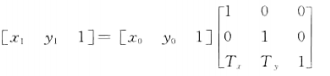
java3D与计算机图形学期末复习 第五章 坐标变换与基本形体

重点复习：

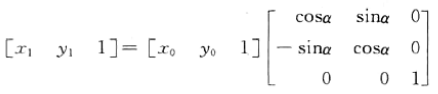
1、图形坐标变换

初始点A坐标为（x0,y0），变换后点A'坐标为（x1,y1）

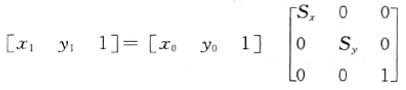
（1）平移变换

矩阵表示：，其中Tx=（x1-x0）,Ty=（y1-y0）。

（2）旋转变换

矩阵表示：，其中α为点A围绕圆心逆时针旋转的角度。（如果是顺时针旋转α角，将上面的α全部替换成-α即可）

（3）等比变换

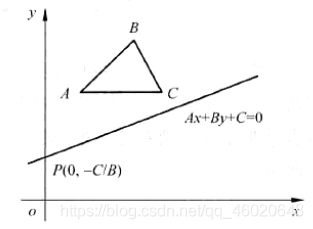
矩阵表示：，其中Sx= x1/x0，Sy= y1/y0。

（4）轴对称变换

轴对称变换是特殊的等比变换，Sx=1，Sy=-1，是对于x轴对称；Sx=-1，Sy=1，是对于y轴对称。

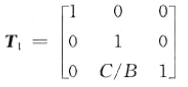
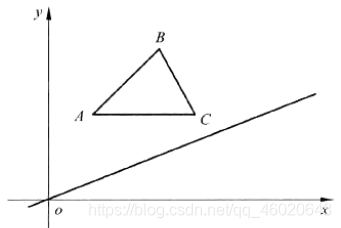
2、组合变换

将上述四种变换，综合起来使用就可以求出复杂的变换。接下来，以求三角形ABC关于直线Ax+By+C=0对称的三角形A'B'C'为例：

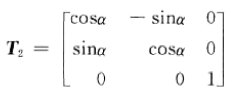
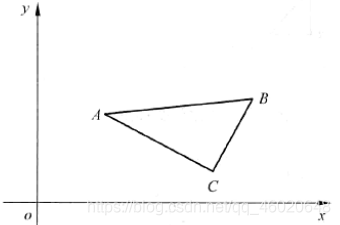


具体步骤：

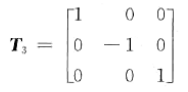
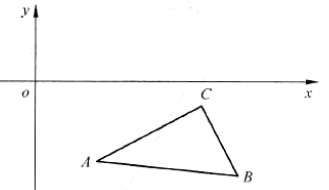
（1）求出直线与y轴的交点坐标（0，-C/B），将三角形向下平移C/B，得出第一个矩阵T1。

，变换后结果为

（2）标记直线与x轴的夹角为α角，将三角形顺时针旋转α角，得出第二个矩阵T2。

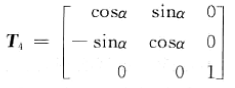
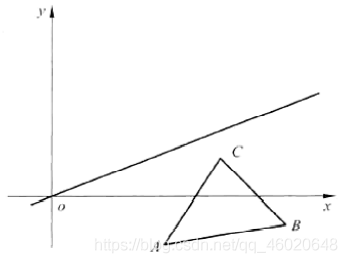
，变换后结果为

（3）对三角形求关于x轴对称，得出第三个矩阵T3。

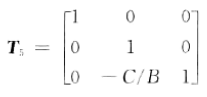
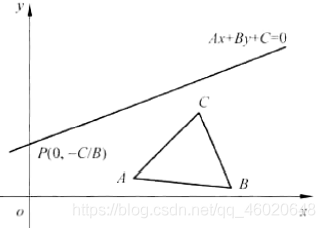
，变换后结果为

当前，直线Ax+By+C=0是与x轴重合的，我们接下来需要将它变换到原来的位置。

（4）逆时针旋转α角，得出第四个矩阵T4。

，变换后结果为

（5）最后向上平移C/B，得出第五个矩阵T5。

，变换后结果为

（6）最后将所有矩阵相乘，可以求出组合变换矩阵。



3、Transform3D类

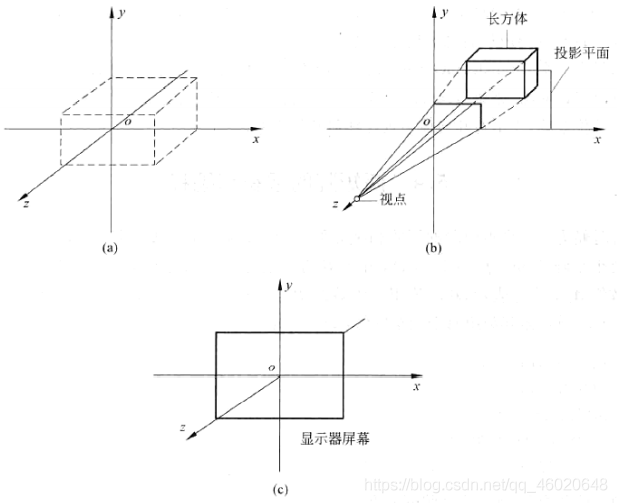
通常一个Transform3D类的对象是一种内部表达为4×4的双精度类型的矩阵。一个Transform3D类的对象用来执行平移、旋转、变比例等坐标变换。

4、TransformGroup类

所有表示各种几何变换的Transform3D类的对象必须包含在一个TransformGroup类的对象中，表示对在该TransformGroup类的对象中的三维图形所进行的坐标变换。

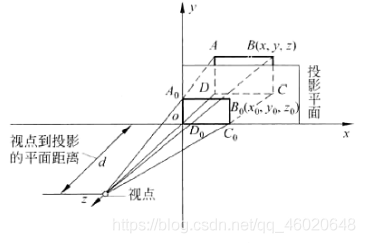
如果表示一种几何变换的Transform3D对象没有加入到该TransformGroup类的对象中,则该变换对在该TransformGroup类的对象中的三维图形不起作用。

5、世界坐标系、观察坐标系与Java3D显示器坐标系

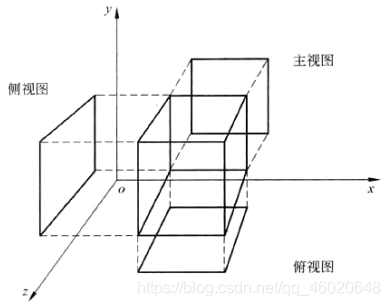


6、透视投影、平行投影及其计算

透视投影利用相似三角形原理计算。



平行投影——就是求三视图



7、隐藏线、隐藏面消除算法

当给定视点与观察方向后，对视点来讲，并不是所有的面都可见，可见面是由视点与观察方向及面的法向量决定。只需显示可见的面，无须显示隐藏的面，这种算法叫做隐藏面消除算法。

隐藏面的消除与隐藏线消除有时是相互关联的。由隐藏线所围成的面是隐藏面，两个隐藏面所共有的线为隐藏线。

8、Z Buffer隐藏面消除算法

在显示器的像素点阵建立两种缓冲存储单元阵列：Z Buffer 和C Buffer阵列。

简单地说：从视点出发通过每个像素点发出一条射线，Z Buffer阵列存储距离视点最近的多边形与射线交点z坐标，C Buffer阵列存储对应的交点所在面颜色。

java3D与计算机图形学期末复习 第五章 坐标变换与基本形体

重点复习：

1、同一平面内点与多边形的包含判定

（1）凸多边形的叉积判断法

（补充一个知识点：

叉积的一个重要性质是可以通过它的符号判断两矢量相互之间的顺逆时针关系：

若 P × Q > 0 , 则P在Q的顺时针方向

若 P × Q < 0 , 则P在Q的逆时针方向

若  P × Q = 0 , 则P与Q共线，但可能同向也可能反向）

从给定的一个点向多边形的各个顶点做向量。按顶点顺序逐个计算每相邻两向量之间的叉积。如果所有的叉积符号相同，说明点在多边形内；反之，则在多边形外。

（2）夹角之和检验法

从给定的一个点向多边形的各个顶点做向量。按顶点顺序逐个计算每相邻两向量之间的夹角。如果所有的夹角符号相同，说明点在多边形内；反之，则在多边形外。

（3）交点计数检验法

从要判断的点向右做一条射线，求出射线与多边形交点的个数。如果交点的个数是奇数，则点在多边形内；反之，则在多边形外。

2、局部光照明计算模型

在给定光源（包括一个或多个光源）光源的类型、颜色以及几何体表面的各种特性参数后，不考虑场景中其他几何体的影响，通过某种计算算法计算几何体表面上各点的光颜色与强度，将这种算法称为局部光照明计算模型。

局部光照明计算模型只考虑物体表面对直射光、环境光的反射、散射与透射。

3、Gouraud算法

基本过程：

（1）先计算出各个小多边形及其顶点的单位法线向量，公共顶点处用合成向量。

（2）在给定光源与几何体表面各特性参数基础上，用局部光照计算模型计算每个顶点的光颜色、光强度值。

（3）由两顶点的光颜色、光强度值,通过线性插值计算出两个顶点之间的边上的各点的光颜色、光强度值。

（4）在扫描线填色算法中，每条水平扫描线与多边形的边相交，都有一对交点,通过线性插值求出这些交点的光颜色与光强度值，再由这些交点的光颜色、光强度值，通过线性插值计算出该水平扫描线上位于多边形内部的各像素点的光颜色、光强度值。通过水平扫描线从上向下的移动，也就相应地求出了小多边形平面上所有点的光颜色、光强度值。

优点：计算量小

缺点：高光区域有时会出现异常；当对曲面用不同的多边形进行分割时会产生不同的显示效果；Gouraud明暗处理会造成表面上出现过亮或过暗的条纹，称为马赫带(Mach \_band)效应。

4、Phong算法

基本过程：

（1）计算平面多边形的单位法矢量。

（2）计算多边形各顶点处的单位法矢量,进而计算各公共顶点的合成向量。

（3）通过对多边形各顶点的单位法向量进行双线性插值,计算多边形内部点的单位法向量。

（4）通过平面多边形内部各点处的单位法向量,利用局部光照明模型计算多边形内部各点光颜色,光强度值。

5、Whitted整体关照模型

为了增加三维图形显示的真实性，除了考虑各种类型光源对几何体表面的直接照射外，还必须考虑当前场景中来自于其他几何体的漫射光、镜面反射光和透射光对该几何体表面的照射，这样就会产生比较真实的照明效果，如阴影与透明效果等。将这种光照明计算模型称为整体光照明模型。

Whitted光照明模型是一种整体光照明模型,，它能很好地模拟光能在光滑几何体表面之间的镜面反射和通过透明体产生的规则透射。

6、光线追踪算法（Raytracing）

光线跟踪算法是一种在由多光源与多个几何体组成的场景中对几何体进行消隐与整体光强度计算的算法。

光线跟踪算法沿着到达视点的光线的反方向进行跟踪，经过屏幕上每一个像素点，找出所跟踪的光线与几何体的交点，在该交点处分别沿反射方向与折射方向再进行跟踪，找出影响该点光强度的所有光源，通过迭代、累加计算跟踪点的光颜色与光强度。光强度计算包括直接照射到几何体表面上一点的漫射光源、直射光源、透射光源,还包括来自反射跟踪方向的其他几何体传来的光强与沿着透射跟踪方向的其他几何体传来的光强。

7、Switch类

意为切换，Switch类可以控制哪些节点将被显示。

8、OrderedGroup类

一般情况下，在一个场景图中三维形体的显示顺序是不确定的，这种显示顺序依赖于当时程序的执行效率。但是，OrderedGroup 类可确保在一个场景图中的三维形体的显示顺序。

9、SharedGroup类

SharedGroup类提供了一种可对场景图中的子图进行链接(Link)复制的能力。这样，在生成复杂场景时，就可以通过Link叶子节点来共享与复制场景中的子图。SharedGroup节点允许叶子节点对其子图通过链接(由Link叶子节点实现)方式共享。

10、Link类

Link 叶子节点允许一个应用程序引用一个以SharedGroup为根的共享子图。任何数量的Link 叶子节点都能引用同一个SharedGroup节点。