

iOS开发-OpenGL ES入门教程3



作者 落影loyinglin (/u/815d10a4bdce) + 关注

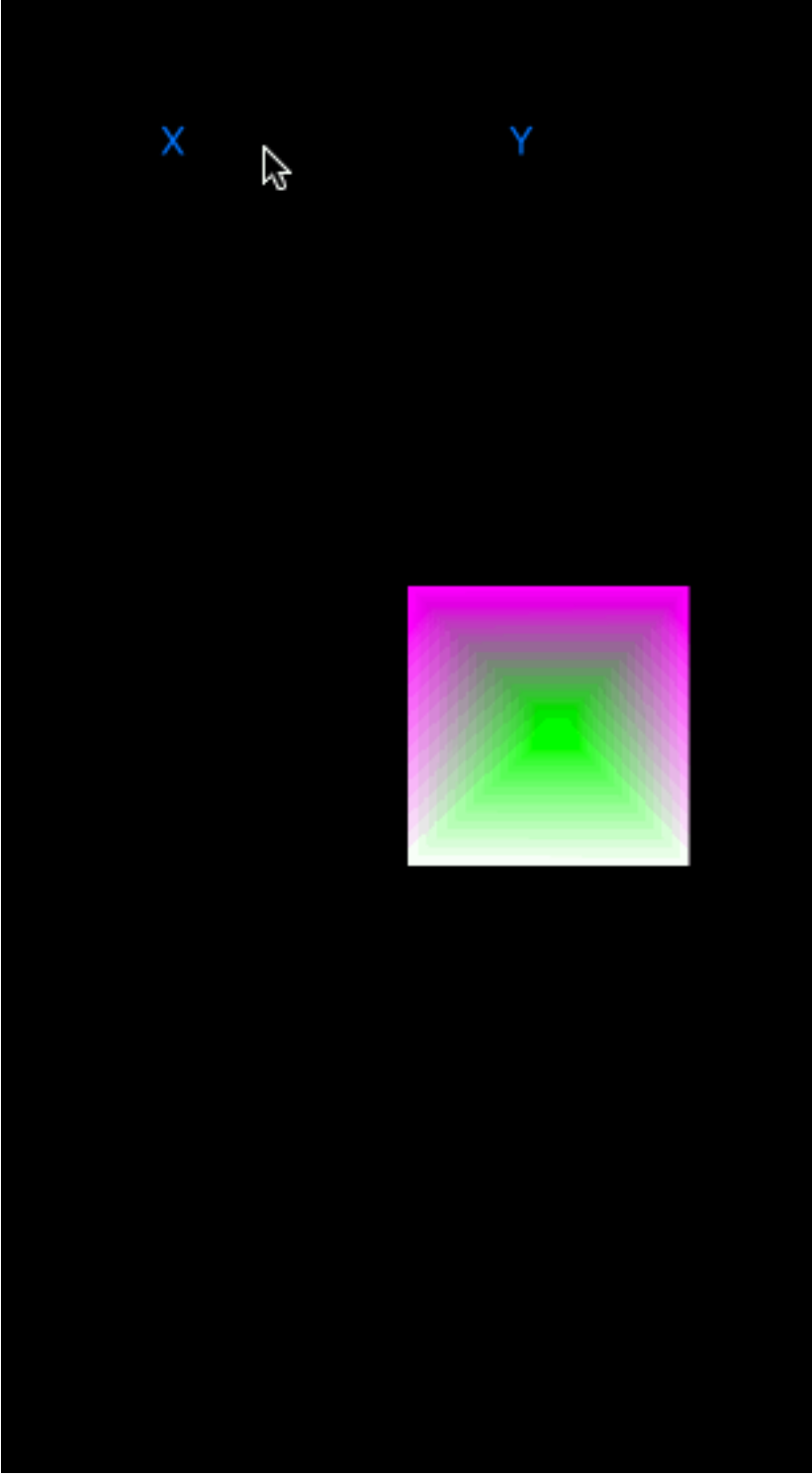
2016.03.29 15:25* 字数 1262 阅读 4176 评论 42 喜欢 16 阅读 4176 评论 42 喜欢 16 (/u/815d10a4bdce)

教程

OpenGL ES入门教程1-Tutorial01-GLKit (<http://www.jianshu.com/p/750fde1d8b6a>)
OpenGL ES入门教程2-Tutorial02-shader入门 (<http://www.jianshu.com/p/ee597b2bd399>)
这次是**三维图形变换**。

OpenGL ES系列教程在这里 (<http://www.jianshu.com/notebooks/2135411/latest>)。
OpenGL ES系列教程的代码地址 (<https://github.com/loyinglin/LearnOpenGLES>) - 你的star和fork是我的源动力，你的意见能让我走得更远。

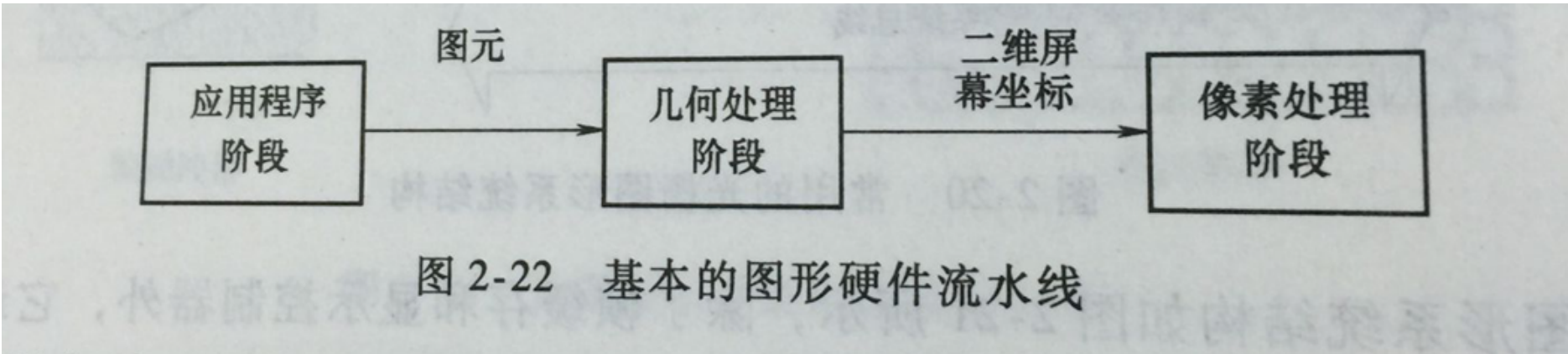
效果展示



概念准备

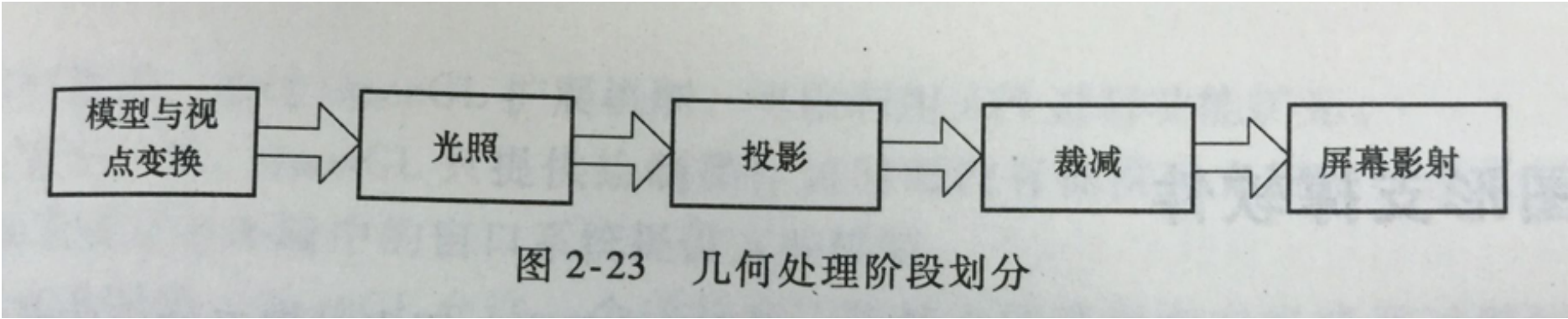
1、计算机图形学

首先了解计算机处理图形的流程，如下图

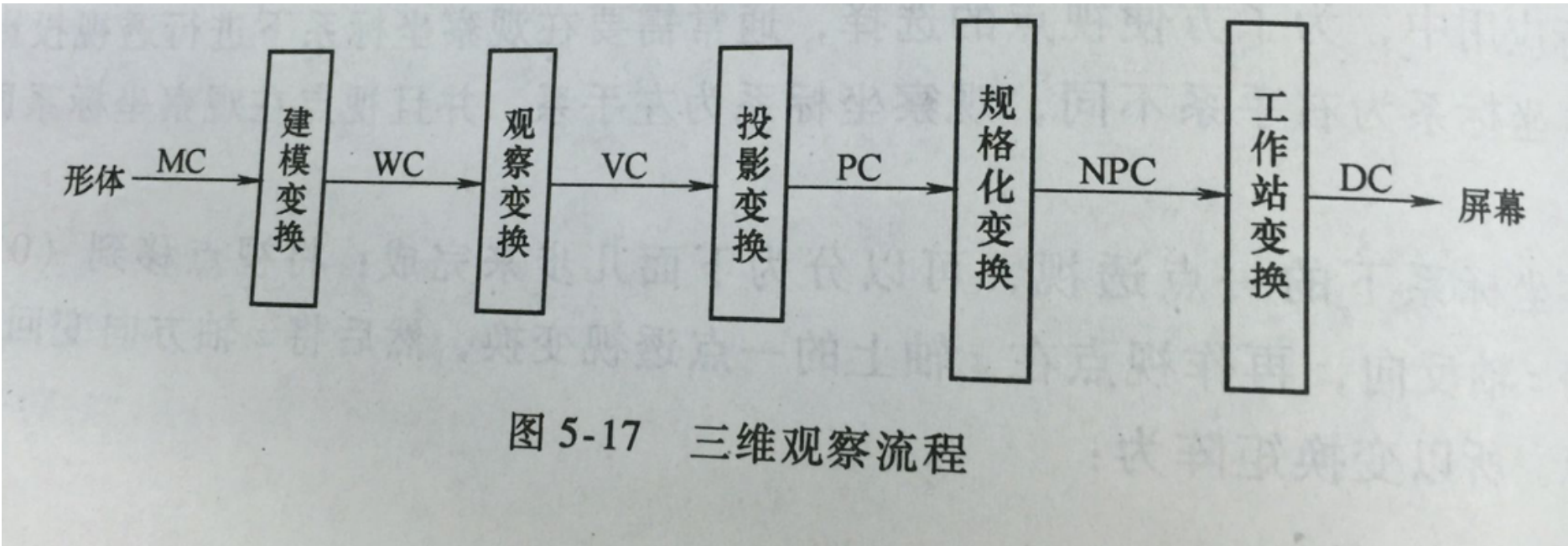


应用程序把数据以图元的方式提供给图形硬件，一般是点、线、多边形、纹理映射图像；基本图元通过几何变换和投影变换，获得二维屏幕坐标；对每一个屏幕像素点进行着色，得到具体的显示帧。

2、几何处理阶段



以顶点为基础，对几何图元进行处理，把三维坐标转变为二维屏幕坐标的过程。
具体的坐标系变换如下：



MC是建模坐标系，WC是世界坐标系，VC是观察坐标系，PC是投影坐标系，NPC是规格化投影坐标系，DC是设备坐标系。

几何变换

a、基本几何变换

平移变换、比例变换、旋转变换、对称变换、错切变换
具体的变换矩阵可以[点这里](http://blog.sina.com.cn/s/blog_620bf89501011f18.html) (http://blog.sina.com.cn/s/blog_620bf89501011f18.html)
或者 [这里](http://www.cnblogs.com/graphics/archive/2012/08/08/2609005.html) (http://www.cnblogs.com/graphics/archive/2012/08/08/2609005.html)

b、复合变换

- 关于任意点的比例、旋转变换

- 1、将任意点P移到原点，作平移变换；
- 2、进行比例、旋转等变换；
- 3、将参考点移到原处；

- 绕任意轴的旋转变换

[看这里](http://www.cnblogs.com/graphics/archive/2012/08/10/2627458.html) (http://www.cnblogs.com/graphics/archive/2012/08/10/2627458.html)

投影变换

把三维物体变为二维图形表示的过程成为投影变换。分类如下：

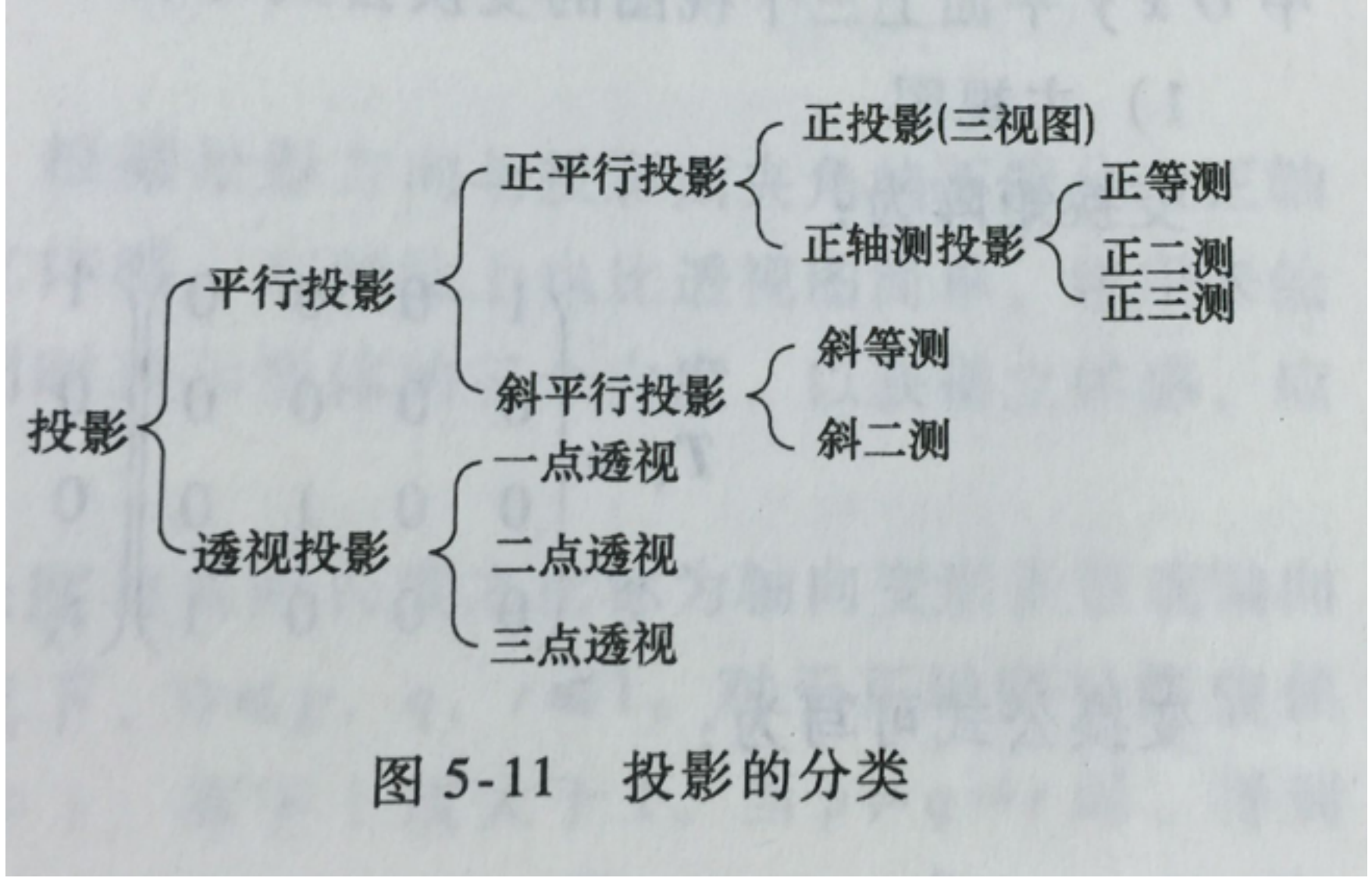


图 5-11 投影的分类

投影中心，也叫投影参考点，相当于人的视点，投影线相当于人的视线。

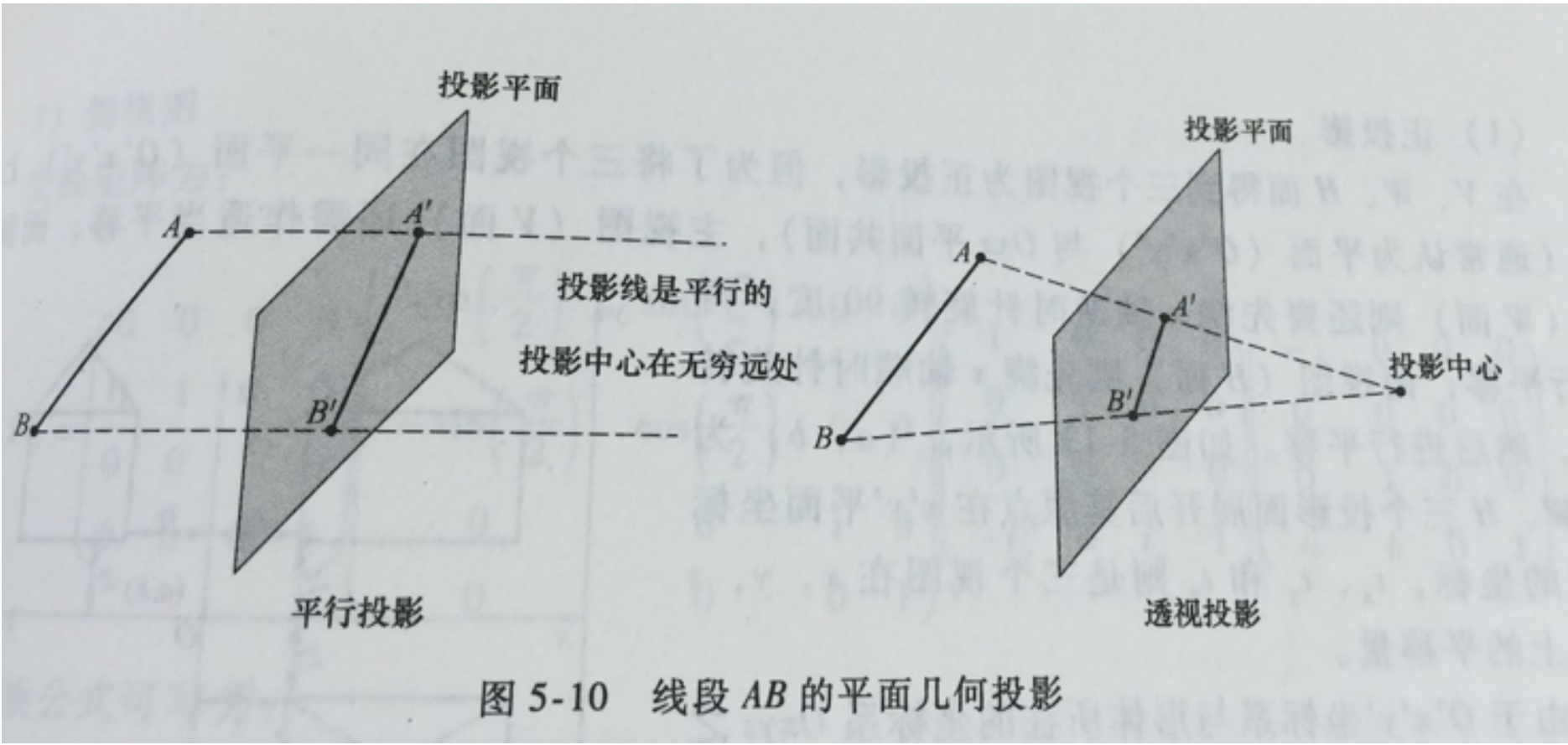


图 5-10 线段 AB 的平面几何投影

平行投影

投影中心和投影平面的距离为无穷大的投影。

- 正平行投影

投影方向垂直于投影平面时称为正平行投影。三视图（主视图、俯视图、侧视图）都属于正平行投影。

- 斜平行投影

投影方向不垂直于投影平面的平行投影称为斜平行投影。

透视投影

投影中心和投影平面的距离是有限的。

透视投影的推导可以看 [这里](#)

(<http://www.cnblogs.com/graphics/archive/2012/07/25/2582119.html>)

OpenGL ES的变换

OpenGL ES通过顶点缓存数组和图元绘制指令，形成基本的图元；图元在顶点着色器会进行顶点变换，也就是几何处理阶段的几何变换和投影变换；到了像素处理阶段，根据之前的结果，通过光照、纹理等对每一个像素点进行着色。

举一个例子，下面的代码用到了透视投影、平移变换、旋转变换。

```
KSMatrix4 _projectionMatrix;
ksMatrixLoadIdentity(&_projectionMatrix);
float aspect = width / height; //长宽比

ksPerspective(&_projectionMatrix, 30.0, aspect, 5.0f, 20.0f); //透视变换，视角30°

//设置glsl里面的投影矩阵
glUniformMatrix4fv(projectionMatrixSlot, 1, GL_FALSE, (GLfloat*)&_projectionMatrix.m[0][0]);

glEnable(GL_CULL_FACE);

KSMatrix4 _modelViewMatrix;
ksMatrixLoadIdentity(&_modelViewMatrix);

//平移
ksTranslate(&_modelViewMatrix, 0.5, 0.0, -10.0);

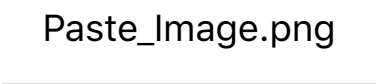
KSMatrix4 _rotationMatrix;
ksMatrixLoadIdentity(&_rotationMatrix);

//旋转
ksRotate(&_rotationMatrix, degree, 1.0, 0.0, 0.0); //绕X轴
ksRotate(&_rotationMatrix, yDegree, 0.0, 1.0, 0.0); //绕Y轴

//把变换矩阵相乘，注意先后顺序
ksMatrixMultiply(&_modelViewMatrix, &_rotationMatrix, &_modelViewMatrix);
```

这里插入一点矩阵的基本性质
(http://www.tongji.edu.cn/~math/xxds/kcja/kcja_a/01.htm):

- (1) $(A^T)^T = A$;
- (2) $(A+B)^T = A^T + B^T$;
- (3) $(\lambda A)^T = \lambda A^T$;
- (4) $(AB)^T = B^T A^T$ 。



ksMatrixMultiply(&_modelViewMatrix, &_rotationMatrix, &_modelViewMatrix); 这一行的代码的精髓在于理解rotation和modelView的先后顺序对最后结果的影响。

ksPerspective(&_projectionMatrix, 30.0, aspect, 5.0f, 20.0f); 这里需要明白5个参数的意义：result矩阵，视角，长宽比，近平面距离，远平面距离，不明白的可以点这里的透视投影推导 (<http://www.cnblogs.com/graphics/archive/2012/07/25/2582119.html>)。

ksTranslate(&_modelViewMatrix, 0.5, 0.0, -10.0); 这里是简单的平移变换，后三个参数为x、y、z的距离。

ksRotate(&_modelViewMatrix, degree, 1.0, 0.0, 0.0); 这里是旋转变换，后三个参数为旋转轴。

glsl代码

```
attribute vec4 position;
attribute vec4 positionColor;
uniform mat4 projectionMatrix;
uniform mat4 modelViewMatrix;

varying lowp vec4 varyColor;

void main()
{
    varyColor = positionColor;

    vec4 vPos;
    vPos = projectionMatrix * modelViewMatrix * position;

    //    vPos = position;

    gl_Position = vPos;
}
```

思考题

- 透视投影里面有一个视锥体的概念，物体不在视锥体内的部分不可见，OpenGL ES是如何判断一个点是否在视锥体内？
- 平移变换里面的z参数为何是负数，它的取值范围是多少？
- ksMatrixMultiply的参数如果颠倒会如何？
- glsl代码里面的projectionMatrix modelViewMatrix position顺序能否交换？要如何交换？

总结

教程2、3是shader的一个分支，内容相对较难，接下的教程主要以GLKit为主。

附上源码地址 (<https://github.com/loyinglin/LearnOpenGLES/tree/master/Tutorial03-%E4%B8%89%E7%BB%B4%E5%8F%98%E6%8D%A2>)
代码里面的util在 [这里](http://www.cnblogs.com/kesalin/archive/2012/12/06/3D_math.html)
(http://www.cnblogs.com/kesalin/archive/2012/12/06/3D_math.html)



落影loyinglin (/u/815d10a4bdce)

写了 171405 字，被 4763 人关注，获得了 2744 个喜欢


(/u/815d10a4bdce) 写了 171405 字，被 4763 人关注，获得了 2744 个喜欢

+ 关注

工程师一枚，喜欢思考，喜欢游戏，喜欢运动。做过什么已经不重要，未来的方向以及当下的准备是生活的...

 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button)


16





更多分享





(<http://cwb.assets.jianshu.io/notes/images/3396371>)


被以下专题收入，发现更多相似内容


 iOS Dev... (/c/3233d1a249ca?utm_source=desktop&utm_medium=notes-included-collection)

 OpenGL ... (/c/044a5240577d?utm_source=desktop&utm_medium=notes-included-collection)

 iOS开发专题 (/c/c258bc0ea6bd?)

-  OPeNGL ... (/c/408442c9c764?
utm_source=desktop&utm_medium=notes-included-collection)
-  OpenGL+... (/c/5d2c87603bd3?
utm_source=desktop&utm_medium=notes-included-collection)
-  OpenGL (/c/c2cd8d71b0c0?utm_source=desktop&utm_medium=notes-included-collection)
-  iOS开发成长之路 (/c/9735671a2cbb?
utm_source=desktop&utm_medium=notes-included-collection)

 登录/注册
为你个性化推荐内容
(/sign_in?utm_source=desktop&utm_medium=notes-bottom-bind)

 下载简书App
随时随地发现和创作内容
(/apps/download?utm_source=desktop&utm_medium=click-note-bottom-bind)