为什么称作Pipeline呢？因为它就像个一条长长的流水线，输入Vertex data，经过一个又一个Stage的转换，最后输出Pixel color。

所谓的对象坐标系，指的是以对象参考点为原点的坐标系。以一个正方体来说，在对象坐标系上(以正方体中心为参考点)，八个Vertex的坐标分别是(-1,-1,1) (1,-1,1)(1,1,1) (-1,1,1) (-1,-1,-1) (1,-1,-1) (1,1,-1) (-1,1,1)。

而所谓的全局坐标系，是整个Scene的坐标系，对所有场景中的对象而言都是同一个，以上例的正方体来说，若它的中心点在World coordinate中是(3,6,4)，假设没有旋转，那么8个Vertex在World coordinate上的坐标分别是(2,5,5) (4,5,5) (4,7,5) (2,7,5)(2,5,3) (4,5,3) (4,7,3) (2,7,3)

|  |
| --- |
|  |
| 1.物体坐标系向世界空间坐标系的转换，平移，旋转，缩放  2.世界坐标系到相机坐标系的转换，通过一个仿射变换设置一个相机位置和朝向，逆这个变换，把物体从世界坐标系转换到相机坐标系  3.根据照明和反射率计算光照，算出颜色  4.相机坐标系转换到裁剪坐标系，2D坐标，透视投影，模仿真实世界的观察  5.把在屏幕视窗外的物体裁剪  6.绘制像素，采样，光栅化，包括纹理，影藏面等等。 |
| 我们输入包括一个相机的位置，朝向，  物体：包括点，线，多边形。几何的属性：位置，法向量， 颜色，纹理坐标等  光源：方向，位置，颜色，强度  纹理。  输出：像素颜色图  把物体变换到世界空间  从世界空间变换到相机空间  顶点颜色计算，更具光照的方向和位置，与转换过的顶点位置，法向量和顶点属性。  投影变换，把视体投影到一个约定的视体，若果是透视投影，就是一个截断的金字塔，投影到立方体，长和宽是2.（-1，-1，-1）， （1,1,1）。  视体有近平面，远平面，左平面，右平面，上平面和下平面。 |
| 物体的设计不用关心场景  设置物体在空间中的位置，大小和朝向 |
| 红色是相机的基，而黑色是世界的基，也就是参考系。小人是世界中的一个物体。相机在移动之前，两个基是重合的。当相机在屏幕中定位时，它首先会进行朝向的确定——旋转，然后进行位置的确定——平移。图中的Rotation和Translation两步就是相机定位时所发生的变换。可以看到相机相对于小人的运动。而当进行相机变换的时候，小人应该从世界基变换到相机的基里面。这样，他应该进行一个相机定位的逆定位，先逆平移小人和相机，然后再逆旋转小人和相机，最后相机归位，小人随相机变到了相机空间。这是由Inverse Translation和Inverse Rotation两个步骤完成的，这两个步骤就是相机变换。现在我们推导这个变换。 |
|  |