国际会议

* 1967年成立计算机图形图象专业委员会，简称SIGGRAPH（the Special Interest Group on Computer Graphics ）
* 1974年开始每年在美国召开（2011，2014年在温哥华举行）
* 会议录用论文代表图形学研究的主流方向
* 2008年 SIGGRAPH Asia
  1. 发展历史

SAGE（Semi Automatic ground Environment System ）—麻省理工学院负责

1950年 图形显示器

1952年 数控铣床

1958年 GerBer 平台式绘图机、Calcomp 滚筒式绘图机

图形学创始人 Ivan E.Sutherland 1963年 MIT的博士学位

博士论文 《Sketchpad: A Man-machine Graphical Communications System 》

3-D 计算机建模, 可视化模拟, CAD, 虚拟现实等概念

1988年,获得图灵奖（以英国科学家阿兰.图灵的名字命名）

William Fetter ： 1960年，Boeing工作时提出

1964 年 Steve Coons 小块曲面表示自由型曲面，使曲面片边界达到任意阶连续

70年代 Integrated Circuit RAM( Random Access Memory)

1971年 Pierre Bezier UNISERFY

五、六十年代 准备和酝酿时期

六十年代 学科建立和进入应用时期

七十年代 蓬勃发展和广泛应用时期

八十年代以后 突飞猛进和成熟化、标准化时期

九十年代以后 集成化、智能化

20世纪80年代末 萌芽阶段 科研院所开始对计算机图形学技术的研究

20世纪90年代 发展阶段 曲线曲面造型，真实感图形显示方面取得一定的成果

2000年 至今 有文章入选SIGGRAPH，作者单位：微软亚洲研究院，中科院深圳先进技术研究院，清华大学，浙江大学

计算机图形学是研究怎样用计算机***生成***、***处理*** 和***显示***图形的一门新兴学科。

计算机图形学是由很多交叉学科形成的。

图像处理（Image Processing）

* 图像的数字化
* 图像的增强
* 图像恢复
* 图像编码
* 图像重建
* 图像分析

图形与图像

* 两个概念间的区别越来越模糊
* 区别：
  + 图象纯指计算机内以位图(Bitmap)形式存在的灰度信息
  + 图形含有几何属性，或者说更强调场景的几何表示，是由场景的几何模型和景物的物理属性共同组成的
  + 更详细的介绍见数字图像部分。。。

模式识别（Pattern Recognition）

研究怎样分析和识别输入的图像，找出其中蕴含的内在联系或抽象模型

计算几何（Computational Geometry）

计算几何是一门研究几何模型和数据处理的学科。着重讨论形体的计算机表示、几何模型的建立、模型数据的存储和管理

计算机图形学的研究内容：

* 图形的输入
* 图形的生成和输出
* 图形结果的处理

计算机图形学的应用

* 计算机辅助设计与制造（CAD/CAM）
* 作战指挥和军事训练
* 计算机动画和艺术
* 地理信息系统（GIS）

CAD/CAM

* CAD/CAM是计算机图形学应用的一个最广泛、最活跃的领域。
* 美国波音公司 1990年10月开始波音777的全新设计，全部设计工作在计算机终端和图形工作站上进行，在全世界范围内首次实现了“无纸设计”。整个设计、生产、试飞周期4年半。
* 中国大飞机计划：

C919 08年开始，2015年11月首架下线

* 驾驶模拟训练

驾驶训练模拟器，也叫“汽车模拟仿真器”。该系统集计算机技术（高端PC工作站）、虚拟现实技术（VR）、自动化技术、多媒体技术为一体，使学员从视觉、听觉和操作感觉上都能体会出与操纵实车一样的感觉

计算机动画和艺术

* 侏罗纪公园 玩具总动员 泰坦尼克号……
* 西游记之大圣归来
* 美国Autodesk 3D Max

加拿大 Alias （被SGI公司收购）

美国 SGI Maya……

GIS

* 1962年 Roger Tomlinson
* 1998年 戈尔 Open GIS Consortium

数字地球是指以地球坐标为依据的，具有多分辨率的，由海量数据组成的，能立体表达的虚拟地球。

* 2010年 IBM 智慧城市
* 实时化数据
* 地理数据：土地利用，路网，POI
* 交通数据：地感线圈、监控摄像，浮动车
* 移动电话：通话数据，短信数据
* 通勤数据：羊城通，咪表
* 环境监测数据：CO2，PM2.5， 噪声
* 社会网络数据: Twitter, Weibo
* ……

可视化

* 将科学计算过程中及计算结果的数据转换成图形或图像在屏幕上显示出来，并进行交互式处理的理论、方法和技术。
* 原始数据和计算结果的可视化
* 地图数据的可视化趋势：2d-> 3d, pc ->Web

*Visualising Spatial and Social Media , CASA Working paper*

逆向工程

* 是相对于“设计思路→产品”的一般产品开发而言。
* 分为三个层次： 原型复制（copy）→改进（redesign）→仿真（simulation）
* 关键技术：数据获取、产品建模
* 原型表面数字化→数据点网格化→几何特征提取→重建CAD模型

虚拟人

* 美国 1989年 “可视人计划”

1991年 人体切片数据库建立

1994年，美国科学家将一具男尸切成1878多片，每片厚度1毫米，数据量15G；

1998年，又将一具女尸切成5190多片，每片厚度0.33毫米，数据量30G

2000年 8月提供人体数据集

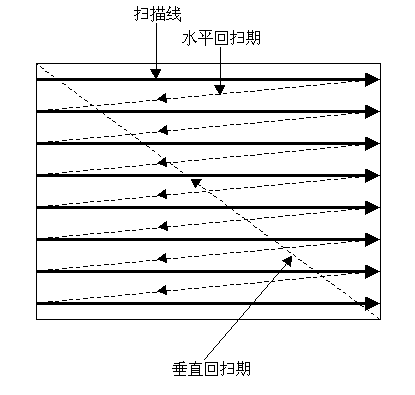
* 韩国 2000年 开始可视人5年计划现已经提前完成
* 中国 2001年11月 “863”项目 中国科学院、首都医科大学、华中科技大学、第一军医大学
  + 虚拟人女I号 切片厚度0.2毫米
  + 虚拟人男I号 切片厚度0.1毫米
  + 钟世镇 广东省科学技术突出贡献奖 （2008）
* 数字可视人→数字物理人→数字生理人→数字智能人

输入设备

* + 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
  + 第五阶段：用户的思维

显示设备的发展

随机扫描显示器：根据需要，电子束在屏幕上按任意位置扫描，扫描方向和顺序不受限制，故称随机扫描。随机扫描得到的图形只能是线条图，所以又称为矢量扫描。

光栅扫描显示器：

硬拷贝设备：打印机

* 绘图仪

平板式 速度慢，精度高

滚筒式 速度快，精度低

* 三种类型的图形软件系统
* 用某种语言写成的子程序包

如: GKS (Graphics Kernel System)

PHIGS( Programmer’s Hierarchical Interactive Graphics system )

GL ( Graphic Library)

便于移植和推广、但执行速度相对较慢，效率低

* 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。

简练、紧凑、执行速度快，但不可移植

* 专用图形系统：效率高，但系统开发量大，可移植性差。

软件标准

通用的、与设备无关的图形包，图形标准

* + GKS (Graphics Kernel System) (第一个官方标准)
  + PHIGS(Programmer’s Hierarchical Interactive Graphics system)
  + IGES (Initial Graphics Exchange Specification)
  + STEP (Standard for the Exchange of Product Model Data)

一些非官方图形软件，广泛应用于工业界，成为事实上的标准

* + - DirectX (MS)
    - Xlib(X-Window系统)
    - OpenGL(SGI)
    - Adobe公司Postscript

OpenGL(Open Graphic Library)

* 1992年 SGI公司推出 是一套独立于操作系统和硬件环境的三维图形库。
* Microsoft、IBM、DEC、SUN等公司将其作为图形标准。
* Visual C++2.0以上版本内置的OpenGL

光栅图形学

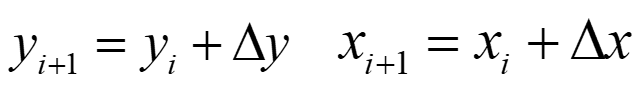
* 显示器是由离散像素组成的矩阵，在绘制具有连续性质的直线、曲线或区域等基本图形时，需要确定最佳逼近它们的像素，这个过程称为光栅化。
* 当光栅化按照扫描线的顺序进行时，它被称为扫描转换。

**生成直线的算法**

1.一般直线 ，从起点到终点，x每次增加(或减少)1，用直线方程计算对应的y值，再用SetPixel(x, int(y+0.5),color)输出该像素。

**复杂度：乘法+加法+取整**

2.DDA算法(Digital Differential Analyzer)

假设直线的端点为：(x1,y1) 、(x2,y2) ，则其微分方程为：

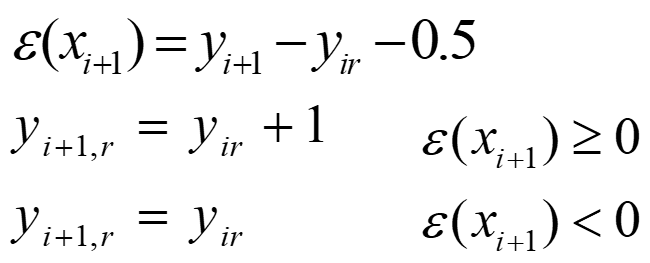
解为：

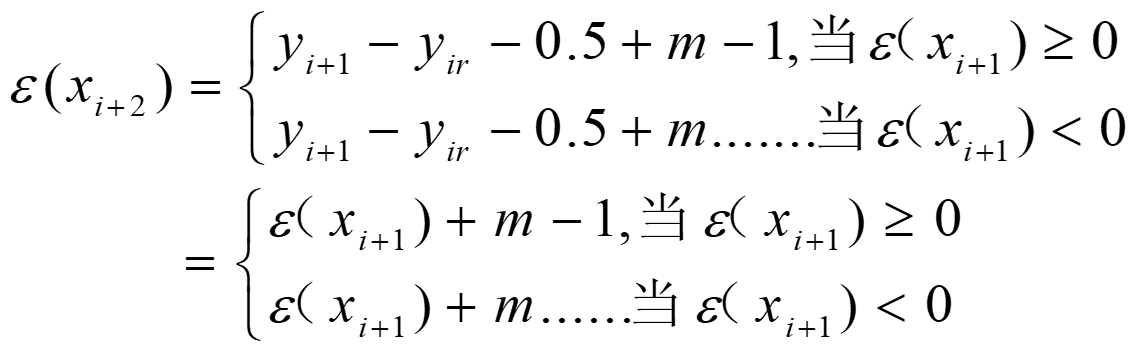


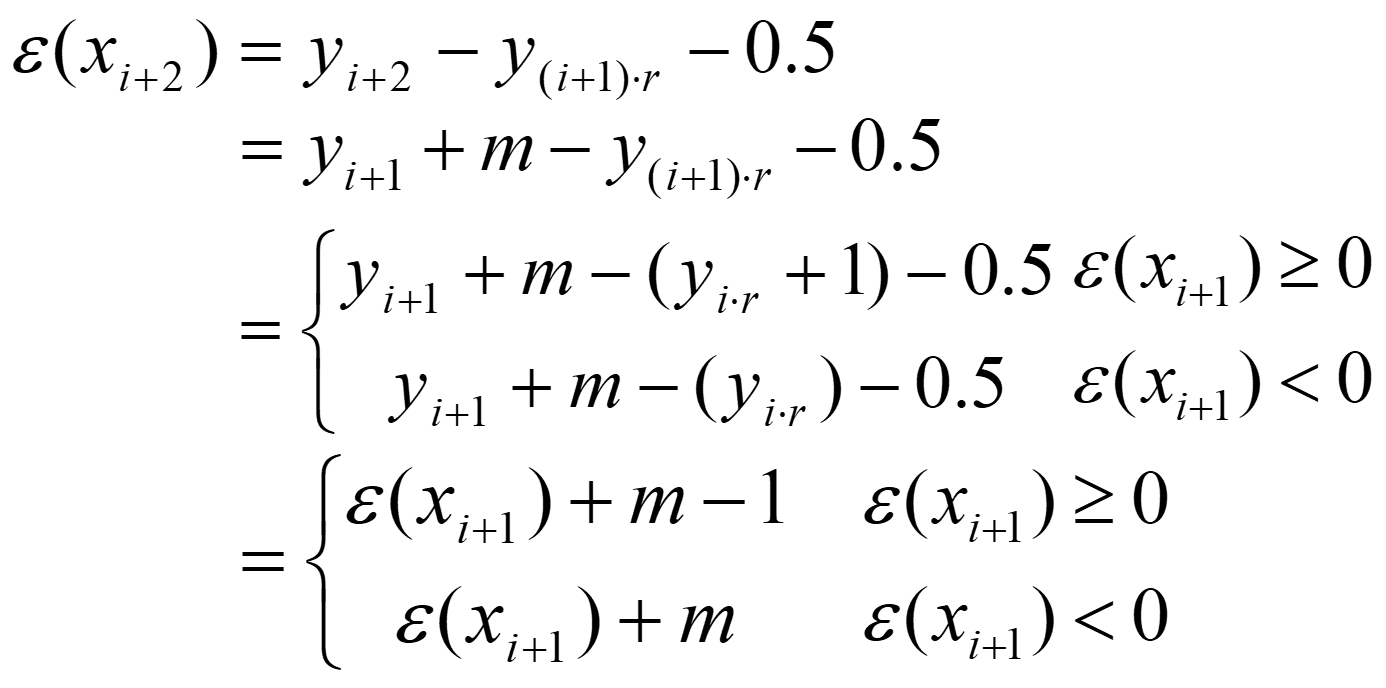
减少了浮点乘法，提高了效率。但是x与dx、y与dy用浮点数表示，每一步要进行四舍五入后取整round().

3.Bresenham算法

设xi列上已经用（xi,yir）表示直线上的点，又设B点是直线上的点，其坐标为（xi+1,yi+1),显然下一个表示直线的点只能在C点或D点中选。设A点为CD边的中点，则B在A的上面则应取D点，否则取C点

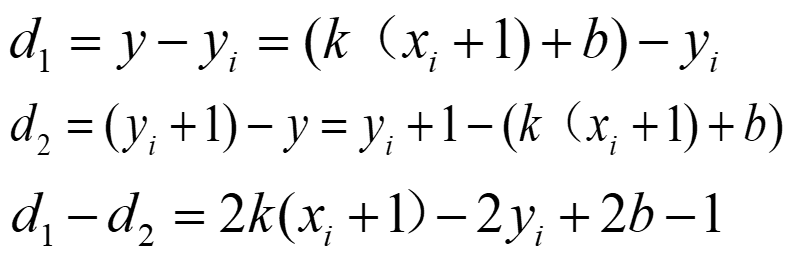
****

****

** 初始e=m-0.5**

整数Bresenham算法

根据误差项d的值来决定是否增1的过程如下：



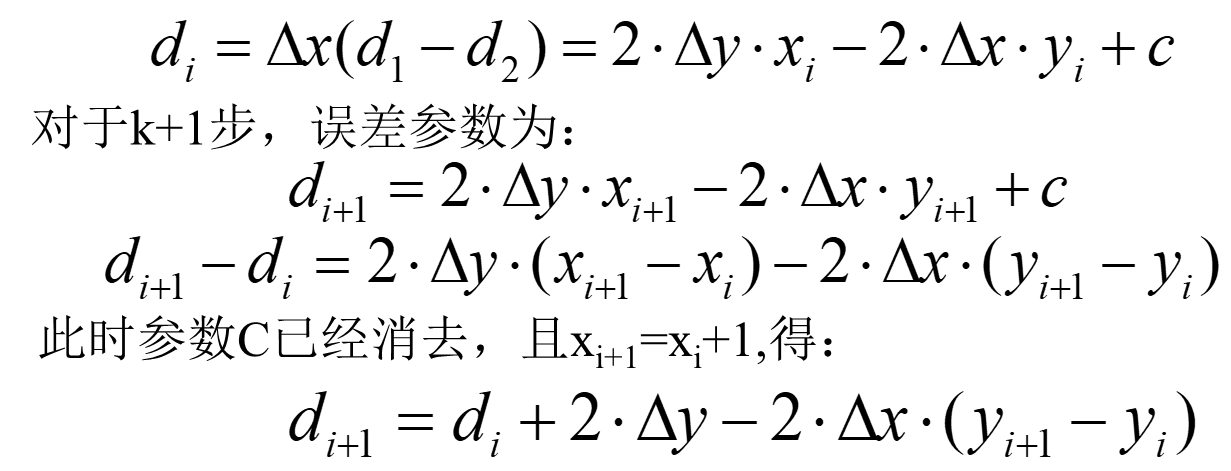
设Δy=y1－y0, Δx=x1－x0，则k=Δy/Δx，代入上式，得；

是常量，与像素位置无关



di的符号与(d1-d2)的符号相同。

* 当di<0时，直线上理想位置与像素(xi＋1，yi)更接近，应取右方像素；
* 当di>0时，像素(xi＋1，yi＋1)与直线上理想位置更接近；
* 当di=0时，两个像素与直线上理想位置一样接近，可约定取(xi＋1，yi＋1)。



**初始d=2Δy-Δx，若yi+1=yi+1，则di+1=di+2Δy-2Δx；若yi+1=yi，则di+1=di+2Δy**

圆弧

1.根据圆的基本方程，可以沿x轴，x从0到 ，以单位步长计算对应的y值来得

到圆周上每点的位置： （只要实现1/8圆）

像素间间隔不一，随着x的增加，间隔越来越大

2.使用极坐标来计算沿圆周的点，此时，圆使用参数方程来表示： ，

像素间间隔不一，随着x的增加，间隔越来越大

3.中点法

以从(0，R)到( ， )的1/8圆为例；

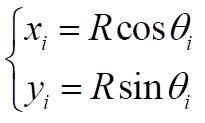
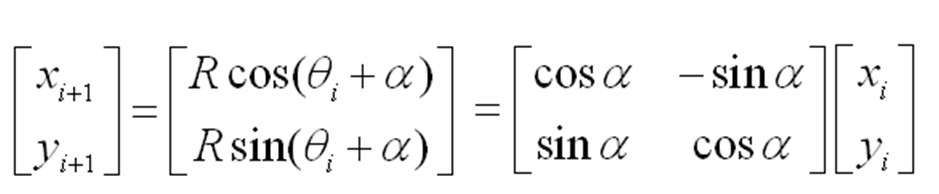
假定当前已确定了圆弧上的一个像素点为P(xp,yp)，那么下一个像素只能是其右方、右下方的点P1、P2，记M为P1、P2中点

F(x,y)=x2+y2+-R2 ，while(x<=y)

* 若F(M)<0，M在圆内，P1点离圆弧更近，取P1为下一个像素；
* 若F(M)>0，M在圆外，P2点离圆弧更近，取P2为下一个像素；
* 若F(M)=0，M在圆上，P1、P2可任取，这里约定取P2 ；



4.多边形逼近

误差 ，限差Delta，

，

裁剪：按预先设置的窗口参数，沿窗口边框线对图形进行裁剪，保留窗口内部图形，裁剪掉窗口外图形的方法

空间任意一点(x,y)是否在窗口内的充分必要条件：



**1.Cohen-Sutherland算法**

关键技术：在于总是要让直线段的一个顶点处于窗口之外，例如P0点。这样P0点到交点P的直线段必然不可见，故可以直接抛弃

* 输入直线段的两端点P1、P2坐标，以及窗口的参数。
* 对P1、P2进行编码；
* Code1、Code2都为0000，则此直线为可见；
* Code1、Code2经位与运算不等于零，则此线段位于窗口的同一侧，为不可见；
* 算出直线与窗口的交点，将直线分割，舍去交点之外的一段，再对另一段重复上述处理。

裁剪直线段时，一般按固定顺序，如左（x=wxl），右（x=wxr）、下（y=wyb）、上（y=wyt）求解窗口边界与直线段的交点。

求交方法为直接代入边界值，求横纵坐标

特点：该算法用编码的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，这使得它在两类裁剪场合中非常高效：

（1）大窗口的场合，其中大部分线段为完全可见；

（2）小窗口的场合，其中大部分线段为完全不可见

* 但是求交计算较复杂

**2.中点分割法**

* 核心思想是通过二分逼近来确定直线段与窗口的交点。
* 是Sproul 和Sutherland为了便于硬件实现而提出的。

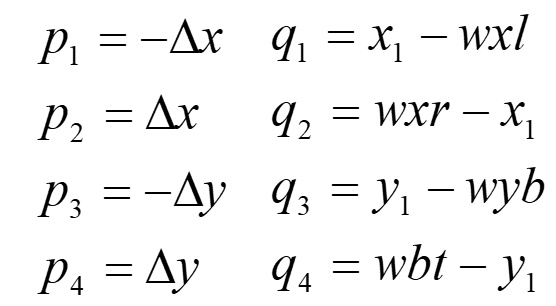
线段既非完全可见，也非完全不可见。求得中点Pm1。两半线段均不能简单接受或拒绝。不断求中点去掉确定的线段部分，最后得到交点

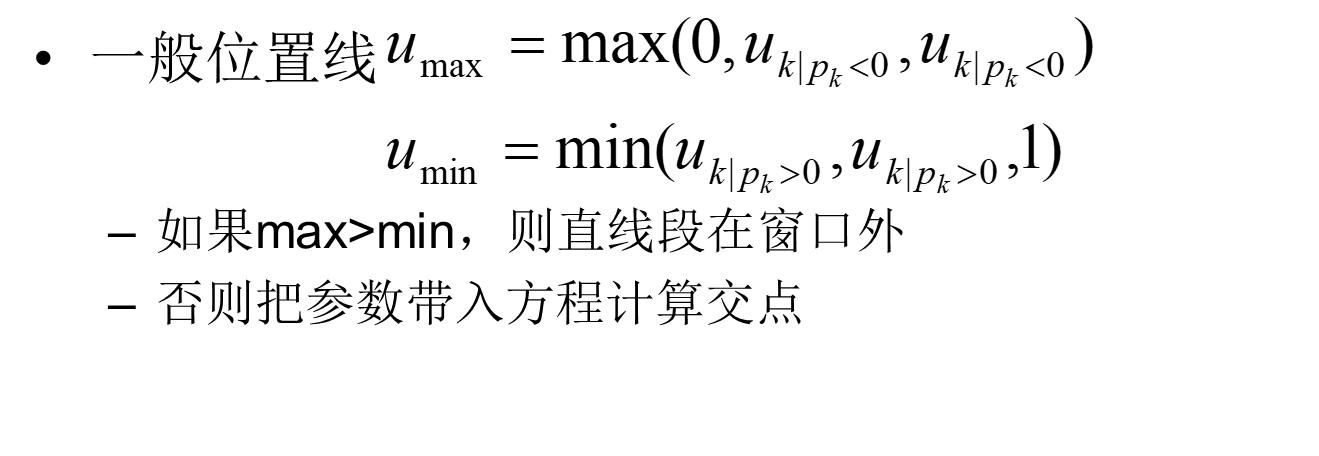
* 对分辩率为2N\*2N的显示器，上述二分过程至多进行N次。
* 主要过程只用到加法和除法运算，适合硬件实现。适合并行计算。

**3.Liang-Barsky算法**

* 标准矩形窗口的直线裁剪算法
* 算法的基本出发点：直线的参数方程



如果直线上的点（x,y）位于窗口内，则满足一下条件：



裁剪窗口的每条边界线将平面分为两个区域。定义裁剪窗口所在侧为可见侧（visible side），另一侧为不可见侧（invisible side），图中阴影线所示为边界线的可见侧。这样，裁剪窗口也可以定义为所有边界线的可见侧。

多边形裁剪

* 一个完整封闭的多边形经窗口边框裁剪后，一般不再封闭。
* 一个完整封闭的非凸多边形，常被窗口边框裁剪成几个独立的小多边形。

Sutherland\_Hodgman算法

* 该算法的基本思想是：通过对单一边的裁剪实现对多边形的裁剪。
* 用窗口的一条边框线对多边形进行裁剪，得到一个或几个新的多边形，再用第二条边框线对这些新的多边形进行裁剪。如此进行四次，从而把多边形相对于窗口的全部四条边框线进行了裁剪。



* 凸多边形用此算法可以获得正确的裁剪结果，而且可以推广到任意凸多边形窗口的情况。
* 在处理凹多边形时，

\* *只能处理裁剪后仍为一个连通图的凹多边形*

\* 裁剪之后产生多个分离部分的凹多边形，该算法会产生一些多余的边

二维填充

**多边形的表示**

* 顶点表示

是用多边形的顶点序列来刻画多边形。这种表示方法直观、几何意义强、占空间少，易于进行几何变换，被广泛地应用于各种几何造型系统中。

* 点阵表示

是用位于多边形内部的像素的集合来刻画多边形。这种表示方法虽然失去了许多重要的几何信息，但它是光栅显示系统显示时所必须的表示形式。

**扫描转换多边形**

* 多边形从顶点表示到点阵表示的转转，称为扫描转换。
* 常用算法有两大类
  + 扫描线算法
  + 种子填充

**逐点判断算法**：逐个判断绘图窗口内的像素，确定它们是否在多边形区域内部，从而求出位于多边形区域内的像素集合

包围盒：包围多边形的最小矩形。只有在包围盒内的点需要检查。

* 对于凸多边形，可以极大地减少所需检查的象素个数，但是对凹多边形，则减少的检查工作量则要少得多。

点与多边形的关系

射线法—从点发出射线与多边形的边相交，若交点的个数为奇数，则点位于多边形内；若为偶数，则点位于多边形外。通常为了简单起见，射线为水平线或者垂直线。

注意射线通过多边形顶点的情况。

**扫描线算法**：扫描线上相邻象素几乎具有相同的特性，这就是所谓扫描线的连贯性。

在给定的扫描线上，象素的这种特性只有在多边形的边和该扫描线交点处才会发生变化。

* 扫描线与多边形求交
* 交点按照x升排序
* 交点两两配对
* 每对交点所确定的区段取多边形的填充色，交点对之间的区间则取背景色。

顶点为局部极值点，则计***两次***，否则计***一次***

**种子填充**：

* 假设在多边形或区域内至少有一个象素是已知的，然后设法找到区域内所有其它象素，并对它们进行填充。（4连接、8连接）

1.种子象素入栈，当栈非空时，重复执行：

* 栈顶象素出栈；
* 将出栈象素置成多边形色；

2.四向检查象素，是否为边界象素或是否已经设置成多边形颜色。若是则忽略不计，否则把该象素压入堆栈

缺点：

* 有些象素会入栈多次，降低算法效率；栈结构占空间。
* 递归执行，算法简单，但效率不高，区域内每一象素都引起一次递归，进/出栈，费时费内存。

**扫描线种子填充**：

Step1：栈顶象素出栈，

Step2：沿扫描线对出栈象素的左右象素进行填充，直到遇到边界象素为止，即填充区间。

Step3：检查上下扫描线，选相应区间的最右象素作种子象素入栈。

**二维图形变换**

**几何变换**：几何变换提供了构造或修改图形的一种方法，图形在方向、尺寸和形状方面的改变都可以通过几何变换来实现。

一般来说图形的几何变换是指对图形的几何信息经过平移、比例、旋转等变换后产生新的图形。

把变换矩阵作为一个算子，作用到图形一系列顶点的位置矢量，得到新的顶点序列。

几何变换是在同一坐标系下进行的，因此，变换中坐标系是静止的，而图形是变动的。

基本几何变换都是相对于原点和坐标轴进行的几何变换，有平移、缩放和旋转等。

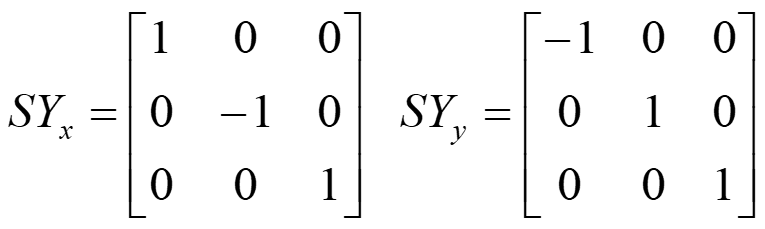
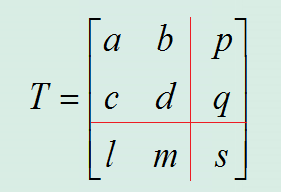
**平移**是指将p点沿直线路径从一个坐标位置移到另一个坐标位置的重定位过程。

齐次坐标: 用n+1维向量表示n维向量

(x,y)点对应的齐次坐标为 ，标准齐次坐标为：(x,y,1)

**变换具有统一表示形式的优点**

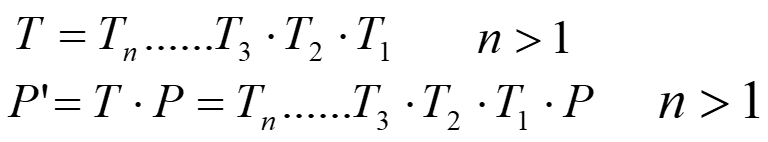
用规范化齐次坐标表示的二维基本几何变换矩阵是一个3×3的方阵，简称为二维变换矩阵





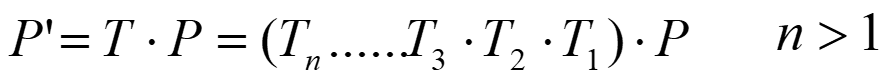
**复合变换**

由于引入了齐次坐标，基本几何变换均可以表示成 的形式

复合变换时指图形作一次以上的几何变换。

复合变换具有同样的形式。所不同的是，此时有：

由于矩阵的乘法满足结合律，因此，通常在计算时先求出T，再与P相乘。即



任意参照点的放缩变换：

（1）把固定点移到坐标系的原点；

（2）把物体相对于坐标原点进行比例变换；

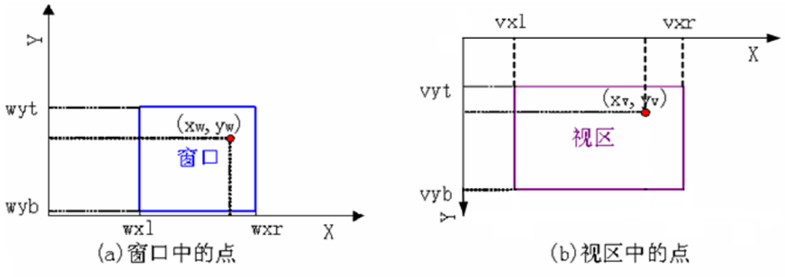
（3）再把固定点移回到原来的位置

**窗口到视口的变换**

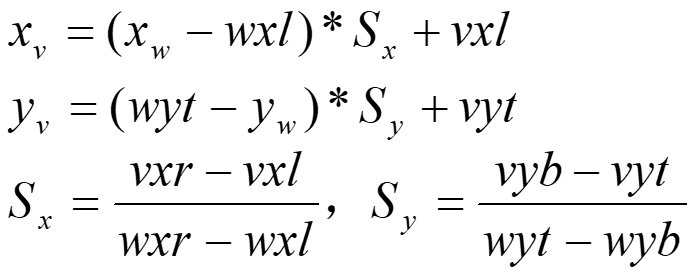
要将窗口内的图形在视区中显示出来，必须经过将窗口到视区的变换（Window-Viewport Transformation）处理。

要将窗口内的点（xw,yw）映射到相对应的视区内的点（xv,yv）需进行以下步骤：

(1) 将窗口左下角点移至用户坐标系的坐标原点

(2) 针对原点进行比例变换

(3) 进行反平移



当Sx=Sy时，X和Y方向的变形相同，变换后图形形状保持不变。

**三维图形的基本几何变换**

基本几何变换都是相对于原点和坐标轴进行的几何变换，有***平移***、***缩放***和***旋转***等。在以下的讲述中，均假设用P(x,y,z)表示三维空间上一个未被变换的点，而该点经过某种变换后得到的新点用P’(x’,y’,z’)表示。

旋转角度正负的确定：当沿坐标轴的正向往坐标原点看过去时，**逆时针**方向旋转的角度为正向旋转角

绕任意轴的旋转

1）使P1点与原点重合（图(b)）；

2）使得轴落入坐标平面内（图(c)）

3）使与z轴重合（图(d)）；

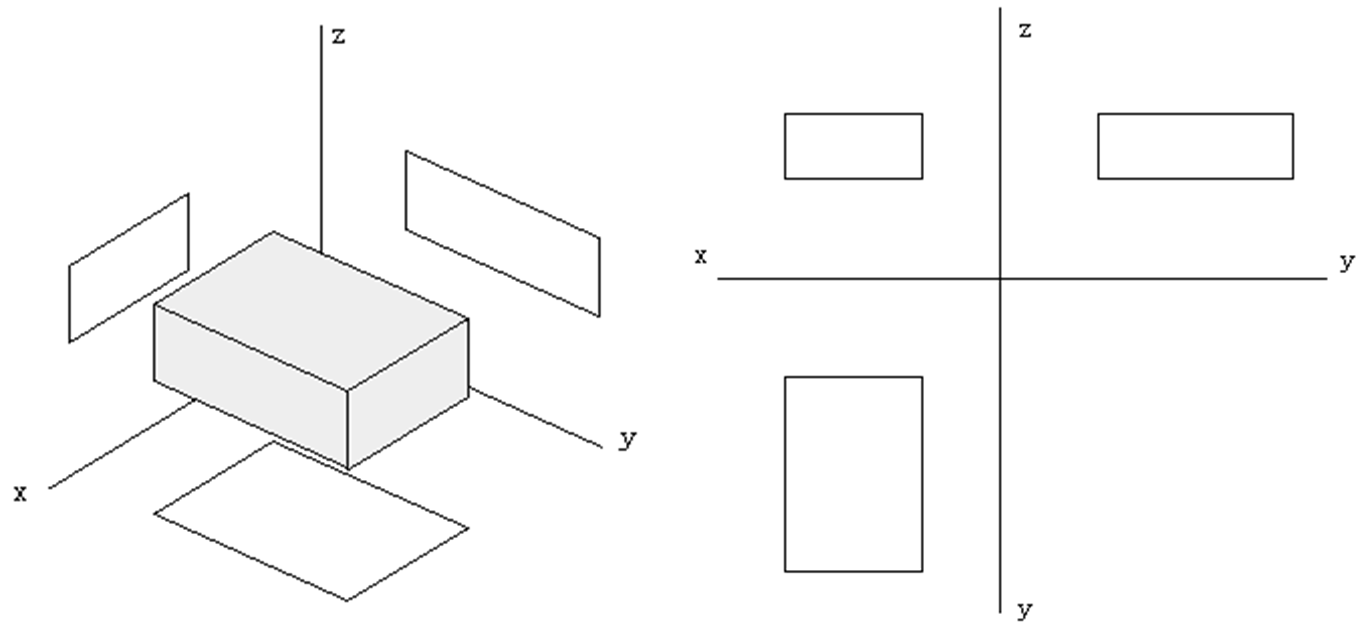
4）执行绕轴的角度的旋转(图(e))；

5）作变换3-2-1的逆变换；

投影：投影将三维物体变换为二维图形表示的过程。

投影变换的主要元素：光源、投影方向、投影平面

平行投影可根据投影方向与投影面的夹角分为两类：正投影和斜投影。当投影方向与投影面垂直时，为正投影，否则为斜投影。



  主视图 俯视图 左视图