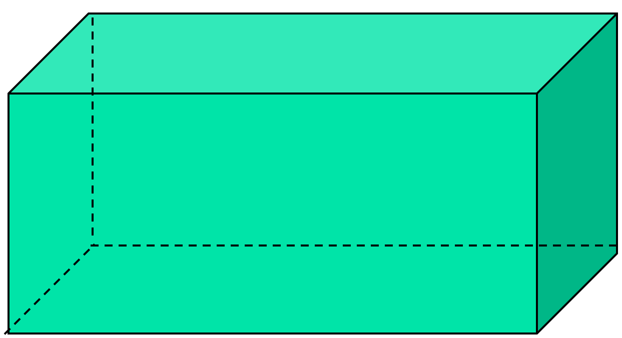
6.2三维图形投影变换技术

南京农业大学

谢忠红

- 三维图形最重要的三个课题。
- (1) 如何定义三维形状？——几何造型技术
- (2)如何作为二维图像在图像显示器等输出装置上表示出来?——三维投影技术
- (3)为了增强三维图形在二维显示器上显示的立体感，还有必要对其进行消隐计算。



- 前面讲的内容解决了如何在计算机中定义一个立体形体，下面我们来解决第二个问题：

- 如何将三维形体作为二维图像
- 在图像显示器等输出装置上
- 表示出来？

任务：将三维空间的点变换成二维空间的点。

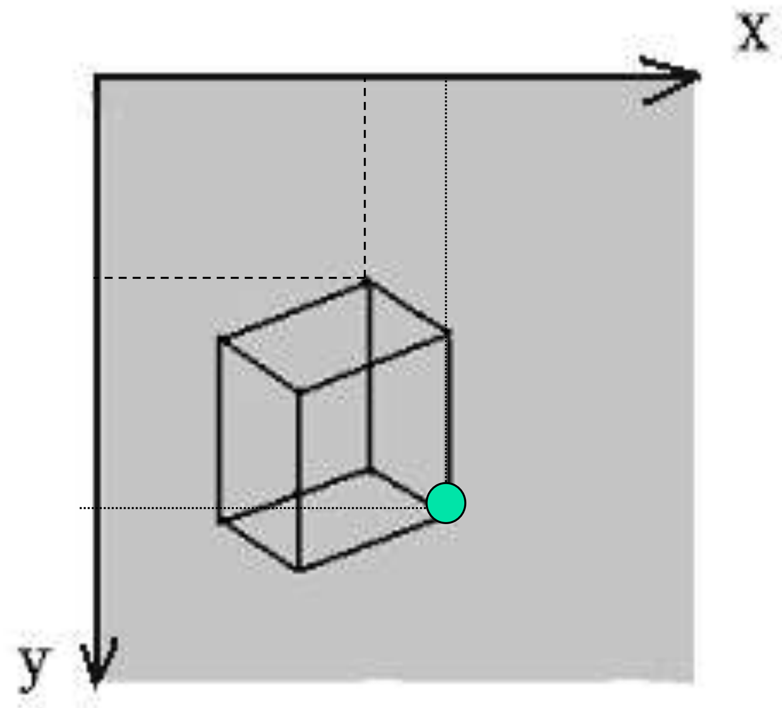
- 如何将三维空间的点
- 变换成二维空间的点?

投影

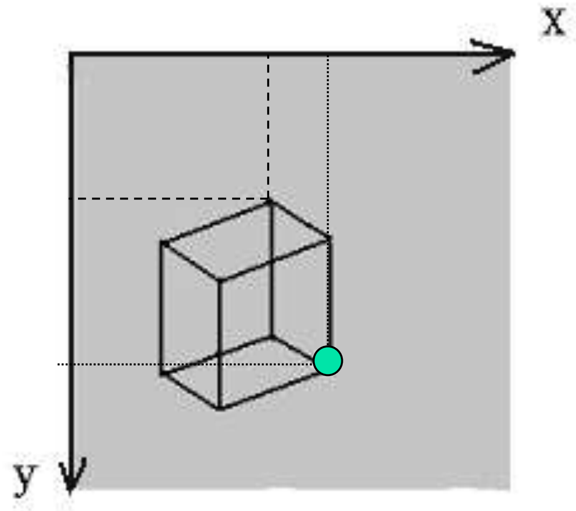
5.2三维形体在二维平面上的投影

DEF:图形输出的设备都是二维的，
要使三维形体输出显示在二维设备上
就需要将三维坐标表示的几何形体
变换成 二维坐标表示的图形，这种
变换称为投影变换 .

二维显示器显示的立体图形



- **DEF:**投影就是将三维坐标系下的图形坐标转换为
- 二维坐标系下的坐标：
- 即将**(x,y,z)------(x,y)** 或 **(y,z)**或 **(x,z)**



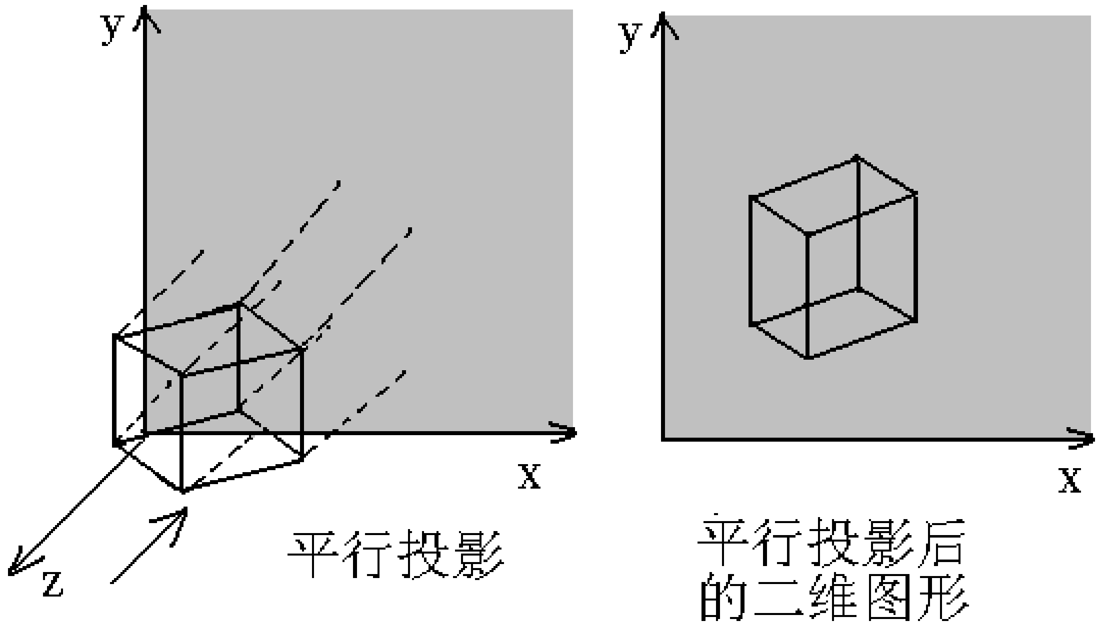
- 如何实现投影呢?

投影的实现

平行投影

透视投影

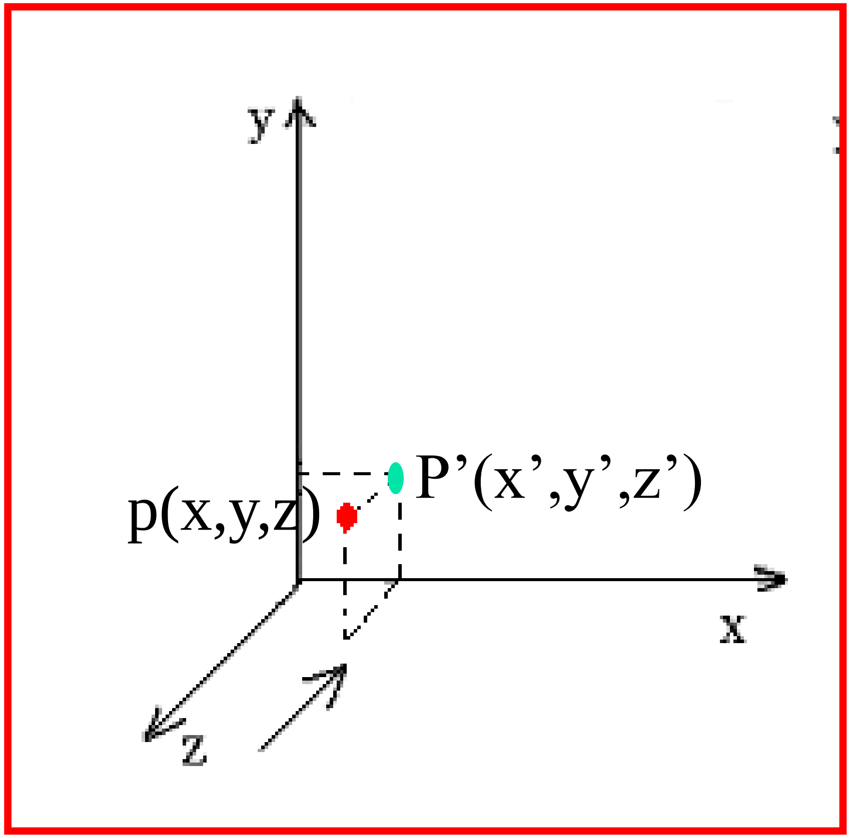
平行投影(投影方向：z轴，投影面o-xy面)



投影方法： 从被投影的形体的各个点，(沿投影方向)向投影平面画平行线，这些平行线和投影面的交点形成投影像。

- 设平行投影方向为**Z轴**，投影面为**o-xy**面，则空间中任意一点**P(x,y,z)**投影到**o-xy**面上获得点**P'(x',y',z')**的关系很显然是

$$\begin{cases} x' = x \\ y' = y \\ z' = 0 \end{cases}$$



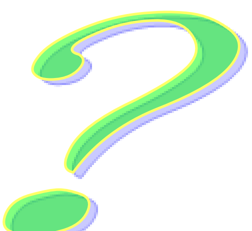
$$\begin{cases} x' = x \\ y' = y \\ z' = 0 \end{cases}$$

用矩阵表示：

$$(x\ y\ z\ 1) * \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (x'\ y'\ z'\ 1)$$

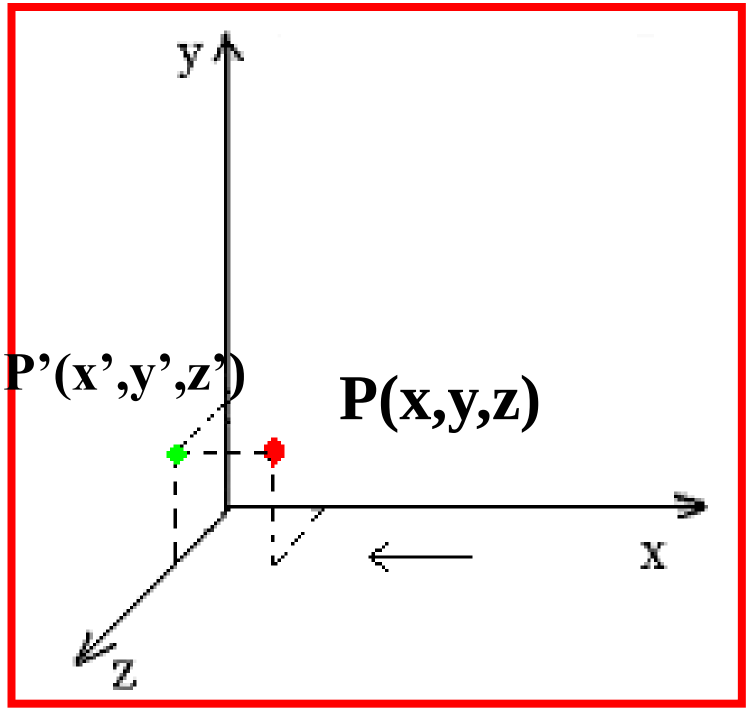
三维坐标　　变换矩阵　　投影后的二维坐标

- 投影方向：x轴，投影面yz面
- 投影方向：y轴，投影面xz面
- 投影矩阵为多少？



•平行投影方向为x轴，投影面为o-yz面，则空间中任意一点P(x,y,z)投影到o-yz面上获得点P’(x’,y’,z’)的关系是

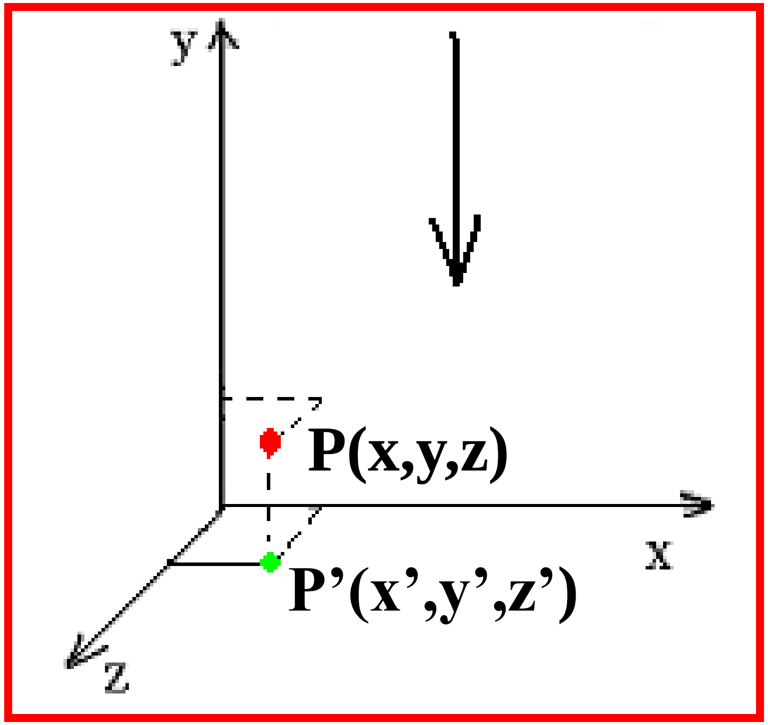
$$\begin{cases} x'=0 \\ y'=y \\ z'=z \end{cases}$$



$$(x\ y\ z\ 1) * \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (x'y'z'1)$$

平行投影方向为Y轴，投影面为o-xz面，

则空间中任意一点P(x,y,z)投影到o-xz面上获得点P’(x’,y’,z’)的关系是

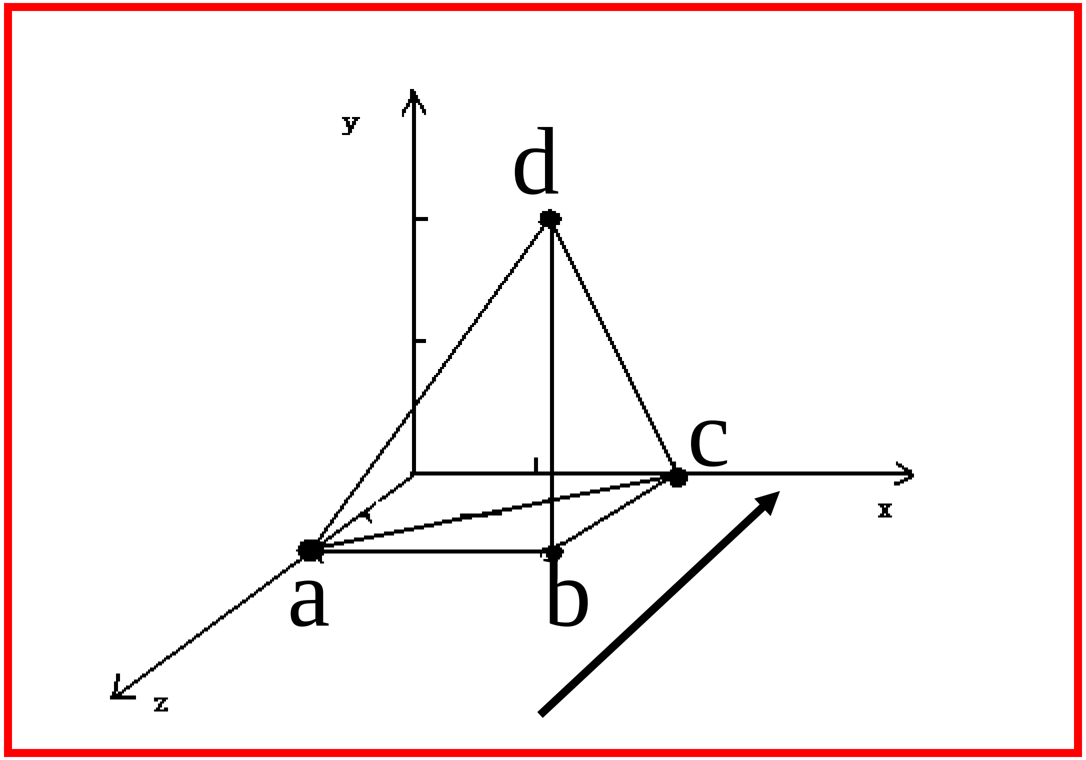


$$\begin{cases} x' = x \\ y' = 0 \\ z' = z \end{cases}$$

$$(x\ y\ z\ 1) * \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (x'\ y'\ z'\ 1)$$

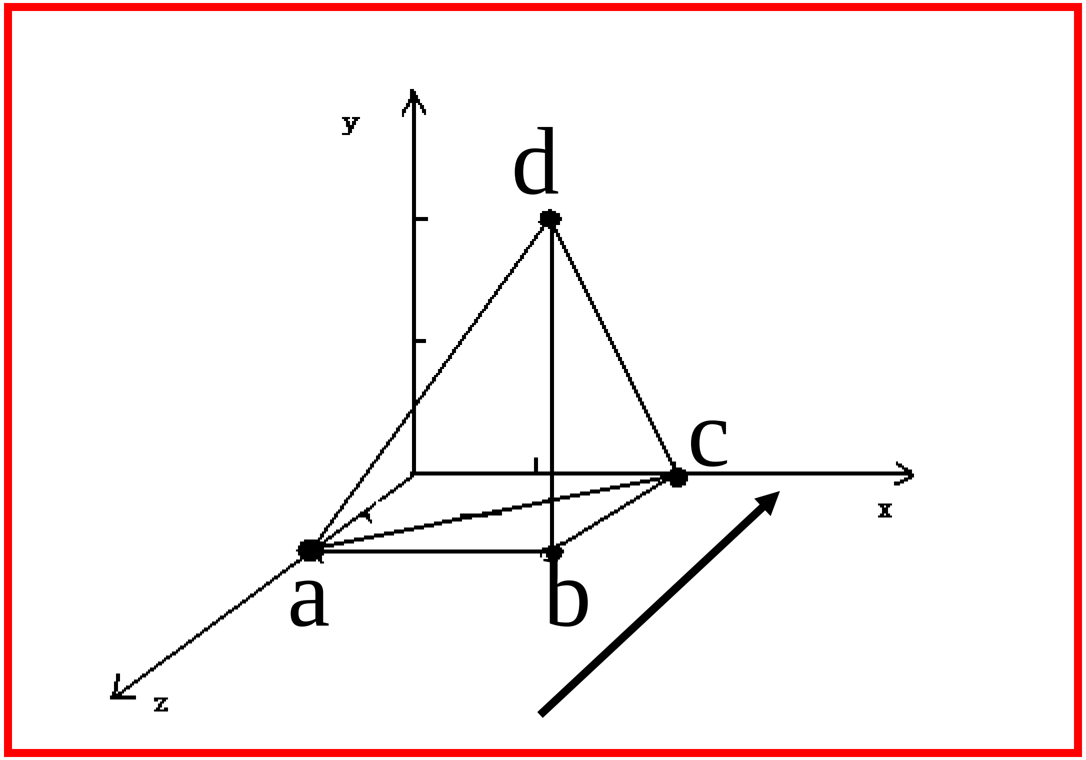
- **例1：** 已知三棱锥各顶点坐标为**(0,0,20), (20,0,20), (20,0,0), (10,20,10)**试从**Z**轴方向向平面**O-XY**作平行投影，求出各顶点投影后的坐标并绘制平行投影图。

$$\begin{pmatrix} 0 & 0 & 20 & 1 \\ 20 & 0 & 20 & 1 \\ 20 & 0 & 0 & 1 \\ 10 & 20 & 10 & 1 \end{pmatrix}$$



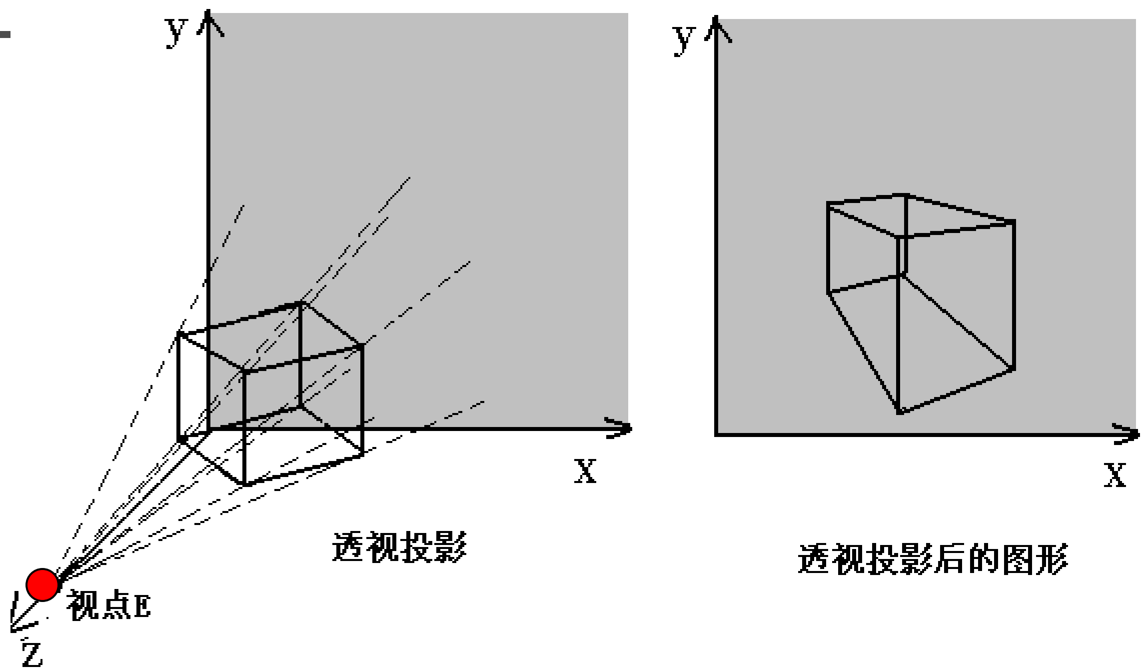
•编程题：

•已知三棱锥各顶点坐标是(0,0,20) (20,0,20) (20,0,0) (10,20,10)试编程绘制从Z轴向平面O-XY作平行投影图.



- 思路点拨：
- (1)建立相应的数据结构表示该三维形体
- (2)根据平行投影的公式将三维坐标—二维坐标
- （注意这里的坐标均为世界坐标系）
- (3)将二维世界坐标转换为屏幕坐标，再根据边表画出对应的边。

透视投影(投影视点E-观察者的眼睛,投影面xy面)



投影方法：从视点E经过形体的各个点，向投影平面画射线，这些射线和投影面o-xy的交点形成投影像（也就是具有真实立体感的二维图形）。

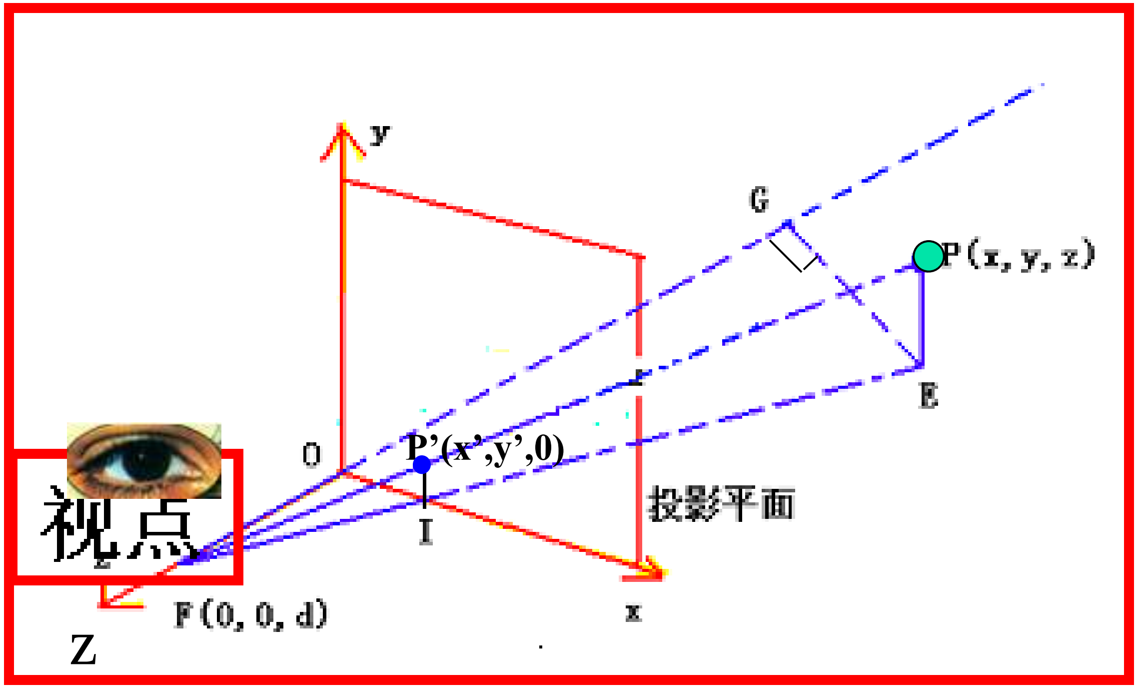
如何得到一点透视投影的坐标呢？

■ 一点透视投影的变换矩阵及特征

假设视点F在Z轴上其坐标为(0,0,d)

向平面O-XY作透视投影

P是三维形体上的任意一点



由左图可知：

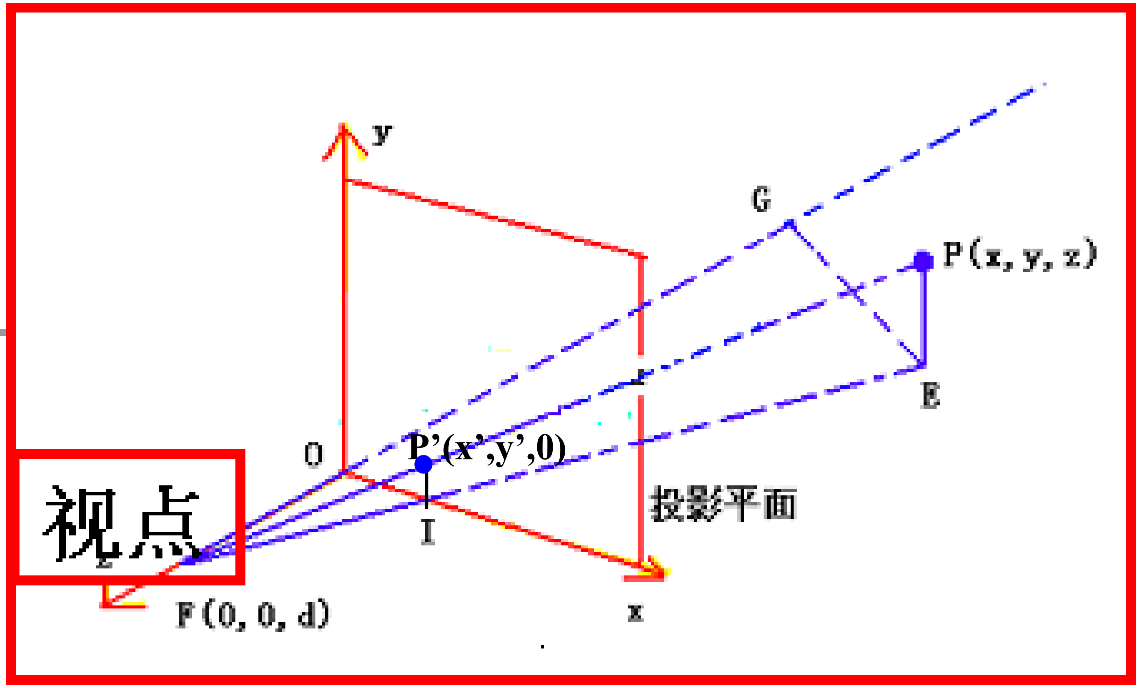
ΔPEF与 ΔP'IF相似

ΔGEF与ΔOIF相似

由左图可知：

ΔPEF与 ΔP’IF相似

ΔOIF与ΔGEF相似



$$\begin{cases} x/x' = (d-z)/d \\ y/y' = (d-z)/d \end{cases}$$

$$\begin{cases} x' = x*d/(d-z) \\ y' = y*d/(d-z) \\ z' = 0 \end{cases}$$

写成矩阵形式(计算验证)

$$(x\ y\ z\ 1) * \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/d \\ 0 & 0 & 0 & 1 \end{pmatrix} = (x'\ y'\ z'\ 1)$$

其中：$T_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/d \\ 0 & 0 & 0 & 1 \end{pmatrix}$

T_1: 为透视点(0,0,d),透视平面 O-XY平面的透视投影变换矩阵

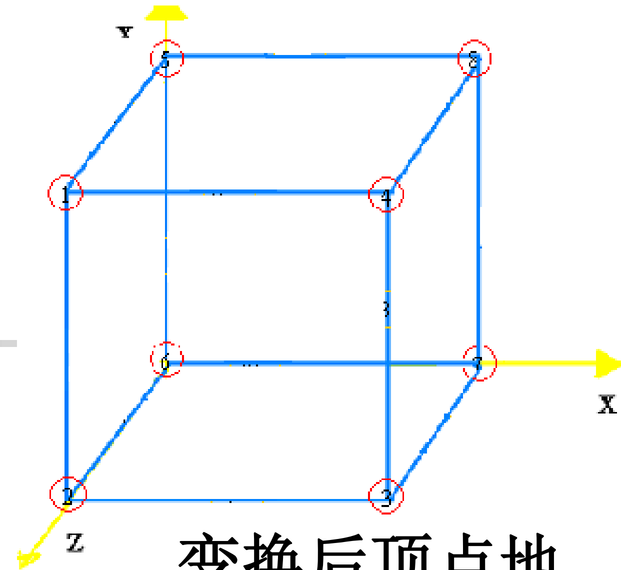
如果视点在y轴坐标（0,d,0）向平面O-XZ作透视投影

透视投影变换矩阵为: $T_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

同理，如果视点在x轴坐标（d,0,0）向平面O-YZ作透视投影

透视投影变换矩阵为: $T_3=\begin{pmatrix} 0 & 0 & 0 & -1/d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

- **例1：** 已知某立方体棱长均为100，试从（0，0，400）处向平面O-XY作透视投影，求出个顶点投影后的坐标并绘制投影图。



立体图形顶点坐标矩阵

透视变换矩阵

变换后顶点地二维坐标矩阵

$$
\begin{pmatrix}
0 & 100 & 100 & 1 \\
0 & 0 & 100 & 1 \\
100 & 0 & 100 & 1 \\
100 & 100 & 100 & 1 \\
0 & 100 & 0 & 1 \\
0 & 0 & 0 & 1 \\
100 & 0 & 0 & 1 \\
100 & 100 & 0 & 1
\end{pmatrix}
*
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & -1/400 \\
0 & 0 & 0 & 1
\end{pmatrix}
=
\begin{pmatrix}
\text{计} \\
\text{算} \\
\text{出} \\
\text{结} \\
\text{果}
\end{pmatrix}
$$

投影的特征：

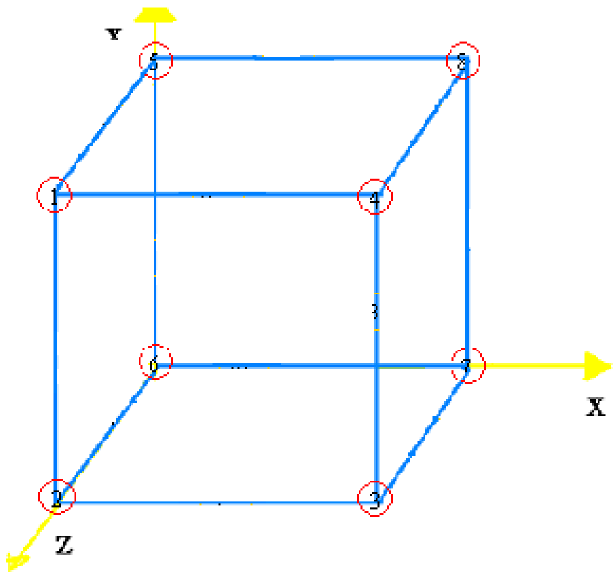
$$\begin{cases} y/y' = (d-z)/d \\ x/x' = (d-z)/d \end{cases} \qquad \begin{cases} x' = x*d/(d-z) \\ y' = y*d/(d-z) \\ z' = 0 \end{cases}$$

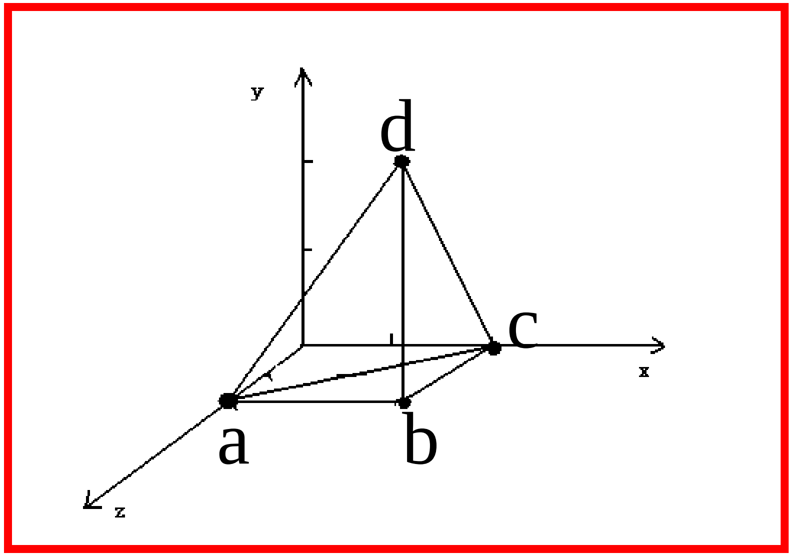
由投影公式可知：

|Z|越大，x’,y’就越小。

即：远处的物体比近处物体的投影要小。

- # 程序设计题：
- （1）按照前面讲的思路，编写程序显示例1所示立方体从视点（0,0,d）处向0-XY平面作的透视图。
- 要求（视点位置d可以任意输入）
- （2）设三棱锥各顶点坐标是(0,0,20)(20,0,20)(20,0,0)(10,20,10)试编程绘制从视点(0,0，120)对平面O-XY的一点透视图





- **思路点拨：**
- **(1)**建立相应的数据结构表示该三维形体
- **(2)**根据一点透视投影的公式将三维坐标二维坐标
- （注意这里的坐标均为世界坐标系）
- **(3)**将二维世界坐标转换为屏幕坐标，再根据边表画出对应的边。