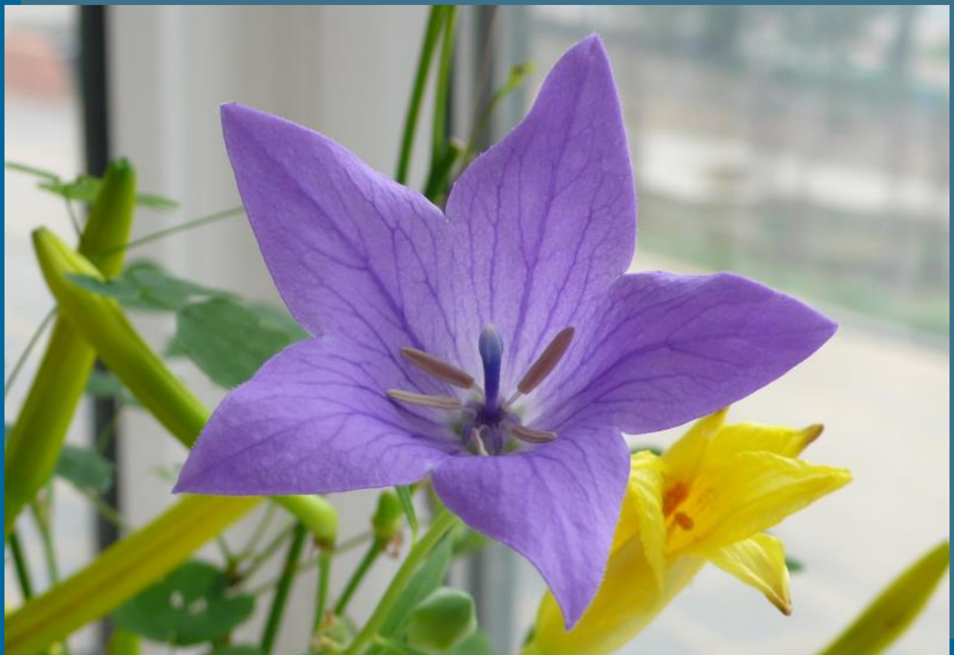
图形通常由点、线、面、体等几何元素和灰度、色彩、线型、线宽等非几何属性组成。

- 二维图形;
- 三维图形
- 图像处理

图形在计算机中的表示-标量图

- 点阵法
- 是用具有颜色信息的点阵来（枚举）表示图形的一种方法，它强调图形由哪些点组成，并具有什么灰度或色彩。

图像（Image）



图形在计算机中的表示-矢量图

- 参数法：
- 是以计算机中所记录图形的形状参数与属性参数来表示图形的一种方法。
  - 形状参数（几何要素）——方程或分析表达式的系数，线段的端点坐标等
  - 属性参数（非几何要素）——颜色、材质、线型等

*图形（**Graphics**）*

直线A的端点为(x1, y1), (x2, y2)，红色，实线，宽度2个象素

计算机图形学---研究内容

- 几何：研究面的表示和处理的方法
- 动画：研究移动的表示和操作方法
- 图像拟真/渲染：研究模拟光线传递效果的算法
- 图像：研究图像的获取或是编辑

目的：要利用计算机产生令人赏心悦目的真实感图形。

- 真实感图形学
  - **1970年，Bouknight提出了第一个光反射模型**
  - **1971年　Gourand提出“漫反射模型＋插值”的思想，被称为Gourand明暗处理**
  - **1975年，Phong提出了著名的简单光照模型- Phong模型**

Gourand（哥罗德）法

- 基于平面多面体逼近原型体，计算每个顶点法向量方向的光亮度；
- 然后通过插值计算出多边形内容各点的光亮度；
- 缺点：由于采用了插值方法，使得镜面反射所产生的特亮区域的形状和位置存在很大差异同。

Phong法向插值方法

- 为了克服Gourand法对镜面发射区域所引起的畸变，Phong提出了用法向量来代替光强度插值；
- 这样做将代表亮度值的I（标量）换成了相应法向量N（矢量），计算量会更大些，但是效果更好些。

第一讲 绪论

1.1 发展历史
1.2 研究内容
1.3 应用与研究前沿

1.1 发展历史

1 历史追溯

2 硬件发展

图形显示器的发展

图形输入设备的发展

3 图形软件及软件标准的发展

历史追溯

- 二十世纪50年代（酝酿期）
- 60年代（萌芽期）
- 70年代（发展期）
- 80年代（普及期）
- 90年代（深入发展）

硬件发展

- 图形显示器的发展
  - 60年代中期：画线显示器（亦称矢量显示器）需要刷新。设备昂贵，限制普及
  - 60年代后期：存储管式显示器，不需刷新，价格较低，缺点是不具有动态修改图形功能，不适合交互式。

- 70年代初，刷新式光栅扫描显示器出现，大大地推动了交互式图形技术的发展。
- 特点：以点阵形式表示图形，使用专用的缓冲区存放点阵，由视频控制器负责刷新扫描。

硬件发展

图形输入设备的发展

- 第一阶段：控制开关、穿孔纸等等
- 第二阶段：键盘
- 第三阶段：二维定位设备，如鼠标、光笔、图形输入板、触摸屏等等，语音
- 第四阶段：三维输入设备（如空间球、数据手套、数据衣），用户的手势、表情等等
- 未来发展：用户的思维

图形软件发展及软件标准形成

三种类型的计算机图形软件系统发展：

（1）用某种语言写成的子程序包，如：

GKS (Graphics Kernel System)

PHIGS(Programmer’s Herarchical Interactive Graphics system)

GL，OpenGL，DirectX

特点：平台适应性较好，便于移植和推广；但执行速度相对较慢（事实工业标准得到硬件支持，基本上不存在这个问题）；使用不方便。

图形软件发展及软件标准形成

（2）扩充计算机语言，使其具有图形生成和处理的功能如：

Turbo Pascal、Turbo C，AutoLisp等。

特点：简练、紧凑、执行速度快，但可移植性差。

（3）专用图形系统，如：

AUTOCAD、3DS MAX

特点：效率高，但系统开发量大，可移植性差。

课后习题

1．图形硬件设备主要包括哪些？请按类别举出典型的物理设备?

图形输入设备：鼠标、光笔、触摸屏、数据手套和坐标数字化仪，以及图形扫描仪等。
图形输出设备：CRT、液晶显示器（LCD）、打印机、绘图仪等。
图形处理器：GPU（图形处理单元）、图形加速卡等等。

2．什么是图形软件标准？为什么要制定图形软件标准？

图形软件标准通常是指图形系统及其相关应用系统中各界面之间进行数据传送和通信的接口标准，另外还有供图形应用程序调用的子程序功能及其格式标准。

为了提高计算机图形软件、计算机图形的应用软件以及相关软件的编程人员在不同计算机和图形设备之间的可移植性。

- 3）常用的图形输入设备有：（ ABD ）
- A 数据手套
- B 扫描仪
- C 绘图机
- D 触摸屏

- 4）下列哪些是计算机图形学的应用领域：（ ABCD ）
- A CAD/CAM/CAI
- B 图像处理
- C 数据场可视化
- D 艺术造型和模拟

- 5）下列哪些是常用的图形输出设备：（ ACD ）
- A 阴极射线管（CRT）
- B 触摸屏
- C 绘图仪
- D 打印机

- 6）图形通常由点、线、面、体等几何元素和灰度、色彩、线型、线宽等非几何属性组成。

- 7）图形在计算机中的表示有点阵法（标量图）和参数法（矢量图）

8) 什么是OpenGL（ Open Graphics Library ）

- OpenGL 是行业领域中最为广泛接纳的 2D/3D 图形 API
- OpenGL是个与硬件无关的软件接口，可以在不同的操作系统平台之间进行移植。

9) 什么是DirectX（ Direct eXtension）

- 微软公司创建的多媒体编程接口，由C++编程语言实现。
- 很多API组成的，按照性质分类，可以分为四大部分，显示部分、声音部分、输入部分和网络部分。

11) 虚拟现实技术的发展趋势(ABCD)

- A) 动态环境建模技术
- B) 实时三维图形生成和显示技术
- C) 适人化、智能化人机交互设备的研制
- D) 大型网络分布式虚拟现实的研究与应用

- **12) 智能CAD 系统可以实现从概念设计到结构设计的全过程。**

- **13)什么是科学计算可视化？**
  用图形直接反映科学计算的结果，如分子模型、核爆炸过程、大气科学等。

14）试列举出几种图形学的软件标准?

- ACM成立图形标准化委员会，1977年制定CGS（“核心图形系统”--Core Graphics System）
- ISO发布CGI、CGM、GKS-3D、PHIGS等
- 工业界事实上的标准OpenGL，DirectX

15）常用的真实感图形绘制方法包括（AB）

A）Gourand（哥罗德）算法

B）Phong算法

C）Liang-Barsky算法

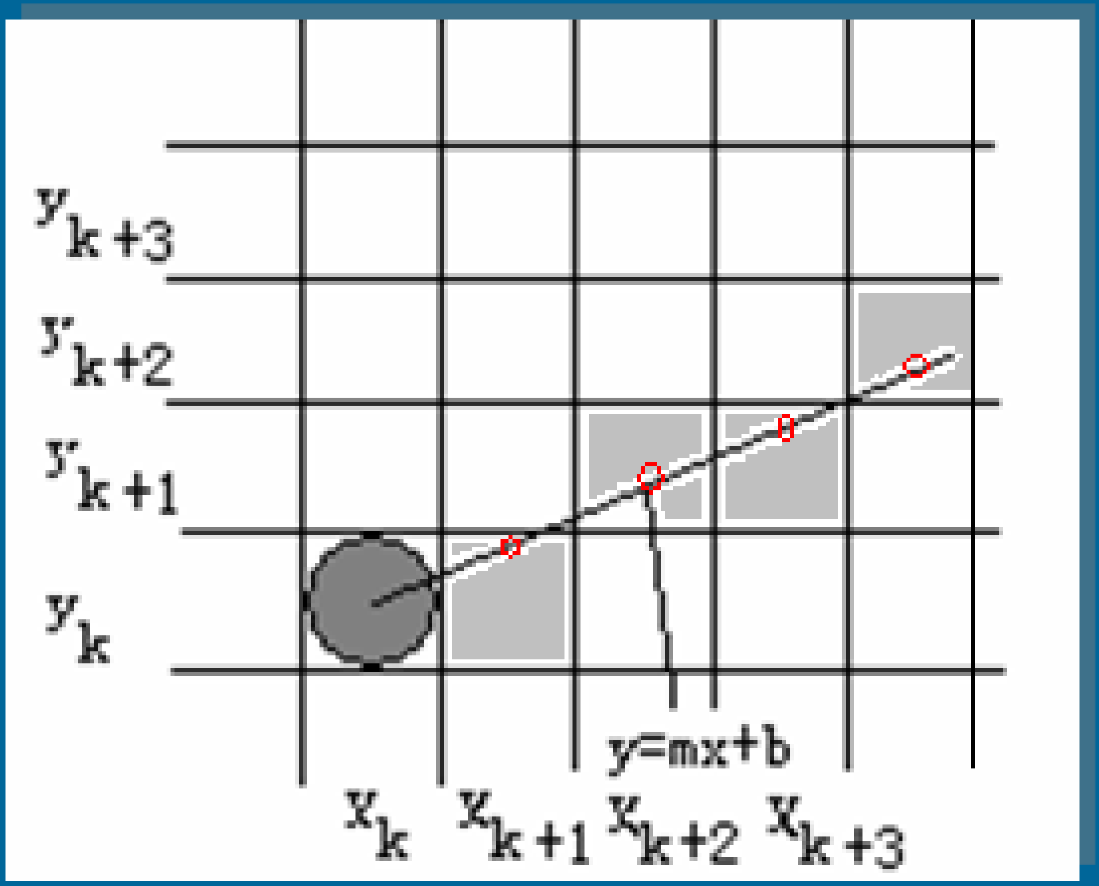
D）Weiler-Atherton算法

第二讲　光栅图形学

　　光栅图形显示器可以看成是由许多可发光的离散点（即像素）组成的矩阵，它需要专门的算法来生成直线、圆弧和曲线等等图形。

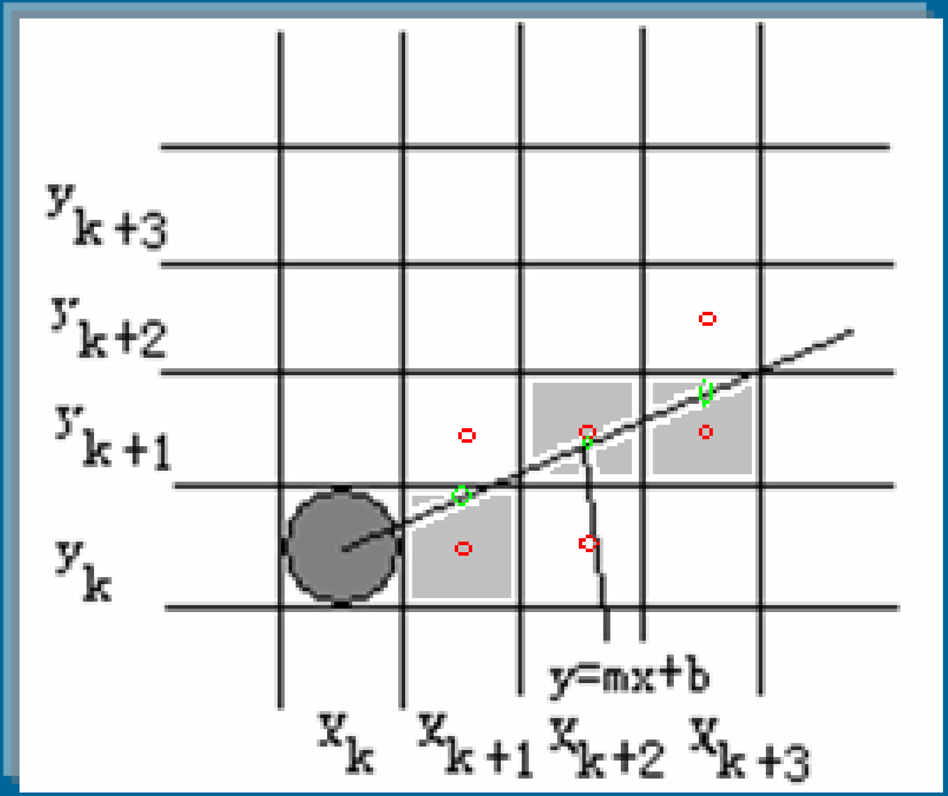
1）DDA算法（digital differential analyzer）

思想：在一个坐标轴上以单位间隔对线段取样，从而确定另一个坐标轴上最靠近线路经的对应位置。



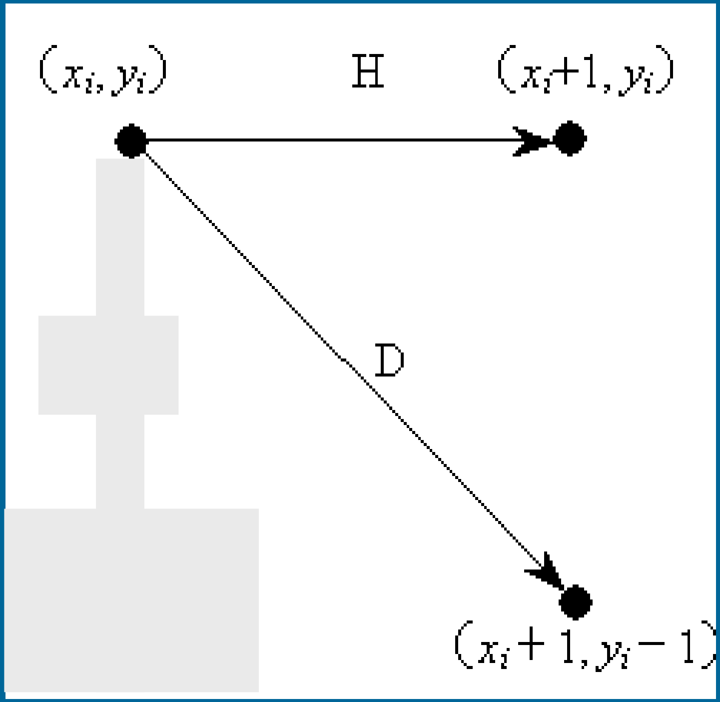
2）Bresenham算法 —常用算法

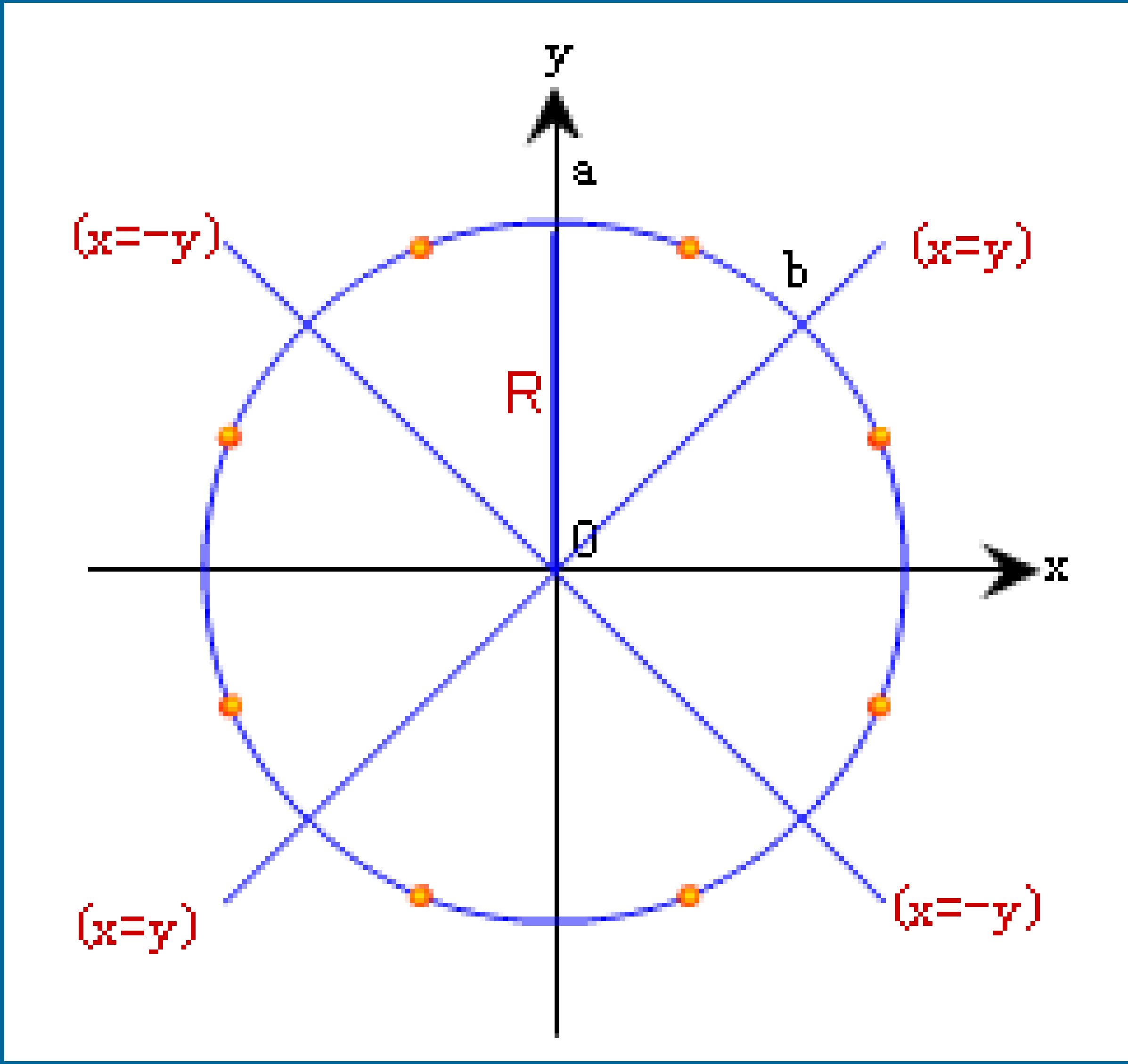
思想：以单位x取样，并且需要确定每一次取样时，两个可能的像素位置哪一个更接近线路经。采用增量计算，使得对于每一列，只要检查一个误差项的符号，就可以确定该列像素中与此交点最近的像素。



3）中点画圆算法

中心画圆算法和Bresenham算法原理类似，只是搜索的像素点范围不同。





$P_2 = C(x, -y)$，

$P_3 = C(-x, y)$，

$P_4 = C(-x, -y)$，

$P_5 = C(y, x)$，

$P_6 = C(-y, x)$，

$P_7 = C(y, -x)$，

$P_8 = C(-y, -x)$。

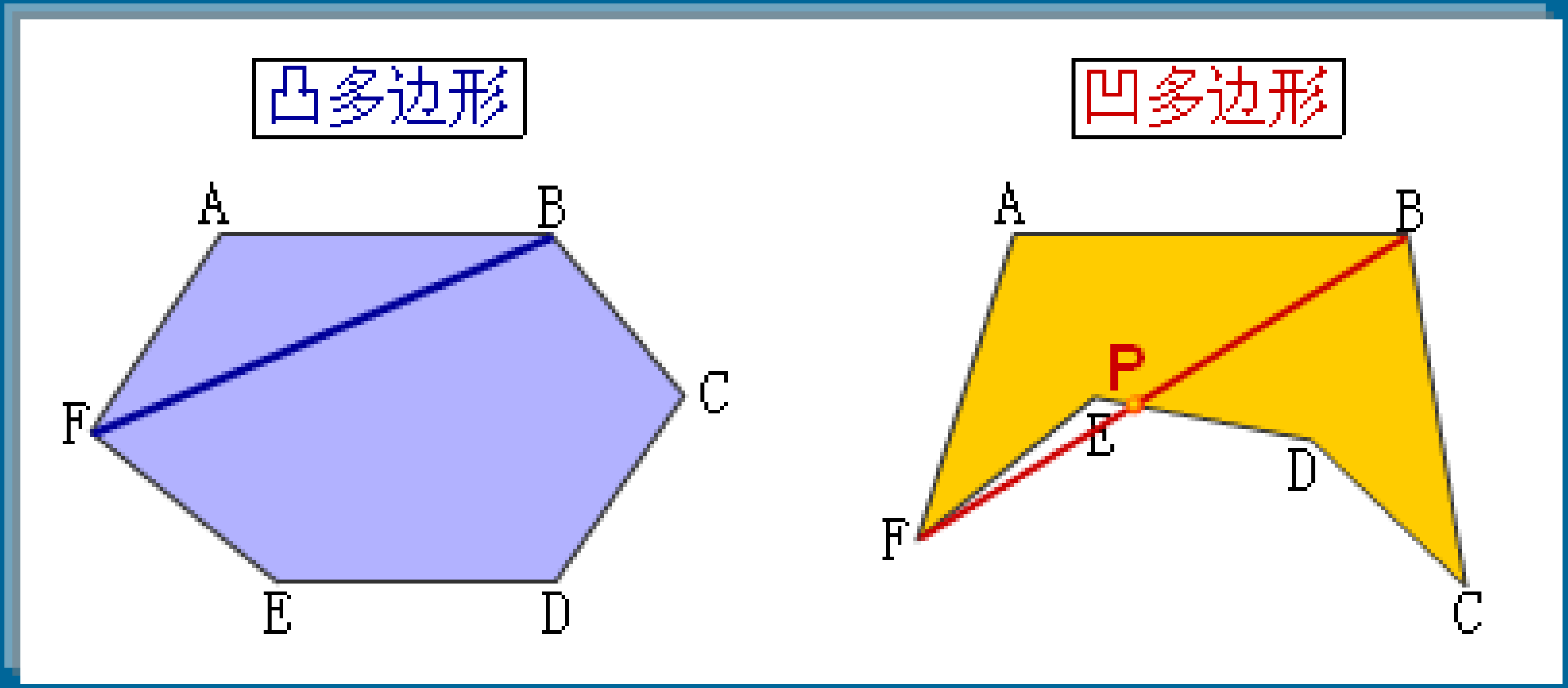
4）区域填充

1）扫描线多边形填充算法

2）种子填充算法

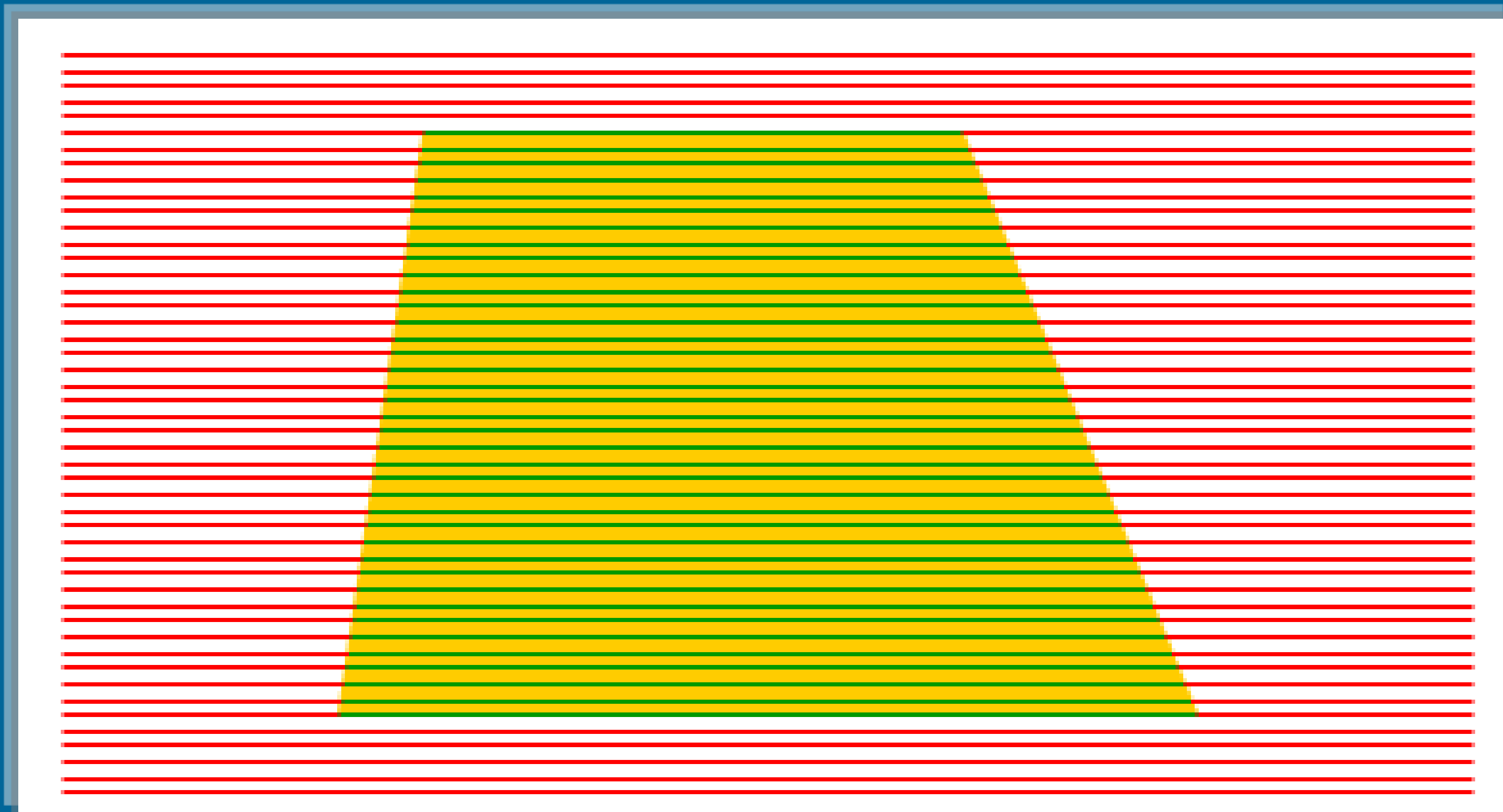
多边形

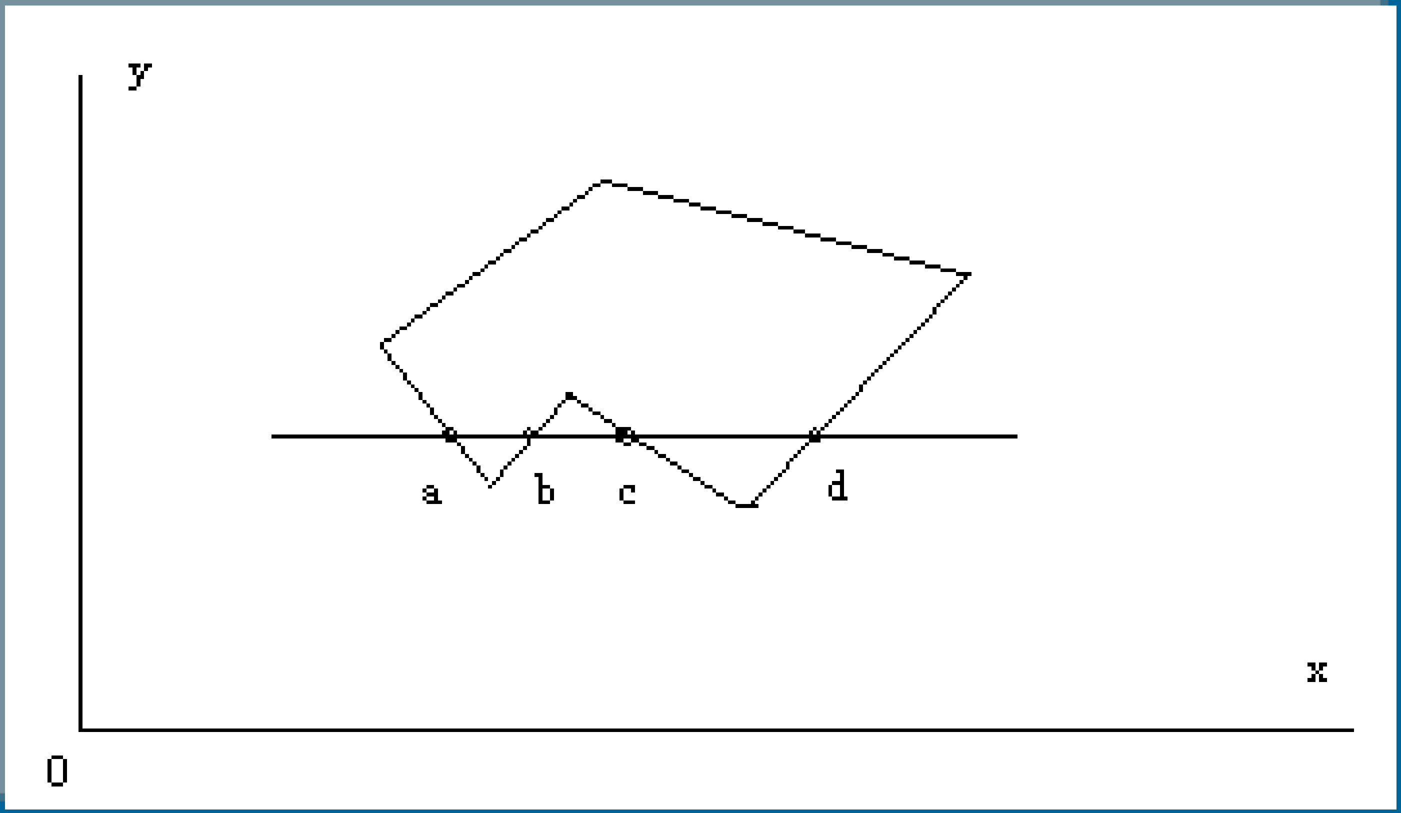
由一系列首尾相连的直线段构成的图形称为多边形。如果在多边形内任意选取不相同的两点，其连线上的所有点均在该多边形内，这样的多边形称为凸多边形；否则，称为凹多边形。



扫描线多边形填充算法

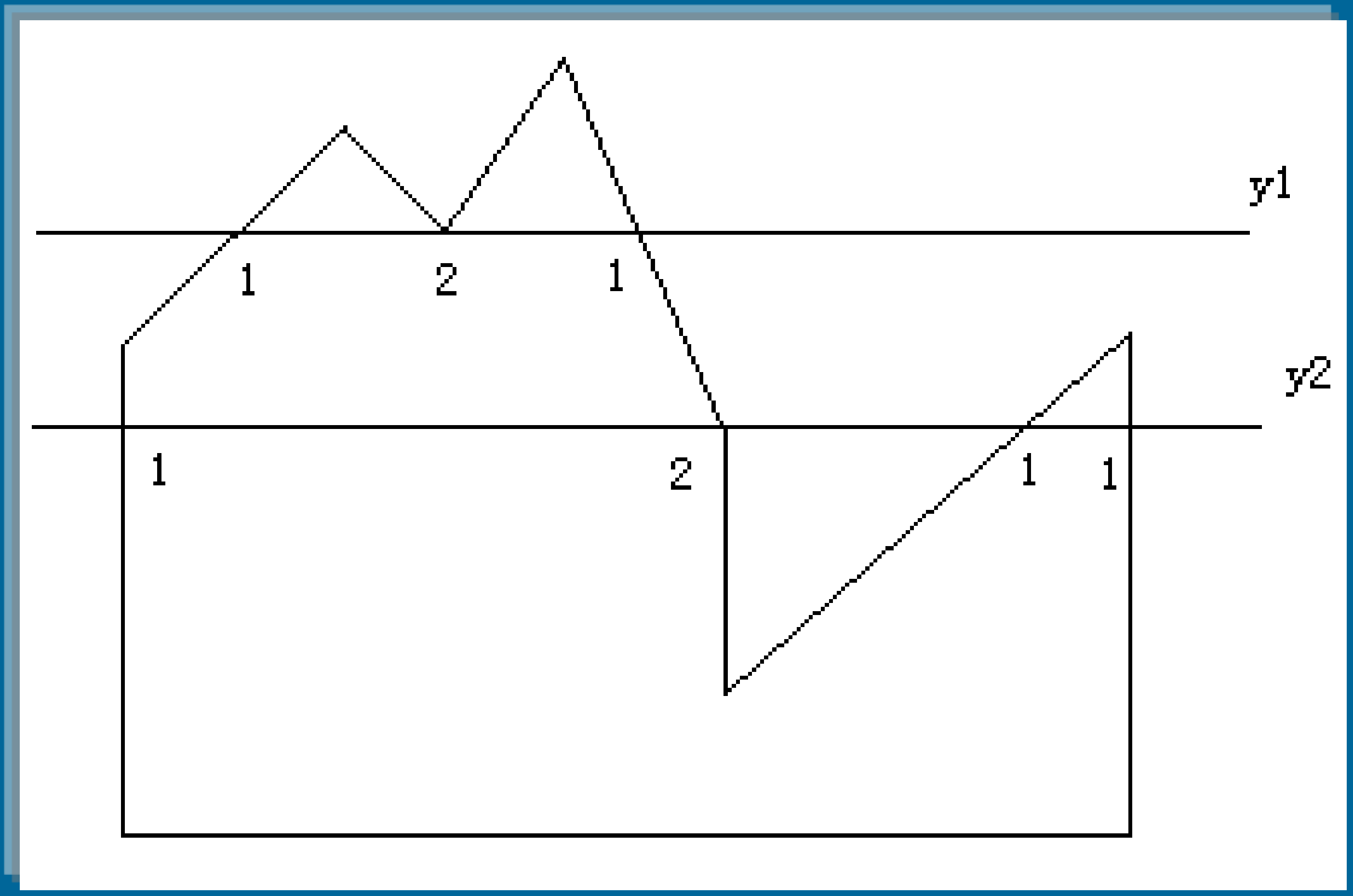
扫描线填充算法是一个常用的多边形填充算法。其基本思想是：对于一个给定的多边形，用一组水平或垂直的扫描线进行扫描，分别求出每条扫描线与多边形的交点，这些交点将扫描线分割为相间排列的落在多边形内和多边形外的线段，将落在多边形内的所有线段上的每个像素点赋以给定的多边形填充色。



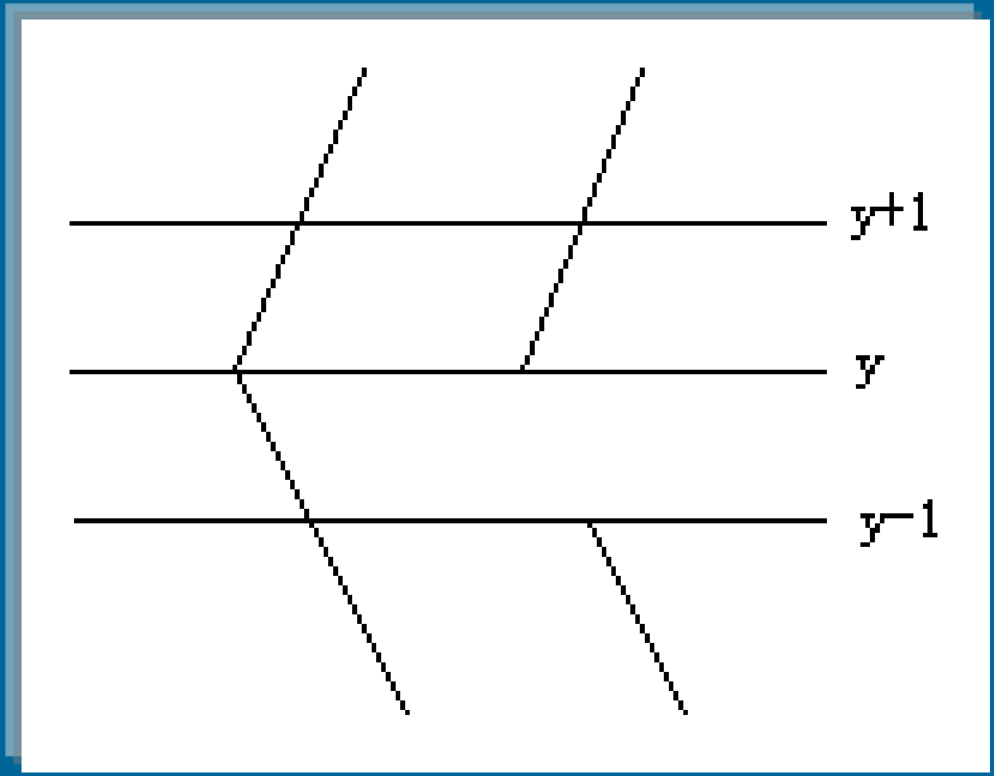


y
a
b
c
d
x
0

有一些扫描线会通过多边形的顶点，这时需要特殊处理。



解决将顶点计为一个或者两个交点的问题常用的一种方法是将多边形某些边缩短。



注意：由于是在像素级别，因此视觉上还是连续的

2）种子填充算法

种子填充算法又称为边界填充算法。其基本思想是：从多边形区域的一个内点开始，由内向外用给定的颜色画点直到边界为止。如果边界是以一种颜色指定的，则种子填充算法可逐个像素地处理直到遇到边界颜色为止。

种子填充算法常用四连通域和八连通域技术进行填充操作。

像素的邻域

4-邻域

	r	
r	p	r
	r	

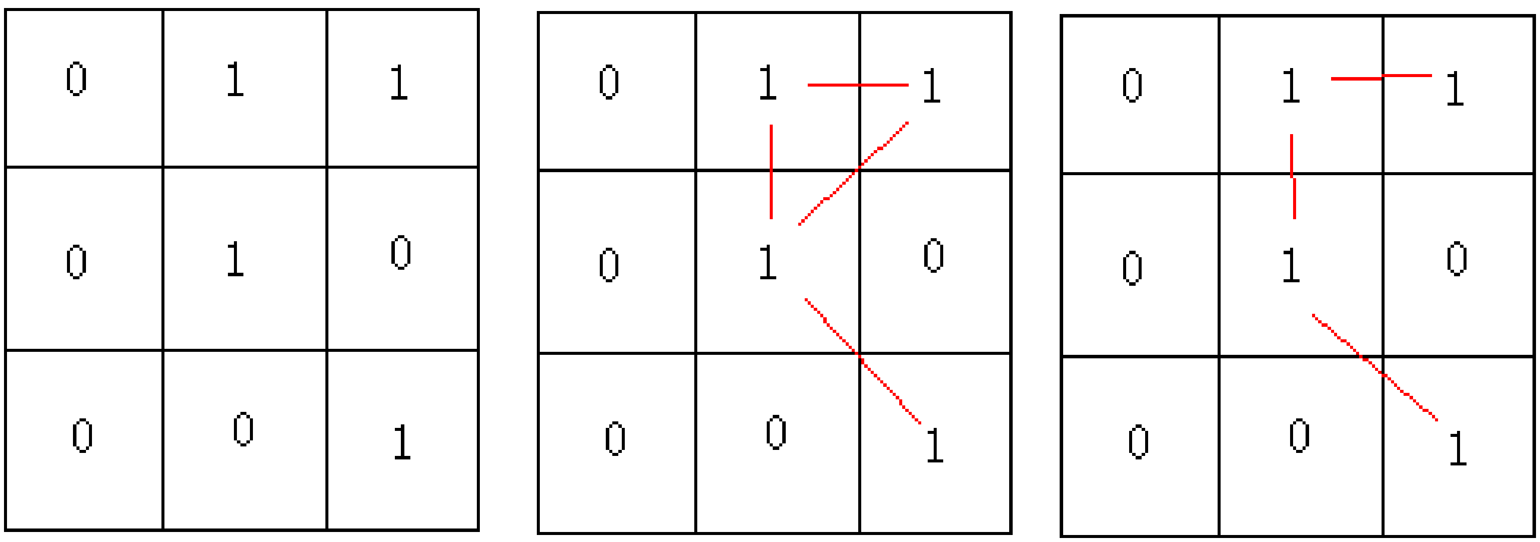
8-邻域

s	r	s
r	p	r
s	r	s

像素的邻域

连通性

在一幅二值图中（0和1），2个4-邻域像素只有在它们具有相同的灰度值时才可以说是连通的



像素间的联接

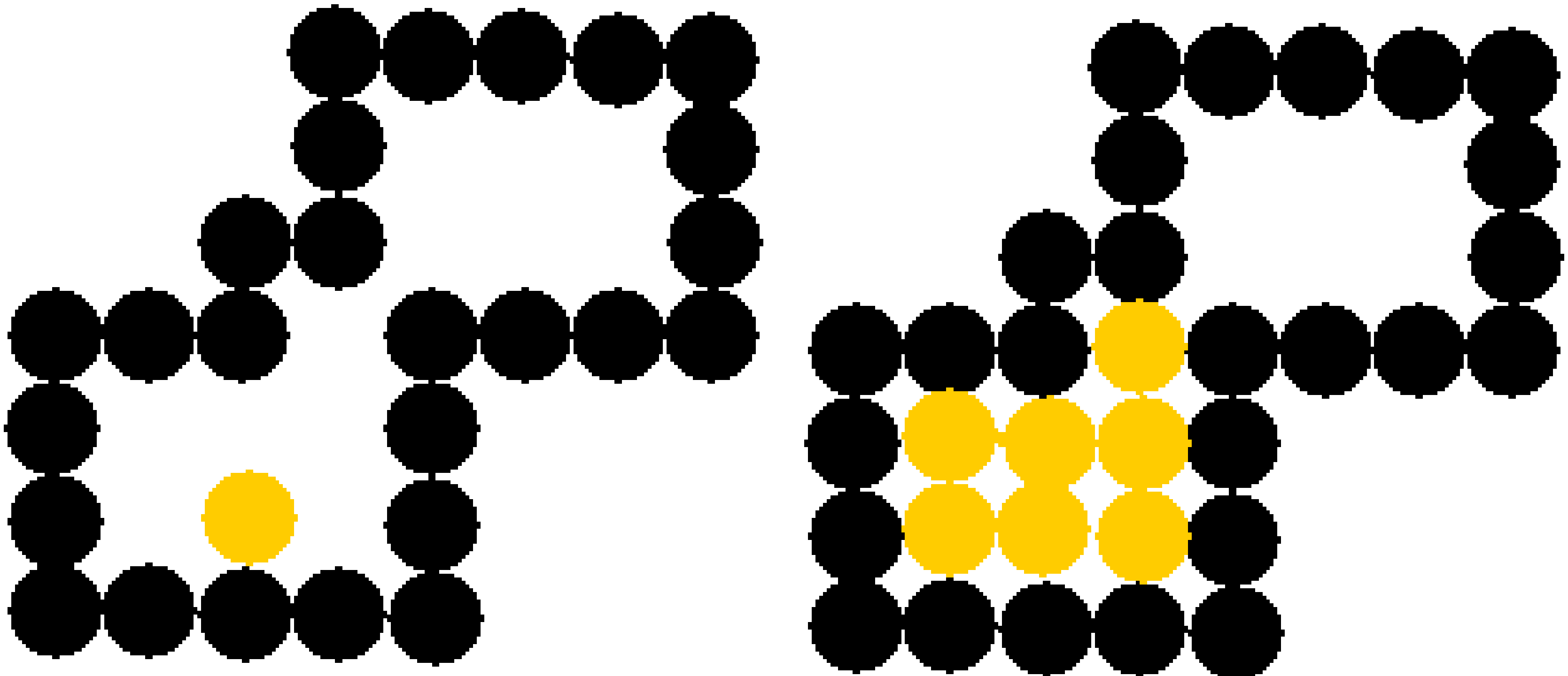
四连通域

从区域内任意一点出发，通过上、下、左、右四个方向到达区域内的任意像素。用这种方法填充的区域就称为四连通域；这种填充方法称为四向连通算法。

八连通域

从区域内任意一点出发，通过上、下、左、右、左上、左下、右上和右下八个方向到达区域内的任意像素。用这种方法填充的区域就称为八连通域；这种填充方法称为八向连通算法。

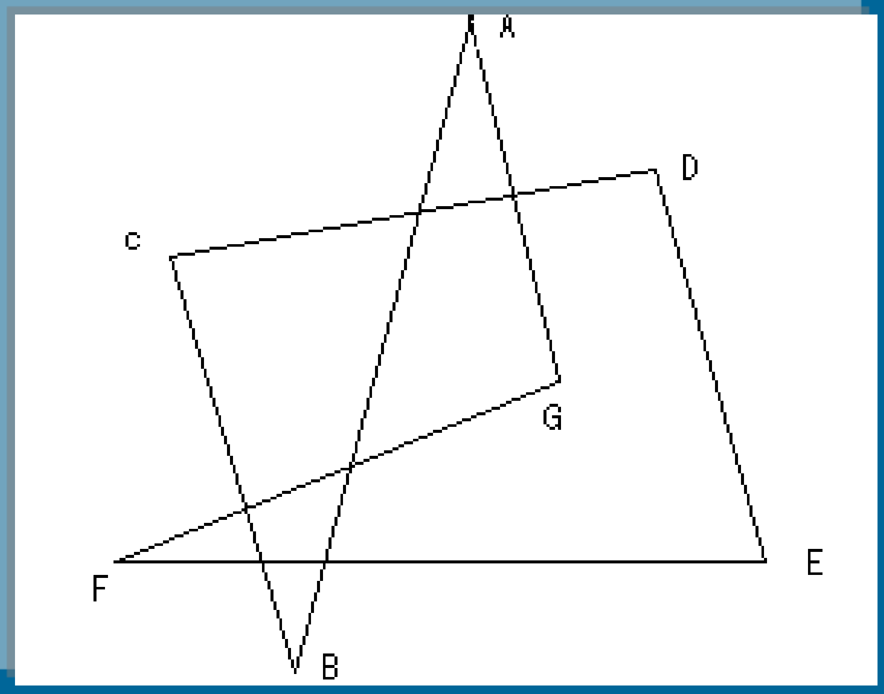
四向连通填充算法



a）连通域及其内点　　b）填充四连通域

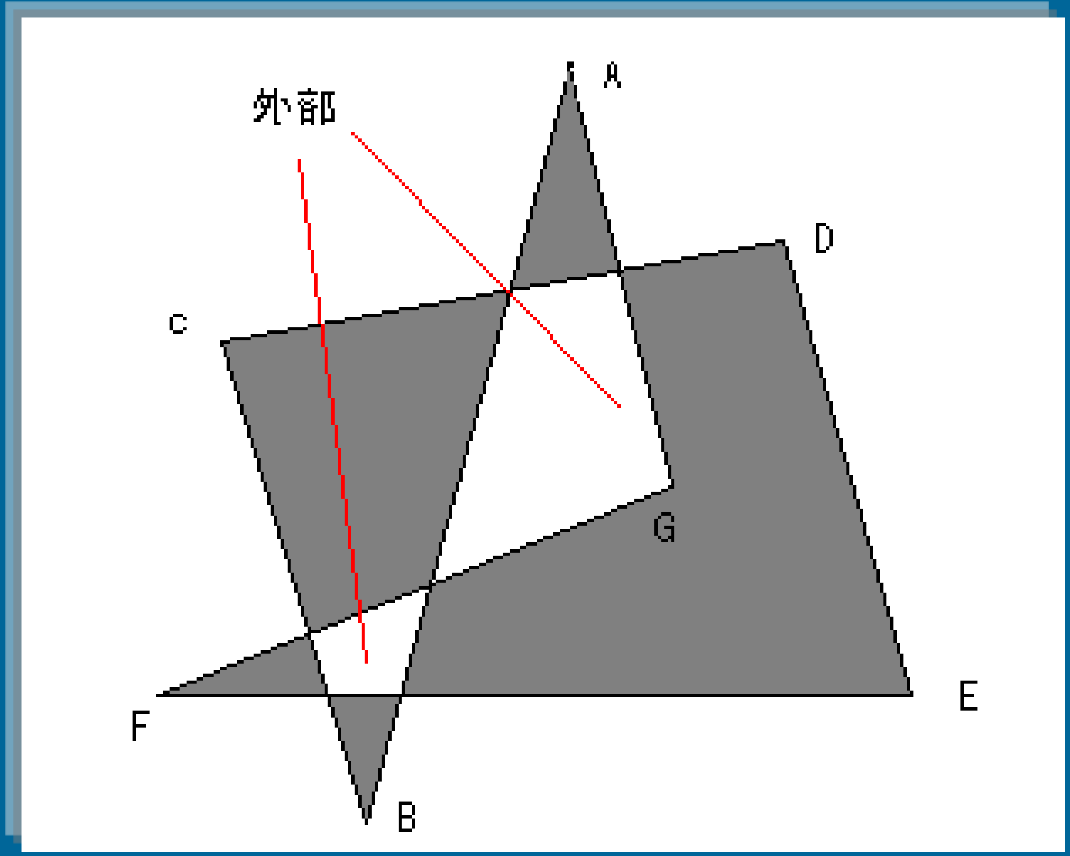
内—外测试

在基本的几何形体中，通常定义多边形为不自交。标准多边形有三角形，四边形、八角形等等。这些多边形的组成边仅在顶点出相连，而在平面内没有其他公共边。对于有些多边形哪个区域为“内侧”和“外侧”则并非是一目了然。



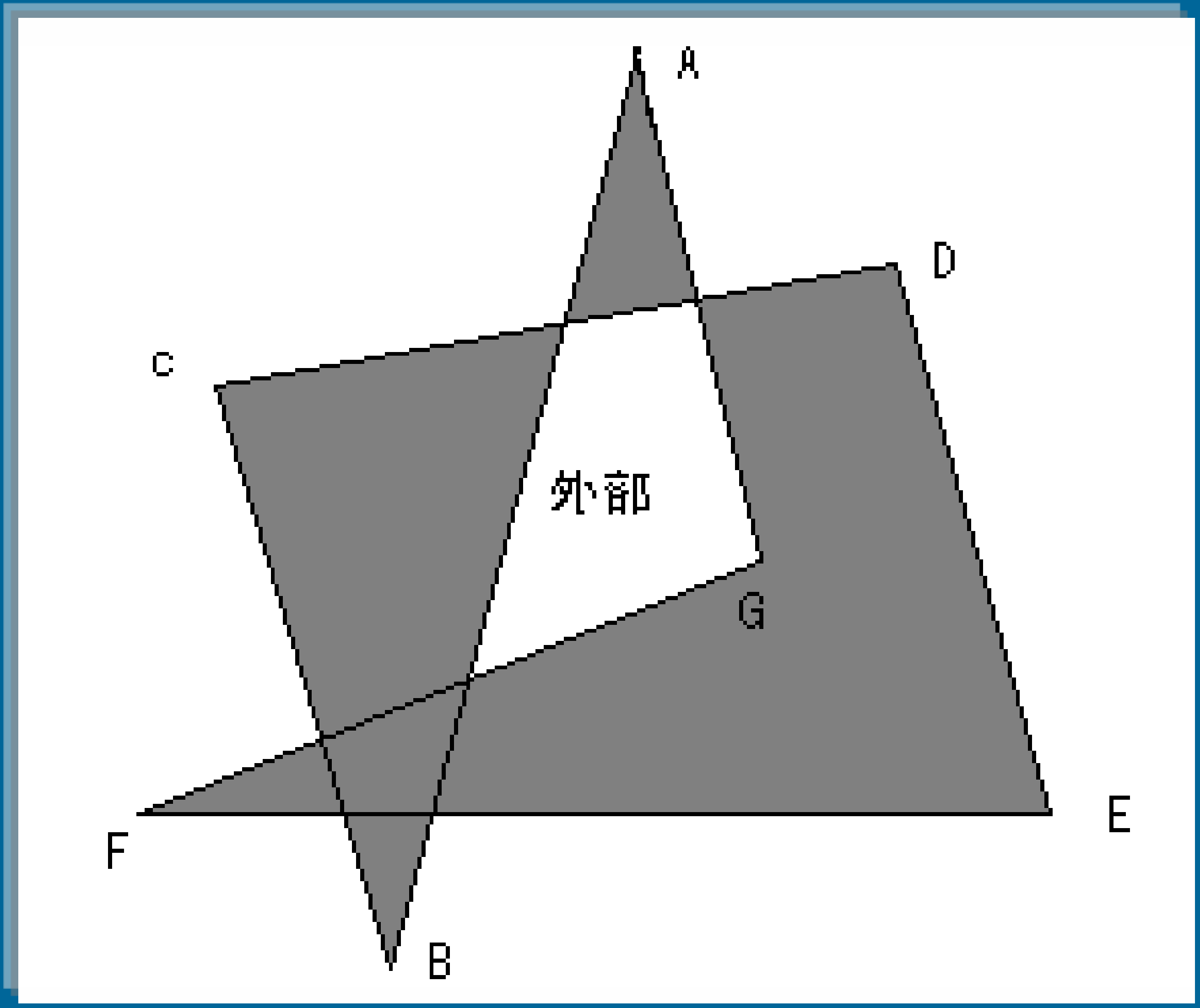
奇数规则

也称奇偶性规则或者奇偶规则。从概念上讲，该规则从任一位置P到对象坐标范围一位的远点划一条线（射线），然后统计该射线与各边的焦点数目。假如交点数为奇数，则P为内部点，否则P为外部点。为了得到精确的边数，必须确认所花的直线不与任何多边形顶点相交。



非零环绕数（nonzero winding number）

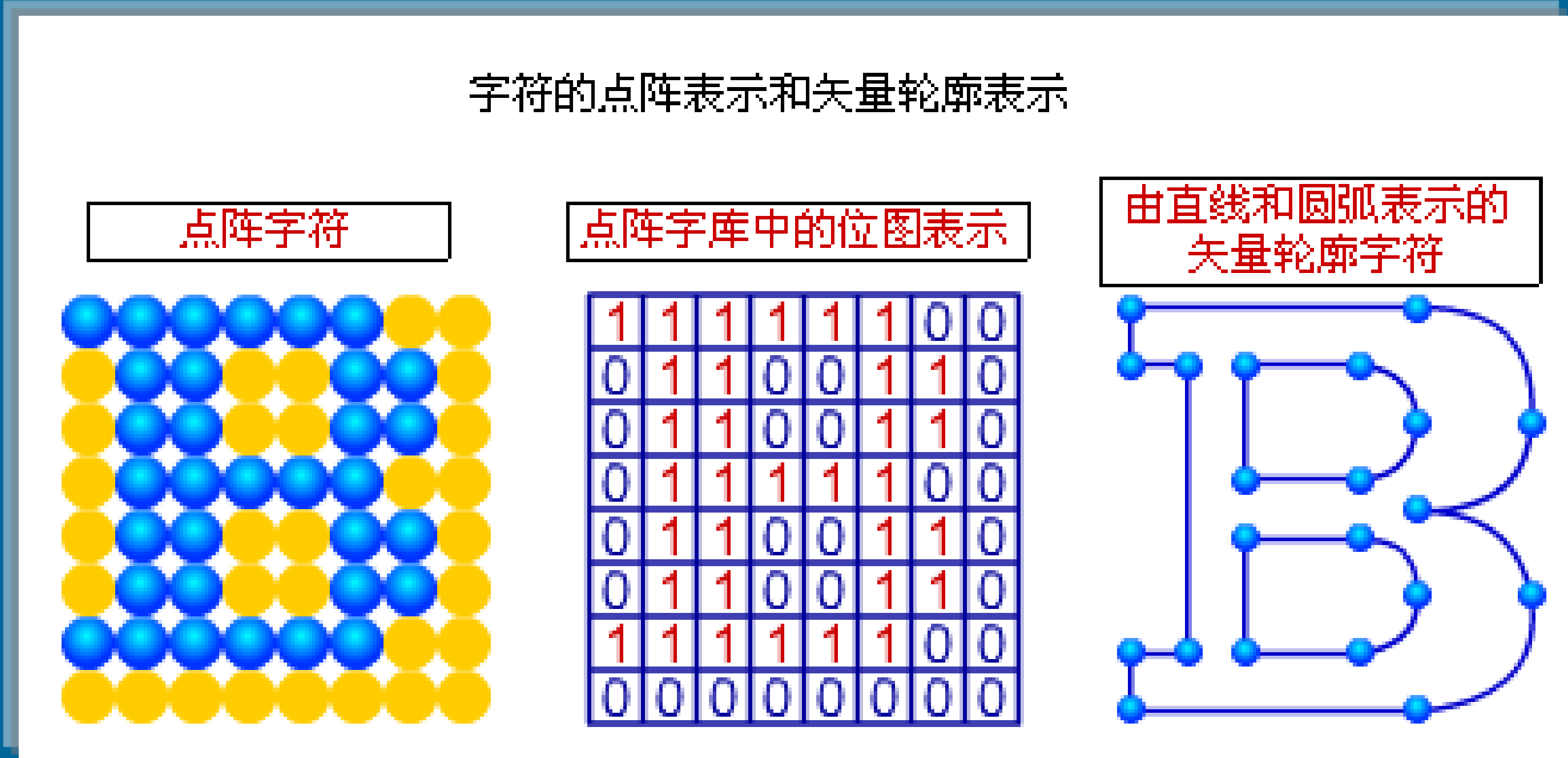
该方法统计多边形以逆时针方向环绕某一特定点的次数。这个数称为环绕数。将二维物体的内部点定义为具有非零值的环绕数。在对多边形应用非零环绕数规则时，将环绕数初始化为零。并假设从任意位置P画一条射线，所选择的射线不能与多边形的任意定点相交。当多边形从右至左穿过射线时，边数加1；从左至右时，变数减去1。在所有穿过的边都已经计数后，环绕数的最后值决定P的相对位置。环绕数为非零，则P将定义为内部点，否则为外部点。



环绕数为非零，则P将定义为内部点，否则为外部点。

2.7 字符生成

为了在显示器等输出设备上输出字符，计算机系统中必须安装相应的字库。字库分为点阵字库（位图字体）和矢量字库（轮廓字体）两种，用于存储每个字符的形状信息。



5）光栅图形反走样基础

Bresenham直线算法生成的直线图形一般都呈阶梯状，实际上这是光栅图形的一种走样现象。这种走样现象是由于采用离散量表示连续量引起的。通常，我们把由离散量表示连续量引起的失真称为走样；把减少或克服走样效果的技术称为反走样技术，简称反走样。

光栅图形的走样有如下几种：

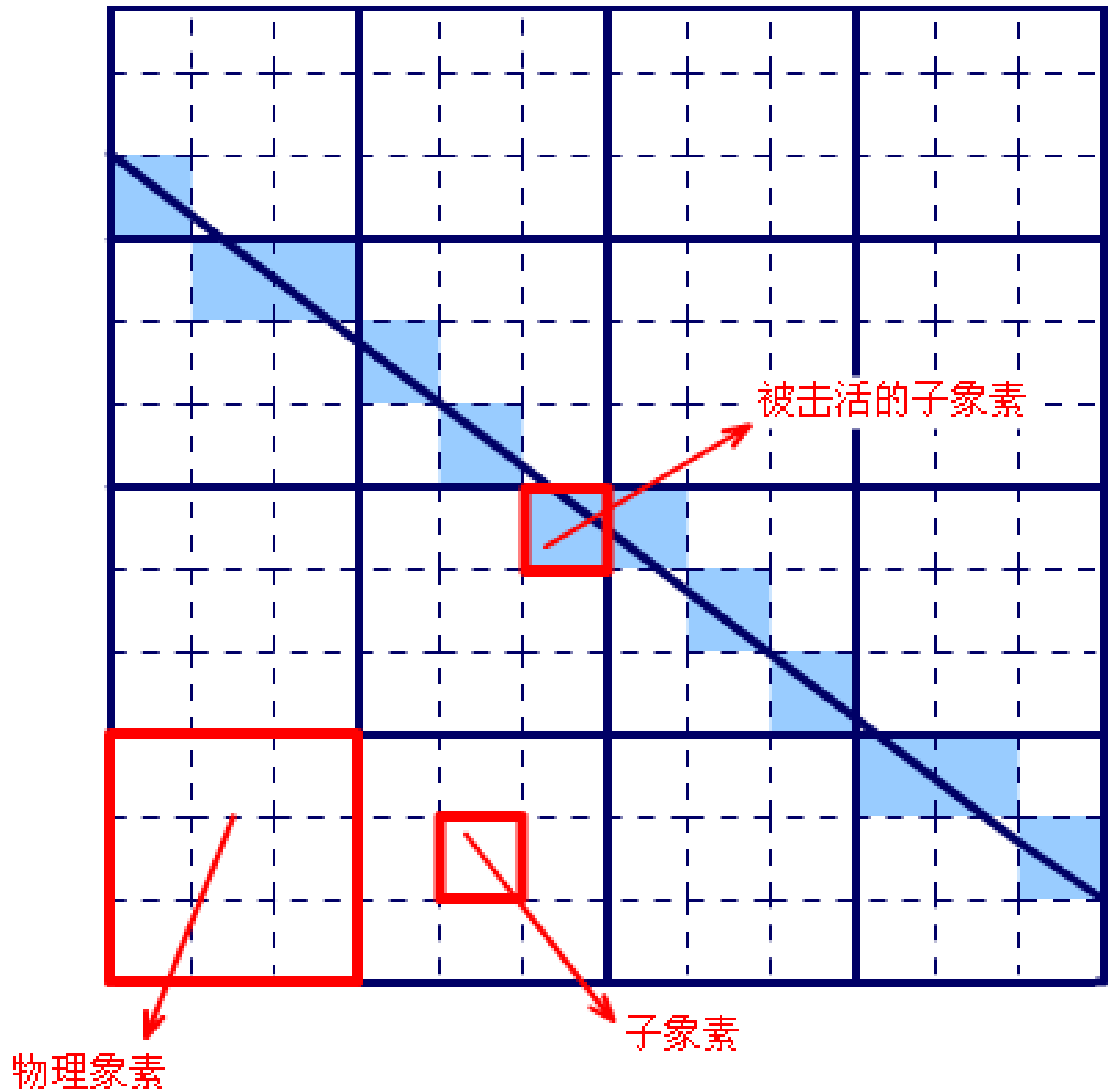
a) 产生阶梯或锯齿形；
b) 细节或纹理绘制失真；
c) 狭小图形遗失；
d) 实时动画忽隐忽现、闪烁跳跃。

反走样原理：

对于能显示两级以上亮度（颜色或者灰度等级）的光栅系统，可以使用反走样方法来修改图像亮度。通过适当地改变图元边界的像素亮度，可以平滑边界以减小锯齿现象。

光栅图形的反走样方法主要有两类：

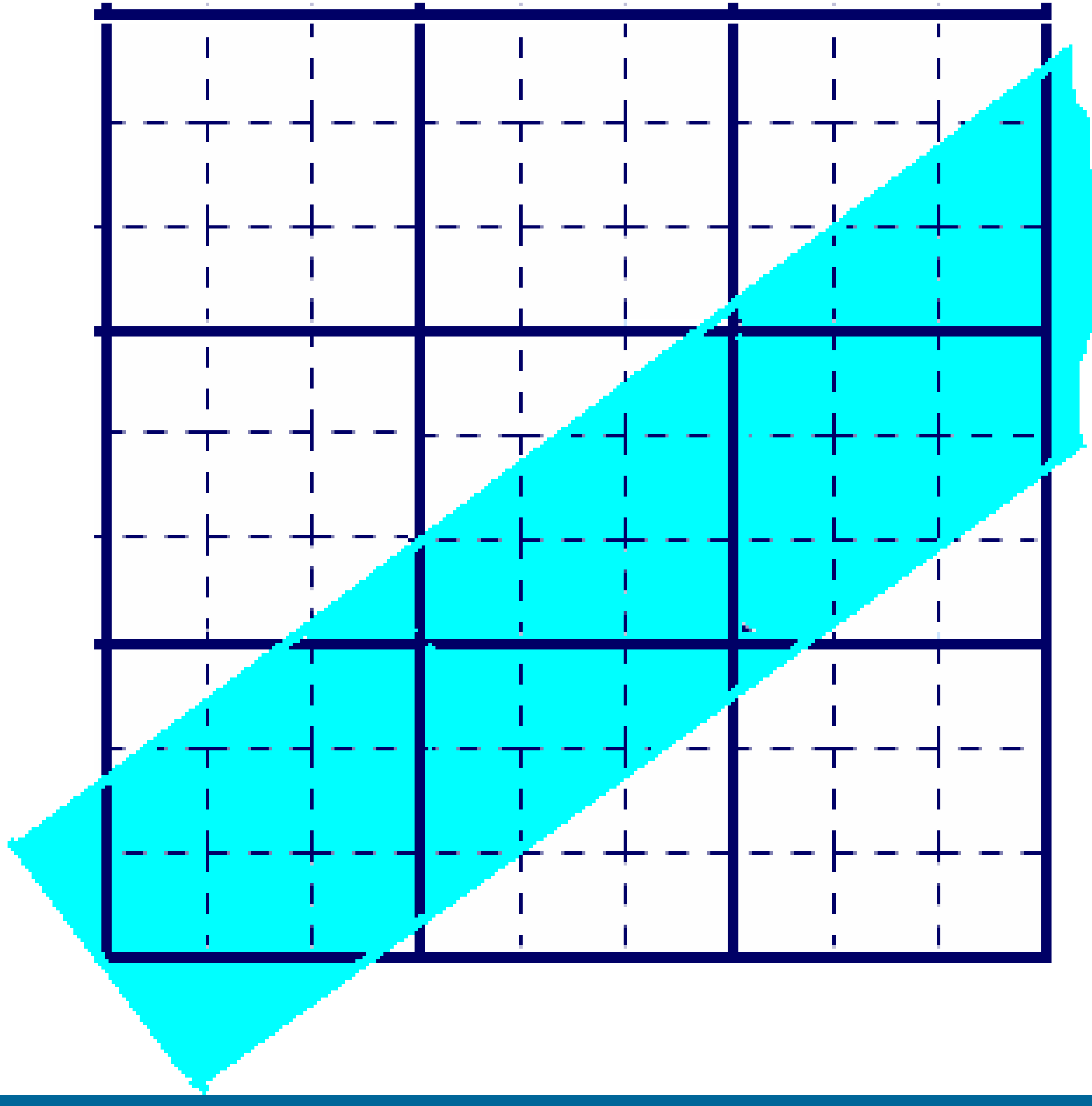
第一类是超采样（过取样）或称后置滤波。这类算法的基本思想着眼于提高分辨率，虽然采用高分辨率的光栅图形显示器也是一个选择，但它受到客观条件的限制，而且也不经济。因此，我们往往采用软件实现的方法，即：将低分辨率的图形像素划分为许多子像素，在较高分辨率上对各子像素的颜色值或灰度值进行计算，然后采用某种平均算法，将原像素内的各子像素的颜色值或灰度值的平均值作为该像素显示的颜色值或灰度值，在较低分辨率的光栅图形设备上进行显示。



反走样细直线的超采样

光栅图形的反走样方法主要有两类：

第二类方法称为前置滤波（区域取样）。即：通过计算待显示的每一个像素在对象上的讣告区域，从而确定像素亮度。像素覆盖区域通过确定对象边界与单个像素边界的相交位置而得到。注意这时像素是有几何尺寸的。



课后习题

- 1）常用的区域填充方法有：（ **BC** ）
- A **Bresenham**算法
- B 扫描线算法
- C 种子填充算法
- D **Cohen-SutherLand** 算法

- 2）光栅图形的走样有如下几种：（ **ABCD** ）
- A 产生阶梯或锯齿形；
- B 细节或纹理绘制失真；
- C 狭小图形遗失；
- D 实时动画忽隐忽现、闪烁跳跃。

- 3）输出一条任意斜率的直线，其显示效果一般受到哪些因素影响？（ **ABCD** ）
- **A** 光栅图形显示器的分辨率；
- **B** 线宽、线型；
- **C** 显示器的所能显示的颜色或灰度级别；
- **D** 直线的扫描转换的算法。

- 4）在计算机图形学中，字符作为一个基本图元，按照存储形式的不同，可以分为位图字体和轮廓字体两类。
- 5）在计算机图形学中，多边形分为凸多边形和凹多边形两类。
- 6）像素的连接方式有4-连接、8-连接和混合连接三种类型

7）请说明DDA画线算法的算法思想。若直线起始点在终止点的左侧，请写出DDA画线算法的算法实现步骤。

1） 如果斜率 m 小于 1，则存在如下递推公式：

$$x_{k+1} = x_k + 1$$

$$y_{k+1} = y_k + m$$

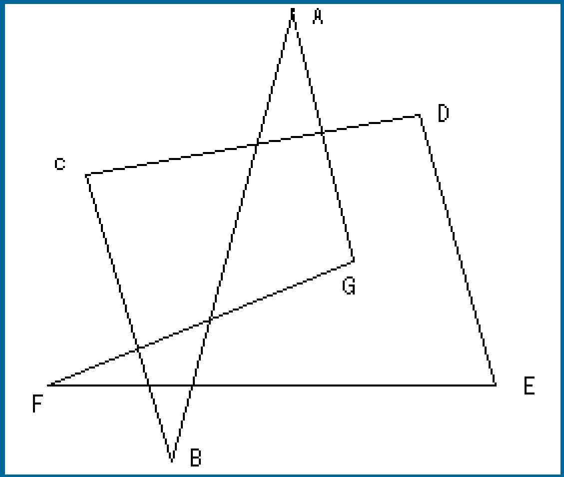
k 取整数值，从第一个点 1 开始递增直到最后一个端点，m 是 0 和 1 只见的任意实数，所以计算的 y 必须取整

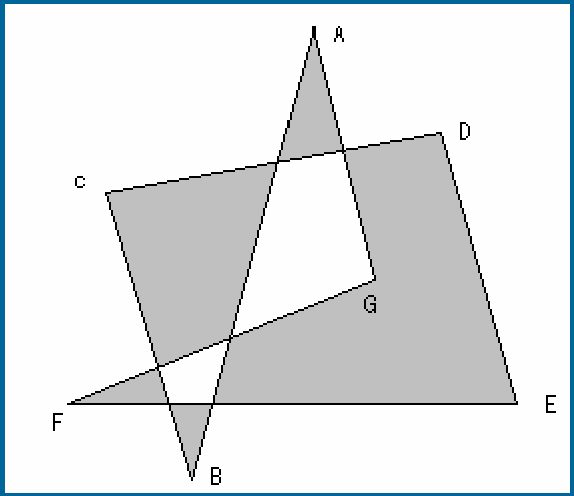
2） 如果斜率 m 大于 1，则存在如下递推公式：

$$x_{k+1} = x_k + \frac{1}{m}$$

$$y_{k+1} = y_k + 1$$

8）试说明在进行区域填充时，用来判别复杂多边形内侧与外侧点的奇数规则的算法思想。并且利用奇数规则标出下面这个多边形的内-外侧。





答：奇数规则也称奇偶性规则或者奇偶规则。从概念上讲，该规则从任一位置P到对象坐标范围一位的远点作一条线（射线），然后统计该射线与各边的交点数目。假如交点数为奇数，则P为内部点，否则P为外部点。为了得到精确的边数，必须确认所画的直线不与任何多边形顶点相交。

9）试说明直线段的过取样反走样算法原理。

- 答：通过适当地改变图元边界的像素亮度，可以平滑边界以减小锯齿现象，从而减小直线的走样现象。过取样算法原理如下：将每个像素分成n×n个子像素，然后在子像素级对直线进行光栅化，这样就可以得到每个像素中被激活的子像素的个数。在n×n伪光栅上，可以光栅化的子像素最多为n个。每个物理像素的光强与其被激活的子像素数与n的比值成正比。假设一个物理像素中被激活的子像素有m个，其可能的最大光强为Imax，则该像素的光强（亮度）$I = \frac{m}{n} I_{max}$

10）解释一下光栅图形走样产生的原因，以及反走样的含义。

- 答：光栅图形的走样现象是由于采用离散量表示连续量引起的。通常，我们把由离散量表示连续量引起的失真称为走样；把减少或克服走样效果的技术称为反走样技术，简称反走样。

- 11) 在光栅图形设备上显示一个点时，实际上它是用一个发光的矩形区域来表示的。

- 12) Bresenham算法使得每次只需检测决策参数的符号就能决定直线上的下一个像素的位置。

- 13) 为了提高了画圆速度，中点画圆算法以点（0，R）为起点按顺时针方向生成圆，则其他七个分圆中的位置可以由对称性得到。。

- 14）请解释一下光栅图形学的含义？

  光栅图形显示器可以看成是由许多可发光的离散点（即像素）组成的矩阵，它需要专门的算法来生成直线、圆弧和曲线等等图形。

- 15）常用的画线算法有：（ AC ）
  - A Bresenham算法
  - B 扫描线算法
  - C DDA算法
  - D Cohen-SutherLand 算法

- 16）带自交的多边形内外-侧检验方法有<u>奇数规则</u>和<u>非零环绕数</u>两种。
- 17）常用的区域填充算法有哪些，各有什么特点？
  - 1）扫描线多边形填充算法
  - 速度快，适合于填充多边形、圆、椭圆等简单图形；
  - 2）种子填充算法
  - 填充有复杂形状边界的图形，多用于交互式涂色系统。

- 18）常用的光栅图形的反走样方法主要有两类：超采样（后置滤波）和区域取样（前置滤波）。

第三讲 二维图形变换和显示

3.1　二维几何变换
3.2　二维观察
3.3　裁剪

1）齐次坐标

齐次坐标表示法，就是用n+1维向量表示一个n维向量，即n维空间中的点的位置向量$(P_1, P_2, \ldots P_n)$被表示为具有n+1个坐标分量的向量$(wP_1, wP_2, \ldots, wP_n, w)$。例如：二维空间坐标（x，y）的齐次坐标表示为[X，Y，w]，则x=X/w，y=Y/w。

对于平移变换，则有：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{T}_t \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \text{其中，} \quad \mathbf{T}_t = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}.$$

统坐标原点的旋转变换也可表示为：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad P' = R(\theta)\cdot P$$

相对于坐标原点的缩放变换表示为：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad P' = S(s_x, s_y)\cdot P$$

2）复合变换

利用齐次坐标，通过计算单个变换的矩阵乘积，将任意的变换序列组合成复合变换矩阵。形成变换矩阵的乘积经常称为矩阵的连结或复合。对于坐标位置的列矩阵表达式，通过以从右向左的顺序进行矩阵相乘而形成复合变换，即每个随后的变换矩阵左乘前面的变换矩阵乘积。

3）连接特性

由比例、平移和旋转变换构成的复合变换，仅在某些情况下交换次序不影响总的变换效果，它们是哪几种两个变换的组合。

（1） 两个连续的平移变换；
（2） 两个连续的比例变换；
（3） 两个连续的旋转变换；
（4） 比例系数相等的比例变换和旋转变换。

4）反射

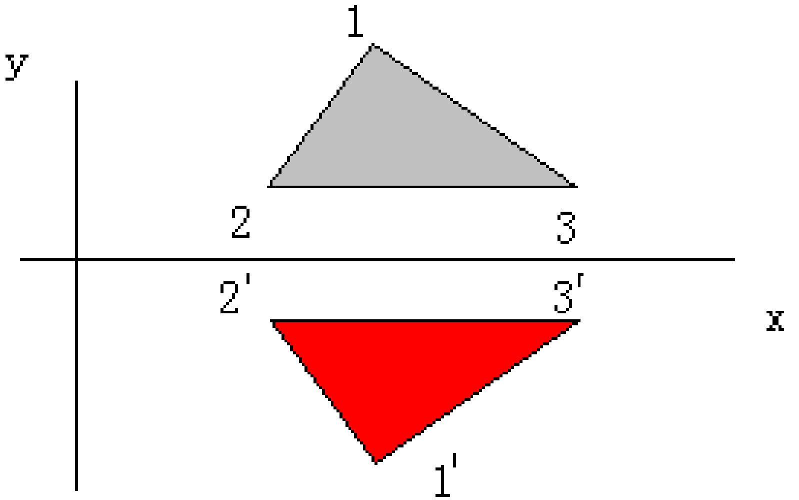
反射是产生对象的镜像的一种变换。相对于反射轴的二维反射镜像是通过将对象绕反射轴旋转180度而生成的。

对于x轴的反射

关于直线 y=0 的反射，可以由下列变换矩阵来完成：

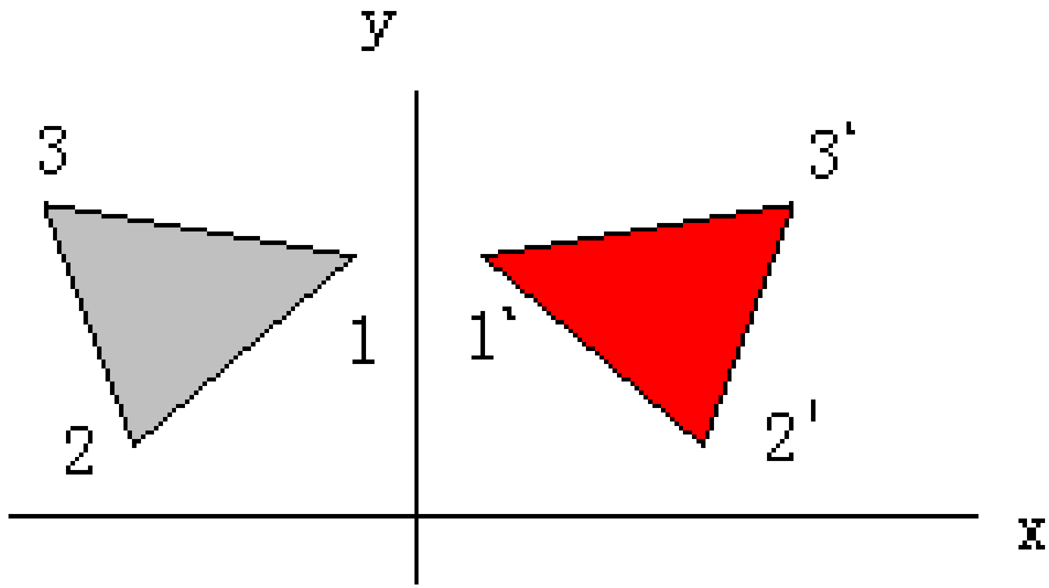
这个变换保持了 x 值的相同，但"翻动"y 坐标位置的值。对于这种变换，平面上的对象移出 xy 平面，通过三维空间绕 x 轴旋转 180 度再回到 x 轴另一侧的 xy 平面。

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



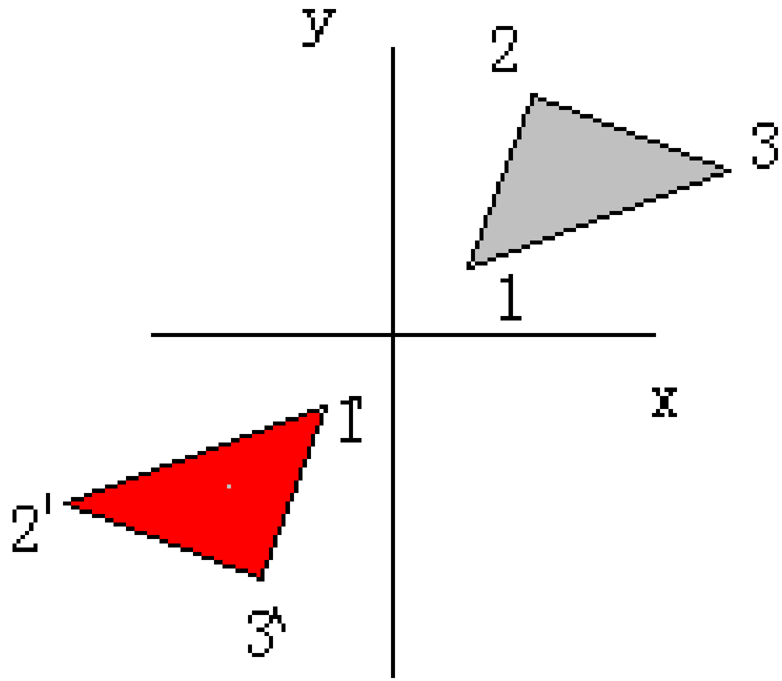
关于y轴的反射

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



对象相对于垂直xy平面并通过坐标原点的轴进行反射

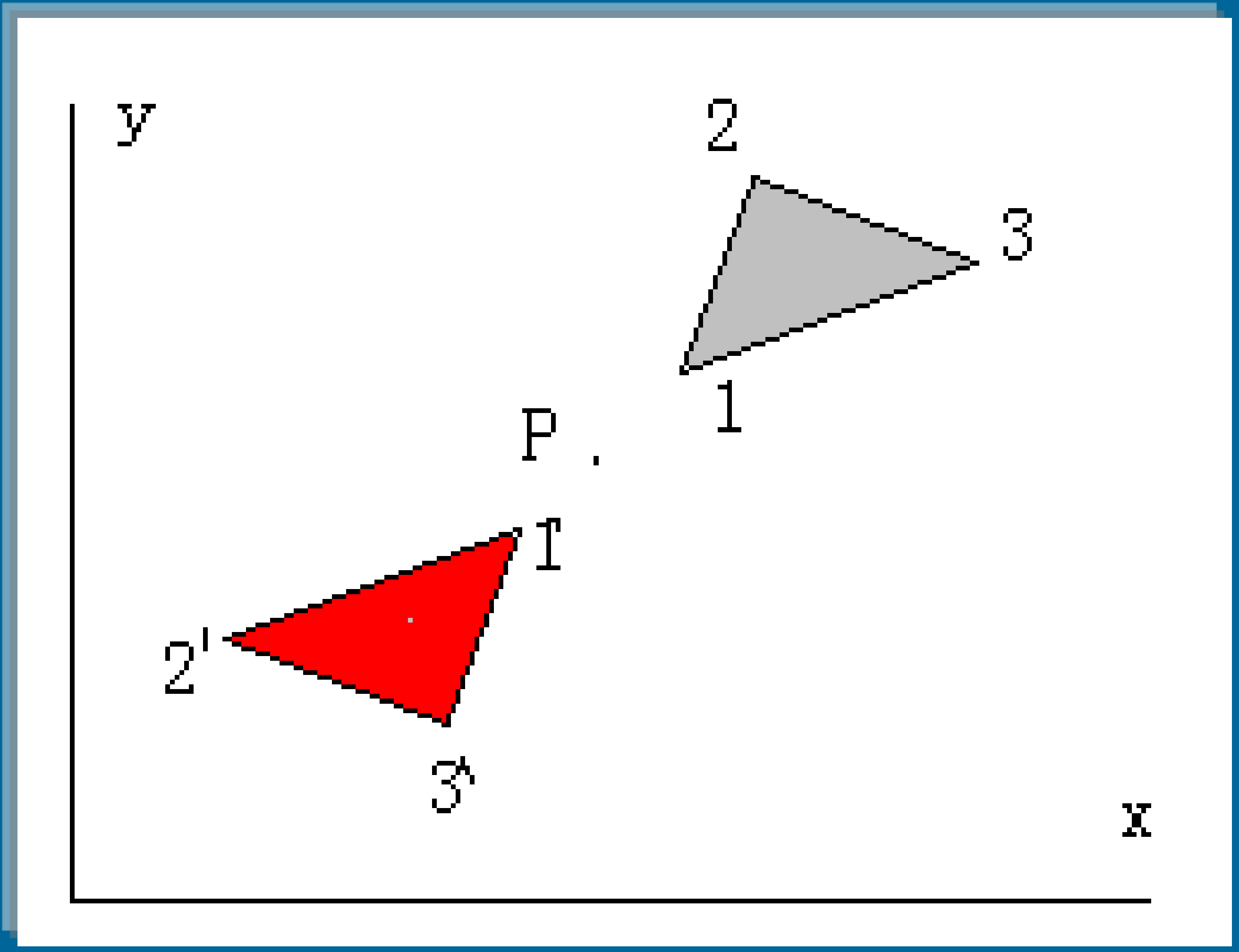
$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



上面的反射矩阵是 $\theta = 180^\circ$ 的旋转矩阵 $R(\theta)$，

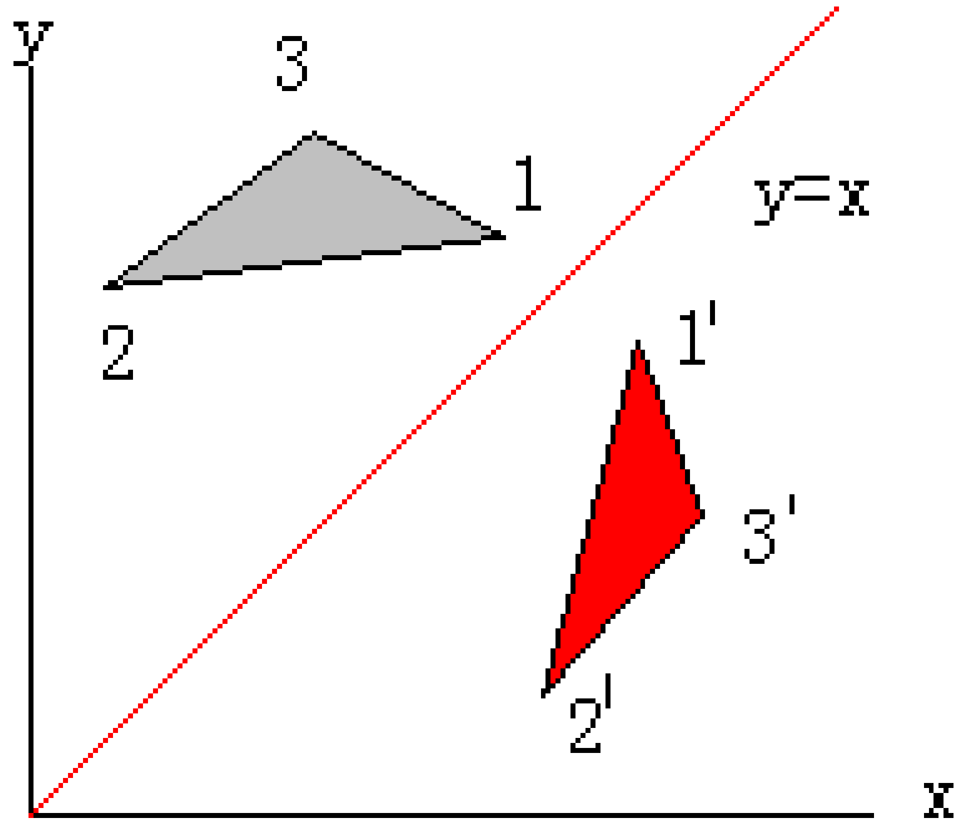
也就是将 xy 平面内的对象绕原点旋转半圆

对象相对于垂直xy平面并通过点P的轴的反射



对象对于直线y=x的反射

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



5）复合变换

对于坐标位置的列矩阵表达式，通过以从右向左的顺序进行矩阵相乘而形成复合变换，即每个随后的变换矩阵左乘前面的变换矩阵乘积。

平移复合

假如将两个连续的平移向量 (t_{x1}, t_{y1}) 和 (t_{x2}, t_{y2}) 用于坐标位置 P，那么最后的变换位置 P' 可以计算为：

$$P' = T(t_{x2}, t_{y2}) \cdot T(t_{x1}, t_{y1}) P = \begin{bmatrix} 1 & 0 & t_{x2} \\ 0 & 1 & t_{x1} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{x1} \\ 0 & 1 & t_{y1} \\ 0 & 0 & 1 \end{bmatrix} \cdot P$$

$$= \begin{bmatrix} 1 & 0 & t_{x1} + t_{x2} \\ 0 & 1 & t_{y1} + t_{y2} \\ 0 & 0 & 1 \end{bmatrix} \cdot P = T(t_{x1} + t_{x2}, t_{y1} + t_{y2}) P$$

旋转复合

应用于点 P 的两个连续旋转产生的变换位置为:

$$P' = R(\theta_2) \cdot R(\theta_1) \cdot P = R(\theta_2 + \theta_1) \cdot P$$

缩放复合

两个连续缩放操作的变换矩阵连接，将产生下列复合缩放矩阵：

$$\begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x1} \cdot s_{x2} & 0 & 0 \\ 0 & s_{y1} \cdot s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= S(s_{x2}, s_{y2}) \cdot S(s_{x1}, s_{y1}) = S(s_{x1} \cdot s_{x2}, s_{y1} \cdot s_{y2})$$

这种情况下的结果矩阵表明，连续缩放操作是相乘的，假如我们要连续两次将对象尺寸放大3 倍，那么其最后的尺寸将是原尺寸的 9 倍。

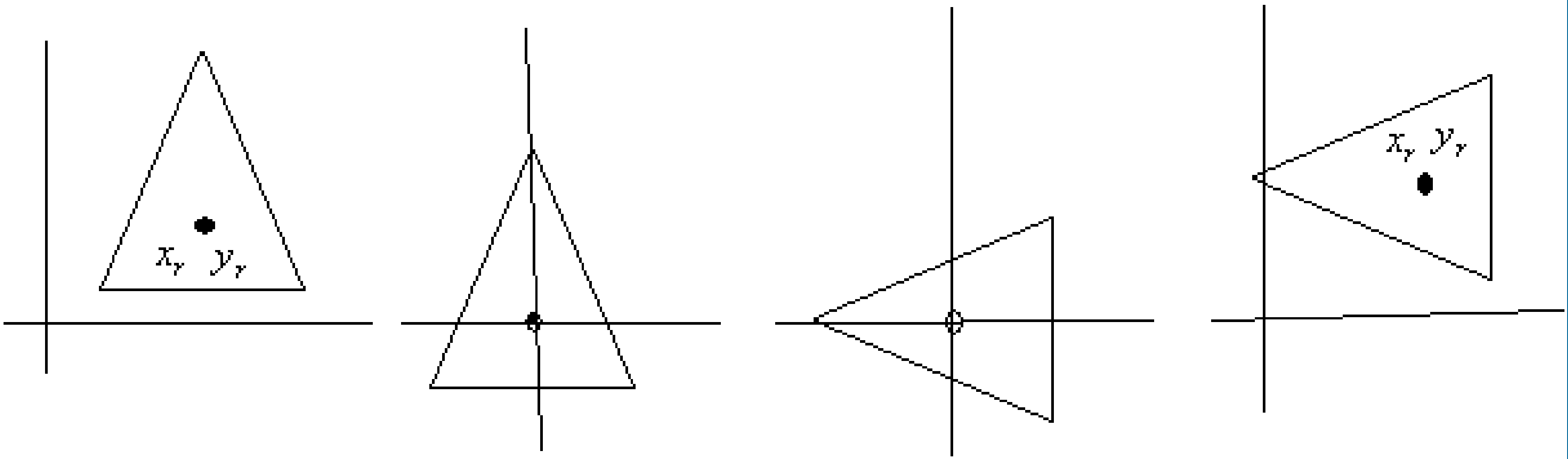
绕任意基准点的旋转

1） 平移对象使基准点位置移动到坐标原点；

2） 绕坐标原点旋转；

3） 平移对象使基准点回到其原始位置。

$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta)+y_r\sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta)-x_r\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$



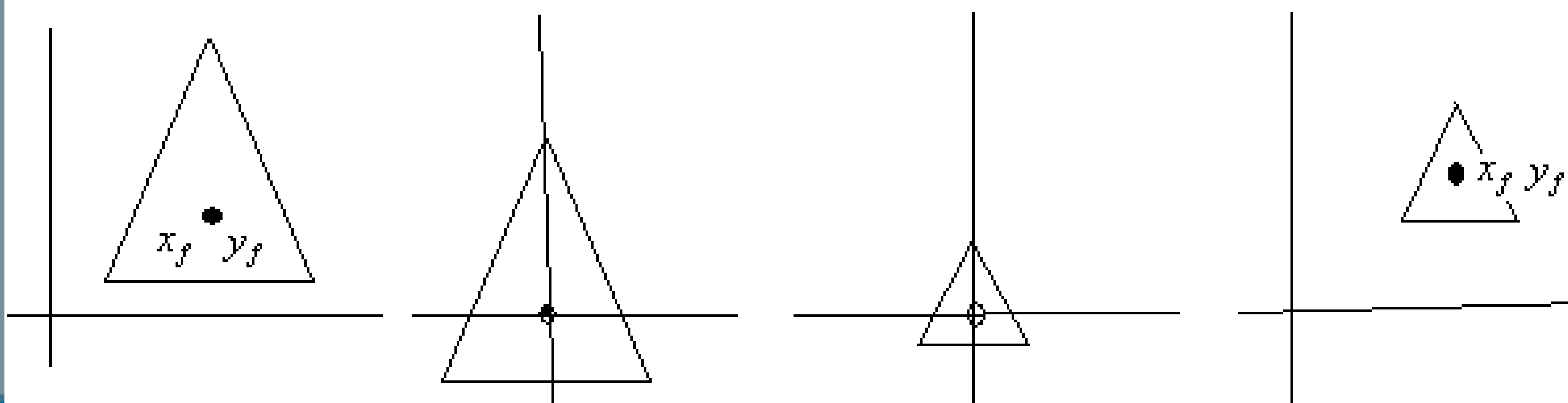
绕任意基准点的缩放

1） 平移对象使基准点位置移动到坐标原点；

2） 基于坐标原点缩放；

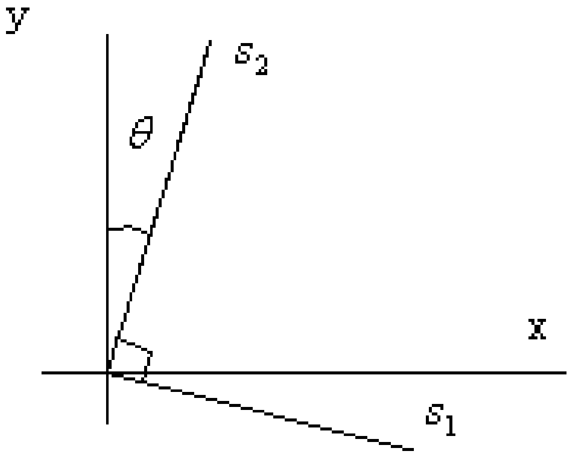
3） 使用步骤 1 的方向平移将对象返回到原始位置。

$$\begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}$$



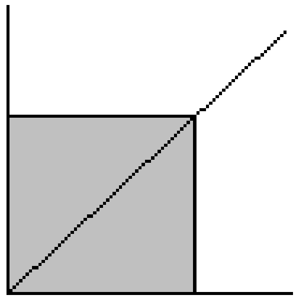
通用定向缩放

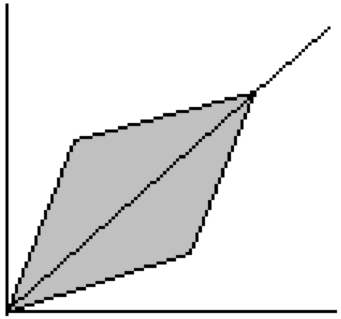
参数 s_x, s_y 沿着 x 和 y 方向缩放对象，可以通过在应用缩放变换之前，将对象所希望的缩放方向旋转到与坐标轴一致而在其他方向上缩放对象。假如我们要在下图所示的方向上使用 s_1, s_2 作为缩放系数，为了完成这种缩放而不改变对象方向，应当首先完成旋转操作，从而使 s_1, s_2 的方向分别与 x 和 y 轴重合，然后应用缩放变换，再进行反向旋转以回到其原始位置。其复合矩阵如下：$R^{-1}(\theta) \cdot S(s_1, s_2) \cdot R(\theta)$



例如：通过沿（0，0）到（1，1）的对角线将单位正方形拉长，使其转变为平行四边形。

这里使用的参数为$\theta=45^\circ, s_1=1, s_2=2$将对角线旋转到 y 轴，并将其长度加倍。





$$R(-\theta)\cdot S(s_1,s_2)\cdot R(\theta)$$

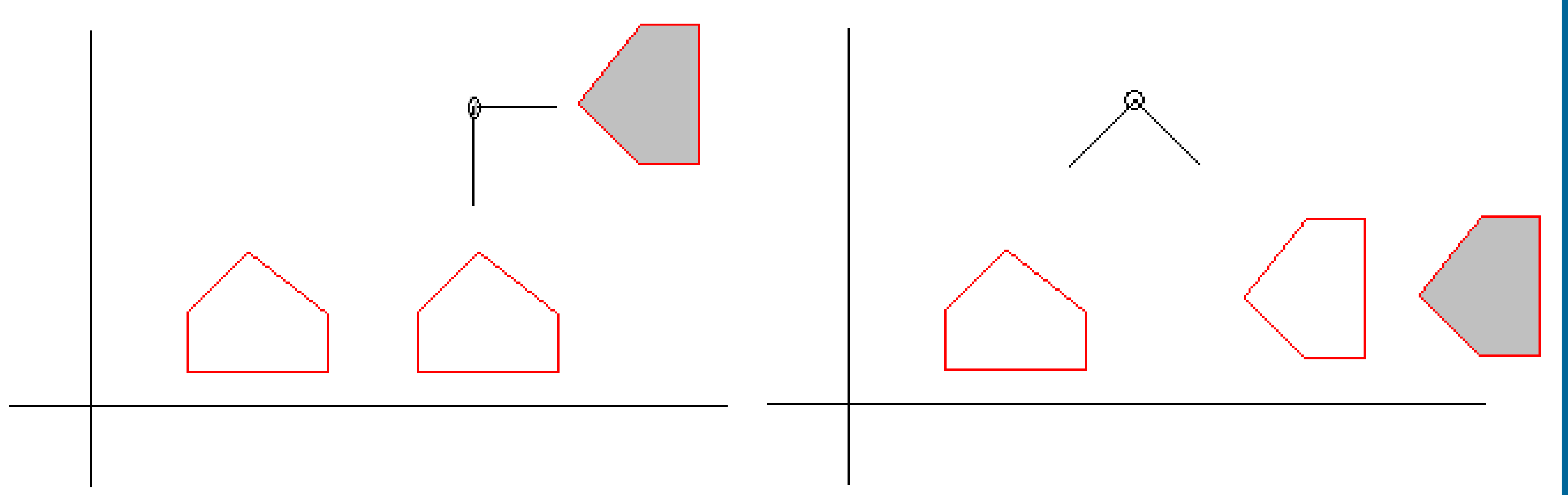
连接特性

矩阵相乘符合结合律：

$$A \cdot B \cdot C = (A \cdot B) \cdot C = A \cdot (B \cdot C)$$

但是，变换积通常不可以交换顺序，矩阵积 $A \cdot B$ 不等于 $B \cdot A$

这说明如果要平移和旋转对象，必须注意符合矩阵求值的顺序。



仿射变换（affine transformation）

形式为：

$$x' = a_{xx}x + a_{xy}y + b_x$$
$$y' = a_{yx}x + a_{yy}y + b_y$$

的坐标变换称为二维仿射变换。参数 a_{ij} 和 b_k 是由变换类型确定的常数变换的坐标都是原始坐标的线性函数。放射变换具有平行线变换成平行线、有限点映射到有限点的一般特性。

注意

平移、旋转、缩放、反射和错切是二维仿射变换的特例，任何常用的二维仿射变换总是可以表示为这5种变换的组合。仅包含平移、旋转和反射变换的仿射变换将保持角度、长度以及平行线。对于这三种变换、变换前后两直线间的角度和长度相同。

二维几何变换具有如下一些性质

- 直线的中点不变性
- 平行直线不变性
- 相交不变性
- 仅包含旋转、平移和反射的仿射变换维持角度和长度的不变性
- 比例变化可改变图形的大小和形状
- 错切变化引起图形角度关系的改变，甚至导致图形发生畸变。

光栅变换

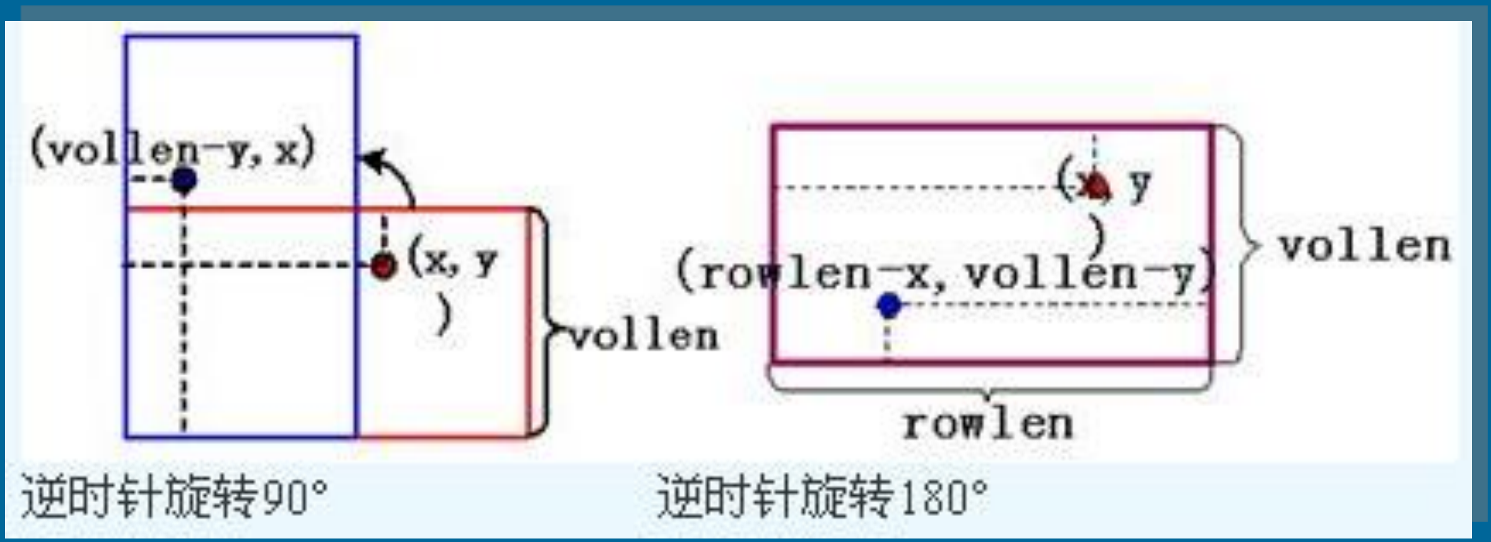
- 直接对帧缓存中像素点进行操作的变换称为光栅变换。

1. 光栅平移变换

- 通过像素块的移动来完成，即首先从光栅帧缓存中读出指定的像素块的内容，然后将像素块的内容复制到另一光栅区域，随后擦除原光栅区域中的像素块内容。

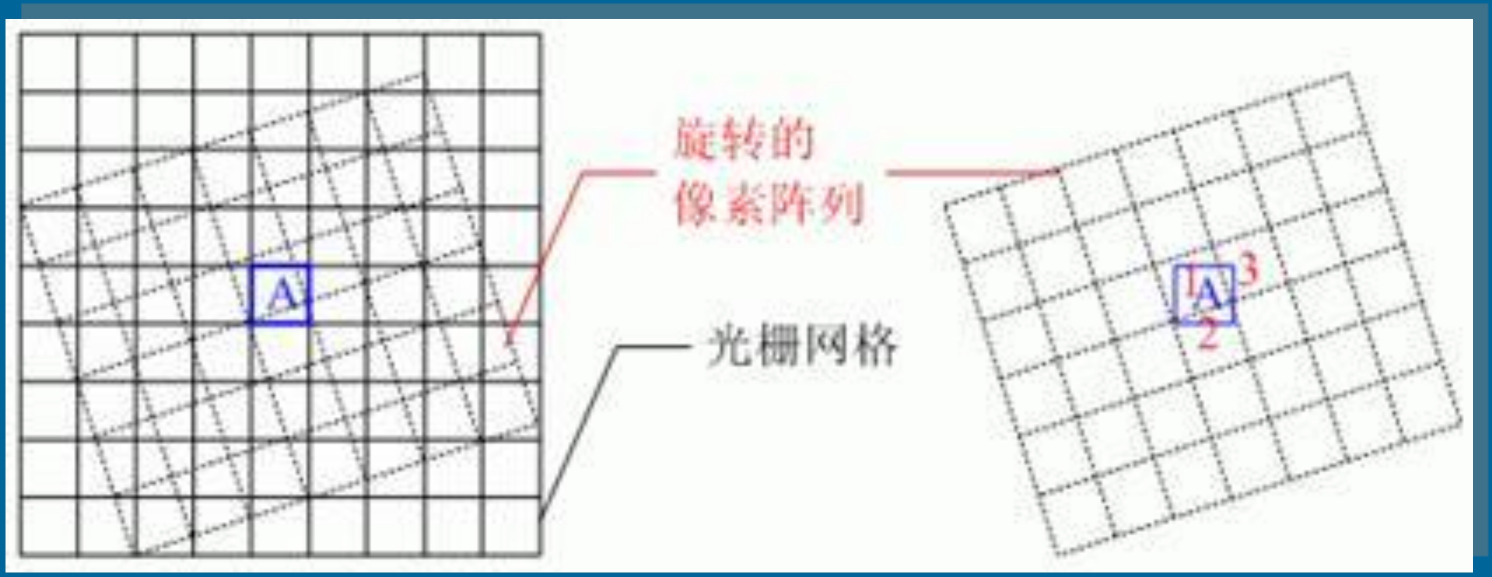
2. 90°、180°和270°的光栅旋转变换

- 利用像素块的移动还可容易地完成90°、180°和270°的光栅旋转变换



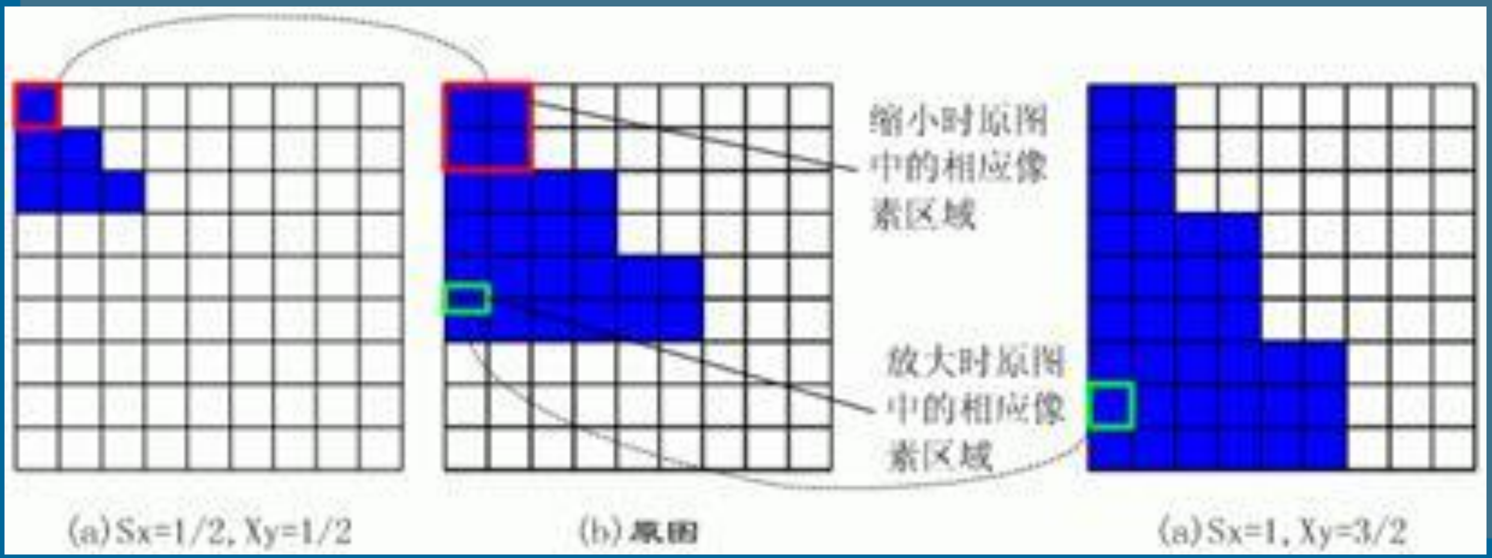
3.任意角度的光栅旋转变换

- 像素点A的亮度由其在旋转像素阵列中区域1，2和3上的覆盖量来决定，即将区域1，2和3的亮度加权平均可以求得像素A的亮度值，其中权值就是区域A在区域1，2，3上的覆盖量。

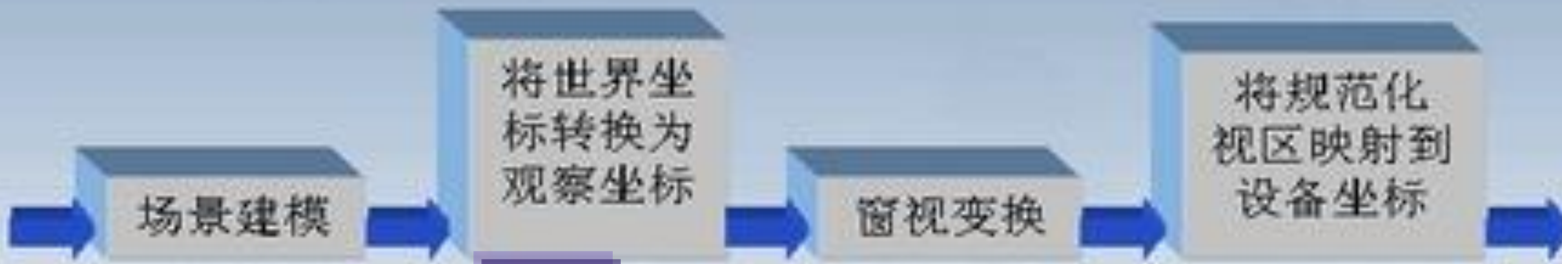


4. 光栅比例变换

- 根据Sx和Sy的值取出原图中的相应像素区域，对应变换后图像中的一个像素点，将原图中的相应像素区域的像素点的亮度值进行加权平均即可得到变换后像素点的亮度值，其中权值为原图中的相应像素区域在像素点上的覆盖量



5）观察流程



二维观察流程

a　　世界坐标系：

b　　局部坐标系：

c　　观察坐标系：

d　　视平面（成像面）坐标系：

e　　屏幕坐标系(设备坐标系) 在世界坐标系中要显示的区域称为窗口，而显示器上相应的图形输出区域称为视区。将世界坐标系中的一部分区域中的场景映射到设备坐标系的过程称为观察变换；将二维观察变换简单地称为窗口到视区的变换，简称为窗视变换。

6）线段的裁剪

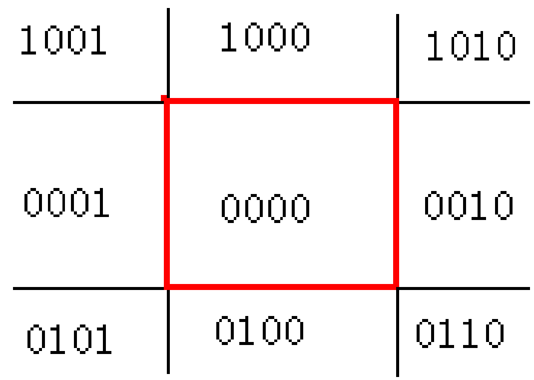
1）首先测试一个给定线段，判断它是否完全落在裁减窗口之内。

2）如果没有完全落在裁剪窗口之内，再判断是否完全落在窗口之外。

3）对于既不能确定完全落在窗口内又不能确定完全落在窗口外的线段，要计算它与一个或者多个裁剪边界的交点。

Cohen-Sutherland线段裁剪算法

区域编码



假定以相对的两个角点（x_{min}，y_{min}）和（x_{max}，y_{max}）来表示一个矩形裁剪区域。将二维平面分 区，中间的矩形为裁剪区域。每个区以四位代码（称为区域码）表示，代码的编号从右到左，各位与坐标区域的关系为：

位1：左；

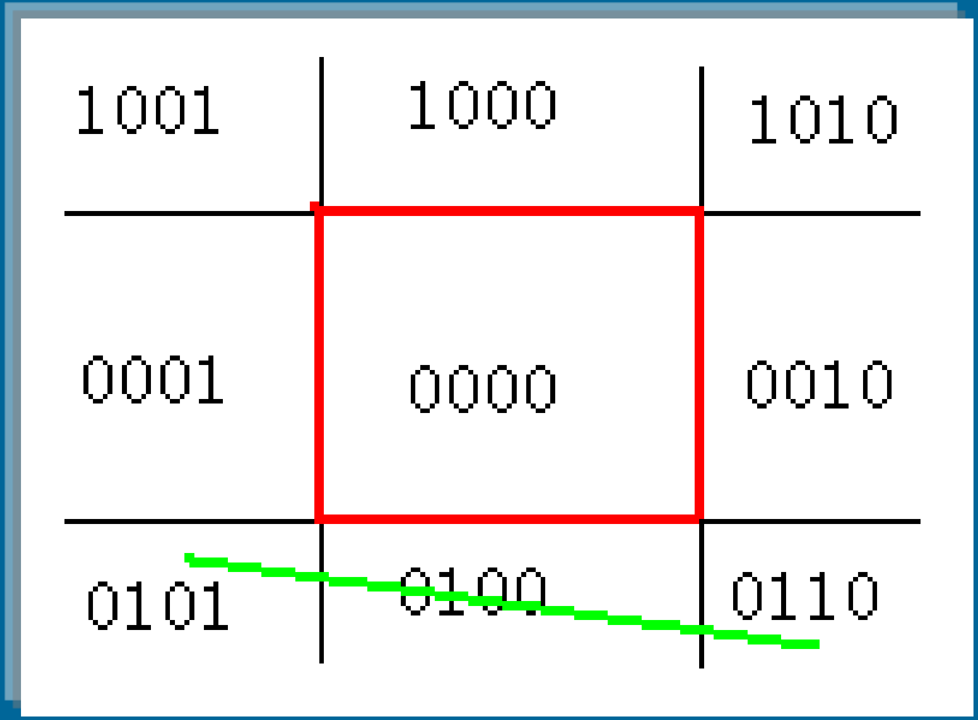
位2：右；

位3：下；

位4：上。

如果某一位赋值为1，则表示端点落在相应的位置上，否则该位值为0。

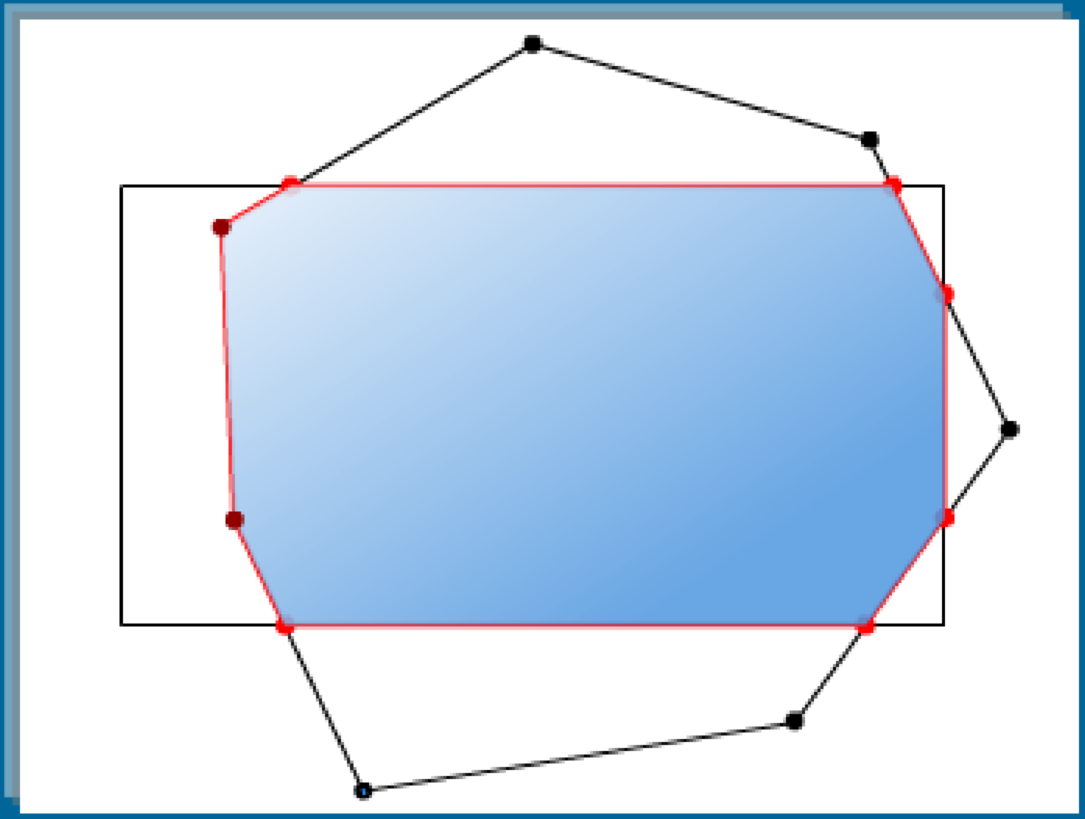
两个端点的区域码的相同位的值都为1的线段则完全落在裁剪区域之外，该线段应当去除。



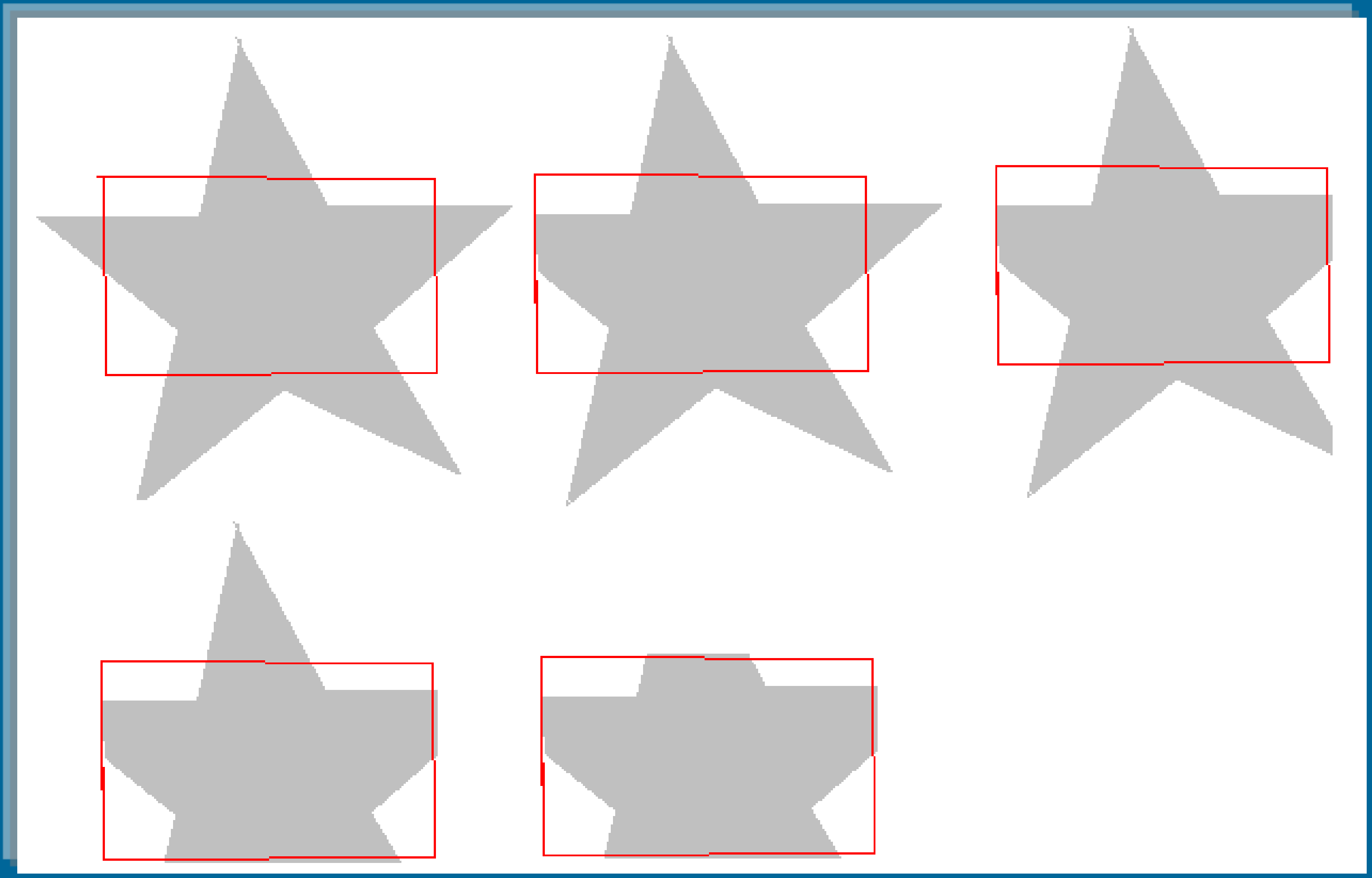
例如，若线段的两端点的区域码分别为：
code1=0101,code2=0110，则code1 & code2=0100，表示该线段位于裁剪窗口的下方。

3）多边形裁剪

多边形裁剪的裁剪区域是一个矩形窗口。多边形裁剪就是去除矩形裁剪窗口以外的部分，保留或显示其位于矩形裁剪窗口之内的部分。

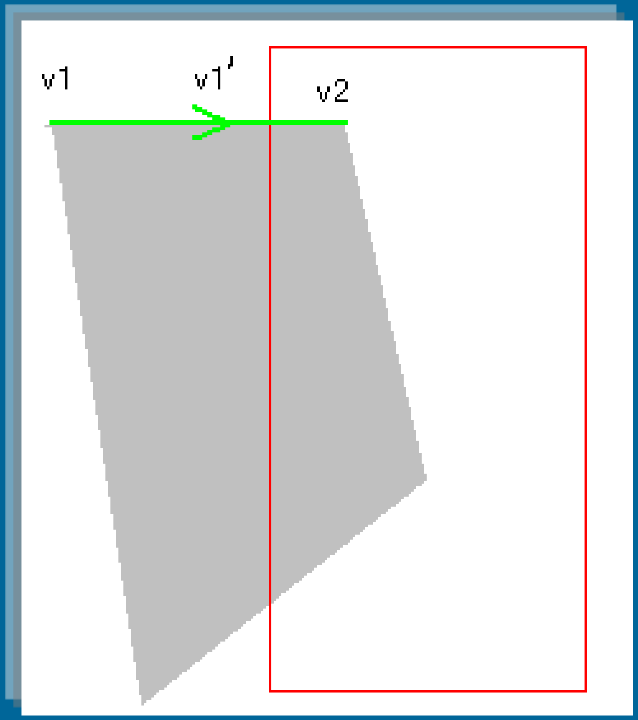


Sutherland-Hodgeman多边形裁剪

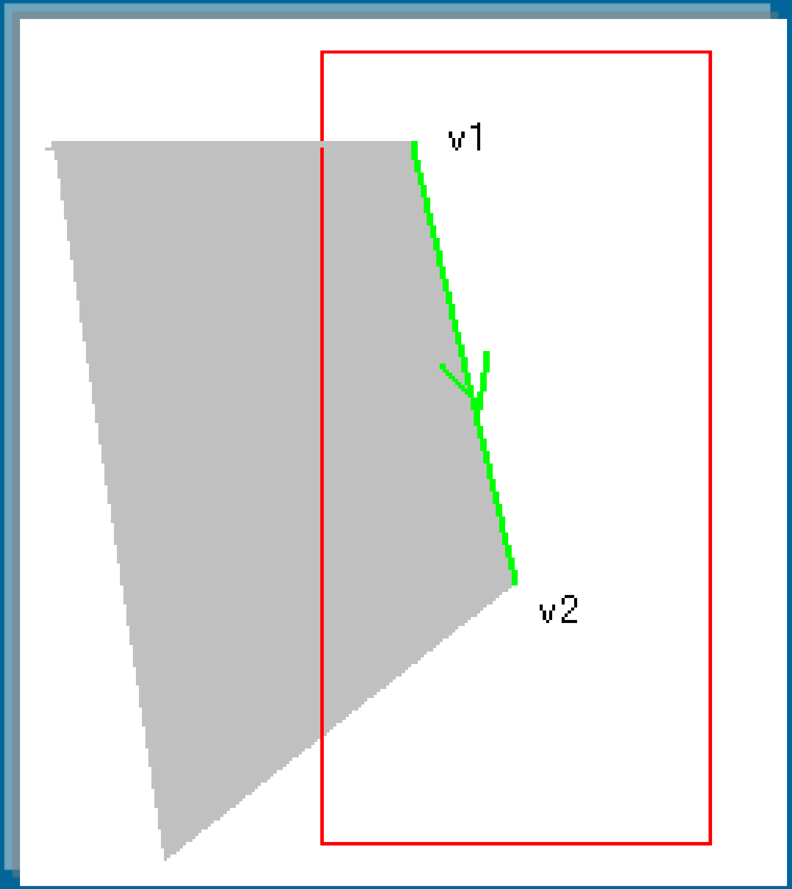


沿着多边形依次处理顶点遇到的4种情况

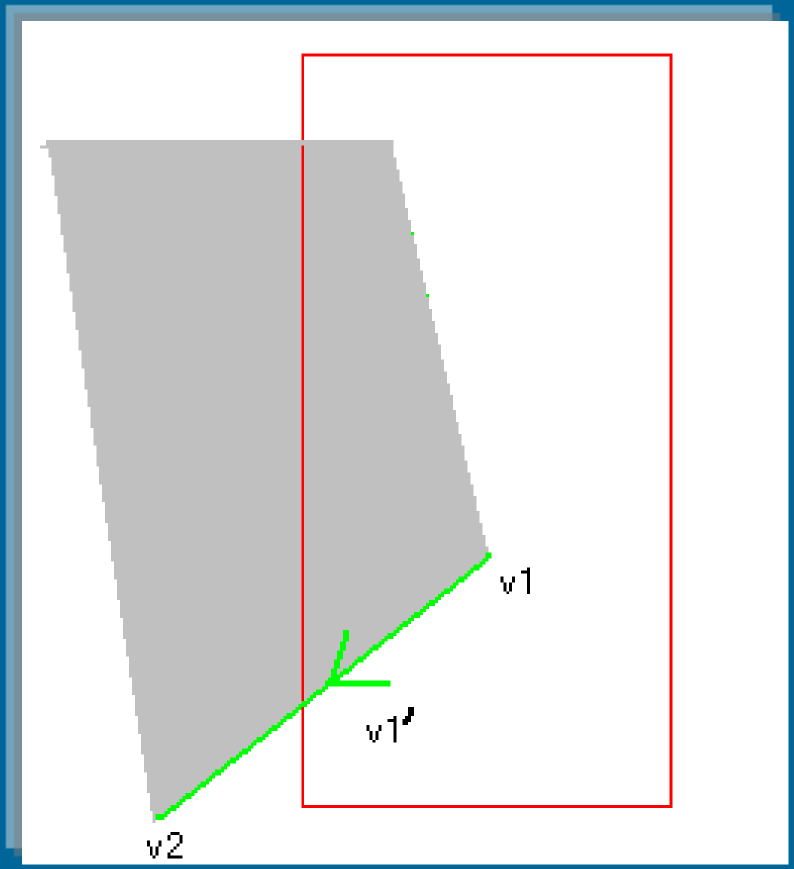
（1）如果第一个点在窗口边界外而第二个点在窗口边界内，则将多边形的这条边与窗口边界的交点和第二点加到输出顶点表中；



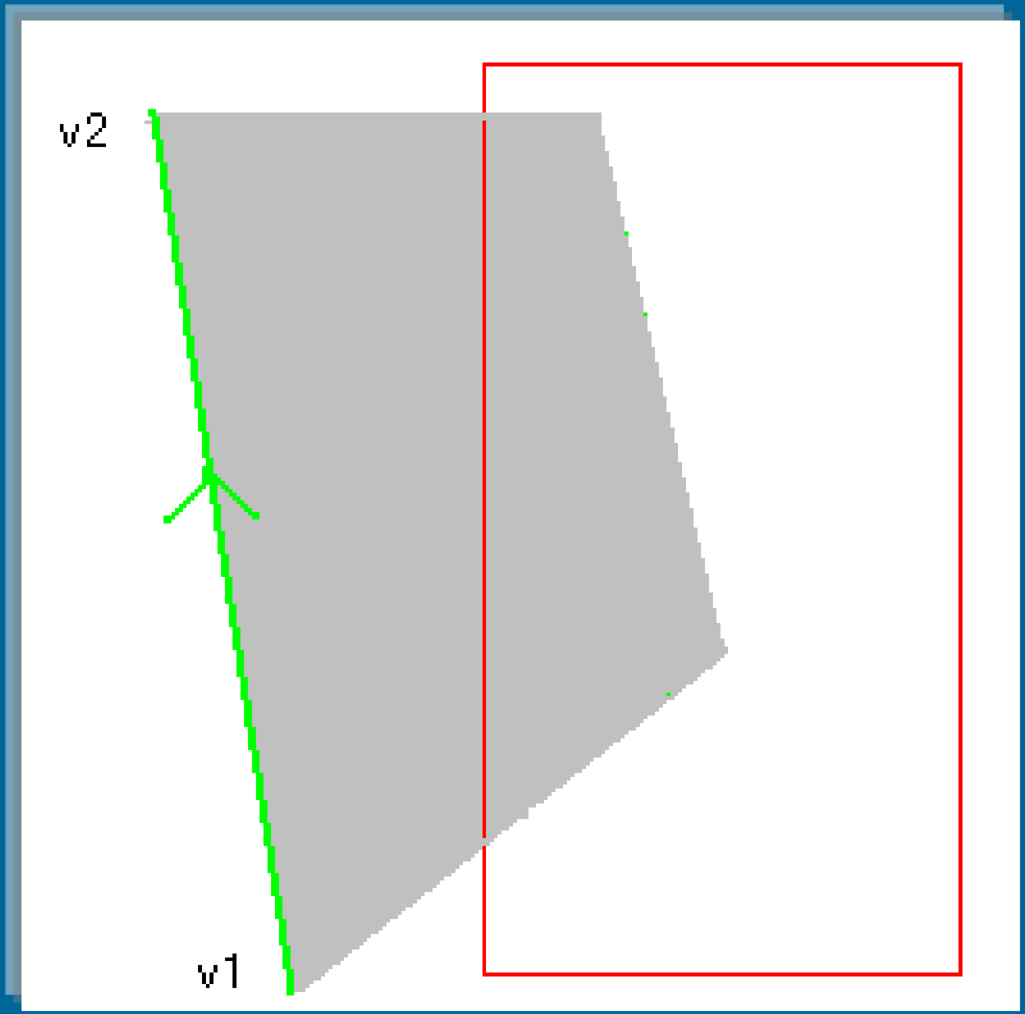
（2）如果两个顶点都在窗口边界内，则只有第二点加到输出顶点表中；

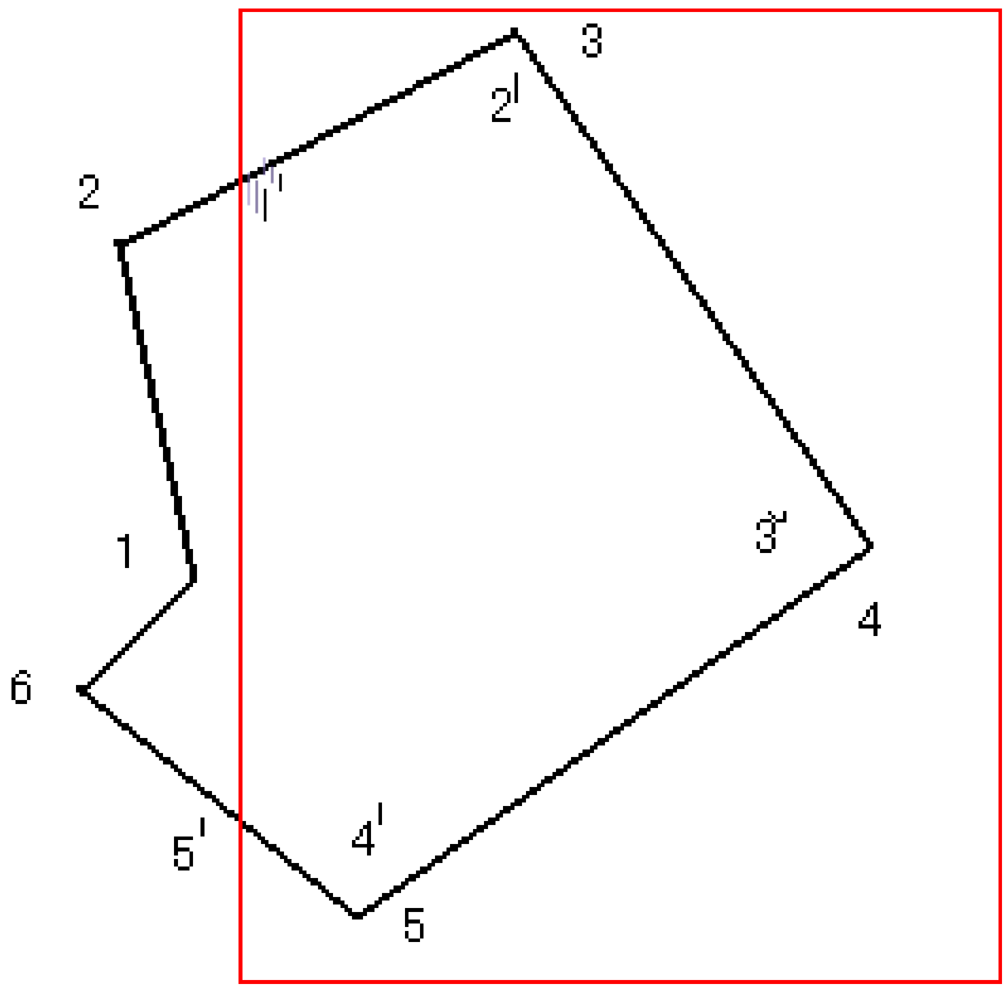


（3）如果第一点在窗口边界内而第二个点在窗口外，则只有在窗口边界的交点加到输出顶点表中；

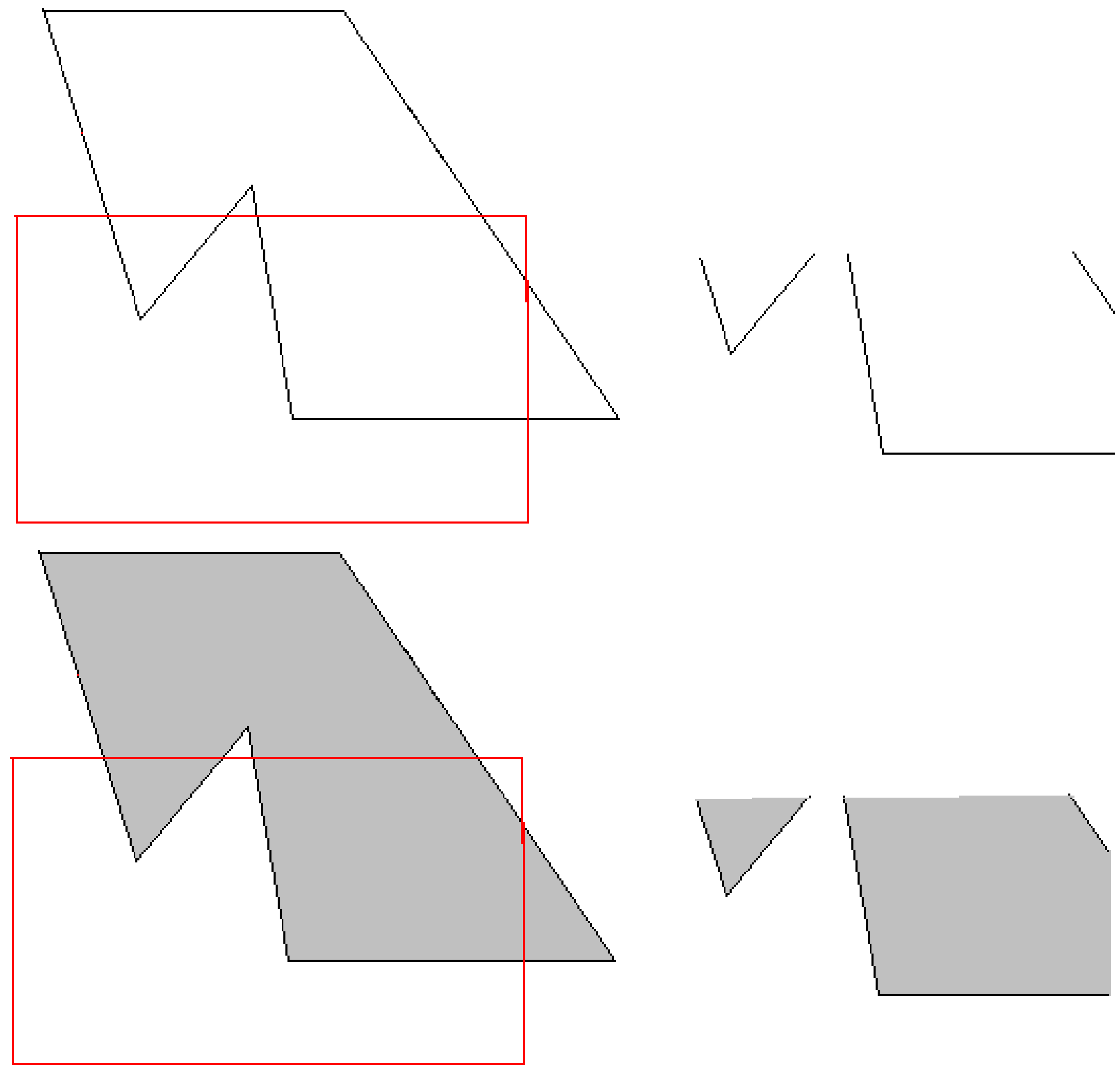


（4）如果两个点都在窗口边界外，那么输出顶点表中不增加任何点。

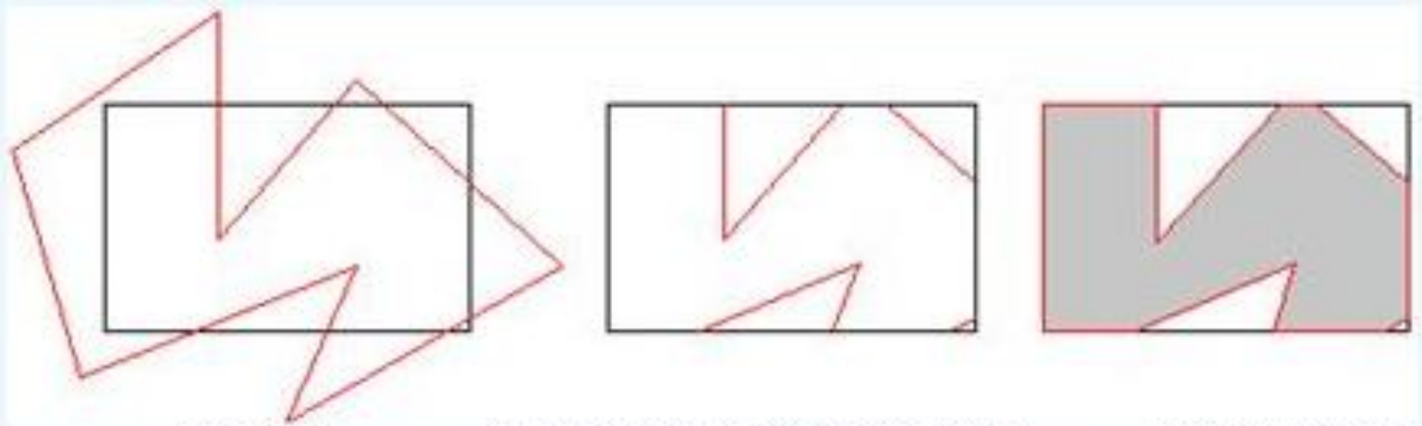




从顶点1开始裁剪，带撇号的数字表示用于标识输出顶点表中的点



多边形裁剪输出应当是：定义裁剪后的<u>多边形边界的顶点序列</u>。

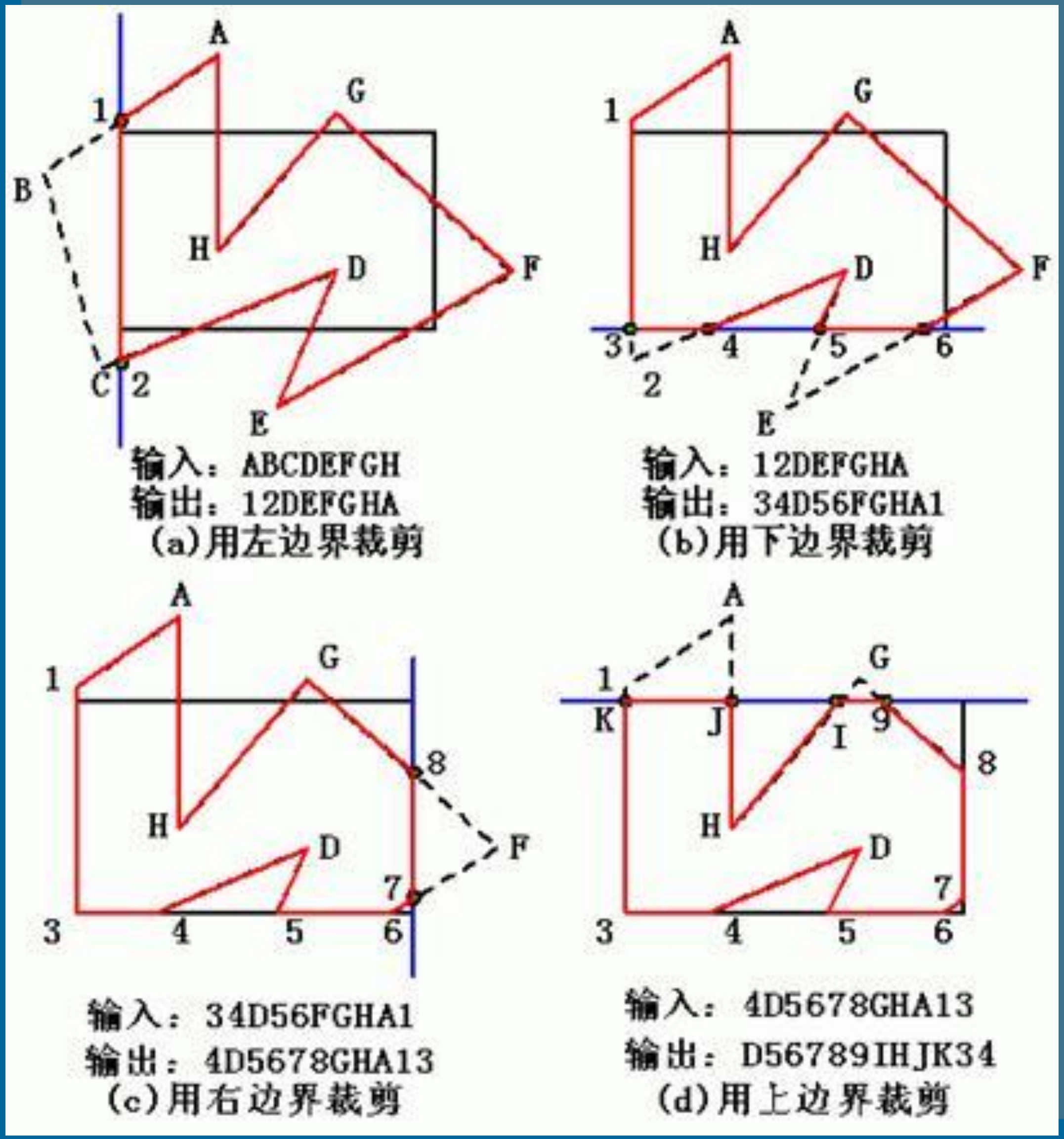


(a)裁剪前　(b)直接采用直线段裁剪的结果　(c)正确的裁剪结果

Sutherland-Hodgeman多边形裁剪实例

1) 均在可见侧，输出P2；
2) 均在不可见侧，没有输出；
3) P1可见，P2不可见，则输出线段P1P2与裁剪边界的交点；
4) P1不可见，P2可见，则输出线段P1P2与裁剪边界的交点和P2点。

Sutherland-Hodgeman多边形裁剪实例



文字及其裁剪

全部保留

全部舍弃

在计算机图形学中，字符作为一个基本图元，按照存储形式的不同，可以分为位图字体和轮廓字体两类

课后习题

- 1）下列哪些是裁剪的应用：（ **ABCD** ）
- **A** 选择图形的一部分以便进行复制、移动或者删除操作
- **B** 从定义的场景中抽取出用于观察的部分
- **C** 在三维视图中标识出可见面
- **D** 显示多窗口环境

- 2）从算法实现的角度看，多边形裁减后的输出应当是裁减后的<u>多边形边界的顶点序列</u>。

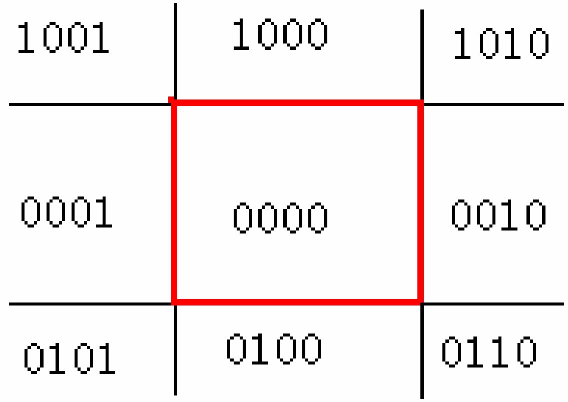
3）二维几何变换具有如下哪些性质（ABCD）

- A 平行直线不变性
- B 相交不变性
- C 仅包含旋转、平移和反射的仿射变换维持角度和长度的不变性
- D 比例变化可改变图形的大小和形状

4）请画图说明Cohen-Sutherland线段裁剪算法中区域编码方法，并简要说明该算法的算法步骤。

答：

区域编码



Cohen-Sutherland 线段裁剪算法的算法步骤：

1、对线段的两个端点的区域码进行位与运算，若其结果不为 0000，则线段完全位于裁剪窗口之外；

2、若两个端点的区域码均为 0000，则线段位于裁剪窗口内；

3、否则，需要进一步计算线段与裁剪边界的交点，然后才能确定线段是完全位于裁剪窗口之外，还是一部分位于裁剪窗口之外另一部分位于裁剪窗口之内。

5）在齐次坐标系中，写出下列变换矩阵：

（a）整个图象放大2倍；

（b）y向放大4倍和x向放大3倍；

（c）图象上移10个单位和右移5个单位；

（d）保持x=5和y=10 图形点固定，图象y向放大2倍和x向放大3倍；

（e）图象绕坐标原点顺时针方向转π/2;

（f）图象绕点x=2和y=5反时针方向转π/4。

1. 在齐次坐标系中，写出下列变换矩阵：

（a）整个图象放大2倍；

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

（b）y向放大4倍和x向放大3倍；

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

（c） 图象上移10个单位和右移5个单位；

$$\begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix}$$

（d） 保持x=5和y=10 图形点固定，图象y向放大2倍和x向放大3倍；

(1) 将坐标系平移到点（5，10）

$$T1 = \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -10 \\ 0 & 0 & 1 \end{bmatrix}$$

(2) 将图象y向放大2倍和x向放大3倍

$$S = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3) 平移坐标系至（-5，-10）点

$$T2 = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T = T2 \cdot S \cdot T1 = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -5 \\ 0 & 1 & -10 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 0 & -10 \\ 0 & 2 & -10 \\ 0 & 0 & 1 \end{bmatrix}$$

（e） 图象绕坐标原点顺时针方向转$\pi/2$：

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

（f） 图象绕点x=2和y=5反时针方向转π/4。

(1) 将坐标系平移到点（2，5）；

$$T1 = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix}$$

(2) 将图象绕原点反时针方向转π/4；

$$R = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3) 将坐标系平移到点（-2，-5）；

$$T2 = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T = T2 \cdot R \cdot T1 = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2+\sqrt{2}/2 & -\sqrt{2}/2 & -2+3\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 & -7\sqrt{2}/2+5 \\ 0 & 0 & 1 \end{bmatrix}$$

6） 由比例、平移和旋转变换构成的复合变换，仅在某些情况下交换次序不影响总的变换效果，它们是哪几种两个变换的组合。（**ABCD**）

（**A**） 两个连续的平移变换；

（**B**） 两个连续的比例变换；

（**C**） 两个连续的旋转变换；

（**D**） 比例系数相等的比例变换和旋转变换。

7）写出下列二维图形的齐次坐标变换矩阵：

a） y向放大4倍和x向放大3倍；

b） 图象上移10个单位和右移5个单位；

c） 图象绕坐标原点顺时针方向转π/2；

答：a） y 向放大 4 倍和 x 向放大 3 倍

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

c） 图象绕坐标原点顺时针方向转 π /2

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

b） 图象上移 10 个单位和右移 5 个单位

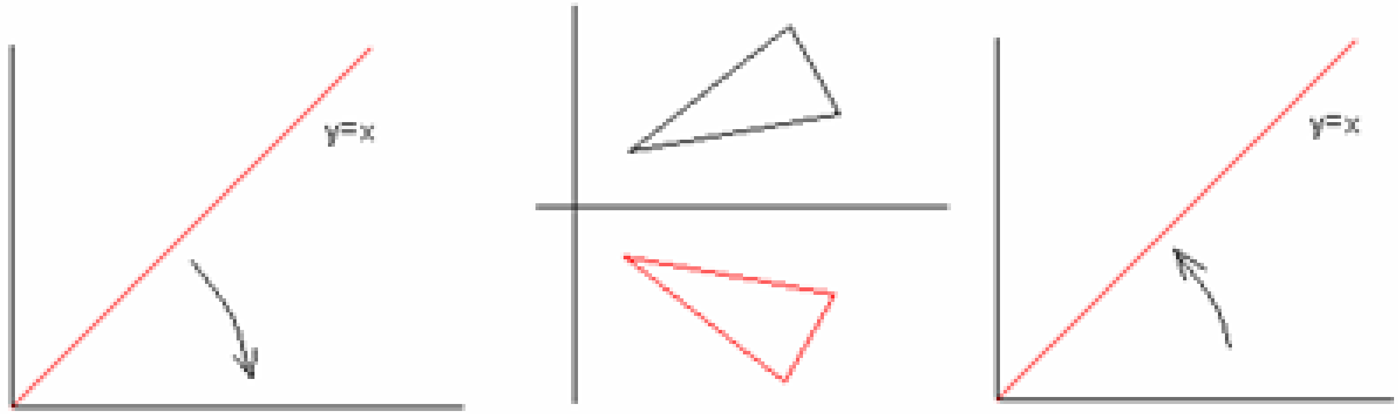
$$\begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix}$$

- 8）针对一个二维几何图形，在齐次坐标系中，写出下列变换矩阵：
- a) 关于直线y=0（x轴）的反射；
- b) 关于直线x=0（y轴）的反射；
- c) 以关于直线y=0的反射和关于直线x=0（y轴）的反射以及关于坐标原点的旋转变换作为基本变换，写出关于直线y=x的复合变换反射，并画出图形变换效果图（以三角形示意，要求注明各个端点的标号）。

答：a）关于直线 y=0 的反射 $\begin{bmatrix}1 & 0 & 0\\0 & -1 & 0\\0 & 0 & 1\end{bmatrix}$

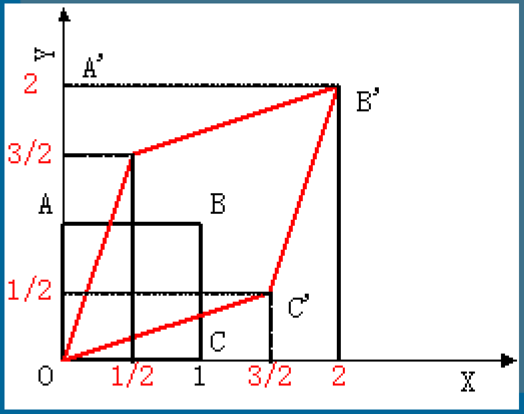
b）关于直线 x=0 的反射 $\begin{bmatrix}-1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$

c)关于直线 y=x 的复合变换反射，



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

9） 将正方形ABCO各点沿(0, 0)→(1, 1)方向进行拉伸，写出其变换矩阵和变换过程。



10）简述什么是光栅变换。

- 直接对帧缓存中像素点进行操作的图形变换称为光栅变换。

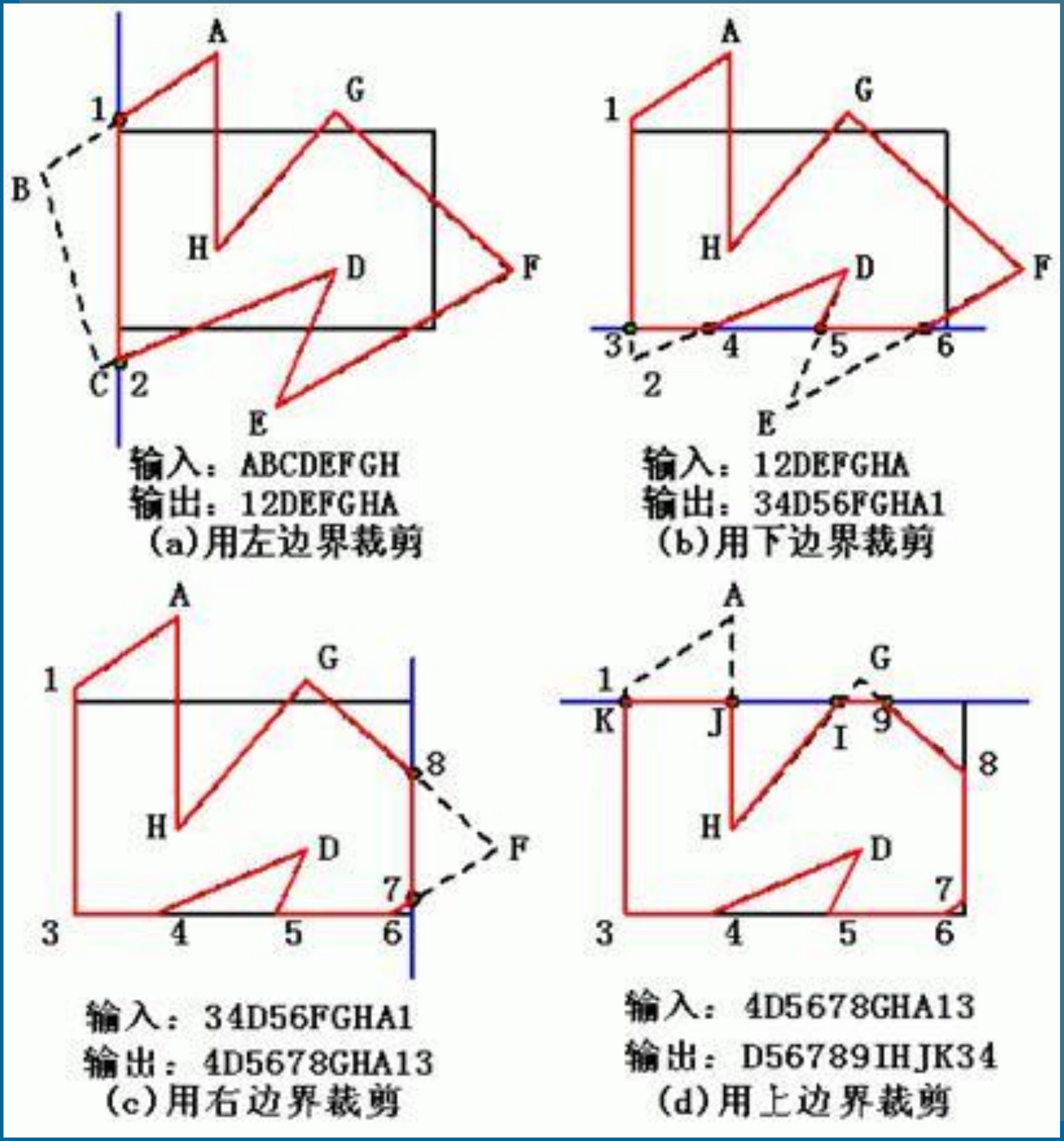
11）Cohen-Sutherland线段裁剪算法的特点

1. 通过初始测试来快速判断线段与视区的关系
2. 用编码方法实现了对完全可见和不可见直线段的快速接受和拒绝
3. 减少线段求交的次数，从而提高裁剪算法的速度。

12）Sutherland-Hodgeman多边形裁剪实例

1) 均在可见侧，输出P2；
2) 均在不可见侧，没有输出；
3) P1可见，P2不可见，则输出线段P1P2与裁剪边界的交点；
4) P1不可见，P2可见，则输出线段P1P2与裁剪边界的交点和P2点。

Sutherland-Hodgeman多边形裁剪实例



13）请说明关于xy平面内任意直线y=mx+b的反射的变换步骤，可以使用平移-旋转-反射变换的组合而完成。

- 1）通常先平移直线使其经过原点；
- 2）然后将直线旋转到坐标轴之一；
- 3）关于坐标轴反射；
- 4）最后利用旋转和平移变换将直线还原到原来位置。

14）下列哪些属于刚性变换：（A，D）

- A 反射变换；
- B 比例变换；
- C 错切变换；
- D 比例变换。

- 15）什么是反射变换，二维反射变换包含哪两类，各有什么特点？
- 反射是产生对象的镜像的一种变换。包括相对于反射轴的二维反射和相对于坐标原心的二维反射。相对于坐标原心的二维反射相当于绕圆心旋转180度角。相对于反射轴的二维反射相当于比例变换的特例。

- 16）多边形裁剪算法包括（AB）
- A）Sutherland-Hodgeman算法
- B）Weiler-Atherton算法
- C）Liang-Barsky算法
- D）Cohen-SutherLand算法

第四讲 三维几何和建模变换

4.1　平移
4.2　缩放
4.3　旋转
4.4　反射
4.5　错切
4.6　复合变换

1）平移

在三维齐次坐标表示中，任意点P可以有如下的矩阵来表示。

如果将P点沿三个坐标轴方向分别平移x_t、y_t和z_t到达P′点，则此平移变换可以用齐次坐标表示为：

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_t \\ 0 & 1 & 0 & y_t \\ 0 & 0 & 1 & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \triangleq T(x_t, y_t, z_t) \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

其中，

$$T(x_t, y_t, z_t) = \begin{bmatrix} 1 & 0 & 0 & x_t \\ 0 & 1 & 0 & y_t \\ 0 & 0 & 1 & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

称为平移变换矩阵。

2）缩放

以坐标原点为参考点、沿x、y和z坐标轴分别独立地缩放α_x、α_y和α_z倍的比例变换，其变换矩阵为：

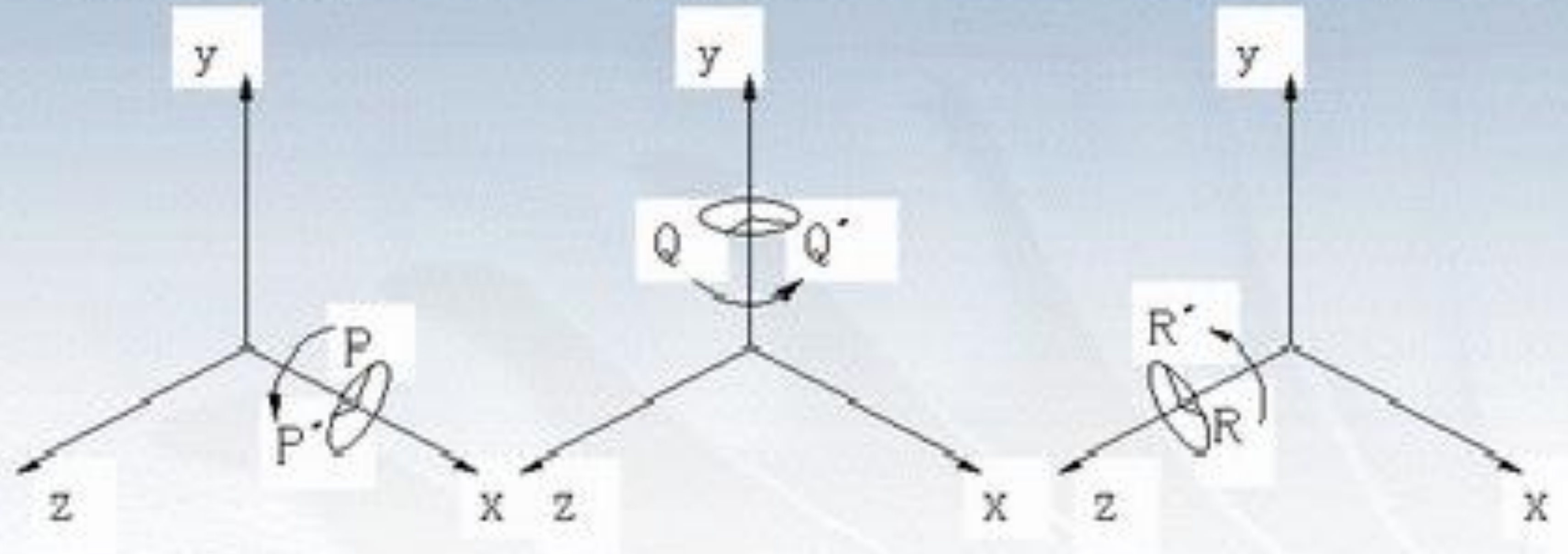
$$\mathbf{S}(\alpha_x,\alpha_y,\alpha_z)=\begin{bmatrix}\alpha_x & 0 & 0 & 0\\ 0 & \alpha_y & 0 & 0\\ 0 & 0 & \alpha_z & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

其逆变换的变换矩阵为：

$$\mathbf{S}^{-1}(\alpha_x,\alpha_y,\alpha_z)=\begin{bmatrix}1/\alpha_x & 0 & 0 & 0\\ 0 & 1/\alpha_y & 0 & 0\\ 0 & 0 & 1/\alpha_z & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

3）旋转

坐标轴旋转



a）绕x轴旋转θ角　b）绕y轴旋转θ角　c）绕z轴旋转θ角

绕x、y和z轴分别旋转α 、 β 和 θ 角的旋转变换矩阵：

$$\mathbf{R}_x(\alpha)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\alpha & -\sin\alpha & 0\\ 0 & \sin\alpha & \cos\alpha & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{R}_y(\beta)=\begin{bmatrix}\cos\beta & 0 & \sin\beta & 0\\ 0 & 1 & 0 & 0\\ -\sin\beta & 0 & \cos\beta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{R}_z(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0 & 0\\ \sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

4.4 反射

3维反射可以是关于给定的反射轴，或者是关于给定的反射平面。

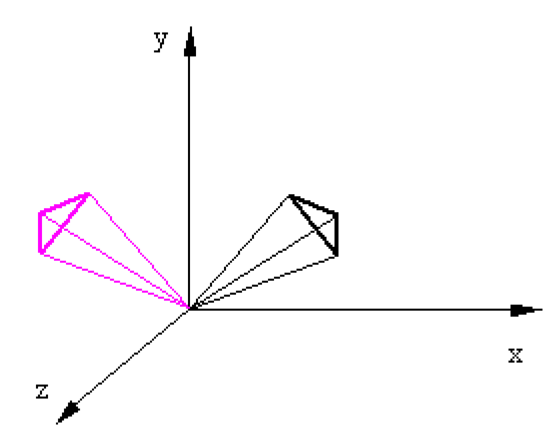
关于给定轴的反射等价于绕此轴旋转180度，

关于平面的反射可以看成是比例变换的特例。

以坐标原点为参考点、沿x、y和z坐标轴分别独立地缩放α_x、α_y和α_z倍的比例变换，其变换矩阵为：

$$S(\alpha_x,\alpha_y,\alpha_z)=\begin{bmatrix}\alpha_x & 0 & 0 & 0\\ 0 & \alpha_y & 0 & 0\\ 0 & 0 & \alpha_z & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

例如：令 $\alpha_x=-1,\alpha_y=\alpha_z=1$，所表示的就是相对于坐标平面yoz的反射变换矩阵



课后作业

- **1）将x和y放大为原来的三倍，且图形点（0.5，0.2，－0.2）保持不动；（注意相对于任意点的缩放这一类问题 ）**

$$T1 = \begin{bmatrix} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & -0.2 \\ 0 & 0 & 1 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T1 = \begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0.2 \\ 0 & 0 & 1 & -0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = T_2\, S\, T_1 = \begin{bmatrix} 3 & 0 & 0 & -1 \\ 0 & 3 & 0 & -0.4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- 2）三维反射可以分为：关于给定的反射轴的反射和关于给定的反射平面的反射。

- 3）三维反射中关于给定轴的反射等价于绕此轴旋转180度，关于平面的反射可以看成是比例变换的特例。

- 4）一般旋转（旋转绕坐标原点的任意旋转）可以看成是相继绕三个坐标轴旋转变换的复合变换结果。当旋转的顺序不同时，其结果是相同的。

5）在三维几何变换中，物体绕与坐标轴不平行的轴旋转的变换步骤是什么？

1）平移物体，使得旋转轴通过坐标原点；

2）旋转物体使得旋转轴与某一个坐标轴重合；

3）绕坐标轴完成指定的旋转；

4）利用逆旋转使旋转轴回到其原始方向；

5）利用逆平移使旋转轴回到其原始方位。

第五讲 三维物体的表示

5.1　多边形表面
5.2　二次曲面
5.3　超二次曲面
5.4　样条
5.5　Bezier曲线和曲面
5.6　结构实体几何表示法
5.7　八叉数
5.8　二叉空间分割（BSP）树
5.9　分形几何方法

　　在计算机图形学系统中，场景中的三维物体的表示方法通常分为两大类：

——边界表示（**Boundary representation, B-reps**）

——空间分区（**Space partitioning**）表示 。

边界表示

边界表示用一组平面或曲面来描述三维物体，它采用表面模型将物体分为内部和外部。表面模型能够满足平面和曲面求交、消隐、明暗绘制、生成数控加工路径等等需要。

空间分区

空间分区表示用来描述物体的内部性质，它将物体所在的空间区域划分成一组小立方体。三维物体的空间分区描述是八叉树表示。

实体的表示

计算机中表示三维形体的模型，按照几何特点进行分类，大体上可以分为三种：

——线框模型

——表面模型

——实体模型

线框模型和表面模型存贮的三维几何信息都不是十分完整，但实体模型能够完整地、准确地表示三维形体。

平面

假定已知三维空间中不共线的三个点P_i（x_i, y_i, z_i），i＝1，2，3。它们所在的平面可以用平面方程表示：

$$Ax+By+Cz+D=0$$

其中，A，B，C和D是描述该平面的空间特征的常数，可以从下列有关A/D，B/D和C/D的线性方程中解出：

$$(A/D)x_i+(B/D)y_i+(C/D)z_i = -1,$$

i=1,2,3。

解上面的三元一次线性方程組，得：

$$A=\begin{vmatrix}1 & y_1 & z_1\\ 1 & y_2 & z_2\\ 1 & y_3 & z_3\end{vmatrix}=y_1(z_2-z_3)+y_2(z_3-z_1)+y_3(z_1-z_2),$$

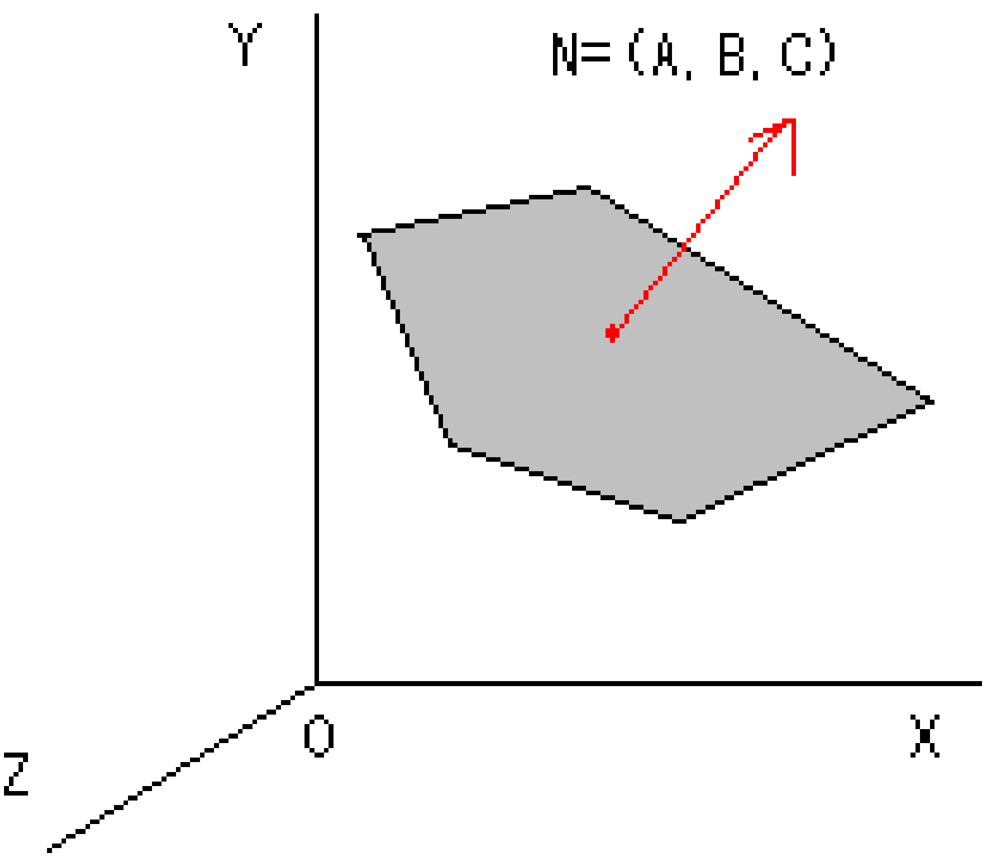
$$B=\begin{vmatrix}x_1 & 1 & z_1\\ x_2 & 1 & z_2\\ x_3 & 1 & z_3\end{vmatrix}=z_1(x_2-x_3)+z_2(x_3-x_1)+z_3(x_1-x_2),$$

$$C=\begin{vmatrix}x_1 & y_1 & 1\\ x_2 & y_2 & 1\\ x_3 & y_3 & 1\end{vmatrix}=x_1(y_2-y_3)+x_2(y_3-y_1)+x_3(y_1-y_2),$$

$$D=-\begin{vmatrix}x_1 & y_1 & z_1\\ x_2 & y_2 & z_2\\ x_3 & y_3 & z_3\end{vmatrix}=-x_1(y_2z_3-y_3z_2)-x_2(y_3z_1-y_1z_3)-x_3(y_1z_2-y_2z_1)$$

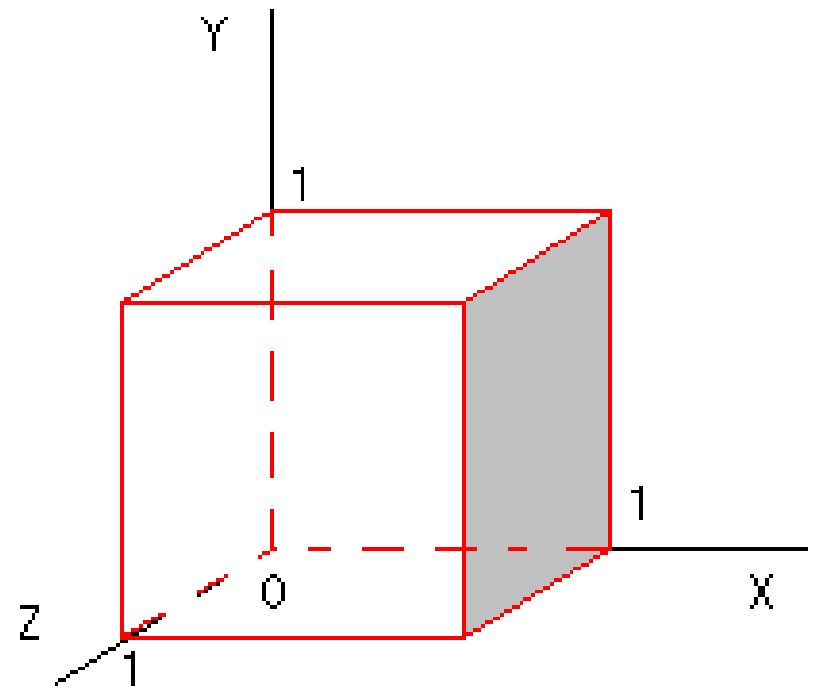
平面的空间方向用它的三維空间的法向量表示：

$$\mathrm{N} = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$



“内侧”面和“外侧”面

面向物体内部的面称为“内侧”面，向外的面成为“外侧”面。如果将多边形顶点指定为沿逆时针方向，则在右手坐标系中观察平面的外侧时，法向量方向由里向外。



单位立方体中阴影多边形表面的平面方程为x-1=0，法向量N=（1，0，0）

平面的空间方向用它的三维空间的法向量，N也可以用平面上的两个向量的叉积来计算：

$$N=(P_2-P_1)\times(P_2-P_3)$$

可得出平面参数A、B和C。如果平面上的三点P_1、P_2和P_3按逆时针方向排列，则在右手坐标系中观察平面时，N指向平面的外侧（或称为正面）。相应地，把平面的另一面称为平面的内侧（或称为反面）。

利用平面方程可以识别空间中的点与平面的相对位置关系：

如果Ax+By+Cz+D>0，则点（x，y，z）在平面的外侧（N指向同一侧）；

如果Ax+By+Cz+D<0，则点（x，y，z）在平面的内侧（与N的方向相反）。

5.4 样条表示

样条是通过一组给定点集而生成的平画曲线的柔性带。数学上使用分段3次多项式来描述样条曲线，其特点是在曲线的连接处又连续的一次和二次倒数。

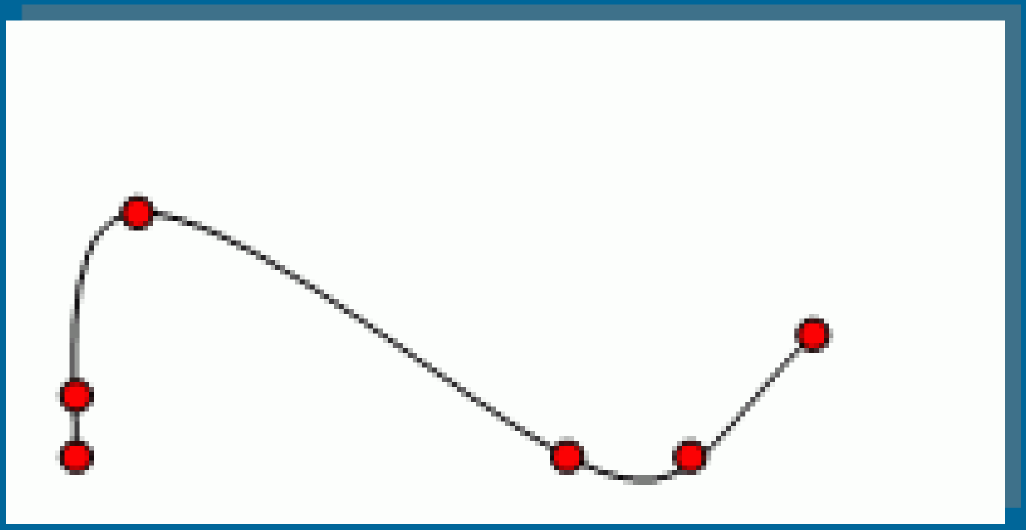
在计算机图形学中，样条曲线指的是由多项式曲线段来年接而成的曲线，在每段的边界处满足特定的连续性条件。

1）插值和逼近样条

给定一组称为控制点的坐标点，可以得到一条曲线，这些点给出了曲线的大致形状。

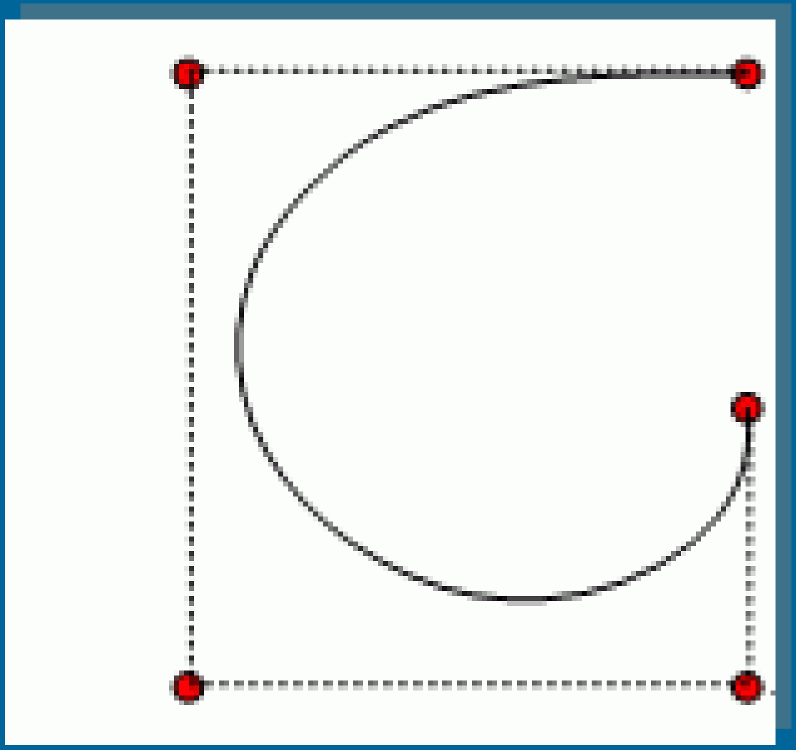
曲线的拟合（插值样条曲线）

- 当用一组型值点来指定曲线曲面的形状时，形状完全通过给定的型值点列。用该方法来得到曲线曲面称为曲线曲面的拟合。



曲线的逼近（逼近样条曲线）

- 当用一组控制点来指定曲线曲面的形状时，求出的形状不必通过控制点列，该方法称为曲线曲面的逼近。



Bezier曲线有不足之处

- 控制多边形的顶点个数决定了Bezier曲线的阶数，即n+1个顶点的控制多边形必然会产生n次bezier曲线，而且当n较大时，控制多边形对曲线的控制将会减弱。
- Bezier曲线不能作局部修改，任何一个控制点位置的变化对整条曲线都有影响。
- B样条方法保留了Bezier方法的优点，克服了其由于整体表示带来的不具备局部性质的缺点，具有表示与设计自由型曲线曲面的强大功能。

分形几何方法

分形几何（**Fractal Geometry**）方法是为了描述形状不规则、但具有自相似性的物体所采用的一种造型方法。

分形几何中的物体具有两个特点：

1）每一个点具有无限的细节

2）物体整体和局部特性之间的自相似性。

课后作业

- 1）三维实体表示方法通常分为两大类：边界表示和空间分区。

- 2）样条分为插值样条和逼近样条

- 3）空间区分表示方法有（CD）
- A) 超二次曲面
- B) Bezier曲线和曲面
- C) 结构实体几何表示法
- D) 八叉数

- 4）计算机中表示三维形体的模型，按照几何特点进行分类，大体上可以分为哪几种（ ABD ）：
- A）实体模型
- B）表面模型
- C）光照模型
- D）线框模型

- 5）计算机中表示三维形体的模型中，线框模型和表面模型存贮的三维几何信息都不是十分完整，但实体模型能够完整地、准确地表示三维形体。

- 6） Bezier曲线在进行二维或三维物体表示时，其特点是（ABC）

- A） Bezier曲线不能作局部修改，任何一个控制点位置的变化对整条曲线都有影响。
- B）控制多边形的顶点个数决定了Bezier曲线的阶数，即n+1个顶点的控制多边形必然会产生n次bezier曲线；
- C）当Bezier曲线的n较大时，控制多边形对曲线的控制将会减弱；
- D）当用一组控制点来指定Bezier曲线曲面的形状时，形状完全通过给定的控制点。

7）三次样条插值在进行二维或三维物体表示时，其特点是（ACD）

- A）在灵活性和计算速度之间提供了一个合理的折衷方案；
- B）当用一组控制点来指定曲线曲面的形状时，求出的形状不必通过控制点列；
- C）与更高次多项式相比，三次样条只需要较少的计算和存储空间；
- D）与低次多项式相比，三次样条在模拟任意曲线形状时更灵活。

8）如何定义平面的“内侧”面和“外侧”面

- 面向物体内部的面称为“内侧”面，向外的面成为“外侧”面。如果将多边形顶点指定为沿逆时针方向，则在右手坐标系中观察平面的外侧时，法向量方向由里向外。

9）分形几何中的物体具有两个特点：

答：1）每一个点具有无限的细节；2）物体整体和局部特性之间的自相似性。

10）分形包括：自相似分形和自仿射分形

11）分形维数用于表示物体的细节变化，它是物体粗糙性或细碎性的度量，具有较大锯齿形的物体其分形维数较大。

12）什么是结构实体几何表示法（CSG），并说明其过程。

- 通过对体素进行集合运算而得到新的形体的一种表示方法。
- 一般过程可以简单地描述如下：
- 1）在三维空间中选两个基本体素；
- 2）选择组合两个基本体素的集合运算（并、交、差），这时除了基本体素外，增加了一个新的物体；
- 3) 利用基本体素和每一步新创建的物体的组合构造新物体，直至形成最后的结果物体。

- 13）计算机中表示三维形体的模型，按照几何特点进行分类的3类模型中，线框模型和表面模型存贮的三维几何信息都不是十分完整，但实体模型能够完整地、准确地表示三维形体。
- 表面模型能够满足平面和曲面求交、消隐、明暗绘制、生成数控加工路径等等需要。

14）解释一下三维物体的表示方法中边界表示的含义是什么？

边界表示用一组平面或曲面来描述三维物体，它采用表面模型将物体分为内部和外部。表面模型能够满足平面和曲面求交、消隐、明暗绘制、生成数控加工路径等等需要。

15）解释一下三维物体的表示方法中空间分区的含义是什么？

空间分区表示用来描述物体的内部性质，它将物体所在的空间区域划分成一组小立方体。三维物体的空间分区描述是八叉树表示。

- 16) 三维建模过程的边界表示**方法有（A）**
- **A) 样条**
- **B) CSG**
- **C) BSP树**
- **D) 八叉数**

17）解释一下分形维数的含义

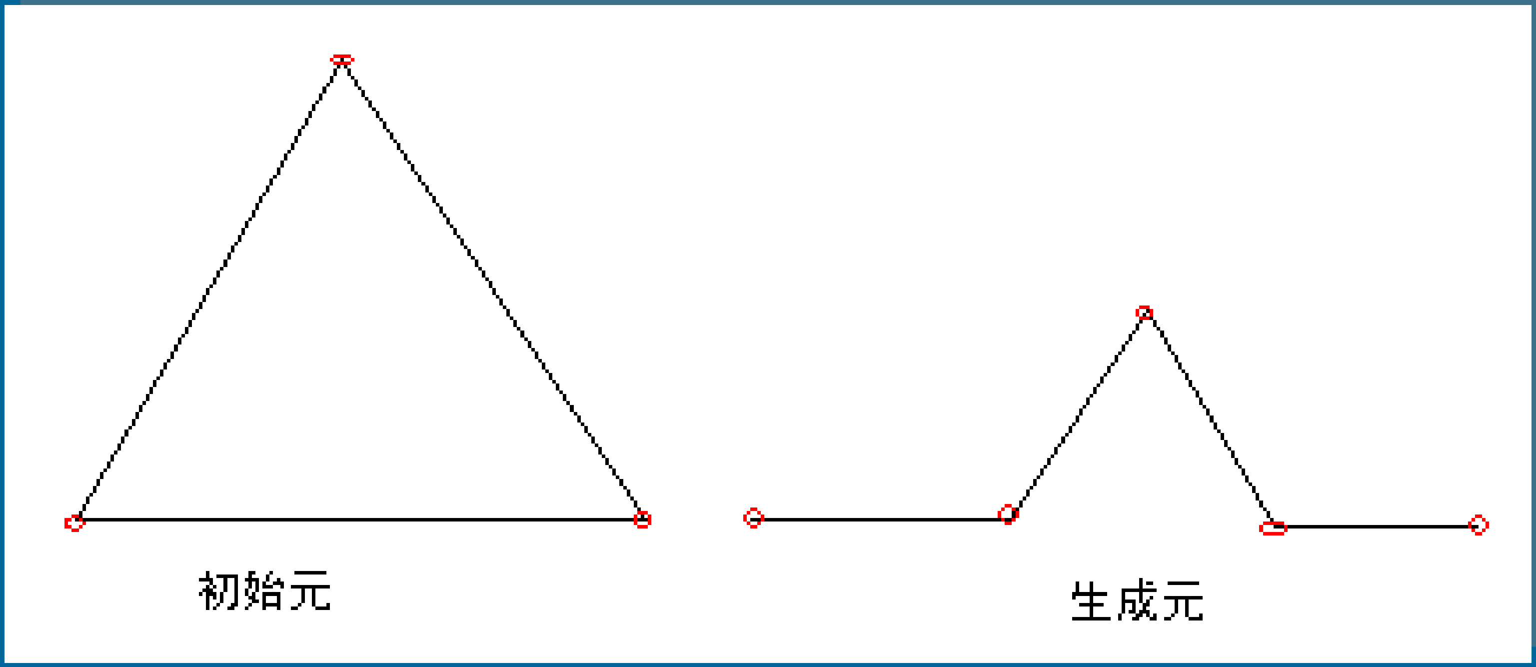
分形维数用于表示物体的细节变化，它是物体粗糙性或细碎性的度量，具有较大锯齿形的物体其分形维数较大。

分形维数：$D = \dfrac{\ln(n)}{\ln(1/s)}$

其中：

n 表示单位线段、正方形、和立方体的再分数目

s 表示缩放因子



再分数目n=4——初始元中的每一个线段每一次用四个相等的线段代替

缩放因子s=1/3

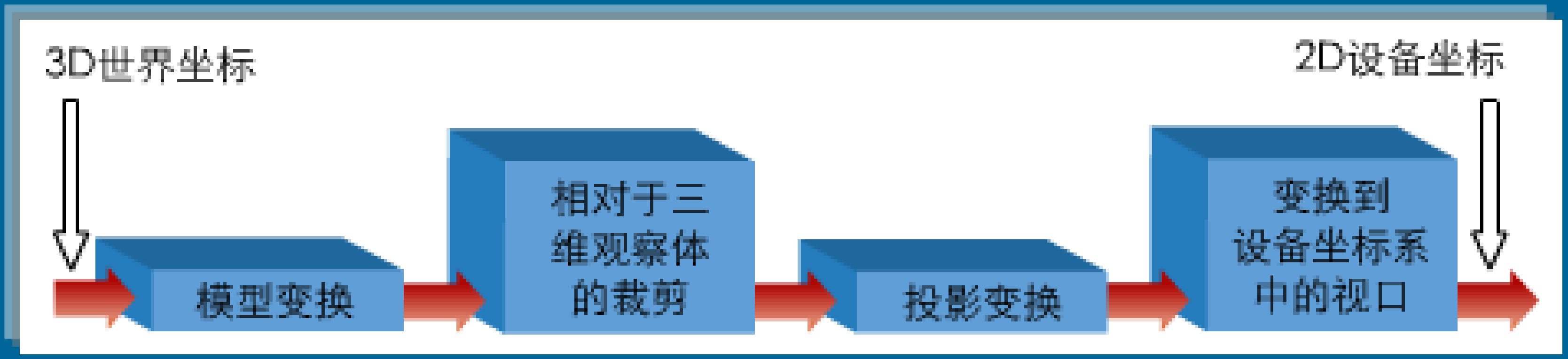
第六讲 三维观察

6.1　观察流水线
6.2　观察坐标系
6.3　投影
6.4　观察体和一般投影变换
6.5　裁剪

三维观察 vs 二维观察

在三维观察过程中，观察窗口和物体是三维的，但显示平面是二维的，这种几何空间的维数的不匹配大大增加了三维观察的难度。

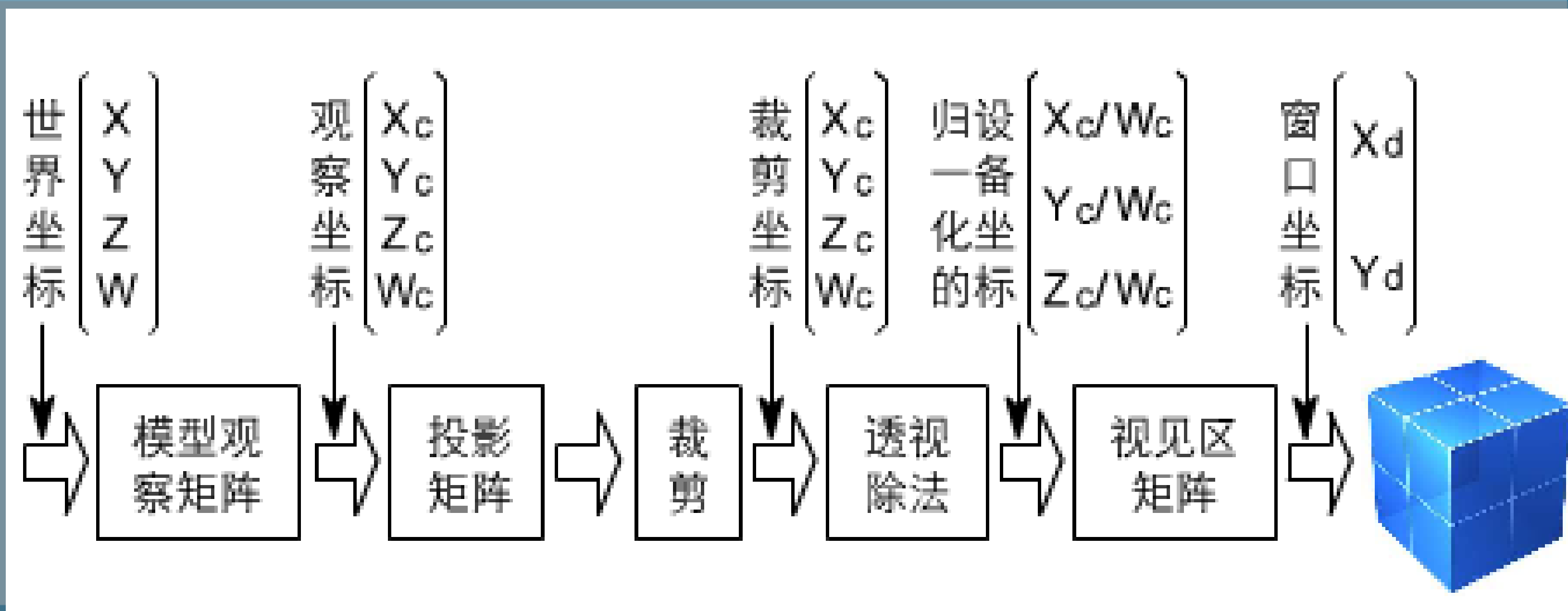
6.1观察流水线(三维观察流程)



观察流水线是指系统的图形输出流程。它由三部分组成：模型变换、取景变换和图象变换（光栅变换）。

- 模型变换--在世界坐标系的变换
- 图象变换--在标准设备坐标中的变换
- 取景变换--介于这两者之间的变换

三维观察流程


世界坐标
X
Y
Z
W
模型观察矩阵
观察坐标
Xc
Yc
Zc
Wc
投影矩阵
裁剪
裁剪坐标
Xc
Yc
Zc
Wc
透视除法
归一化的设备坐标
Xc/Wc
Yc/Wc
Zc/Wc
视见区矩阵
窗口坐标
Xd
Yd

6.3 投影

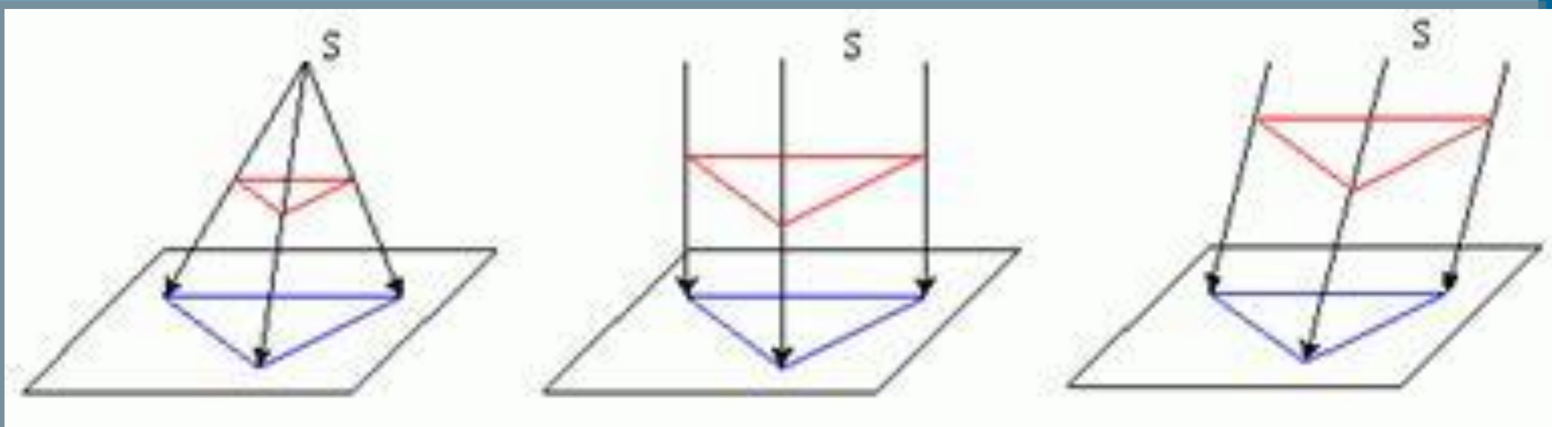
　　投影变换就是把三维立体（或物体）投射到投影面上得到二维平面图形。

---平面几何投影

---观察投影

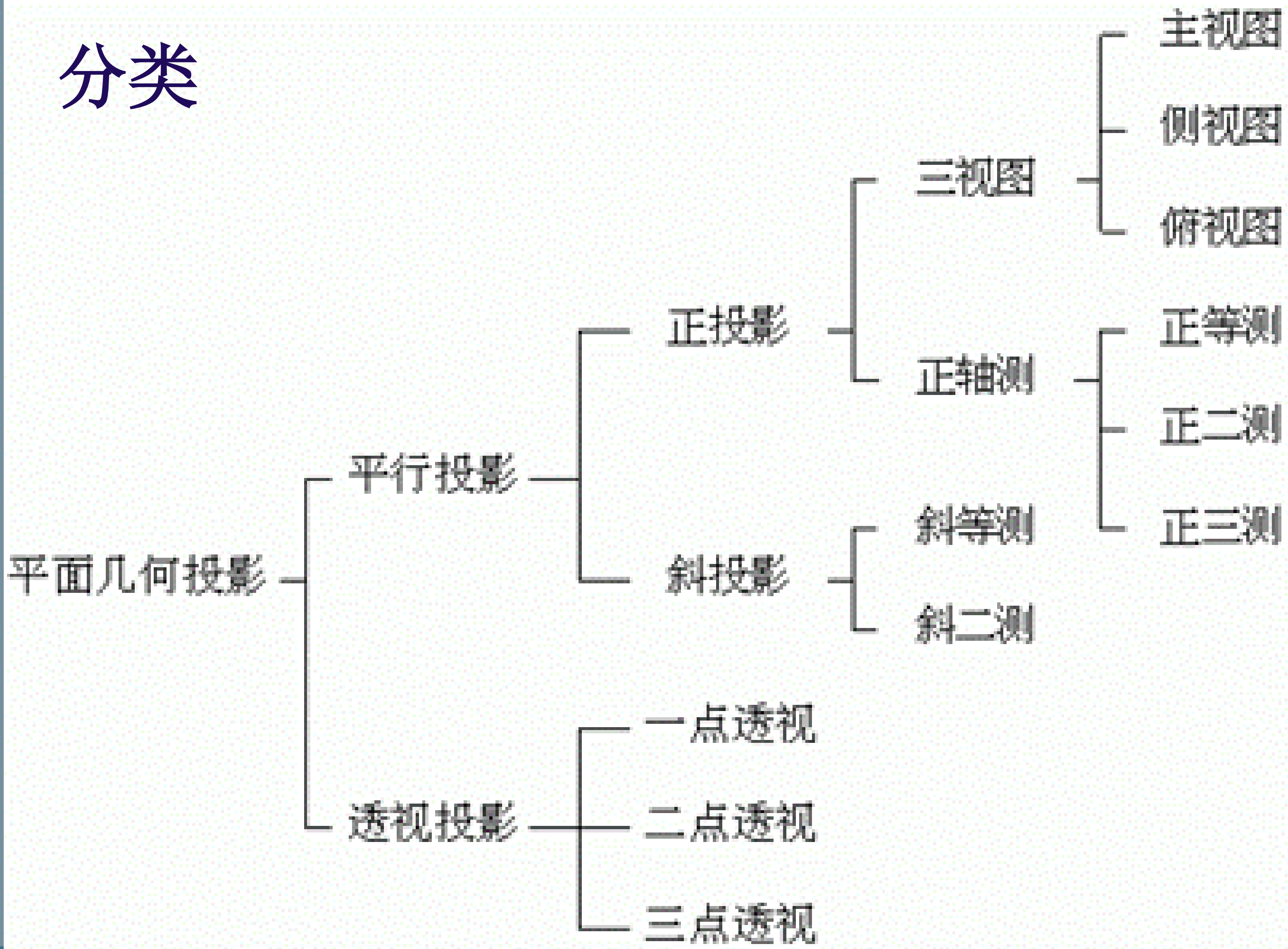
平面几何投影

- 平面几何投影可分为透视投影和平行投影两大类，透视投影的投影中心到投影面之间的距离是有限的。平行投影的投影中心到投影面之间的距离是无限的。



(a)透视投影　　(b)正投影　　(c)斜投影

分类



平面几何投影
平行投影
透视投影
正投影
斜投影
三视图
正轴测
主视图
侧视图
俯视图
正等测
正二测
正三测
斜等测
斜二测
一点透视
二点透视
三点透视

观察投影

- 在显示三维形体的过程中，在用户坐标系下定义一个观察空间，将观察空间外的物体裁剪掉，只对落在观察空间内的物体作投影变换并予以显示。

平行投影/透视投影

平行投影--坐标位置沿着平行线变换到观察平面上。

透视投影--物体位置沿着收敛于某一点的直线变换到观察平面上。

(a) 透视投影　　(b) 平行投影

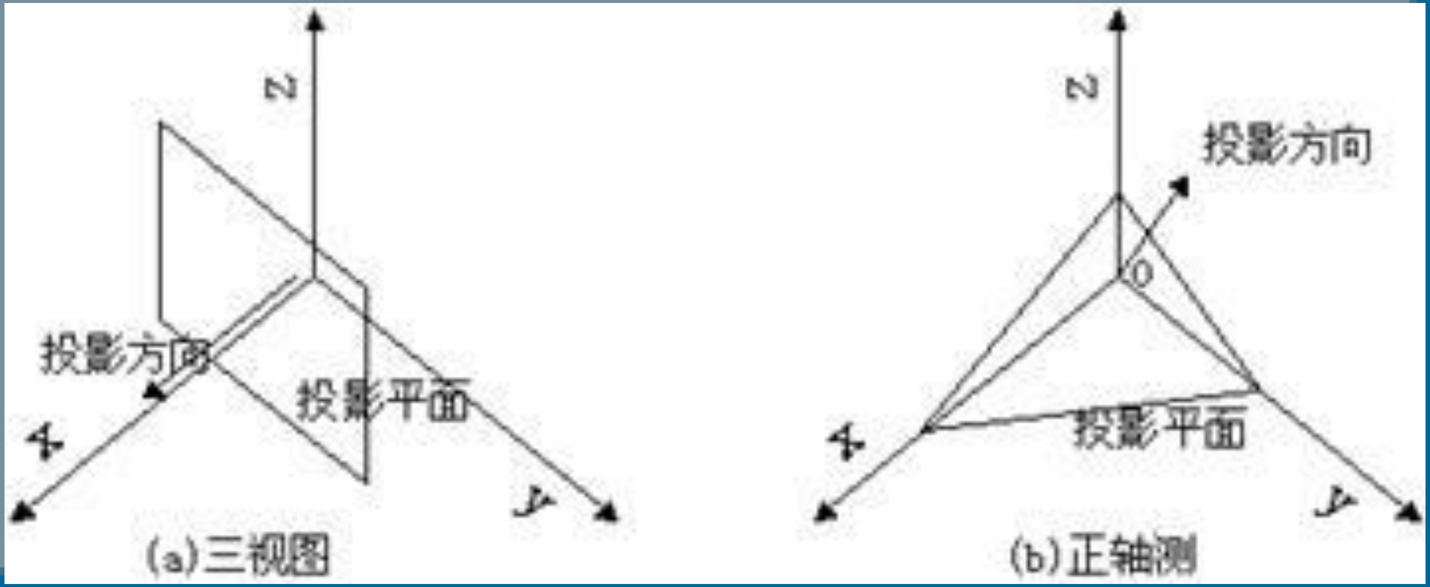
1）平行投影

在平行投影中，三维物体的坐标沿平行线投影到观察平面上，它保持物体的有关比例不变，能够精确地反映物体的实际尺寸。在三维空间平行的直线经平行投影后依然保持平行。

正投影　　　　斜投影

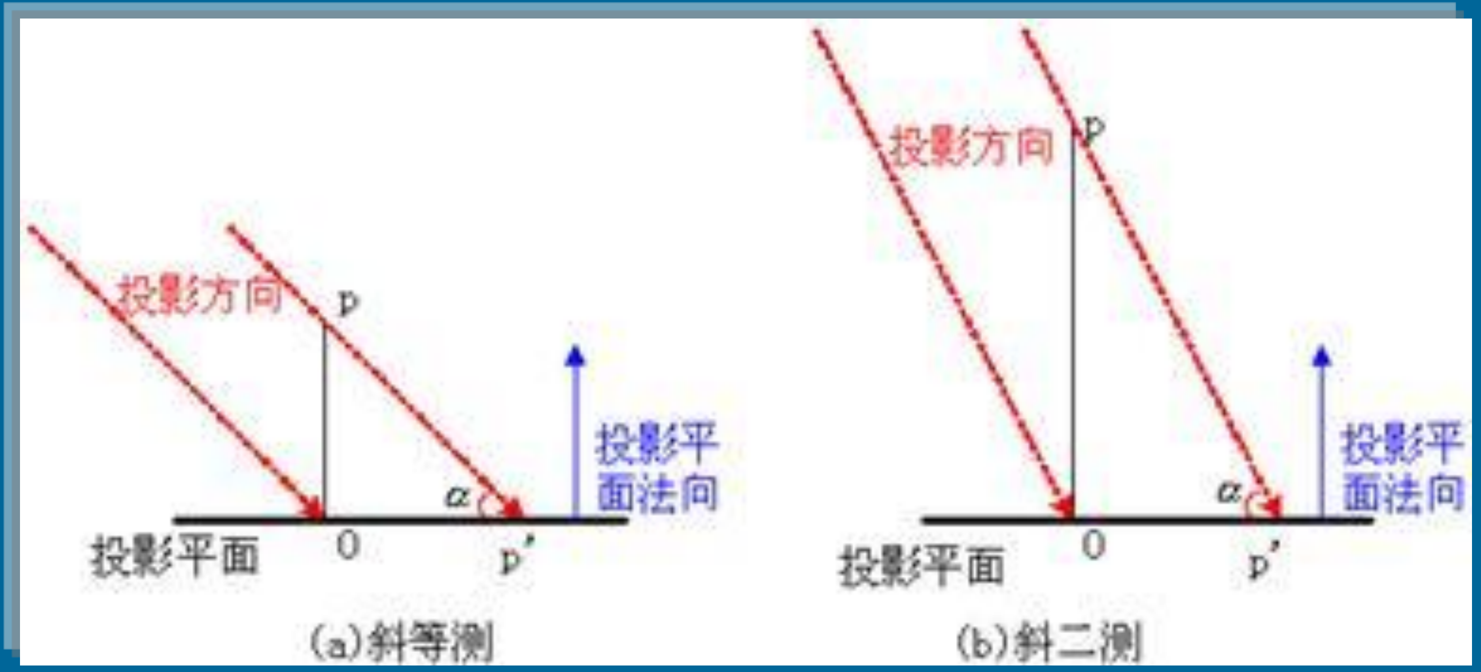
正投影

　　正投影又可分为：三视图和正轴测。当投影面与某一坐标轴垂直时，得到的投影为三视图；否则，得到的投影为正轴测图



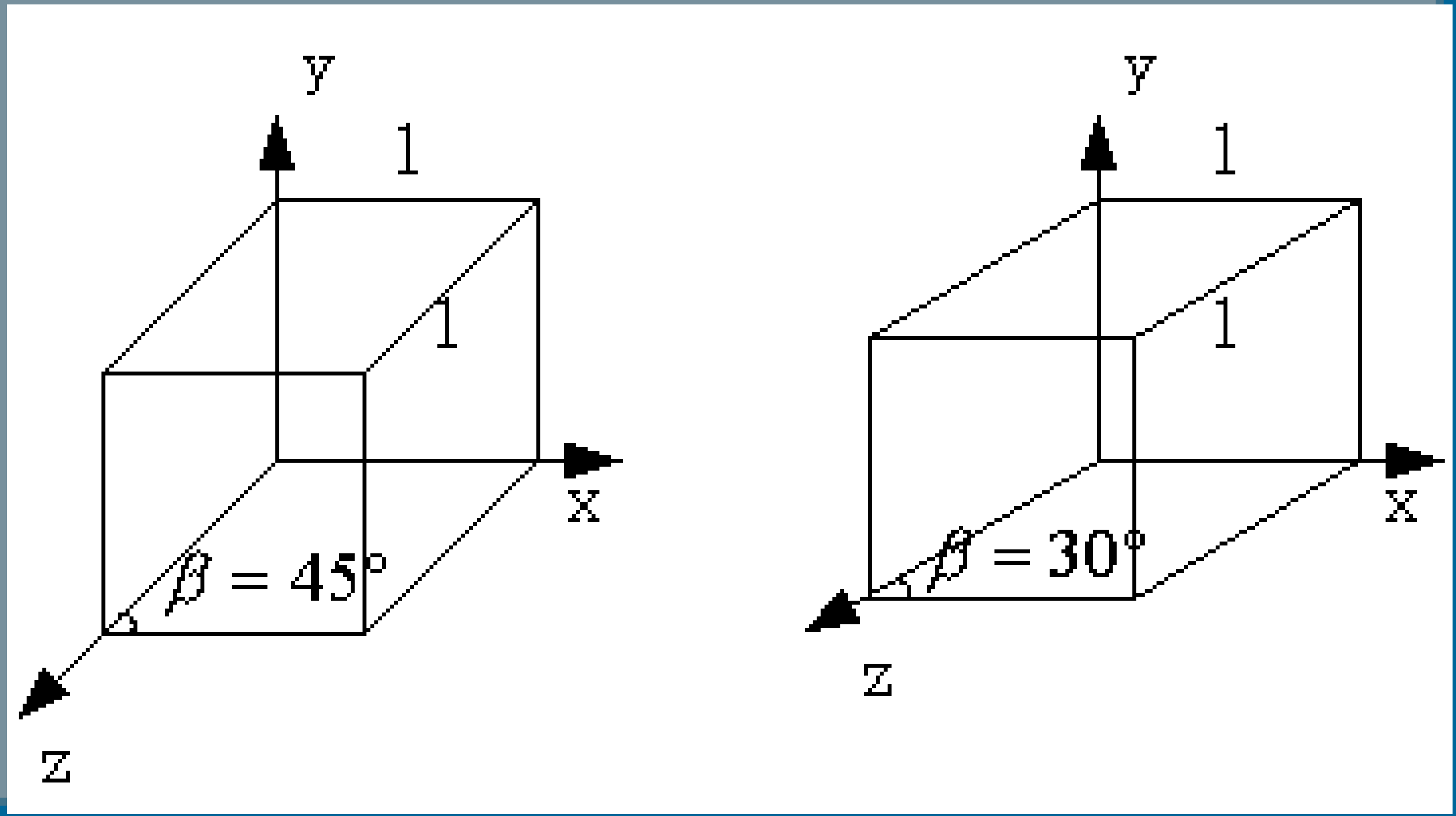
斜投影

- 常用的斜轴测图有斜等侧图和斜二侧图
- 斜投影图既可以进行测量又可以同时反映三维形体的多个面，具有立体效果。



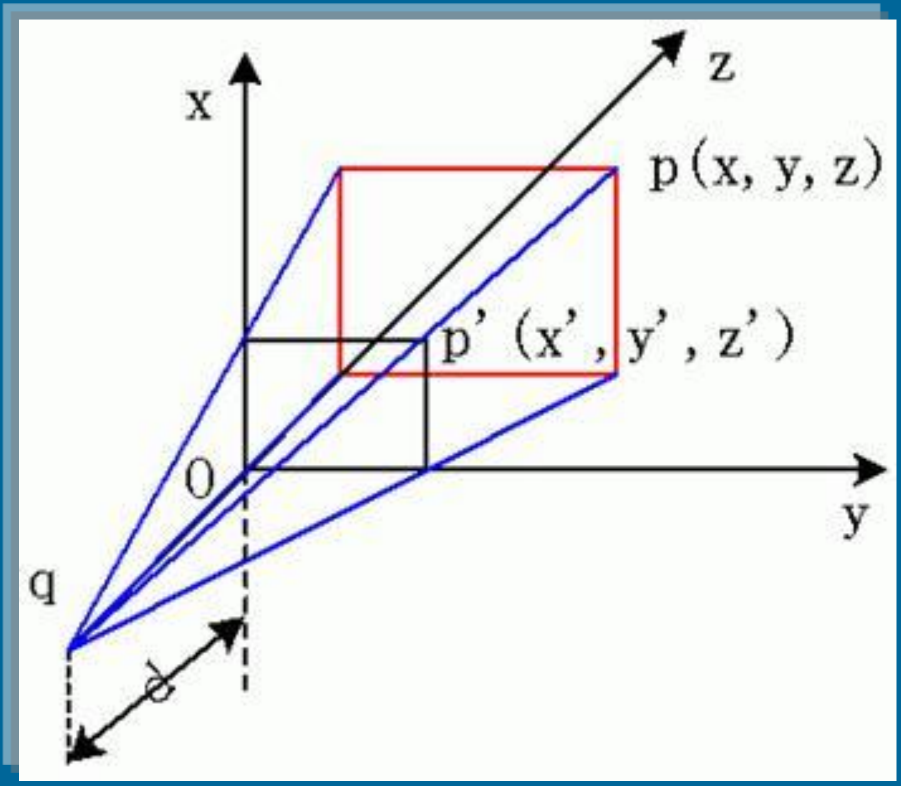
单位立方体的斜平行投影

斜等测　　　　斜二测



2）透视投影

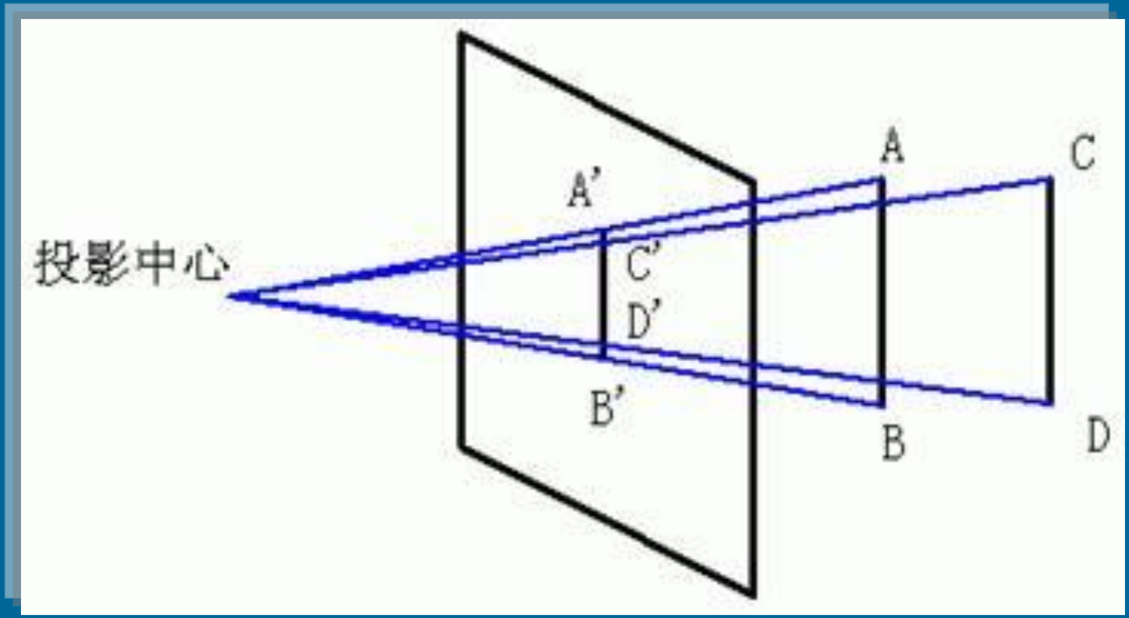
透视投影是沿着相交于视点（即：投影参考点 Projection Reference Point）的投影线进行投影。



特点：透视投影的深度感更强，更加具有真实感，但透视投影不能够准确反映物体的大小和形状。

透视投影（特点）

- 透视投影的深度感更强，更加具有真实感，但透视投影不能够准确反映物体的大小和形状。
- 物体的透视投影的大小与物体到投影中心的Z方向距离成反比。（透视缩小效应）

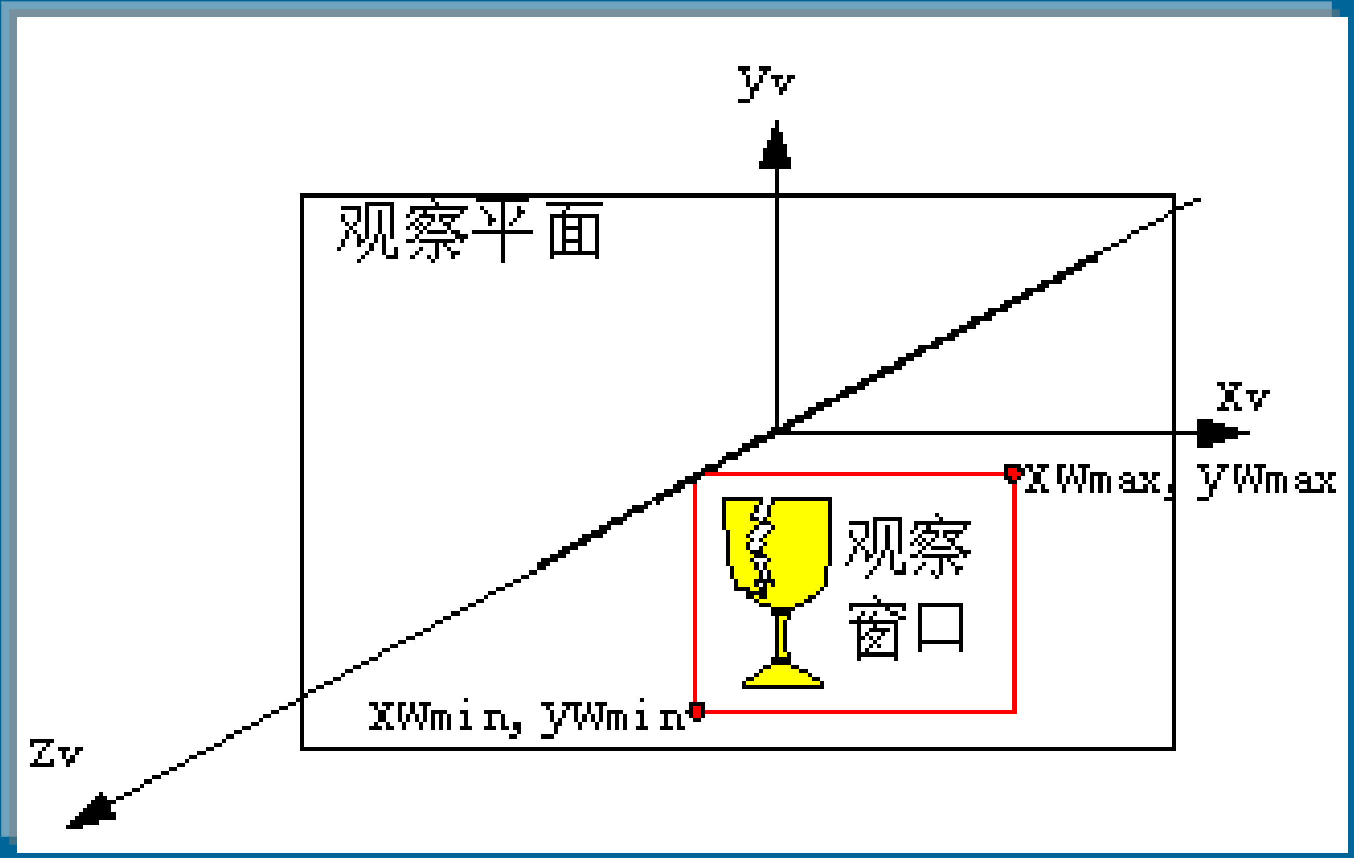


透视投影（特点）

- 一组平行线若平行于投影平面时，它们的透视投影仍然保持平行。
- 只有当物体表面平行于投影平面时，该表面上的角度在透视投影中才能被保持。

6.4　观察体和一般投影变换

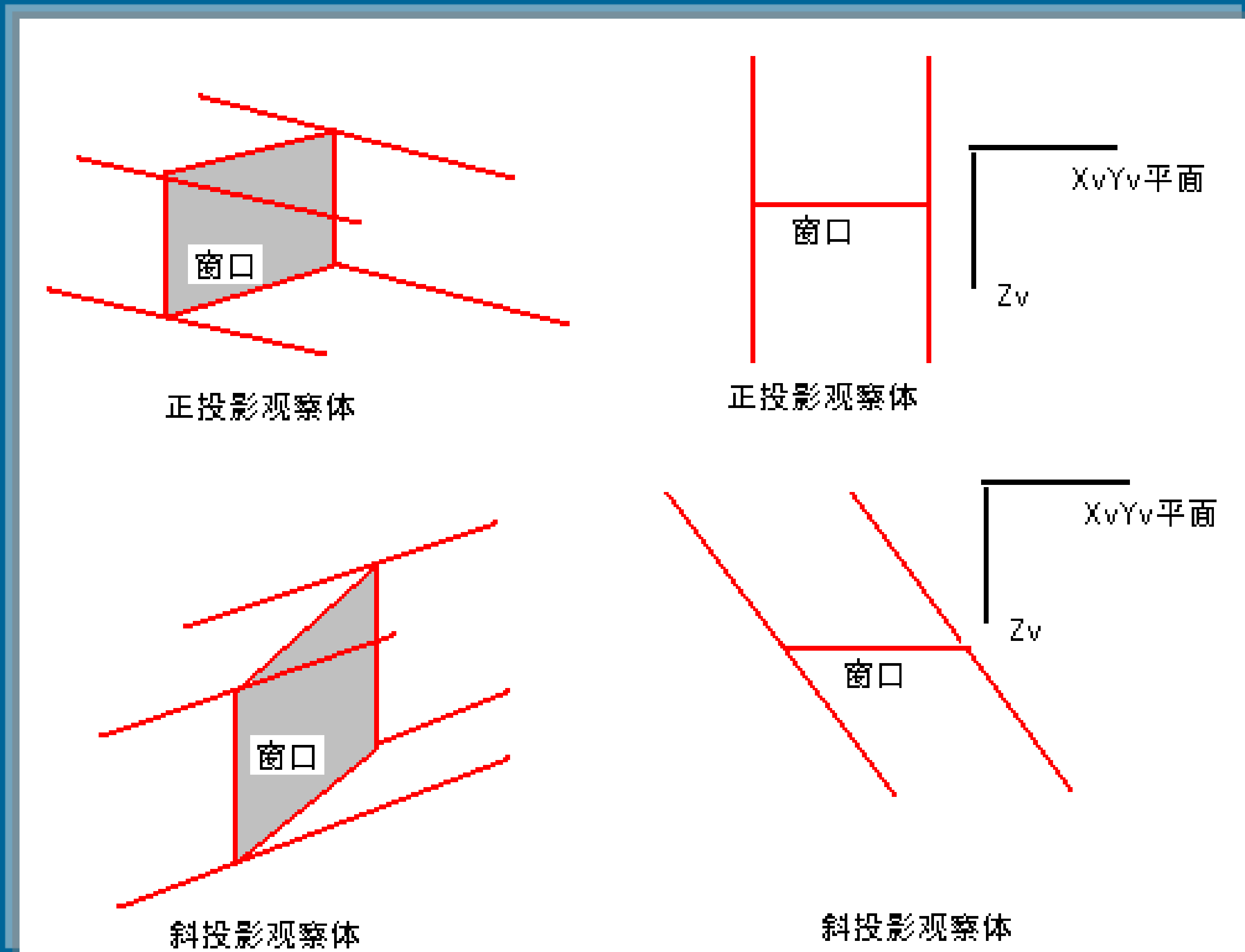
3-D场景中的物体按观察体进行裁剪。可以利用窗口边界来设置观察体，只有在观察体中的那些物体才在输出设备中显示，其他部分均被裁剪掉。



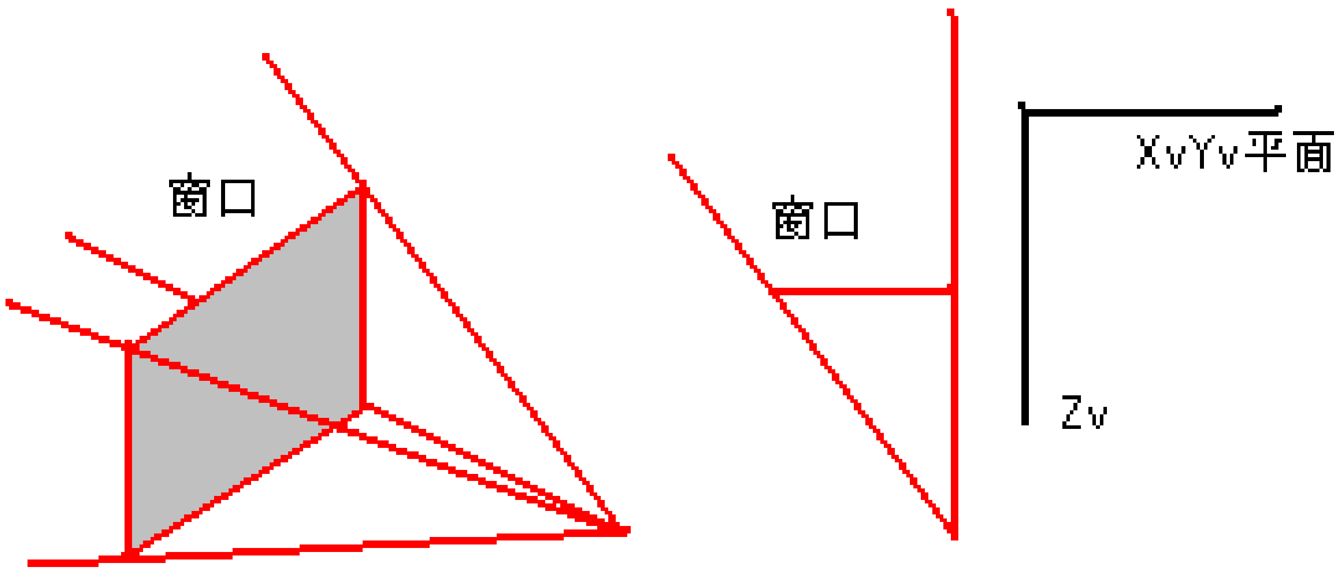
观察体（观察空间）

- 将观察窗口沿投影方向作平移运动产生的三维形体。
- 观察体的大小依赖于窗口的大小，观察体的形状依赖于生成显示结果的投影类型。
- 对于任何情况，体的四个侧面是经过窗口的平面。

对于平行投影，观察体的四侧面形成了无限长的平行六面体。



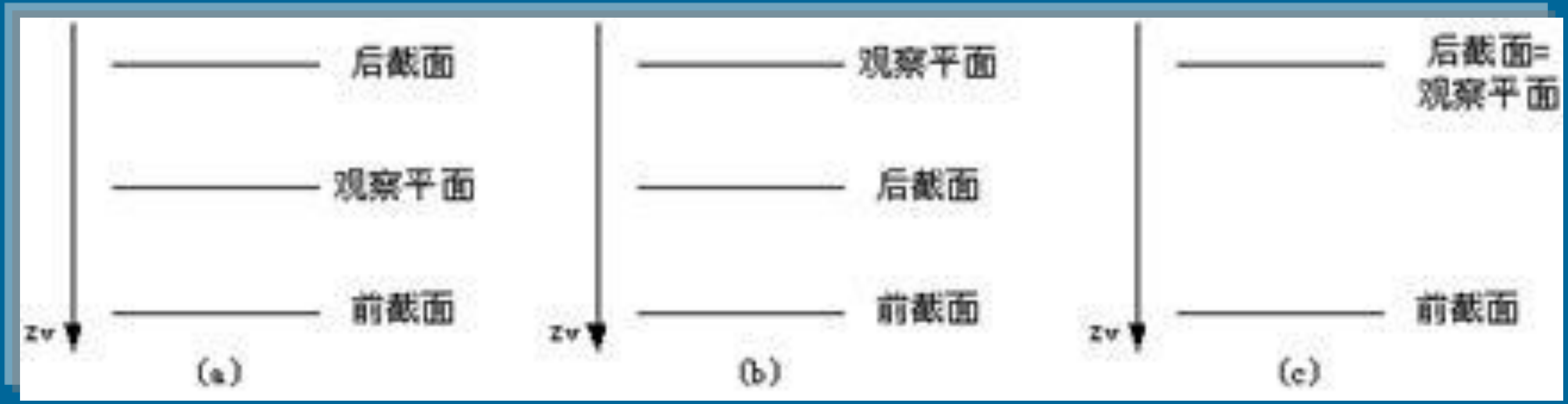
对于透视投影，观察体是顶点在投影参考点处的棱锥。



透视投影观察体

--正平行投影不受观察平面位置的影响。

--对于透视投影，前截面必须在投影中心和后截面之间。



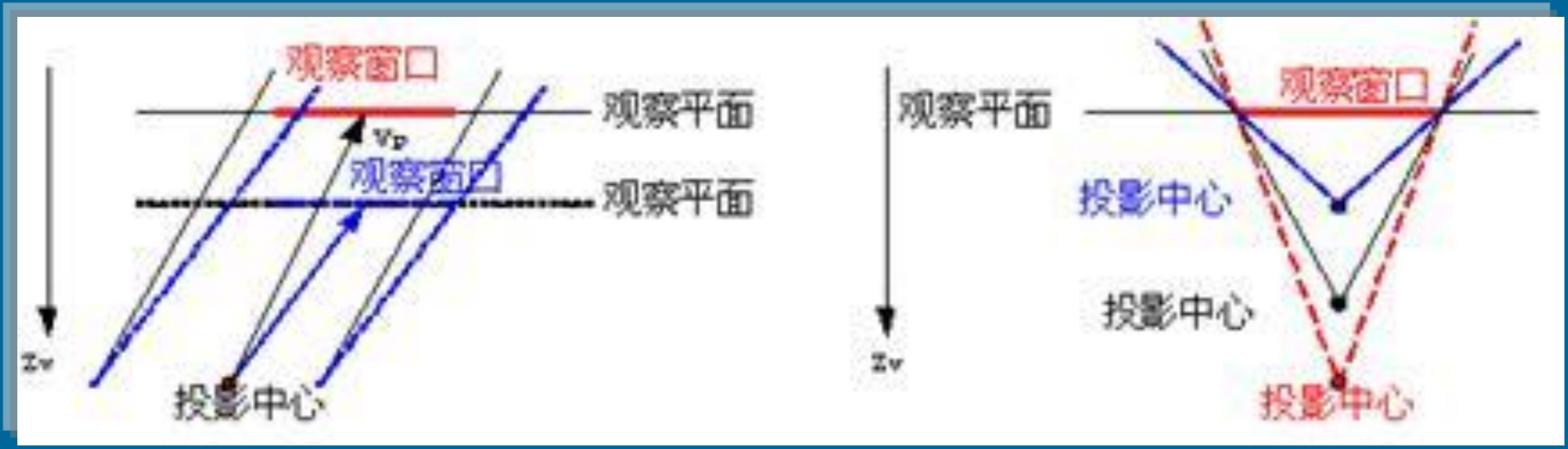
观察平面及前后截面的位置安排

投影中心与观察平面的位置决定了透视投影效果：

--由于正投影的投影线与观察平面垂直，因而正投影不受观察平面位置的影响；

--斜投影可能受到观察平面位置的影响；

--透视投影也会受到投影中心与观察平面位置的影响。



实质：投影线的变换导致观察空间发生了变化

视口裁剪（Cohen－Sutherland算法）

Cohen－Sutherland算法中的区域码的概念就可以推广到三维的情况：对场景中的各线段的每个端点(x,y,z)用6位而不是4位区域码来识别点与视口的位置关系。根据端点的位置，从右到左对区域码的6个二进制位置位。

如果$x<x_{min}$（端点位于窗口的左边），第一位值1；

如果$x>x_{max}$（端点位于窗口的右边），第二位值1；

如果$y<y_{min}$（端点位于窗口的下边），第三位值1；

如果$y>y_{max}$（端点位于窗口的上边），第四位值1；

如果$z<z_{min}$（端点位于观察体的前面），第五位值1；

如果$z>z_{max}$（端点位于观察体的后边），第六位值1。

课后习题

- 1）在进行投影变换时，常见的投影方式有平行投影和透视投影两类。
- 2）在进行投影变换时，常见的正投影方式有正平行投影和斜平行投影两类。
- 3）正投影可分为：三视图和正轴测图。当投影面与某一坐标轴垂直时，得到的投影为三视图；当投影面不与任意坐标轴垂直时，得到的投影为正轴测图

- 4）透视投影可分为：一点透视、两点透视和三点透视。当投影面与某一坐标轴垂直且与另外两个坐标轴平行时，得到一点透视；当投影面与两个坐标轴相交且与一个坐标轴平行时，得到两点透视；当投影面与三个坐标轴都相交时，得到的投影为三点透视。
- 5）在进行观察投影时，观察体的大小依赖于窗口的大小，观察体的形状依赖于生成显示结果的投影类型。

6）平行投影的特点是（BCD）

- A) 平行投影的深度感更强，更加具有真实感。
- B)保持物体的有关比例不变。
- C)能够精确地反映物体的实际尺寸。
- D)在三维空间平行的直线经平行投影后依然保持平行。

7）透视投影的特点是（ABCD）

- A) 透视投影的深度感更强，更加具有真实感，但透视投影不能够准确反映物体的大小和形状。
- B) 物体的透视投影的大小与物体到投影中心的Z方向距离成反比。
- C) 一组平行线若平行于投影平面时，它们的透视投影仍然保持平行。
- D) 只有当物体表面平行于投影平面时，该表面上的角度在透视投影中才能被保持。

8）什么是平面几何投影中平行投影？它有什么特点。

- 在平行投影中，三维物体的坐标沿平行线投影到观察平面上，它保持物体的有关比例不变，能够精确地反映物体的实际尺寸。在三维空间平行的直线经平行投影后依然保持平行。

9）简述什么是透视缩小效应

- 物体的透视投影的大小与物体到投影中心的Z方向距离成反比。

10）简述平面几何投影中透视投影的特点。

- 透视投影是沿着相交于视点（即：投影参考点Projection Reference Point）的投影线进行投影。
- 透视投影的深度感更强，更加具有真实感，但透视投影不能够准确反映物体的大小和形状。
- 物体的透视投影的大小与物体到投影中心的Z方向距离成反比。
- 一组平行线若平行于投影平面时，它们的透视投影仍然保持平行。
- 只有当物体表面平行于投影平面时，该表面上的角度在透视投影中才能被保持。

11）正轴测有等轴侧、正二侧和正三侧三种。当投影面与三个坐标轴之间的夹角都相等时为等轴侧；当投影面与两个坐标轴之间的夹角相等时为正二侧；当投影面与三个坐标轴之间的夹角都不相等时为正三侧。

12）什么是观察投影？观察体的结构依赖哪些因素？

- 在显示三维形体的过程中，在用户坐标系下定义一个观察空间（观察体），将观察空间外的物体裁剪掉，只对落在观察空间内的物体作投影变换并予以显示。
- 在进行观察投影时，观察体的大小依赖于窗口的大小，观察体的形状依赖于生成显示结果的投影类型。

13)什么是投影?

- 投影变换就是把三维立体（或物体）投射到投影面上得到二维平面图形。

第七讲 可见面判别算法（消隐）

7.1　可见面判别算法的分类
7.2　后向面判别
7.3　深度缓存算法
7.4　区域细分算法
7.5　扫描线算法
7.6　深度排序算法（画家算法）

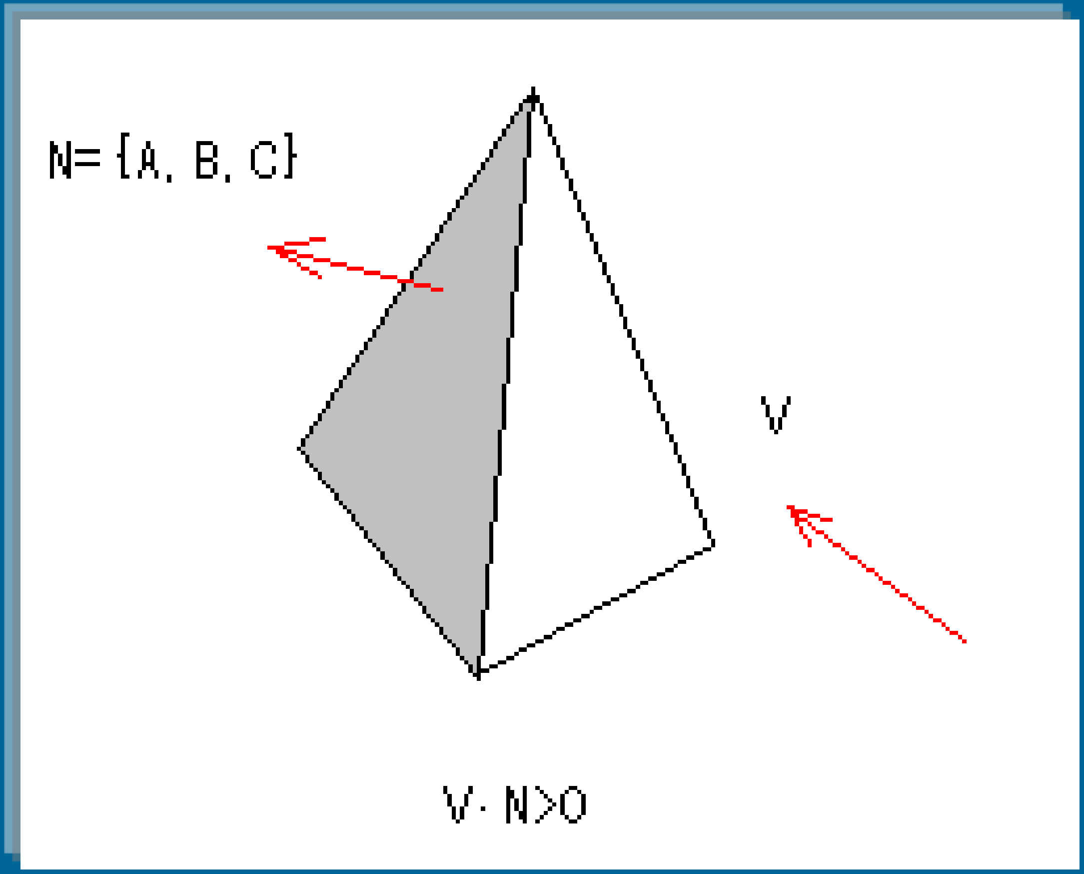
1）物空间算法

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。

2）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。

3）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

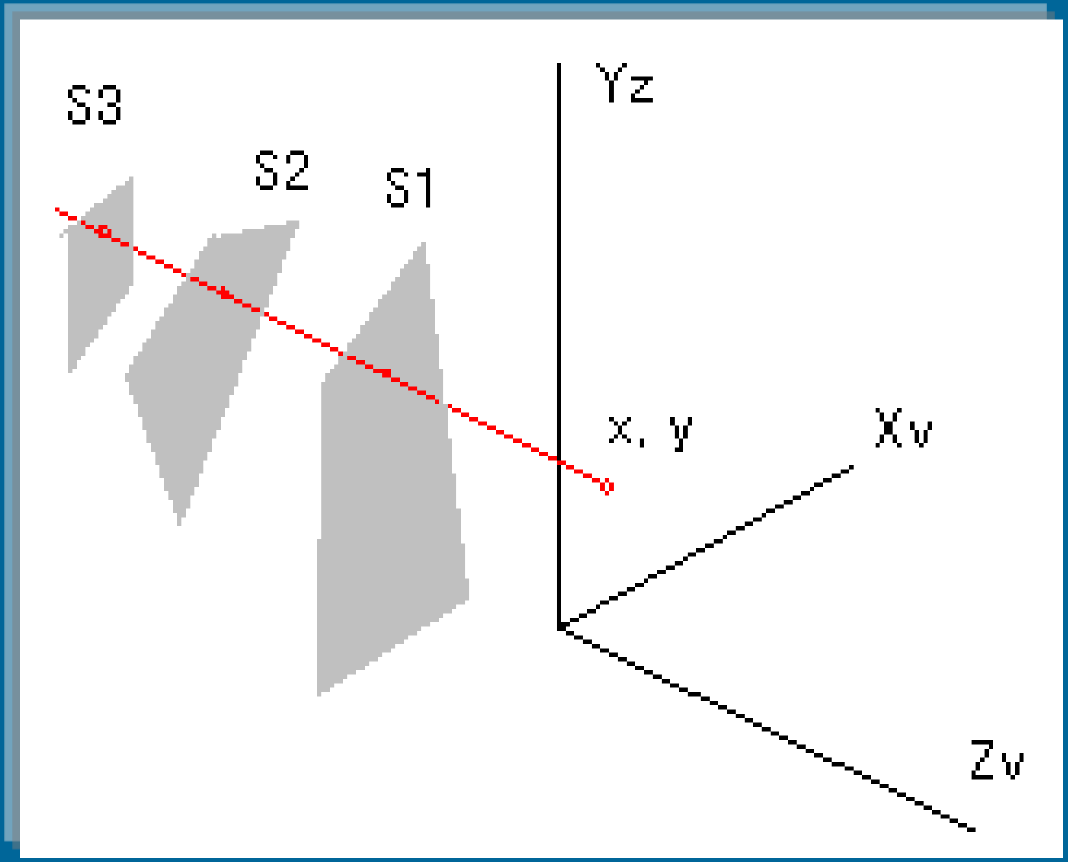
1） 在一个沿着 Z_v 轴反向观察的右手观察系统中，

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，盖面也是后向面。

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。

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

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

课后作业

- 1）按照消隐的工作空间进行分类，可以将消隐算法分为物空间算法和像空间算法两大类。
- 2）如果一个点（x，y，z）满足平面方程Ax+By+Cz+D>0，则该点在平面的外侧（N指向同一侧）。

课后作业

- 3）消隐的结果与被观察的物体有关，也与视点的位置有关。
- 4）通常按照消隐对象的不同，有两类消隐问题：线消隐和面消隐

5）解释一下消隐的含义。

- 答：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。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。

7）解释一下消隐中的像空间算法含义。

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。

8）下列哪些算法属于消隐算法（AB）

A)深度缓存算法

B)后向面判别

C)八叉数

D)结构实体几何（CSG）

- **9）下列哪些算法属于像空间消隐算法（AC）**
- **A)深度缓存算法**
- **B)后向面判别**
- **C)扫描线算法**
- **D)结构实体几何（CSG）**