

课程说明

➤ 主要讲述图形绘制管道

❖ 模型坐标变换、视点矩阵变换、消隐、光照、纹理、扫描转换、平面剪切等计算机图形处理的基础知识。

➤ OpenGL编程语言

➤ 图形绘制基础知识：图形变形、曲线、曲面、动画

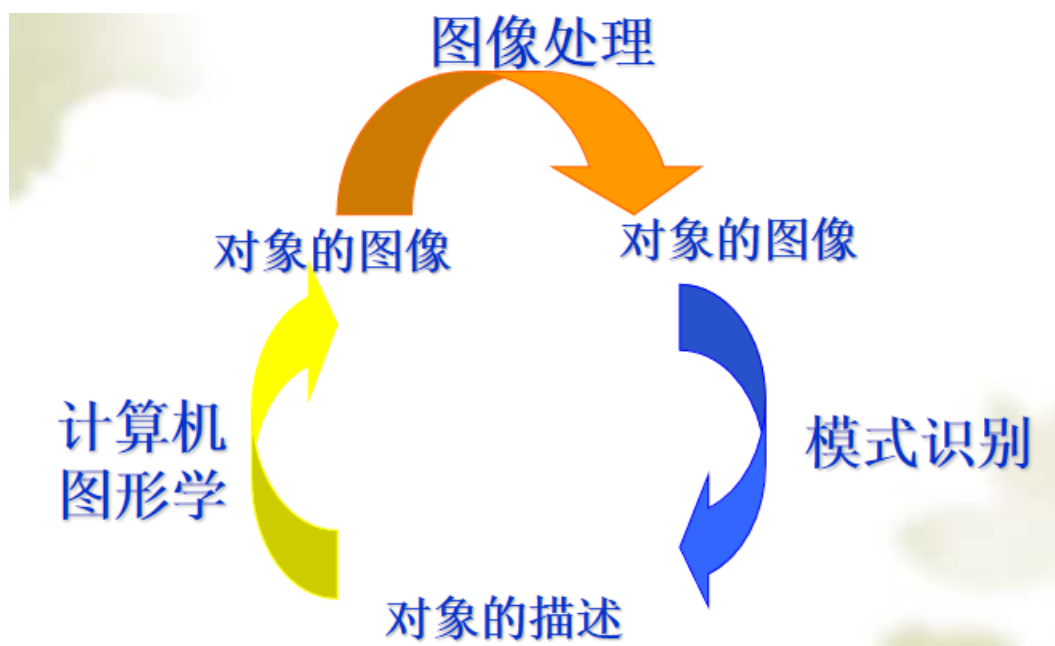
主要知识点：

1,数字图像处理、 计算机图形学、模式识别三门课之间的关系

图像处理(image processing)：利用计算机对图像进行分析处理，继而再现图像。

计算机图形学(computer graphics)：研究图形的计算机生成和基本图形操作，是从数据描述到图形生成的过程。

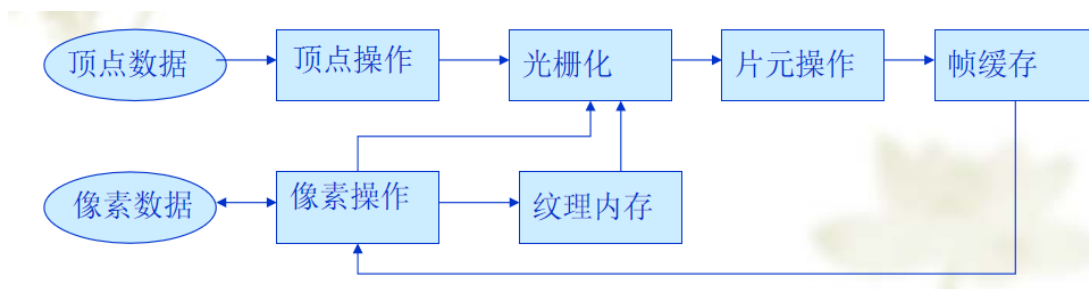
模式识别(pattern recognition)：计算机对图形信息进行的识别和分析描述，是从图形（图像）到描述表达的过程。



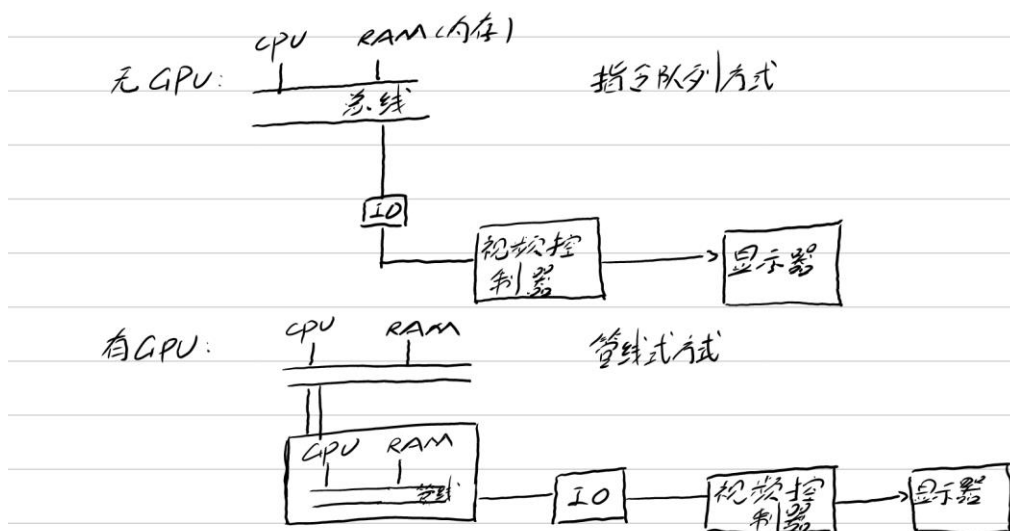
彼此交叉、界线模糊、相互渗透

2,就能够根据图进行分析。图形管线的各个阶段工作。

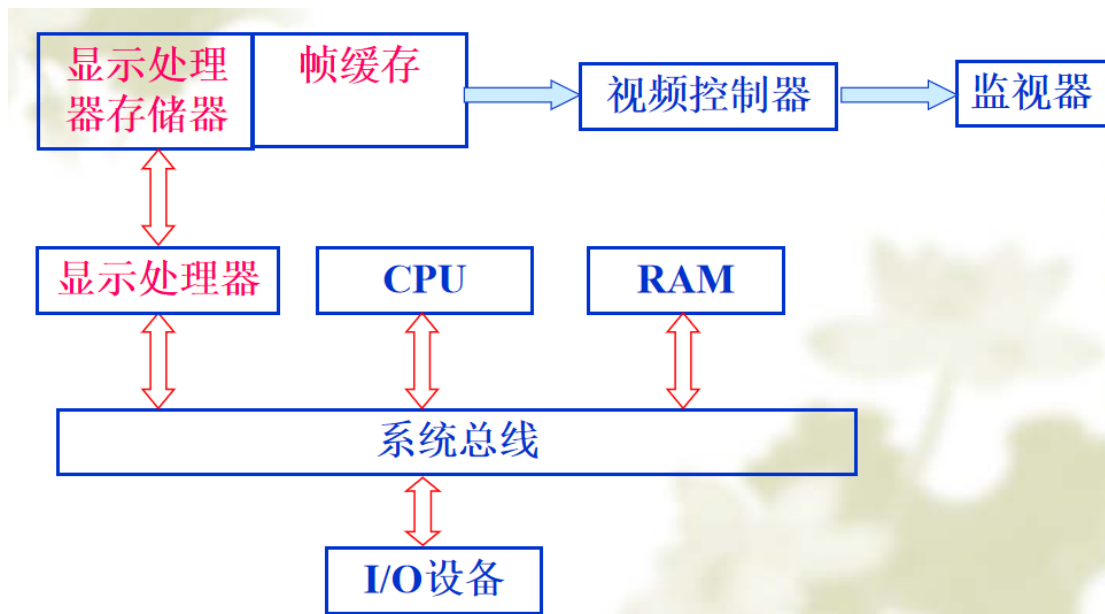
顶点数据 → 顶点处理 → 光栅化 → 片元操作 → 帧缓存
(顶点变换、属性设置)



3,计算机图形处理架构 (有 GPU 模式和无 GPU 模式)



4,简述光栅式扫描显示系统



简单的光栅扫描系统结构:

显示处理器: 可将应用程序给出的图形定义数字化为一组像素强度值, 并存放在帧缓冲器中

帧缓冲存储器(frame buffer): 保存一组对应屏幕所有点的强度值, 经视频控制器可绘制出图形

视频控制器: 有两个寄存器用来存放屏幕像素的坐标, 可实现图形的基本刷新操作

5, 屏幕分辨率三种描述, 扫描频率, 带宽计算

屏幕分辨率三种表示方法:

- (1), 光点直径 (CRT 屏幕上发光荧光层亮点的亮度分布图)
- (2), 水平方向上的光点数 \times 垂直方向上的光点数 $r(x \times y)$
- (3), 显示器精度 dpi

显示器的扫描频率: 也叫刷新率, 分为行频 (水平扫描频率) 和场频 (重绘率/垂直扫描频率/帧频)

行频: 电子枪每秒在屏幕上所扫描过的水平线数

场频: 每秒钟重复绘制显示画面的次数

视频宽带: 单位时间内所扫描的点数

6, 用 bresenham 算法光栅化线段 $P_1(30,20)$, $P_2(40,27)$

Bresenham算法的步骤

- 1) 输入直线端点坐标 (x_0, y_0) , (x_n, y_n)
- 2) 画起始点 (x_0, y_0) ;
- 3) 计算决策参数 $P_0 = 2\Delta y - \Delta x$,
- 4) 以单位步长增加 x 坐标, 计算 P_{k+1} , 确定下一点 (x_{k+1}, y_{k+1})

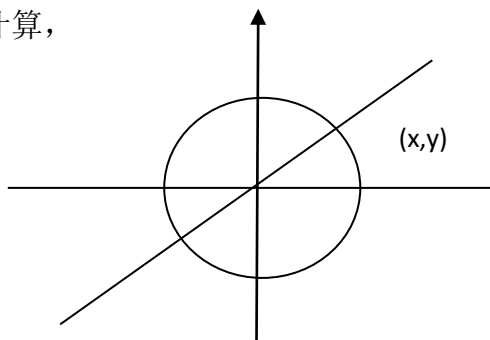
$$P_{k+1} = \begin{cases} P_k + 2\Delta y & (P_k < 0) \quad y_{k+1} = y_k \\ P_k + 2\Delta y - 2\Delta x & (P_k \geq 0) \quad y_{k+1} = y_k + 1 \end{cases}$$

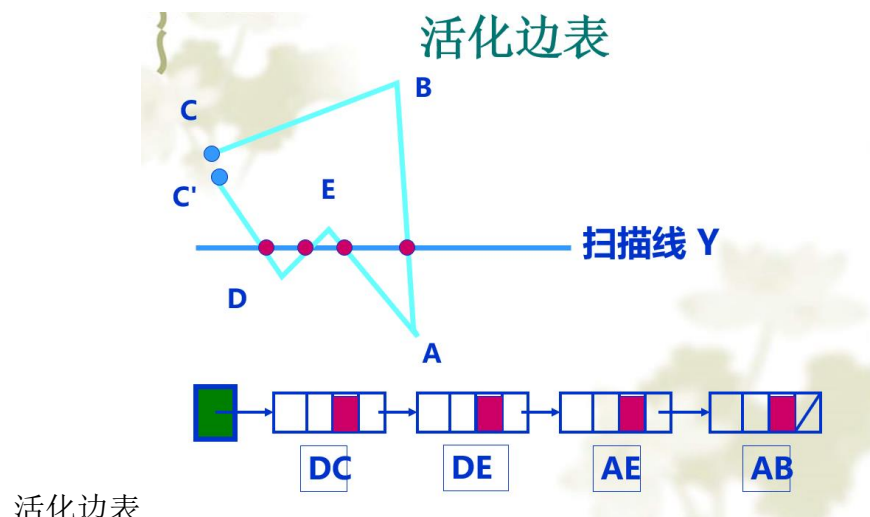
- 5) 重复第4步, 直至 $x_k = x_n$

$$\begin{aligned} \Delta y &= 7, \Delta x = 10 \\ 2\Delta y - 2\Delta x &= -6 \\ P_0 = 2\Delta y - \Delta x &= 4 \end{aligned}$$

step k	P_k	(x_{k+1}, y_{k+1})
0	4	(31, 21)
1	-2	(32, 21)
2	12	(33, 22)
3	6	(34, 23)
4	0	(35, 24)
5	-6	(36, 24)
6	8	(37, 25)
7	2	(38, 26)
8	-4	(39, 26)
9	10	(40, 27)

7, 画图描述圆对称性坐标计算,





3) 根据活化边表进行配对并填色，更新活化边表并重新排序；然后进入下一条扫描线，重复计算交点、建立活化边表并进行填充，直至扫描线值为最高顶点的 y 坐标值

算法描述

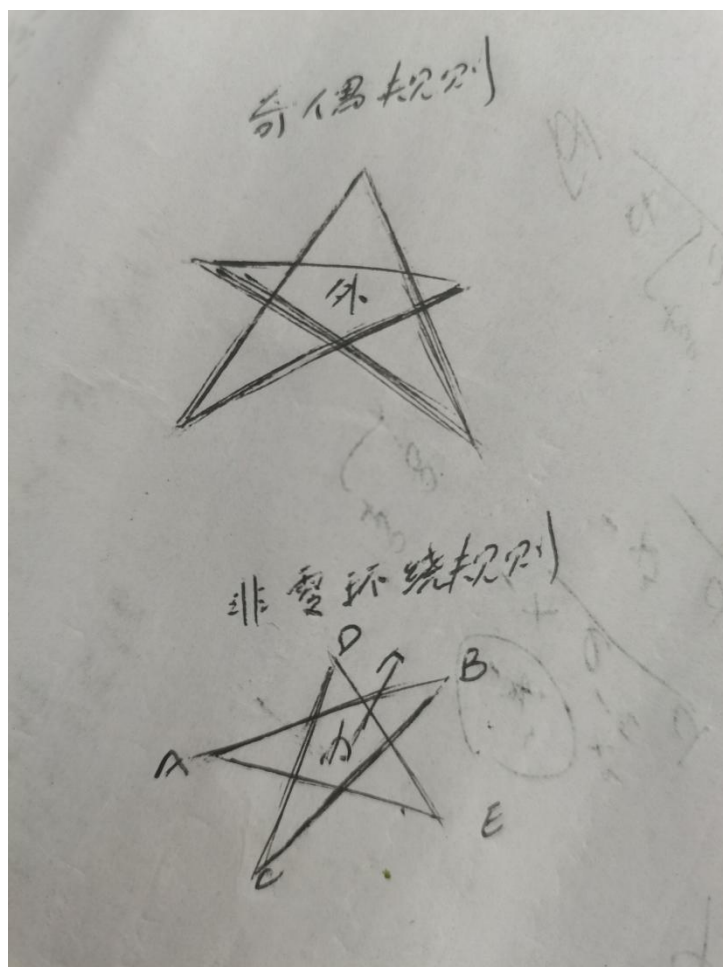
- 1) 输入多边形顶点数及顶点坐标
- 2) 建立有序边表
- 3) 根据当前扫描值建立活化边表
- 4) 填充
- 5) 更新活化边表并重新排序
- 6) 进入下一条扫描线，重复步骤3，直至扫描线值为最高顶点的 y 坐标值

9, 分别用奇偶规则和非零环绕规则判断下列内外区域

奇偶规则：从任意位置 P 作不经过顶点的射线，计算射线穿过的多边形边的数目，奇数为内部点（有进有出），否则为外部点

非零环绕规则：环绕数初始为零，从位置 P 作不经过顶点的射线，多边形边从右至左穿过射线，

加 1，多边形边从左至右穿过射线，减 1；非零为内部点（有来无回）；否则为外部点（有来有回）

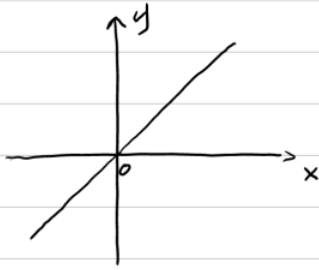


10,构造大矩阵实现,视点 $(0,0,0)$ 从 z 轴负方向,到 y 轴上一点 $(0,5,0)$ 向 $(0,0,0)$ 观察变换。

物体: $T(0,5,0) \cdot R_x(-90^\circ)$

$$R_x(-90^\circ) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T(0,5,0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

11,在三维平面上,构造大矩阵实现,绕 $y=x$ 轴旋转 45 度角的旋转矩阵。



$$R(45^\circ) = R^T(45^\circ) \cdot R(45^\circ) \cdot R(45^\circ)$$

$$R(45^\circ) = \begin{pmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R^{-1}(45^\circ) = \begin{pmatrix} \cos 45^\circ & -\sin 45^\circ & 0 \\ \sin 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\therefore R(45^\circ) = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

12,描述观察变换的坐标变换关系

已知 $w_1=10, w_2=20, w_3=40, w_4=80,$

$v_1=-10, v_2=20, v_3=10, v_4=120,$

窗口中一点 $P(15, 60)$, 求视口中的映射点 P'

窗口到视口的坐标变换

保持视口与窗口中的对象具有同样的相对位置, 必须满足

$$(X_w - W_1)/(W_2 - W_1) = (X_v - V_1)/(V_2 - V_1)$$

$$(Y_w - W_3)/(W_4 - W_3) = (Y_v - V_3)/(V_4 - V_3)$$

窗口到视口的坐标变换

$$X_v = AX_w + B$$

$$Y_v = CY_w + D$$

$$A = (V_2 - V_1)/(W_2 - W_1)$$

$$B = (W_2 * V_1 - W_1 * V_2)/(W_2 - W_1)$$

$$C = (V_4 - V_3)/(W_4 - W_3)$$

$$D = (W_4 * V_3 - W_3 * V_4)/(W_4 - W_3)$$

$$\begin{aligned} (15-10)/(20-10) &= (X_V-10)/(20-10) \\ (60-40)/(80-40) &= (Y_V-10)/(20-10) \\ \therefore X_V &= 5, Y_V = 65 \\ P' &= (5, 65) \end{aligned}$$

13, 已知线段的两个端点

$P_1(-3/2, 1/6), P_2(1/2, 3/2)$

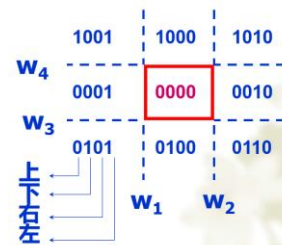
窗口边界 $x=-1, x=1, y=-1, y=1$

用 CS 算法对线段进行剪裁

[\(41 条消息\) Cohen-Sutherland 线段裁剪算法 霍格沃茨研究生的博客-CSDN 博客](#)

CS 算法描述

- 计算直线端点区域编码: c_1 和 c_2 ;
- 判断
 - c_1 和 c_2 (逻辑或) 均为 0000, 保留直线
 - $c_1 \& c_2$ (逻辑与) 不为零, 同在某一边界外, 个区域赋予一个 4 位编码(区域码) $b_3b_2b_1b_0$ 删除该直线
 - $c_1 \& c_2$ 为零, 需要进一步求解交点
- 以左、右、下、上为序, 找出端点区域码中第一位为 1 的位, 将窗口边界方程 $x=w_1$ 或 $x=w_2$ 或 $y=w_3$ 或 $y=w_4$ 代入直线方程, 计算直线与窗口边界的交点, 将交点和另一端点形成新的直线, 重复上述过程, 直至线段保留或删除



University of Electronic Science and Technology of China

Handwritten notes showing the application of the CS algorithm to the line segment P_1P_2 :

Initial codes: $C_{P_1} = 0001$, $C_{P_2} = 1000$.
 Since $C_{P_1} \& C_{P_2} \neq 0$, the line is rejected to the left.
 The intersection point P_1' is found on the line $x = -1$.
 New codes: $C_{P_1'} = 0000$, $C_{P_2} = 1000$.
 Since $C_{P_1'} \& C_{P_2} = 0$, the line is accepted.
 The final clipped segment is $P_1'P_2$.

14, 利用 LB 算法实现上述例子

$$\begin{aligned} \Delta x &= 2, \Delta y = \frac{4}{3} \\ p_1 &= -2, a_1 = -\frac{1}{2}, R_1 = \frac{1}{4} \\ p_2 &= 2, a_2 = \frac{5}{8}, R_2 = \frac{5}{8} \\ p_3 &= -\frac{4}{3}, a_3 = \frac{7}{6}, R_3 = -\frac{7}{8} \\ p_4 &= \frac{4}{3}, a_4 = \frac{5}{6}, R_4 = \frac{5}{8} \end{aligned}$$

$$\begin{aligned} u_1 &= \max\{0, \frac{1}{4}, -\frac{7}{8}\} = \frac{1}{4} \\ u_2 &= \min\{1, \frac{5}{8}, \frac{5}{8}\} = \frac{5}{8} \\ p_1': x &= x_1 + u_1 \Delta x = -\frac{3}{2} + \frac{1}{4} \times 2 = -1 \\ y &= y_1 + u_1 \Delta y = \frac{1}{6} + \frac{1}{4} \times \frac{4}{3} = \frac{1}{2} \\ &\text{即 } (-1, \frac{1}{2}) \\ p_2': x &= x_1 + u_2 \Delta x = -\frac{3}{2} + \frac{5}{8} \times 2 = -\frac{1}{4} \\ y &= y_1 + u_2 \Delta y = \frac{1}{6} + \frac{5}{8} \times \frac{4}{3} = 1 \\ &\text{即 } (-\frac{1}{4}, 1) \end{aligned}$$

15, 简述平行投影，透视投影概念

1) 平行投影

将物体表面上的点沿平行线投影到显示平台上

三维场景中的平行线在投影到二维显示平面中后仍然是平行线



7.5.1 平行投影分类

- 正投影：投影向量垂直于观察平面
- 斜投影：不垂直

分类：

正投影：

● 正投影

- 正三面投影：三视图(正视图、俯视图、侧视图)，投影平面法向量与三个坐标轴一个平行
- 轴测正投影：显示物体多个侧面的正投影
- 等轴测投影：与每个坐标轴的交点离原点距离相等

轴测投影

语音

编辑

讨论

上传视频

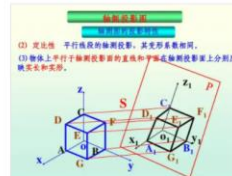
★ 收藏 42 2

本词条缺少信息栏，补充相关内容使词条更完整，还能快速升级，赶紧来编辑吧！

轴测投影是平行投影的一种。

将物体放在三个坐标面和投影线都不平行的位置，使它的三个坐标面在一个投影上都能看到，从而具有立体感，称为“轴测投影”。这样绘出的图形，称为“轴测图”。轴测图在工程技术及其他科学中常有应用。

在轴测图中，物体上与任一坐标轴平行的长度均可按一定的比率来量度。三轴向的比率都相同时称为“等测投影”，其中两轴向比率相同时称为“二测投影”，三轴向比率均不同时称为“三测投影”。轴测投影中投射线与投影面垂直的称为“正轴测投影”，倾斜的称为“斜轴测投影”



面之投影

面之投影

斜投影：

● 斜投影

- 斜等测投影： α 等于45度，生成的视图
- 斜二测投影：投影角满足 $\tan\alpha=2$ ，生成的视图

(2) 透视投影

沿会聚路径将点投影到显示平面上

远小近大，平行线投影后成了会聚线

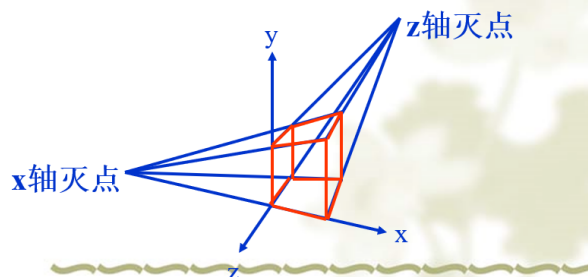
透视投影：物体位置沿收敛于某点的线变换到观察平面

● 灭点

一组平行线投影后收敛于一点称之为灭点。

● 主灭点

物体中平行于某一坐标轴的平行线的灭点



16,投影的三要素

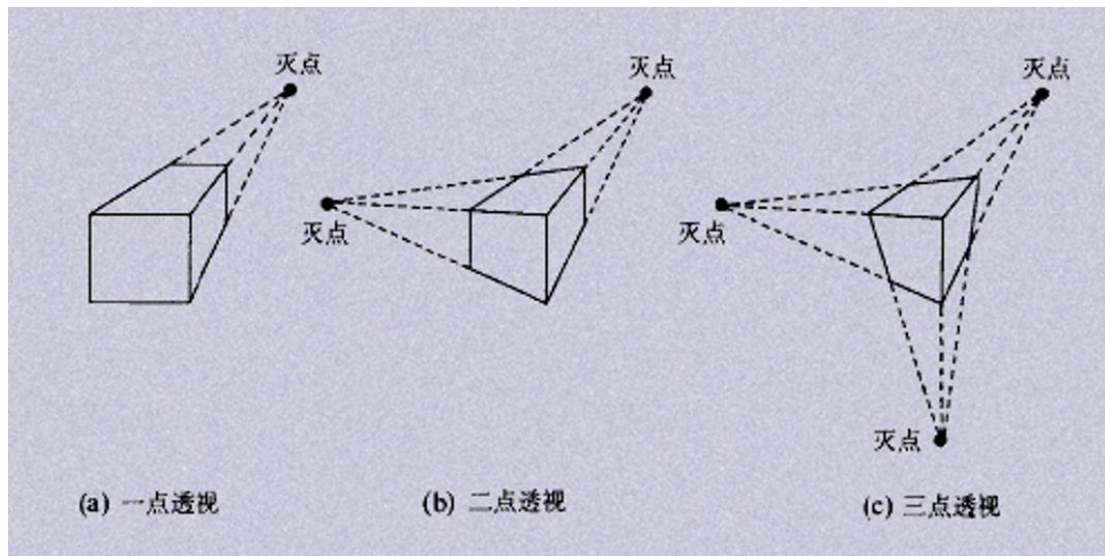
投影中心（平行投影的投影中心在无穷远处）、投影平面和投影射线

17,主灭点，一点透视，两点透视，三点透视

灭点：一组平行线投影后收敛于一点称之为灭点

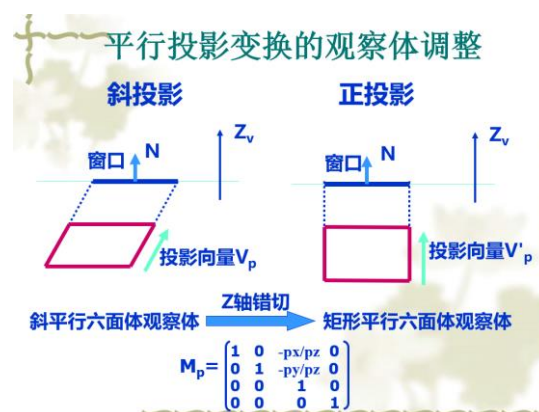
主灭点：物体中平行于某一坐标轴的平行线的灭点

透视投影按照主灭点数目分为一点透视、二点透视、三点透视



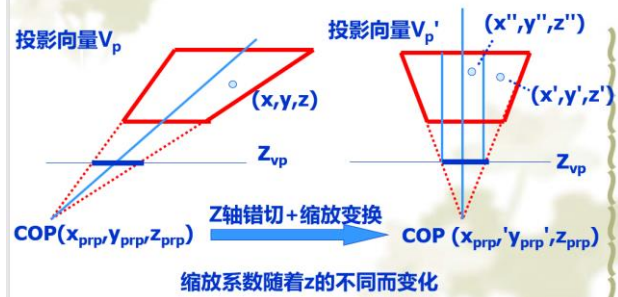
18,观察体调整

调整的意义：使得计算、处理方便快捷



正平行投影：观察体为正平行六面体，不用调整

透视投影变换的观察体调整



斜平行投影和透视投影的观察体均需要调整

19,概念，多边形网格模型表示中，基本数据表形式，顶点表，边表，面表

多边形表数据表分为两组进行组织：

几何表：顶点坐标和用来标识多边形表面空间方向的参数

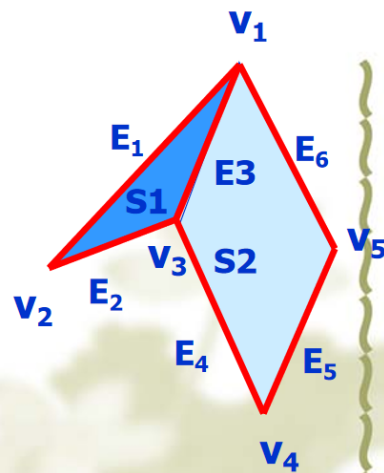
属性表：指明物体透明度及表面反射度的参数和纹理特征

多边形表面

顶点表	
序号	点坐标
1	x_1, y_1, z_1
2	x_2, y_2, z_2
3	x_3, y_3, z_3
4	x_4, y_4, z_4
5	x_5, y_5, z_5

边表	
序号	顶点号
1	v_1, v_2
2	v_2, v_3
3	v_3, v_1
4	v_3, v_4
5	v_4, v_5
6	v_5, v_1

多边形面表	
序号	边序号
1	E_1, E_2, E_3
2	E_3, E_4, E_5, E_6



20,概念, 用函数描述的二次曲面模型, 如球体表面, 如何进行绘制

给定二次曲面函数, 图形包将指定曲线方程投影到显示平面上, 且沿着投影函数路径绘制像素位置, 最终绘制成二次曲面。

13.2 曲线和曲面

- 给定函数方式

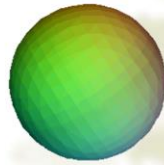
球面

$$x^2+y^2+z^2=r^2 \text{ 或者}$$

$$x=r\cos\Phi\cos\theta \quad -\pi/2 \leq \Phi \leq \pi/2$$

$$y=r\cos\Phi\sin\theta \quad -\pi \leq \theta \leq \pi$$

$$z=r\sin\theta$$



21,概念, 样条曲线, 样条曲面

样条曲线在计算机图形学中的含义: 由多项式曲线段连接而成的曲线、在每段的边界处满足特定的连续性条件

样条曲面: 使用两组正交样条曲线进行描述

22,概念, 样条曲线的两种类型

插值样条曲线: 选取的多项式使得曲线通过每个控制点

逼近样条曲线: 选取的多项式不一定使曲线通过每个控制点

23,概念, 凸壳的概念

包含一组控制点的凸多边形边界

24,概念, 分段连续中连续的定义

参数连续性条件和几何连续性条件

参数连续性条件:

两个相邻曲线段在相交处的参数导数相等

零阶连续(C0 连续): 简单地表示曲线连接, 两个曲线必在相交点处有相同的坐标

一阶连续(C1 连续): 说明代表两个相邻曲线的方程在相交点处有相同的一阶导数 (切线)

二阶连续(C2 连续): 两个曲线段在交点处有相同的一阶和二阶导数, 交点处的切向量变化率相等

几何连续性条件:

两个相邻曲线段在相交处的参数导数成比例

零阶连续 (G0 连续): 与 0 阶参数连续性相同, 即两个曲线必在公共点处有相同的坐标

一阶连续 (G1 连续): 表示一阶导数在两个相邻曲线的交点处成比例

二阶连续 (G2 连续): 表示两个曲线段在相交处的一阶和二阶导数均成比例

25, 简述 Bezier 的几个特点

Bezier 曲线总是通过第一个和最后一个控制点

Bezier 曲线在第一个控制点 P_0 处与直线 P_0P_1 相切, 在最后一个控制点 P_n 处与直线 $P_{n-1}P_n$ 相切。

Bezier 曲线总是落在控制点的凸壳内, 保证了曲线沿控制点的平稳前进

26, threejs 的浏览器中如何调试的方法, 文件相对路径关系 (src=“/images/ddd.jpg”与 “./images/ddd.jpg” 与 “./images/ddd.jpg” 与 “images/ddd.jpg” 的区别)

- 调试的方法: 选择更多工具, 选择开发者工具或者 F12 进入
- ./ 表示当前文件所在目录: “./images/ddd. jpg” 与 “images/ddd. jpg”
(./可省略)
- ../ 表示当前文件所在目录的上级目录: “../images/ddd. jpg”
- / 表示根路径: “/images/ddd. jpg” (绝对路径)

27, VBO, PBO, FBO 的概念

VBO (vertex buffer object) 是 GPU 上存储顶点数据的高速缓存。

PBO (pixel buffer object) 是 GPU 上存储像素数据的高速缓存, 类似于 VBO 存储顶点数据。

FBO (frame buffer object) 是 GPU 上存储渲染后的帧缓存。

28, 如何创建一个 vbo, 并基于 vbo 进行绘制 (写出相应的伪代码过程) (不考)

29, 深度缓存的意义, 基于深度检测的方法

深度缓存的意义: 把一个距离观察平面 (近裁剪面) 的深度值 (或距离) 与窗口中的

每个像素相关联。对投影平面上每个像素所对应的表面深度进行比较。深度缓冲区与帧缓冲区相对应，用于记录上面每个像素的深度值，通过深度缓冲区，我们可以进行深度测试，从而确定像素的遮挡关系，保证渲染正确。

基于深度检测的方法：

深度缓冲器算法

A 缓冲器算法

深度排序算法

可见面判别算法

30,描述固定管线中的光照模型

Lambert 模型(漫反射):

我们可以在建立表面的漫反射模型时假设入射光在各个方向以相同强度发散而与观察位置无关。这样的表面称为理想漫反射体 (ideal diffuse reflector)。因为从表面上任一点反射出的辐射光能量用朗伯余弦定律 (Lambert's cosine law) 计算，因此也称为朗伯反射体 (Lambertian reflector)。该定律表明：在与对象表面法向量夹角为 ϕ_N 的方向上，每个面积为 dA 的平面单位所发散的光线与 $\cos\phi_N$ 成正比 (参见图 10.10)。该方向的光强度可用单位时间辐射能总量除以表面积在辐射方向的投影来计算：

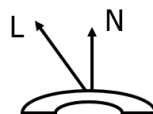
$$\begin{aligned} \text{强度} &= \frac{\text{单位时间辐射能}}{\text{投影面积}} \\ &\propto \frac{\cos\phi_N}{dA\cos\phi_N} \\ &= \text{常数} \end{aligned} \quad (10.6)$$

这样，对于朗伯反射，光强度在所有观察方向都相同。

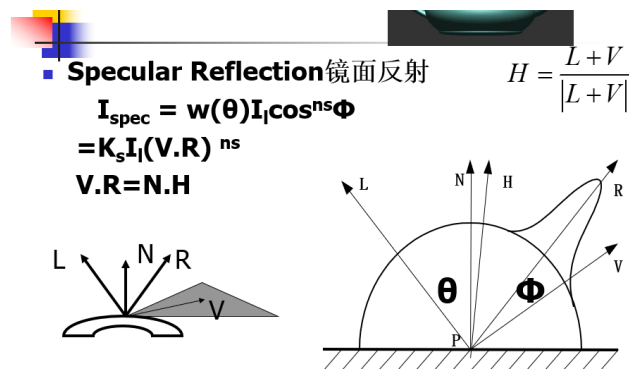
$$I_{\text{Ldiff}} = K_d I_l \cos\theta = K_d I_l (N \cdot L)$$

K_d – diffuse-reflection coefficient

I_l -- intensity of point light source



镜面反射模型：



phenomenon, called specular reflection. It is the result of total, or near total, reflection of the incident light in a concentrated region around the normal. $w(\theta)$ 镜面反射系数
 Φ 为观察方向 V 与镜面反射方向 R 的夹角
 ns 代表镜面反射参数, 由观察物体的表面材质决定, 光滑表面的 ns 值大 (范围小), 而粗糙表面的 ns 值小 (范围大), 影响镜面反射的角度范围。

Phong 模型

Phong Model : 由物体表面上一点 P 反射到视点的光强 I 为环境光的反射光强、理想漫反射光强、和镜面反射光的总和。

$$I = I_{\text{ambdiff}} + I_{\text{ldiff}} + I_{\text{spec}}$$

Ambient Light 环境光: 是场景各个表面的反射光生成的光照效果, 包括

Ambient Light 环境光

$$I_{\text{ambdiff}} = K_a I_a$$

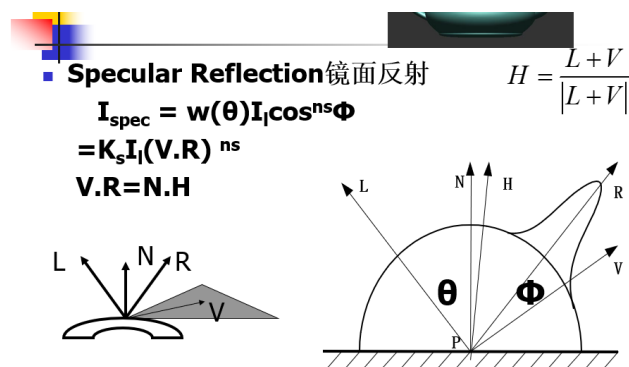
K_a -- ambient-reflection coefficient

I_a -- ambient light intensity

了物体的漫反射和镜面反射。
 反射系数、 I 为环境光强度也就是亮度) (k 为环境

漫反射模型描述见上

镜面反射模型:



5, 利用 `threejs` 设计实现一个简单游戏: 用鼠标控制旋转方向, 键盘控制向前向后移动, 空格键发射子弹, 射线检测碰撞结果

6, 利用 `threejs` 设计实现一个函数曲线的绘制

7, 利用 `threejs` 设计实现一个曲面的绘制