三维图形

生成方法及程序设计

陆玲 汤彬◎著

HEUP哈尔滨工程大学出版社
Harbin Engineering University Press

三维图形生成方法及程序设计

陆玲 汤彬 著

哈尔滨工程大学出版社

内容简介

本书主要介绍一些常用的三维图形的生成方法，以及相应方法的 VC++6.0 程序设计。本书的主要内容包括图形的几何变换及投影变换，规则曲面及非规则曲面的生成方法，真实感图形生成的消隐、光照、纹理、透明与阴影处理，分形造型及在树、山脉中的应用实例，变形造形及在植物果实、花朵等中的应用实例。

本书的特点是所有方法都附有相应 VC++ 源程序代码，且含有 VC++6.0 的编程过程。本书可作为高等学校计算机科学与技术专业、电子信息类专业的本科生和研究生教材，也可作为从事计算机图形学工作的工程技术人员的参考书。

图书在版编目（CIP）数据

三维图形生成方法及程序设计/陆玲，汤彬著．—哈尔滨：哈尔滨工程大学出版社，2011.12
ISBN 978-7-5661-0293-5

Ⅰ．①三…　Ⅱ．①陆…②汤…　Ⅲ．①C 语言－程序设计②计算机图形学　Ⅳ．①TP312②TP391.41

中国版本图书馆 CIP 数据核字（2011）第 267138 号

出版发行　哈尔滨工程大学出版社
社　　址　哈尔滨市南岗区东大直街 124 号
邮政编码　150001
发行电话　0451-82519328
传　　真　0451-82519699
经　　销　新华书店
印　　刷　哈尔滨市石桥印务有限公司
开　　本　787mm×960mm　1/16
印　　张　11.75
字　　数　245 千字
版　　次　2011 年 12 月第 1 版
印　　次　2011 年 12 月第 1 次印刷
定　　价　23.00 元
http://press.hrbeu.edu.cn
E-mail:heupress@hrbeu.edu.cn

前　　言

随着科学技术的不断发展以及人们对计算机图形的不断需求，三维图形及其三维可视化技术正在发挥着越来越重要的作用。计算机图形学已经在各个行业里得到了广泛的重视和应用，并正在向各个学科不断渗透。在信息社会中，计算机图形学无论是在理论上还是在实践上都存在着巨大的发展潜力。

本书是作者在多年研究的基础上，总结过去的一些研究成果，结合教学经验和已出版的相关教材，经过整理与汇总编写而成的。

本书首先介绍了三维图形生成的最基本的几何变换与投影变换，为后续三维图形生成作准备；然后对常用曲面的曲线生成方法进行介绍，主要包括规则曲面的球面及旋转面、非规则曲面的双线性曲面、单线性曲面、Bezier 曲面和 B 样条曲面；再对真实感图形生成中的消隐、光照、纹理、透明与阴影处理进行介绍；最后描述了分形造型和变形造形以及它们的应用实例，例如对山、水、树、植物果实和植物花朵等三维可视化的造形。所有方法都附有相关的VC++ 程序代码，特别适用于初学计算机图形学的读者学习使用。

本书力求体现以下特点：

（1）内容通俗易懂　对读者的要求起点低，只要求读者有 C 语言程序设计的基础。读者通过本书，可掌握三维图形的常用生成方法；

（2）重点难点突出　侧重介绍三维图形生成基本方法思路及实现算法；

（3）程序设计完整　本书涉及的三维图形生成的方法都配有相应的 VC ++ 6.00 程序设计过程及源程序代码，使读者能够掌握图形生成的编程方法。同时，参考或引用了一些国内外专家学者的论著。

本书在编写和出版过程中，得到了很多朋友的支持与帮助，在此一并表示感谢。

由于作者水平有限，书中难免有不妥和错误之处，恳请读者批评指正。

著　者

2011 年 11 月

目　　录

第 1 章　图 形 变 换

图形变换一般是指对图形的几何信息经过几何变换后产生新的图形。图形变换既可以看作是坐标系不动而图形变动，变动后的图形在坐标系中的坐标值发生变化，也可以看作图形不动而坐标系变动，变动后该图形在新的坐标系下具有新的坐标值，这两种情况本质上是一样的。本书所讨论的几何变换属于前一种情况。

1.1　三维图形的几何变换

对于任何图形的变换，通常都是以点变换作为基础。因此，对图形的几何变换实际上就是对点的几何变换。采用齐次坐标表示，用变换矩阵实现对点的变换。假设三维图形中变换前的一点坐标为$[x\ y\ z\ 1]$，变换后为$[x'\ y'\ z'\ 1]$。

1. 变换矩阵

三维图形的几何变换矩阵可用 T_{3D} 表示，其表示式如下：

$$T_{3D}=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

从变换功能上 T_{3D} 可分为 4 个子矩阵，其中$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$产生比例、旋转、错切等几何变换；$[a_{41}\ a_{42}\ a_{43}]$产生平移变换；$\begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}$产生投影变换；$[a_{44}]$产生整体比例变换。

2. 平移变换

设在三个分量上的平移量为(T_x, T_y, T_z)，则平移变换如式(1－1)。

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} = [x + T_x \quad y + T_y \quad z + T_z \quad 1] \qquad (1-1)$$

在程序设计中,平移变换就是在各分量上增加一个平移量。

3. 比例变换

设在三个分量上的比例系数为(s_x, s_y, s_z),相对于原点的比例变换如式(1-2)。

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \cdot \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [s_x x \quad s_y y \quad s_z z \quad 1] \qquad (1-2)$$

在程序设计中,相对于坐标原点的比例变换就是在各分量上乘以一个比例系数。

若比例变换的参考点为(x_f, y_f, z_f),相对于参考点$F(x_f, y_f, z_f)$作比例变换的过程分为以下3步:

①把参考点$F(x_f, y_f, z_f)$平移至原点;

②相对原点作比例变换;

③将参考点F再平移回原点。

其变换矩阵为

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

$$= \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ (1-s_x) \cdot x_f & (1-s_y) \cdot y_f & (1-s_z) \cdot z_f & 1 \end{bmatrix}$$

4. 绕坐标轴的旋转变换

在右手坐标系下相对于各坐标轴旋转变换关系分别如下:

①绕X轴旋转

$$[x'\ y'\ z'\ 1]=[x\ y\ z\ 1]\cdot\begin{bmatrix}1&0&0&0\\0&\cos\theta&\sin\theta&0\\0&-\sin\theta&\cos\theta&0\\0&0&0&1\end{bmatrix}\tag{1-3}$$

②绕 Y 轴旋转

$$[x'\ y'\ z'\ 1]=[x\ y\ z\ 1]\cdot\begin{bmatrix}\cos\theta&0&-\sin\theta&0\\0&1&0&0\\\sin\theta&0&\cos\theta&0\\0&0&0&1\end{bmatrix}\tag{1-4}$$

③绕 Z 轴旋转

$$[x'\ y'\ z'\ 1]=[x\ y\ z\ 1]\cdot\begin{bmatrix}\cos\theta&\sin\theta&0&0\\-\sin\theta&\cos\theta&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\tag{1-5}$$

在三维图形生成过程中，经常使用旋转变换，这里先定义绕三个坐标轴旋转变换的 C 函数，以便后续程序调用。

```
//绕 X 轴旋转函数 revolve_x
//输入参数:cx----旋转角度
//          (xx,yy,zz)----旋转前点坐标
//输出参数:(x,y,z)----旋转后点坐标
void revolve_x(float cx, float xx, float yy, float zz, float &x, float &y, float &z )
{   x = xx ;
    y = yy * cos(cx) - zz * sin(cx) ;
    z = yy * sin(cx) + zz * cos(cx);
}
//绕 Y 轴旋转函数 revolve_y
//输入参数:cy----旋转角度
//          (xx,yy,zz)----旋转前点坐标
//输出参数:(x,y,z)----旋转后点坐标
void revolve_y(float cy, float xx, float yy, float zz, float &x, float &y, float &z )
{   x = xx * cos(cy) + zz * sin(cy) ;
    y = yy ;
    z = - xx * sin(cy) + zz * cos(cy);
```

```
}
//绕 Z 轴旋转函数 revolve_z
//输入参数:cz----旋转角度
//            (xx,yy,zz)----旋转前点坐标
//输出参数:(x,y,z)----旋转后点坐标
void revolve_z(float cz, float xx, float yy, float zz, float &x, float &y, float &z)
{    x = xx * cos(cz)  - yy * sin(cz)  ;
     y = xx * sin(cz) + yy * cos(cz)  ;
     z = zz;
}
```

5. 绕任意轴的旋转变换

设旋转轴 AB 由空间两点 A 与 B 定义,空间一点 P 绕 AB 轴旋转 θ 角到 P',如图 1－1 所示。即要使

$$[x'_p\ y'_p\ z'_p\ 1]=[x_p\ y_p\ z_p\ 1]\cdot R_{AB} \qquad (1-6)$$

其中,R_{AB}为待求的变换矩阵。

求 R_{AB}的基本思想是:将 A 平移到坐标原点,并使 AB 分别绕 X 轴、Y 轴旋转适当角度与 Z 轴重合,再绕 Z 轴转 θ 角,最后再作上述变换的逆变换。

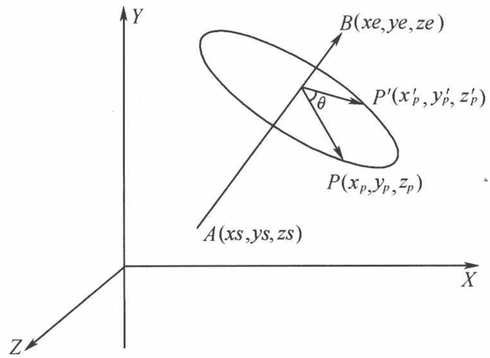

图 1－1　P 点绕 AB 旋转

①使 A 点坐标平移到原点 O,原来的 AB 变为 OB',如图 1－2(a)所示。

②让 OB'绕 X 轴旋转 α 角,如图 1－2(b)所示,经旋转 α 角后,OB''就在 $y=0$ 平面上。

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

③再让 OB''绕 Y 轴旋转 β 角与 Z 轴重合,如图 1－2(c)所示。

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

④经以上 3 步变换后,P 绕 AB 旋转变为 P 绕 X 轴转 θ 角了,如图 1－2(d)所示。

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

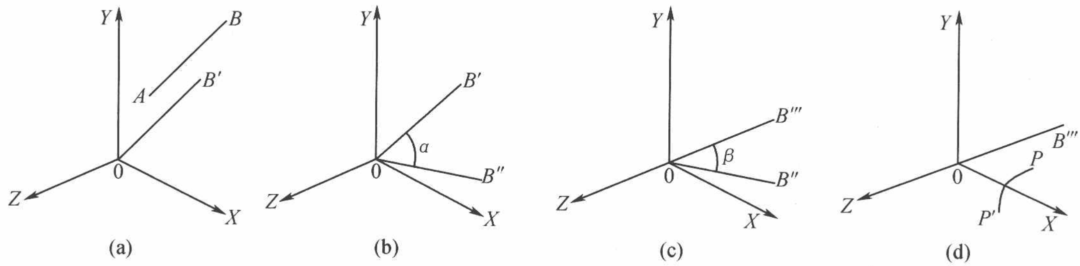

图 1－2　*AB* 经变换与 *Z* 轴重合

⑤求 $\boldsymbol{R}_y,\boldsymbol{R}_x,\boldsymbol{T}_A$ 的逆变换。

$$
\boldsymbol{R}_y^{-1}=\begin{bmatrix}\cos\beta & 0 & \sin\beta & 0\\ 0 & 1 & 0 & 0\\ -\sin\beta & 0 & \cos\beta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}
$$

$$
\boldsymbol{R}_x^{-1}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\alpha & -\sin\alpha & 0\\ 0 & \sin\alpha & \cos\alpha & 0\\ 0 & 0 & 0 & 1\end{bmatrix}
$$

$$
\boldsymbol{T}_A^{-1}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ x_a & y_a & z_a & 1\end{bmatrix}
$$

所以 $\boldsymbol{R}_{AB}=\boldsymbol{T}_A\boldsymbol{R}_x\boldsymbol{R}_y\boldsymbol{R}_x\boldsymbol{R}_y^{-1}\boldsymbol{R}_x^{-1}\boldsymbol{T}_A^{-1}$。

从以上过程可以看出，要实现绕 *AB* 旋转，还需求出相应的旋转角度 α 与 β。具体计算过程如下：

①平移

$$
x=x_p-xs,\ y=y_p-ys,\ z=z_p-zs
$$

②绕 X 轴转 α 角

$$
t=\left|\alpha\tan((ye-ys)/(ze-zs))\right|
$$

$\alpha = t$ $\quad ze - zs \geqslant 0, ye - ys \geqslant 0$

$\alpha = \pi - t$ $\quad ze - zs \leqslant 0, ye - ys \geqslant 0$

$\alpha = \pi + t$ $\quad ze - zs \leqslant 0, ye - ys \leqslant 0$

$\alpha = \pi + t$ $\quad ze - zs \geqslant 0, ye - ys \leqslant 0$

$x' = x$

$y' = y\cos\alpha - z\sin\alpha$

$z' = y\sin\alpha + z\cos\alpha$

③绕 Y 轴转 β 角

$t = |\mathrm{atan}(\mathrm{sqrt}((ye-ys)(ye-ys)+(ze-zs)(ze-zs))/(xe-xs))|$

$\beta = t$ $\quad ze - zs \geqslant 0$

$\beta = \pi - t$ $\quad ze - zs \leqslant 0$

$x = x'\cos\beta + z'\sin\beta$

$y = y'$

$z = x'\sin\beta + z'\cos\beta$

④绕 X 轴转 θ 角

$x' = x$

$y' = y\cos\theta - z\sin\theta$

$z' = y\sin\theta - z\cos\theta$

⑤绕 Y 轴转 $-\beta$ 角

$x = x'\cos(-\beta) + z'\sin(-\beta)$

$y = y'$

$z = x'\sin(-\beta) + z'\cos(-\beta)$

⑥绕 X 轴转 $-\alpha$ 角

$x' = x$

$y' = y\cos(-\alpha) - z\sin(-\alpha)$

$z' = y\sin(-\alpha) + z\cos(-\alpha)$

⑦平移

$x = x' + xs,\ y = y' + ys,\ z = z' + zs$

绕任意轴旋转的程序设计如下：

```
//绕任意轴旋转函数 revolve_rand
//输入参数:(x0,y0,z0)---- 需旋转的点坐标
          c----旋转角度
//        (xs,ys,zs)----旋转轴起点坐标
```

```
//              (xs,ys,zs)----旋转轴终点坐标
//输出参数:(x,y,z)----旋转后点坐标
void revolve_rand(float x0,float y0,float z0,float c,float xs,float ys,float zs,float xe,float ye,float ze,float &x,float &y,float &z )
{   float ax,ay,az,s,dx,dy,dz,cd1,cd3,cd2,xd,yd,zd;
    dx = xe - xs,dy = ye - ys,dz = ze - zs;
    x = x0 - xs,y = y0 - ys,z = z0 - zs;    //平移( -xs, -ys, -zs)
    cd1 = fabs(atan(dy/dz));    //绕 X 轴转 cd1
    if(dz <=0 && dy >=0)
              cd1 =3.1415926 - cd1;
    else if(dz <=0 && dy <=0)
              cd1 =3.1415926 + cd1;
    else if(dz >=0 && dy <=0)
              cd1 =6.283 - cd1;
    revolve_x(cd1,x,y,z,0,0,0,xd,yd,zd);
    cd2 = fabs(atan(sqrt(dy * dy + dz * dz)/dx)); //绕 Y 轴转 cd2
    if(dx <=0 )
              cd2 =3.1415926 - cd2;
    revolve_y(cd2,xd,yd,zd,0,0,0,x,y,z);
    revolve_x(c,x,y,z,0,0,0,xd,yd,zd); //绕 X 轴转 c
    revolve_y( -cd2,xd,yd,zd,0,0,0,x,y,z); //绕 Y 轴转 -cd2
    revolve_x( -cd1,x,y,z,0,0,0,xd,yd,zd); //绕 X 轴转 -cd1
    x = xd + xs,y = yd + ys,z = zd + zs; //平移(xs,ys,zs)
}
```

1.2　三维图形的投影变换

众所周知,计算机图形输出设备,例如,图形显示器的显示屏幕、绘图仪、打印机等都是二维设备。要用这些二维设备显示具有立体感的三维图形,就需要把三维图形上各点坐标变成二维坐标。把三维物体用二维图形表示的过程称为投影变换。

1.2.1　投影变换的分类

投影变换根据投影中心与投影平面之间距离的不同可分为两大类，即平行投影和透视投影。平行投影的投影中心与投影平面之间的距离是无限的，而透视投影的投影中心与投影平面之间的距离是有限的。这两大类根据投影方向、投影面及物体的旋转角度的不同又分为多种类型的投影，如图 1－3 所示。

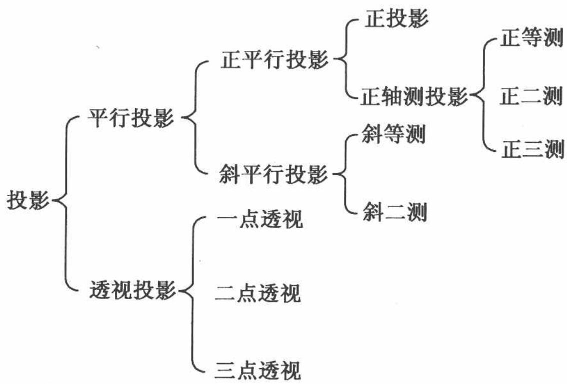

图 1－3　投影的分类

图 1－4 表示了同一条线段 AB 的两类不同投影，其投影线段为 $A'B'$。图 1－4(a)是平行投影，投影线是相互平行的；图 1－4(b)是透视投影，投影线交于投影中心。

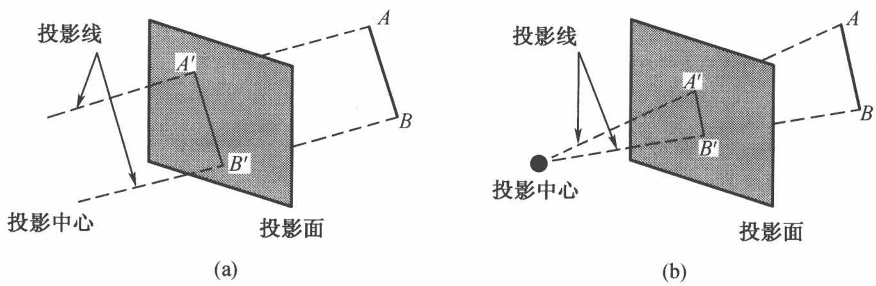

图 1－4　平行与透视投影

(a)平行投影；(b)透视投影

1.2.2　平行投影

平行投影根据投影方向与投影面的夹角不同分为正平行投影与斜平行投影。

1. 正平行投影(三视图)

当投影方向垂直于投影平面时称为正平行投影,通常说的三视图(正视图、俯视图、侧视图)均属正平行投影。如图 1－5 所示,三视图的生成就是把物体投影到分别平行于 3 个坐标面的平面上,一般还需将 3 个视图在一个平面上画出,这时就需要进行坐标变换。

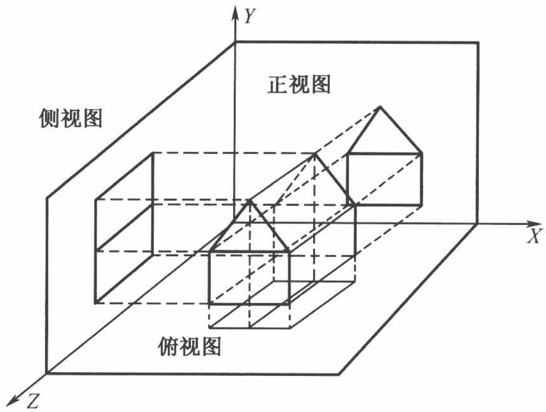

图 1－5　三视图

设投影平面为 $z=0$ 平面,则可将三视图全部投影到 $z=0$ 平面上。

(1)正视图

如图 1－6(a)所示的三维物体,生成正视图的变换过程如下:

①先对三维物体作正投影变换,如图 1－6(b)所示;

②再对投影后的图形进行平移变换,如图 1－6(c)所示。

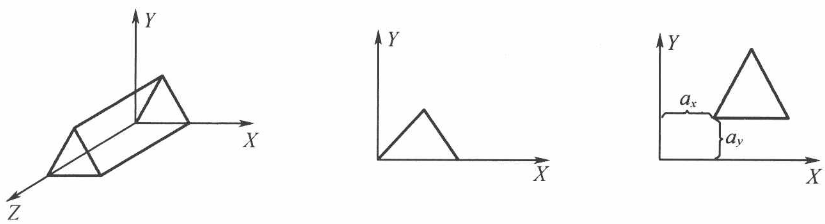

图 1－6　正视图变换过程

以上变换用齐次坐标及变换矩阵表示如下:

$$(x' \; y' \; z' \; 1)=(x \; y \; z \; 1)\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\a_x&a_y&0&1\end{bmatrix}=(x \; y \; z \; 1)\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\a_x&a_y&0&1\end{bmatrix} \tag{1－7}$$

所以正视图的变换式为 $x'=x+a_x, y'=y+a_y, z'=0$。

(2)侧视图

如图 1－7(a)所示的三维物体，生成侧视图的变换过程如下：

①先将三维形体绕 Y 轴旋转 90°，如图 1－7(a)所示；

②再对三维物体作正投影变换，如图 1－7(b)所示；

③对投影后的图形进行平移变换，如图 1－7(c)所示。

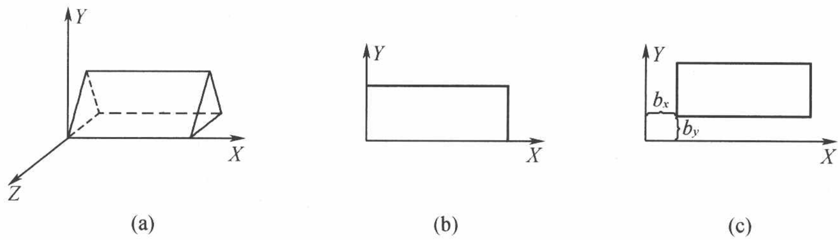

图 1－7　侧视图变换过程

以上变换用齐次坐标及变换矩阵表示如下：

$$(x' \; y' \; z' \; 1) = (x \; y \; z \; 1)\begin{bmatrix} \cos 90° & 0 & -\sin 90° & 0 \\ 0 & 1 & 0 & 0 \\ \sin 90° & 0 & \cos 90° & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b_x & b_y & 0 & 1 \end{bmatrix}$$

$$= (x \; y \; z \; 1)\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ b_y & b_x & 0 & 1 \end{bmatrix} \tag{1-8}$$

所以侧视图的变换式为 $x' = z + b_x$，$y' = y + b_y$，$z' = 0$。

(3)俯视图

如图 1－8(a)所示的三维物体，生成俯视图的变换过程如下：

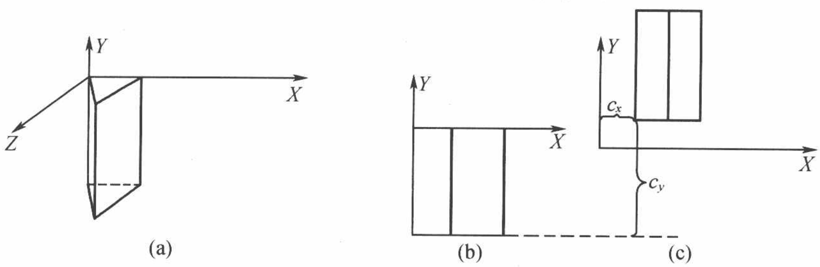

图 1－8　俯视图变换过程

①先将三维物体绕 X 轴旋转 90°，如图 1－8(a)所示；

②再对三维物体作正投影变换，如图 1－8(b)所示；

③对投影后的图形进行平移变换，如图 1－8(c)所示。

以上变换用齐次坐标及变换矩阵表示如下：

$$(x'\ y'\ z'\ 1)=(x\ y\ z\ 1)\begin{bmatrix}1 & 0 & 0 & 0\\0 & \cos 90^\circ & \sin 90^\circ & 0\\0 & -\sin 90^\circ & \cos 90^\circ & 0\\0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 0\\c_x & c_y & 0 & 1\end{bmatrix}$$

$$=(x\ y\ z\ 1)\begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & -1 & 0 & 0\\c_x & c_y & 0 & 1\end{bmatrix} \tag{1-9}$$

所以俯视图的变换式为 $x'=x+c_x, y'=-z+c_y, z'=0$。

(4)正轴测投影

三视图中每个视图只能反映三维物体中的两个坐标方向的实际长度，如果要在一个视图中反映形体的3个坐标方向形状，可采用正轴测投影，如图 1－9(a)所示的三维形体，生成三维形体正轴测投影图的变换过程如下：

①先将三维物体绕 Y 轴旋转 φ 角，如图 1－9(b)所示；

②再将三维物体绕 X 轴旋转 θ 角，如图 1－9(c)所示；

③对三维物体作正投影变换，如图 1－9(d)所示；

④对投影后的图形进行平移变换，如图 1－9(e)所示。

以上变换用齐次坐标及变换矩阵表示如下：

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

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

$$=(x\ y\ z\ 1)\begin{bmatrix}\cos\varphi & \sin\varphi\sin\theta & 0 & 0\\0 & \cos\theta & 0 & 0\\\sin\varphi & -\cos\varphi\sin\theta & 0 & 0\\d_x & d_y & 0 & 1\end{bmatrix} \tag{1-10}$$

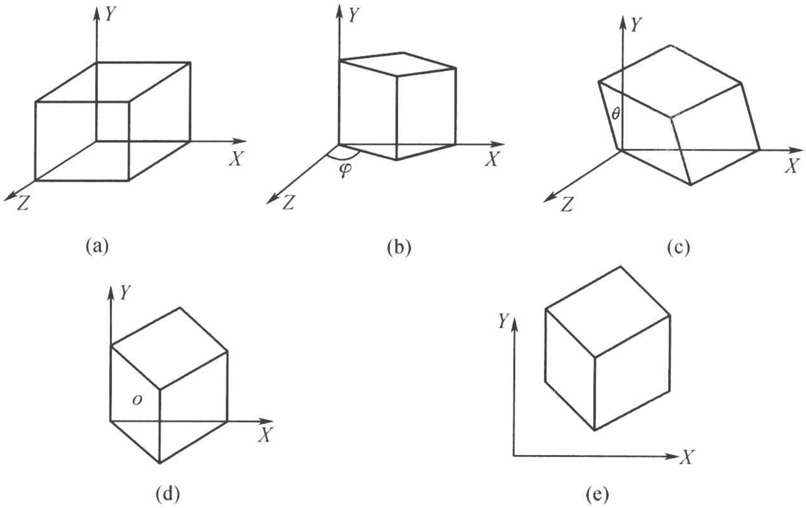

图 1－9　正轴测投影的变换过程

所以正轴测投影的变换式为

$$x' = x\cos\varphi + z\sin\varphi + d_x$$
$$y' = x\sin\varphi\sin\theta + y\cos\theta - z\cos\varphi\sin\theta + d_y$$
$$z' = 0$$

对于正等测投影，其 X,Y,Z 三个方向上的缩放率相等，这时 $\varphi = 45°$，$\theta = 35°16'$；对于正二测投影，其 X,Y,Z 三个方向上的其中两个方向缩放率相等（如 X,Y），这时，$\varphi = 20°42'$，$\theta = 19°28'$。例如一个正方体，如果在正等测投影中，所有边长都相同，如图 1－10(a)所示；在正二侧投影中，有 X,Y 两个方向的长度相同，如图 1－10(b)所示。

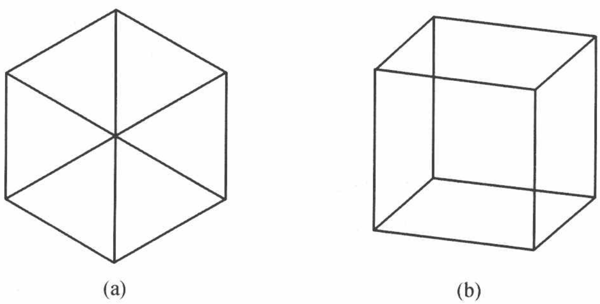

图 1－10　正方体的正等测与正二侧投影

(5)线框图的存储方法

要将三维物体的线框图在投影面显示出来，必须将三维物体的相关数据按一定格式保存，