实验五

实验目的

- 学习光线追踪的原理
- 实现光线的生成
- 用Moller-Trumbore 算法来实现判断光线是否与三角形相交的函数

实验过程

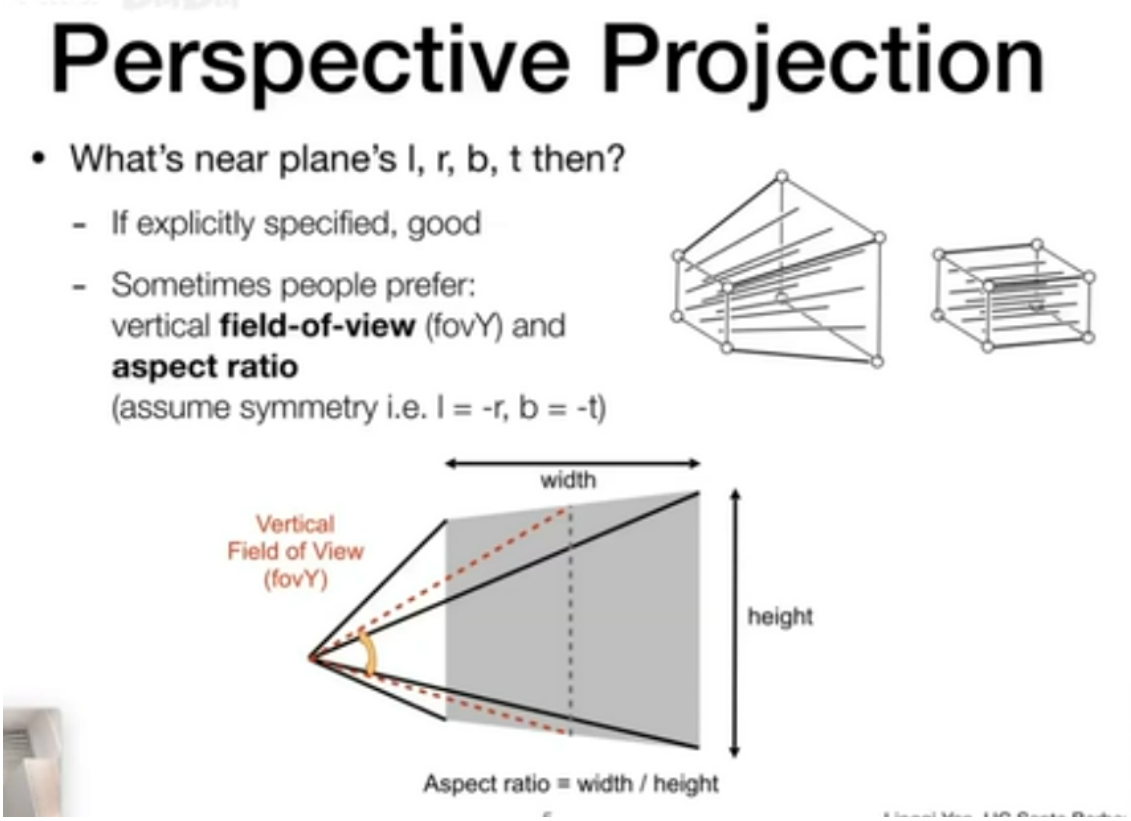
1. 修改完成Render函数

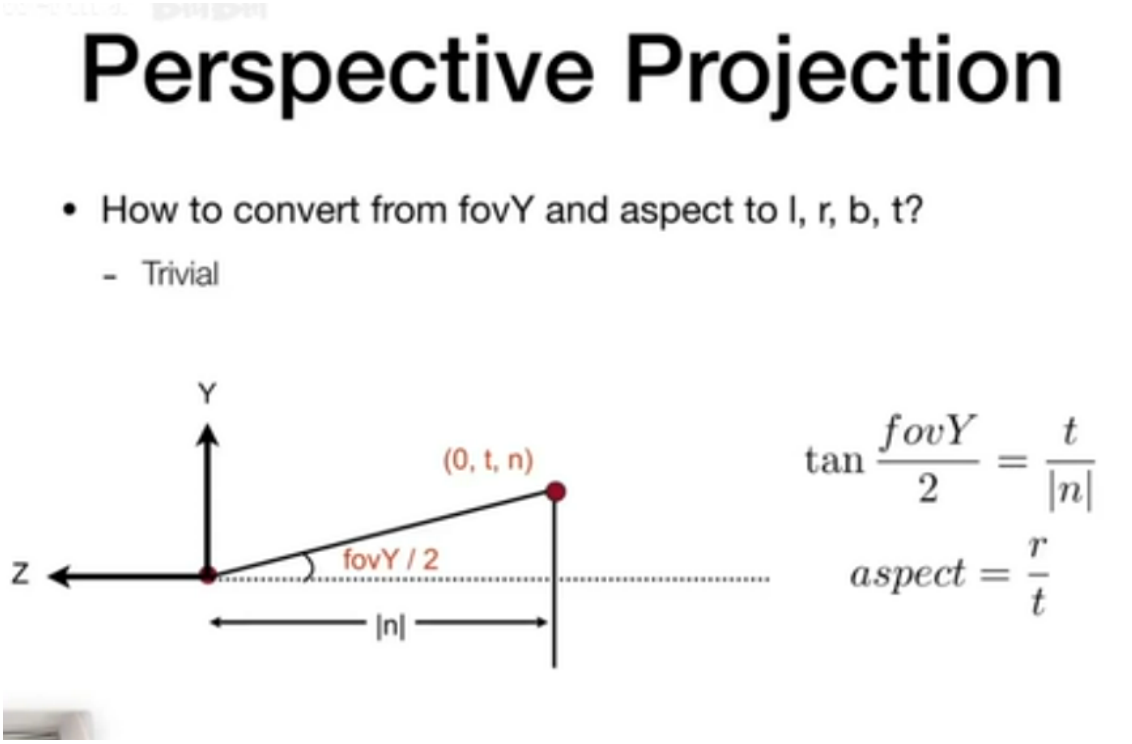
   这里主要的过程是生成视点到屏幕像素的光线。视点的位置已经给出，于是我们需要找到屏幕的位置，根据屏幕的位置，fov，宽高比进而算出每个屏幕像素的具体位置。算出光线的位置后，调用castRay()来得到颜色,最后将颜色存储在帧缓冲区的相应像素中。

```
// [comment]
// The main render function. This where we iterate over all pixels in the
image, generate
// primary rays and cast these rays into the scene. The content of the
framebuffer is
// saved to a file.
// [/comment]
void Renderer::Render(const Scene& scene)
{
    std::vector<Vector3f> framebuffer(scene.width * scene.height);

    float scale = std::tan(deg2rad(scene.fov * 0.5f));
    float imageAspectRatio = scene.width / (float)scene.height; // 屏幕宽高比

    // Use this variable as the eye position to start your rays.
    Vector3f eye_pos(0);
    int m = 0;
    for (int j = 0; j < scene.height; ++j)
    {
        for (int i = 0; i < scene.width; ++i)
        {
            // generate primary ray direction
            float x;
            float y;
            // TODO: Find the x and y positions of the current pixel to get
the direction
            // vector that passes through it.
            // Also, don't forget to multiply both of them with the variable
*scale*, and
            // x (horizontal) variable with the *imageAspectRatio*

            x = scale * 2 * imageAspectRatio / scene.width * i - scale * 2 *
imageAspectRatio / 2 ; // +0.5是像素中心点
            y = - (scale * 2 / scene.height * j - scale * 2 / 2) ;

            // 从dir可以看出屏幕的位置在z=-1处
```

```
            Vector3f dir = Vector3f(x, y, -1); // Don't forget to normalize
this direction!
            dir = normalize(dir); // 单位化向量
            framebuffer[m++] = castRay(eye_pos, dir, scene, 0);
        }
        UpdateProgress(j / (float)scene.height);
    }

    // save framebuffer to file
    FILE* fp = fopen("binary.ppm", "wb");
    (void)fprintf(fp, "P6\n%d %d\n255\n", scene.width, scene.height);
    for (auto i = 0; i < scene.height * scene.width; ++i) {
        static unsigned char color[3];
        color[0] = (char)(255 * clamp(0, 1, framebuffer[i].x));
        color[1] = (char)(255 * clamp(0, 1, framebuffer[i].y));
        color[2] = (char)(255 * clamp(0, 1, framebuffer[i].z));
        fwrite(color, 1, 3, fp);
    }
    fclose(fp);
}
```

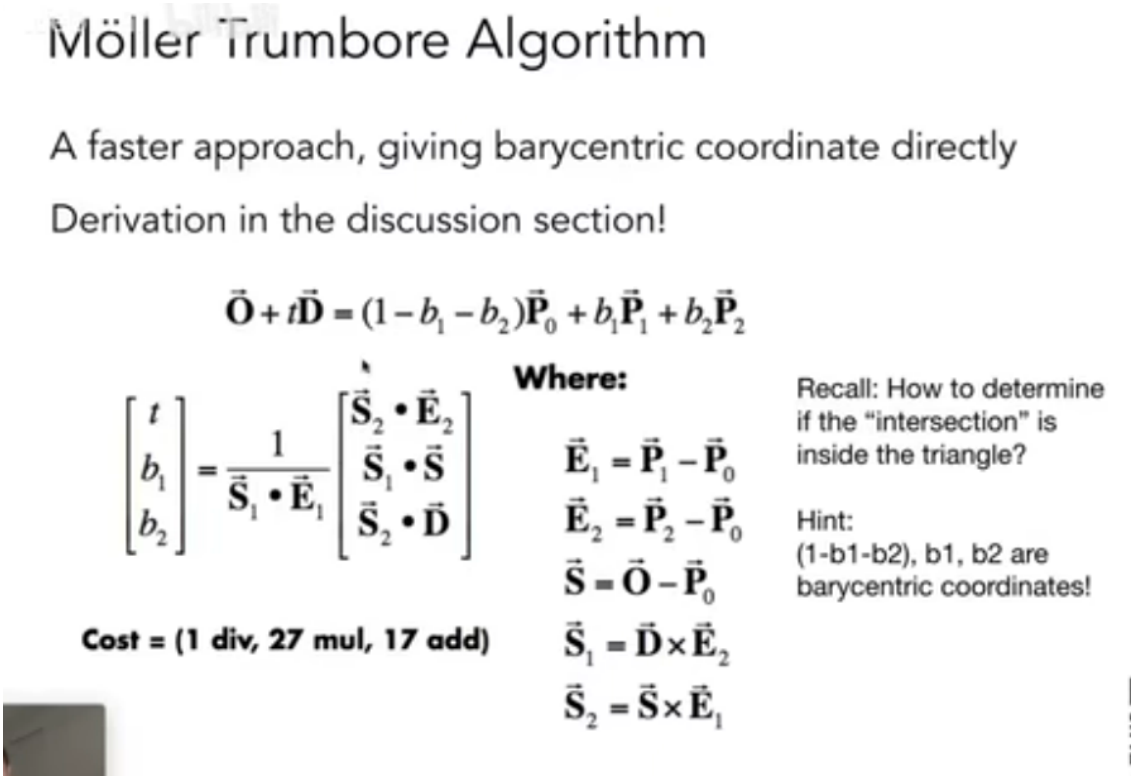
首先从dir可以看出屏幕的位置在z=-1处，默认屏幕是关于x轴和y轴对称的，也就是屏幕的中心在z轴上。循环图像的每个像素点，由于图像上的像素点和屏幕的像素点位置是一一对应的，根据fov和宽高比算出屏幕的大小，就可以找到对应的方法。贴两张图帮助理解：





2. 修改完成rayTriangleIntersect()函数

这个函数的作用顾名思义，判断光线是否与三角形相交。根据课堂内容，相比于光线先与平面相交求交点再判断点是否在三角形内，有更简单的Moller-Trumbore 算法，该方法的原理是利用三角形重心坐标系公式与光线公式联立求参数的方法来判断点是否光线是否与三角形相交，下图是公式原理：

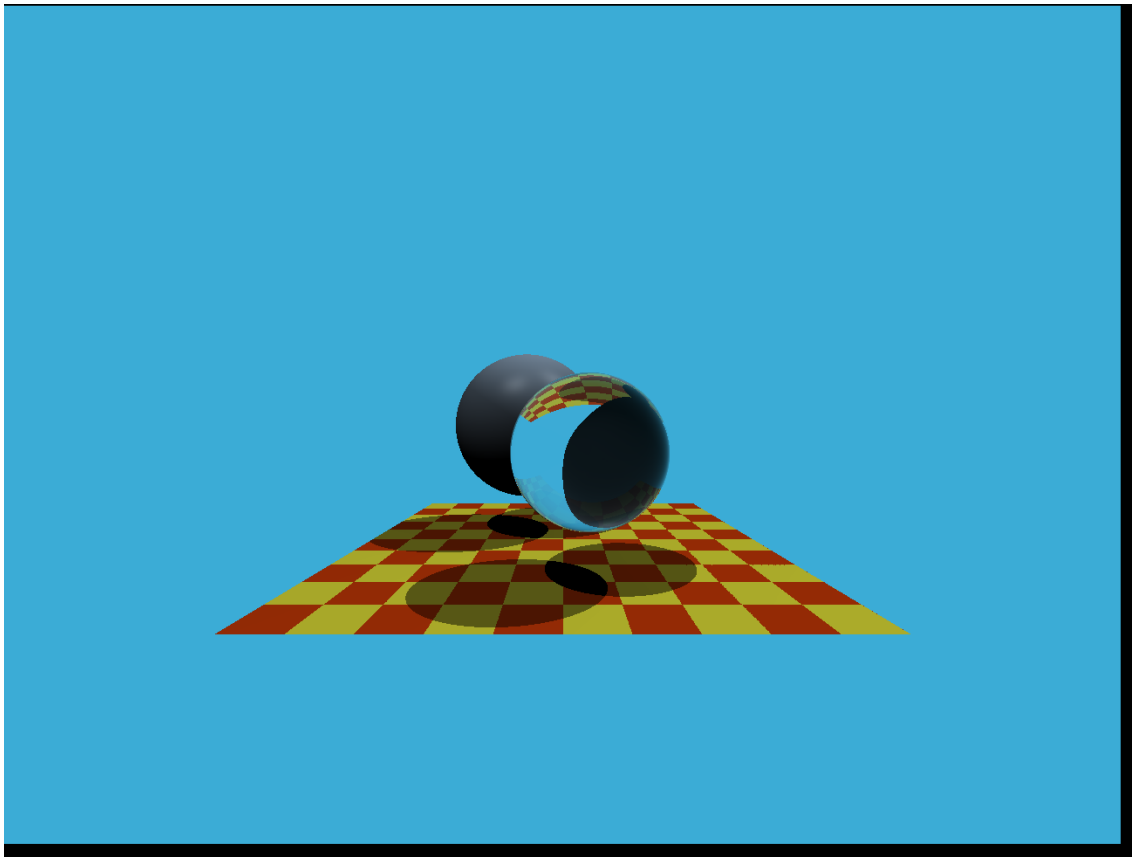


求出t, b1, b2之后，判断参数是否合法即可。参数的解释见GAME101的Lecture 14。

```
bool rayTriangleIntersect(const Vector3f& v0, const Vector3f& v1, const
Vector3f& v2, const Vector3f& orig,
                          const Vector3f& dir, float& tnear, float& u, float&
v)
{
    // TODO: Implement this function that tests whether the triangle
```

```
    // that's specified bt v0, v1 and v2 intersects with the ray (whose
    // origin is *orig* and direction is *dir*)
    // Also don't forget to update tnear, u and v.
    Vector3f E1 = v1 - v0;
    Vector3f E2 = v2 - v0;
    Vector3f S = orig - v0;
    Vector3f S1 = crossProduct(dir, E2);
    Vector3f S2 = crossProduct(S, E1);
    Vector3f tmp(dotProduct(S2, E2), dotProduct(S1, S), dotProduct(S2, dir));
    Vector3f res = 1 / dotProduct(S1, E1) * tmp;
    tnear = res.x;
    u = res.y;
    v = res.z;
    if(tnear >= 0 && u >= 0 && u <= 1 && v >= 0 && v <= 1 && (1 - u - v) >= 0
&& (1 - u - v) <= 1)
        return true;
    else
        return false;
}
```

3. 执行代码



实验总结

“使用光线追踪来渲染图像”，乍一听觉得实验很难，但是其实很多复杂的部分实验都已经完成了，我们需要做的可以说是理解原理后填数学公式就行，具体来说就是图像到屏幕的转换，怎么判断光线是否与屏幕相交，可以说实验设计的很好，把握住了重点。刚开始我也没理解，通过看代码框架和一篇点拨思路的博客就豁然开朗了。

博客链接：https://zhuanlan.zhihu.com/p/438520487#:~:text=GAMES101%20%E4%BD%9C%E4%B8%9A%E4%BA%94%20%E4%BB%A3%E7%A0%81%E6%A1%86%E6%9E%B6%E7%BB%86%E8%8A%82%E7%9A%84%E4%B8%AA%E4%BA%BA%E7%90%86%E8%A7%A3%201%201.%E5%A6%82%E4%BD%95%E8%AE%A1%E7%AE%97%E5%B1%8F%E5%B9%95%E7%9A%84%E5%83%8F%E7%B4%A0%E4%BD%8D%E7%BD%AE%EF%BC%9F%20%E4%BD%9C%E4%B8%9A%E9%87%8C%E9%9D%A2%E9%9C%80%E8%A6%81%E5%AE%9E%E7%8E%B0%E7%9A%84%E4%B8%A4%E5%A4%84%E5%9C%B0%E6%96%B9%EF%BC%8C%E5%85%B6%E4%B8%AD%E4%B8%80%E5%A4%84%E9%9C%80%E8%A6%81%E6%88%91%E4%BB%AC%E8%AE%A1%E7%AE%97%E4%BB%8E%E6%91%84%E5%83%8F%E6%9C%BA%E5%87%BA%E5%8F%91%EF%BC%88%E6%88%96%E8%80%85%E7%9C%BC%E7%9D%9B%E5%87%BA%E5%8F%91%EF%BC%89%EF%BC%8C%E6%89%93%E5%90%91%E6%AF%8F%E4%B8%80%E4%B8%AA%E5%B1%8F%E5%B9%95%E5%83%8F%E7%B4%A0%E7%9A%84%E5%85%89%E7%BA%BF%E7%9A%84%E6%96%B9%E5%90%91%E3%80%82%20%E5%A4%A7%E5%AE%B6%E9%83%BD%E7%9F%A5%E9%81%93%EF%BC%8C%E6%91%84%E5%83%8F%E6%9C%BA%E4%BD%8D%E7%BD%AE%E5%B7%B2%E7%9F%A5%EF%BC%8C%E4%B8%A4%E4%B8%AA%E7%82%B9%E6%89%8D%E8%83%BD%E7%A1%AE%E5%AE%9A%E4%B8%80%E6%9D%A1%E7%9B%B4%E7%BA%BF%EF%BC%8C%E8%80%8C%E5%8F%A6%E5%A4%96%E4%B8%80%E4%B8%AA%E7%82%B9%E4%B8%80%E5%AE%9A%E6%98%AF%E5%B1%8F%E5%B9%95%E5%83%8F%E7%B4%A0%E3%80%82%20%E4%BD%86%E6%98%AF%E5%A6%82%E4%BD%95%E7%9F%A5%E9%81%93%E5%B1%8F%E5%B9%95%E5%83%8F%E7%B4%A0%E7%9A%84%E5%9D%90%E6%A0%87%E5%91%A2%EF%BC%9F,%E4%BB%A3%E7%A0%81%E4%B8%AD%E5%B9%B6%E6%B2%A1%E6%9C%89%E8%A7%84%E5%AE%9A%E5%B1%8F%E5%B9%95%E7%9A%84%E5%9D%90%E6%A0%87%E8%B5%B7%E5%A7%8B%E4%BD%8D%E7%BD%AE%E6%98%AF%E5%B7%A6%E4%B8%8A%E8%A7%92%E8%BF%98%E6%98%AF%E5%B7%A6%E4%B8%8B%E8%A7%92%E3%80%82%20%E4%B8%AA%E4%BA%BA%E7%90%86%E8%A7%A3%EF%BC%8C%E5%9B%A0%E4%B8%BA%E9%A2%9C%E8%89%B2%E4%BF%A1%E6%81%AF%E4%BB%8E%E5%B8%A7%E7%BC%93%E5%AD%98%E4%B8%AD%E5%86%99%E5%88%B0%E6%96%87%E4%BB%B6%E7%9A%84%E9%A1%BA%E5%BA%8F%E6%98%AF%E5%B7%A6%E4%B8%8A%E5%BC%80%E5%A7%8B%E7%9A%84%EF%BC%8C%E6%89%80%E4%BB%A5%E6%8C%89%E8%BF%99%E4%B8%AA%E4%BB%A3%E7%A0%81%E6%A1%86%E6%9E%B6%E4%B8%AD%E9%BB%98%E8%AE%A4%E4%BA%86%E8%BF%99%E4%B8%AA%E9%A1%BA%E5%BA%8F%E3%80%82203%203.castRay%E5%87%BD%E6%95%B0%E5%A6%82%E4%BD%95%E7%90%86%E8%A7%A3%EF%BC%9F%20castRay%E6%98%AF%E7%9C%9F%E6%AD%A3%E5%88%A9%E7%94%A8%E5%85%89%E7%BA%BF%E5%BC%B9%E5%B0%84%E7%9D%80%E8%89%B2%E7%9A%84%E5%87%BD%E6%95%B0%E3%80%82%20%E5%85%B6%E4%BD%9C%E7%94%A8%E5%B0%B1%E6%98%AF%E8%AE%A1%E7%AE%97%E4%BB%8E%E6%91%84%E5%83%8F%E6%9C%BA%E5%87%BA%E5%8F%91%EF%BC%8C%E7%A9%BF%E8%BF%87%E6%AF%8F%E4%B8%AA%E5%83%8F%E7%B4%A0%E5%87%BB%E4%B8%AD%E7%9A%84%E7%89%A9%E4%BD%93%E8%A1%A8%E9%9D%A2%E9%82%A3%E4%B8%AA%E7%82%B9%E7%9A%84%E9%A2%9C%E8%89%B2%E3%80%82%20%E5%A6%82%E6%9E%9C%E6%B2%A1%E6%9C%89%E5%87%BB%E4%B8%AD%E7%89%A9%E4%BD%93%EF%BC%8C%E5%B0%B1%E6%8C%89%E8%83%8C%E6%99%AF%E9%A2%9C%E8%89%B2%E5%A4%84%E7%90%86%E3%80%82%20