

GLSL 核心教程 - 光栅化和插值

本章讨论两个了解图形管线的重要的概念：光栅化和插值。这是固定的图形管线，这部分是不可编程的。

然而，在光栅化之前，需要介绍几个小步骤，裁切和窗口坐标变换。

裁切

这个阶段接收了组装过的图元、变换后的顶点，然后检查顶点是否在裁切体积内。完全在裁切体积内的图元不会有变化。在裁切体积外的图元会被直接丢弃。在裁切体积的边缘上的图元会被裁切。

裁切体积是视锥内的所有点的坐标 (x_c, y_c, z_c, w_c) ，就像这样。

$$-w_c \leq x_c \leq w_c$$

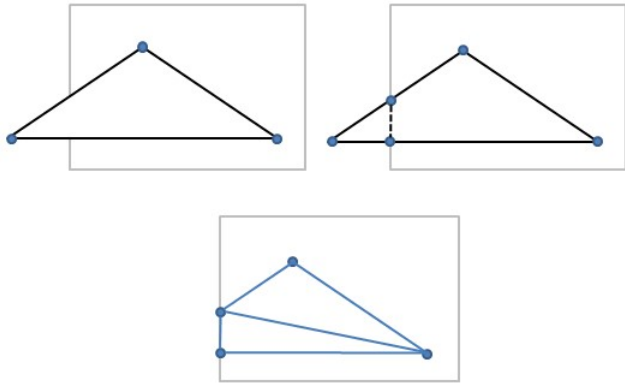
$$-w_c \leq y_c \leq w_c$$

$$-w_c \leq z_c \leq w_c$$

顶点的 `gl_Position` 属性保存了裁切坐标，它还测试了裁切目标。顶点裁切坐标食指 (x_c, y_c, z_c, w_c) 。

对于有边缘的图元，顶点会处在裁切体积边缘的两侧，新的顶点根据裁切平面的相交的点来计算。新的顶点将会成为图元的一部分，裁切体积之外的顶点会被丢弃。因此，裁切可能会为图元增加新的顶点，但是这对程序员来说是透明的。

下面的图演示了一个三角形在裁切时发生了什么。



新的顶点的属性来自原来的边缘，除非平面着色被指定了属性，这种情况下，属性值为常数。

顶点可以转换，确切地说是透视除法，要规格化设备坐标：

$$(x_d, y_d, z_d) = (x_c/w_c, y_c/w_c, z_c/w_c)$$

考虑到规格化设备坐标，裁切体积变为一个坐标范围从 -1 到 1 的立方体（除非禁用深度钳位，这是， z 坐标不会受到限制）。因此，如果有深度裁切，顶点在被裁切后是这样的：

$$-1 \leq x_d \leq 1$$

$$-1 \leq y_d \leq 1$$

$$-1 \leq z_d \leq 1$$

平面着色

当为一个顶点属性，在一个声明中指定平面后，会从一个顶点中复制属性到其余的顶点中。如果没有说，图元的最后一个顶点的属性值会覆盖所有顶点。这个顶点称为 *provoking* 顶点。OpenGL 允许我们指定第一个顶点或者最后一个为 *provoking* 顶点，使用下面的命令：

```
glProvokingVertex(enum provokeMode);
```

provokeMode 可以使用的值有：GL_FIRST_VERTEX_CONVENTION 和 GL_LAST_VERTEX_CONVENTION（最后一个默认值）。

窗口坐标

可以为每一个顶点计算窗口坐标。窗口坐标依赖于视口。考虑视口的高度 h 、宽度 w 和中心 (o_x, o_y) 。在窗口坐标中，所有的值的单位都是像素。

顶点对于规格化的设备坐标 (x_d, y_d) 到窗口坐标 (x_w, y_w, z_w) 的计算方法如下：

$$(x_w, y_w, z_w) = (x_d * (w/2) + o_x, y_d * (h/2) + o_y, (z_d * (f-n) + (n+f))/2)$$

其中 n 和 f 代表 OpenGL 提供的 `glDepthRange*` 函数指定的远近平面。如果没有指定， $n = 0$ ， $f = 1$ ，然后上面的式子就可以写成：

$$(x_w, y_w, z_w) = (x_d * (w/2) + o_x, y_d * (h/2) + o_y, (z_d + 1)/2)$$

光栅化

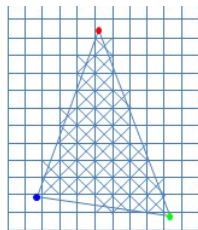
光栅化在设置最终图像的所有像素时进行，是图元的一部分。

多边形的光栅化的第一步是确定方向，也就是哪个是前面，哪个是后面。如果启用了剔除，所有方向不正确的三角形都会被剔除，不参与光栅化过程。

对于方向正确的三角形，每个可以确定的像素都是它的一部分。每个像素的中心点都在三角形的边界内部，这些像素会被进一步处理。在下面的图中，彩色的点表示顶点在屏幕中的位置。注意，顶点所在的像素可能不是三角形的一部分。

- GLSL 核心教程
 - 管线
 - 顶点着色器
 - 图元组合
 - 曲面细分
 - 几何着色器
 - GS 示例





插值

下一步是根据顶点的属性和像素与顶点在屏幕上的距离来计算像素的属性。

中心坐标可用于帮助计算插值。任意在三角形内的点 $p(pa, pb, pc)$ 可以表示为：

$$p = a * pa + b * pb + c * pc$$

where

$$a + b + c = 1$$

and

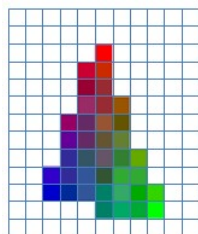
$$0 \leq a \leq 1$$

$$0 \leq b \leq 1$$

$$0 \leq c \leq 1$$

三元 (a, b, c) 中包含了重心坐标 p 。重心坐标可以当作每一个顶点对像素的影响的权重，正如前面所提到的，权重的总值为 1。

对于每一个像素，每个属性的插值计算都要考虑点 p 和中心像素的位置，并且考虑计算权重，也就是重心坐标，并在屏幕空间中显示顶点的属性的值。



翻译自: <http://www.lighthouse3d.com/tutorials/glsl-core-tutorial/geometry-shader/>