《计算机图形学基础》

习题课1

助教　方晓楠

2020年2月29日

联系方式

- 主要邮箱:

  fangxn18@mails.tsinghua.edu.cn

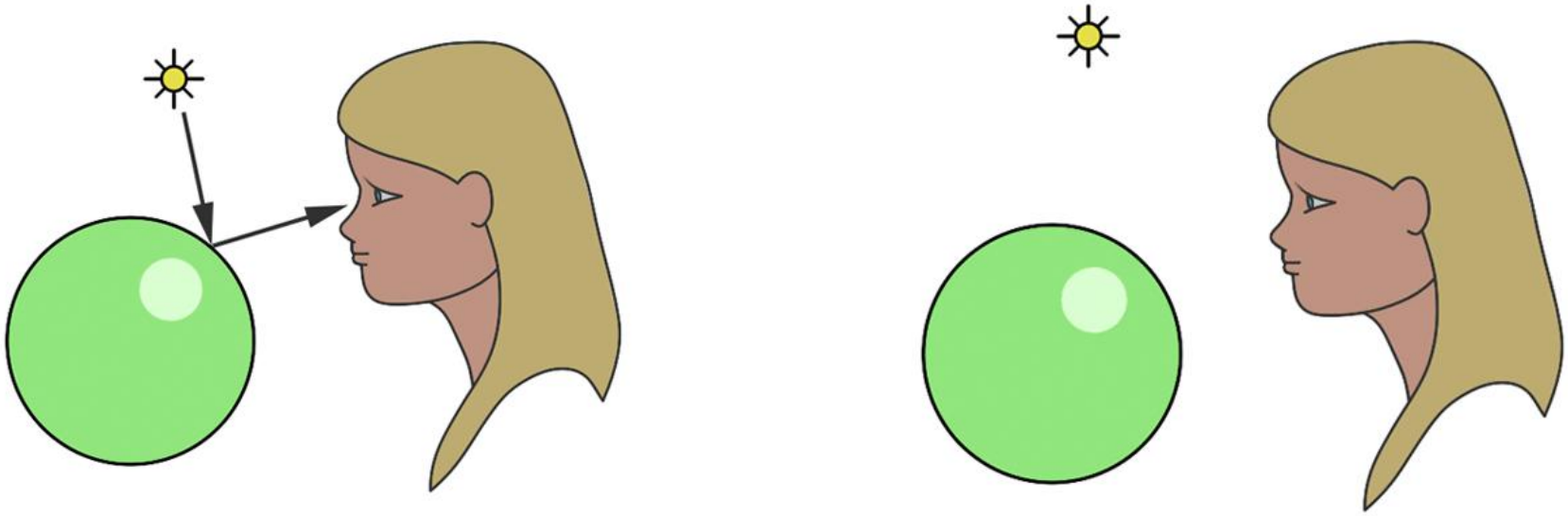
- 位置:
  - FIT楼 3 区 523

习题课1 主要内容

- PA1:RayCasting
  - 光线投射
  - 透视相机
  - 几何求交
  - 代码讲解
  - 环境配置

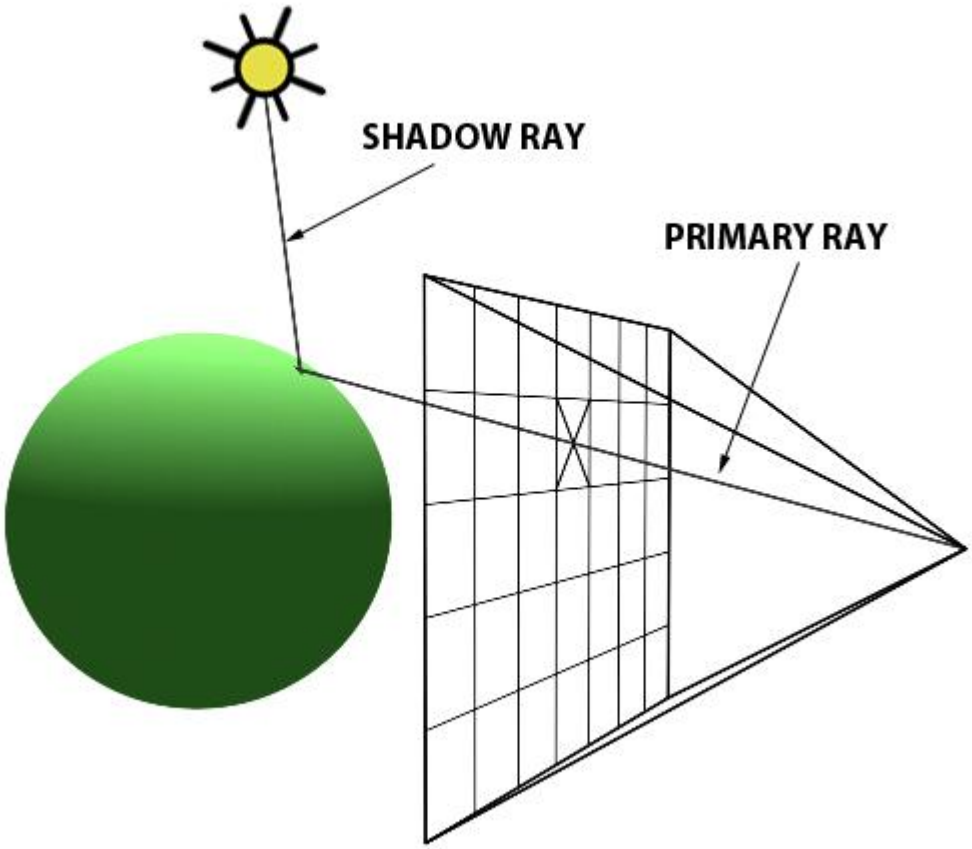
光线投射

光线投射



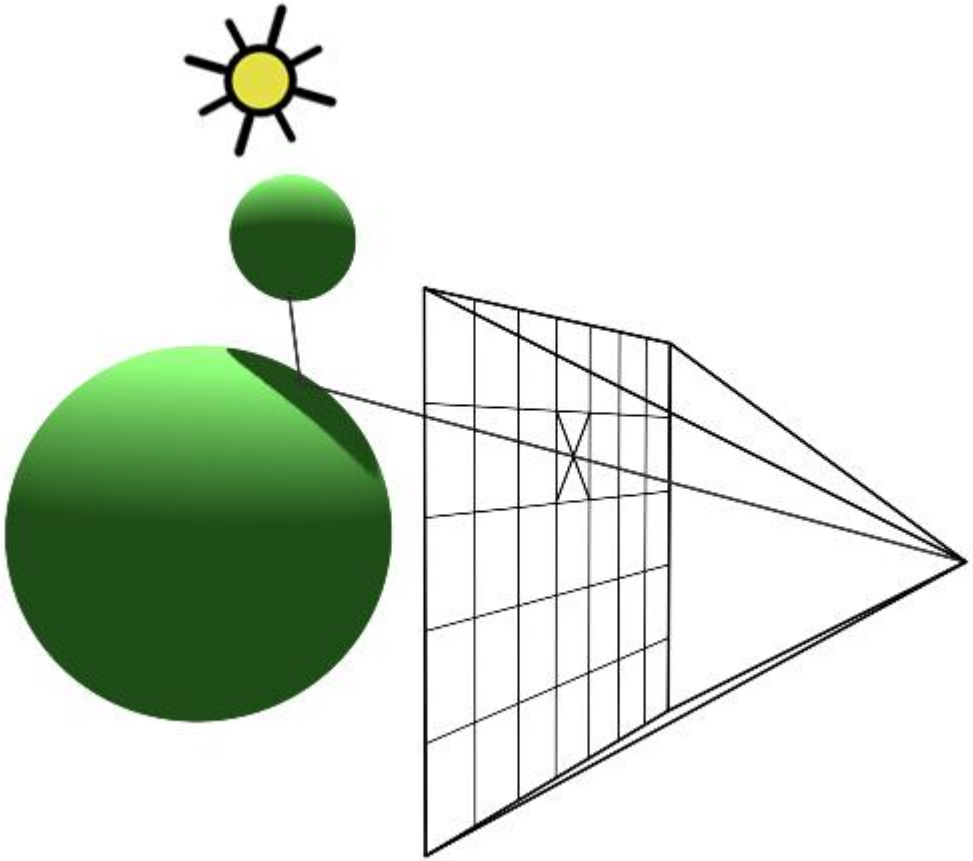
光线投射

- 从视点出发逆向追踪光路



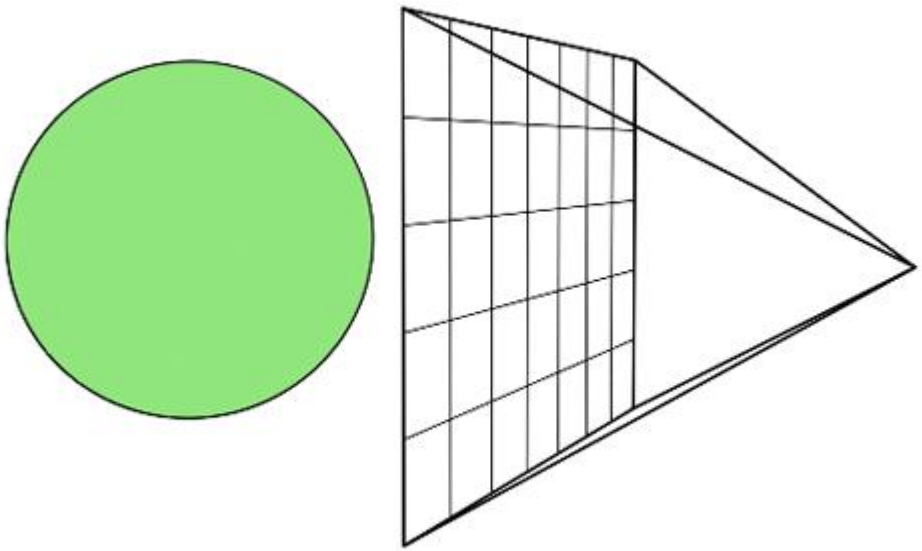
光线投射

- 只计算光源对交点的直接贡献
- 作业中不考虑光源被遮挡的问题

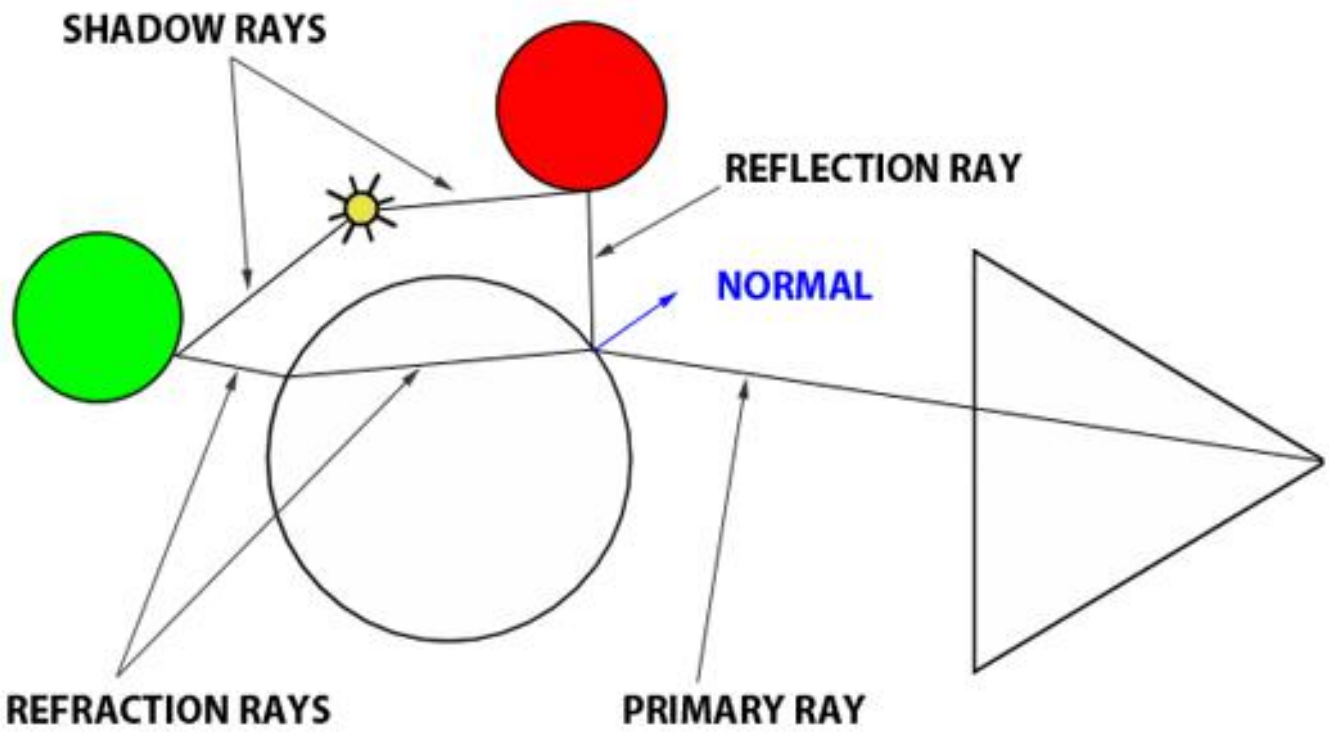


光线投射

- 扫描所有像素求出对应颜色
- 没有交点的位置设为背景色



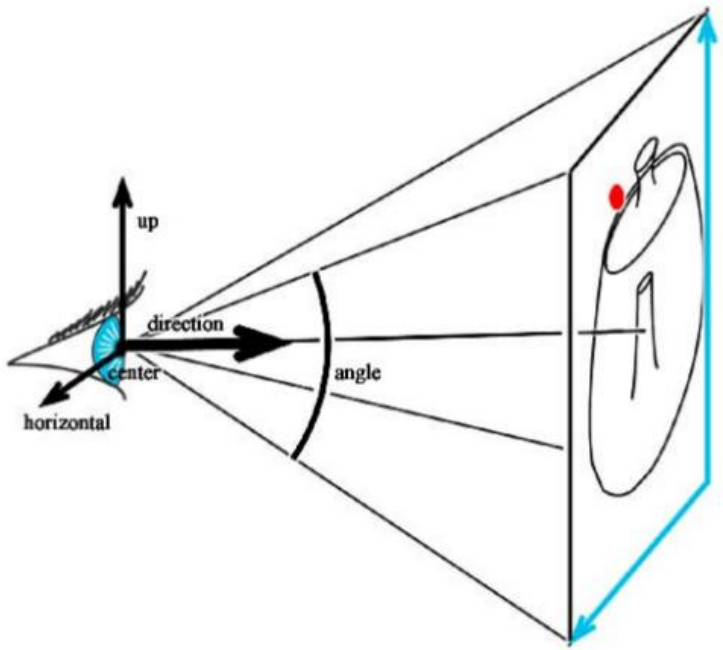
扩展——光线追踪


SHADOW RAYS
REFLECTION RAY
NORMAL
REFRACTION RAYS
PRIMARY RAY

透视相机

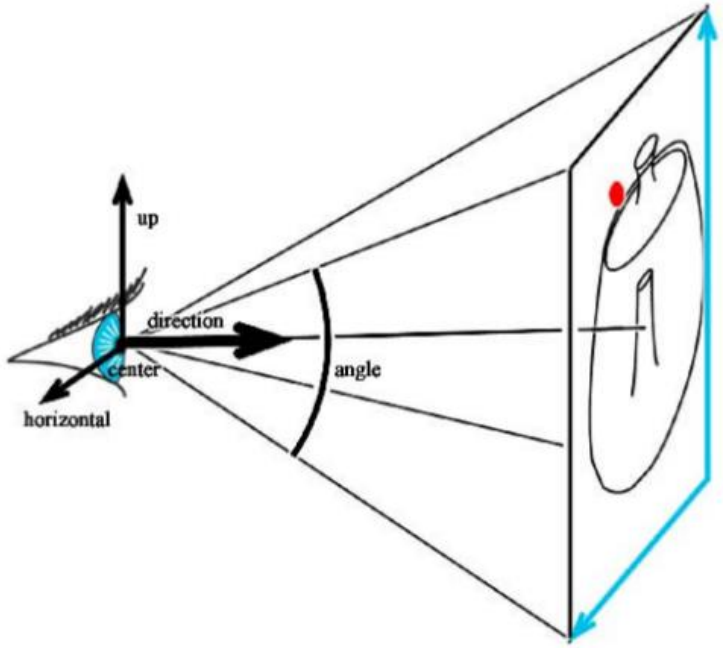
透视相机

- 在视点前放置一块画布，划分成h*w的均匀网格



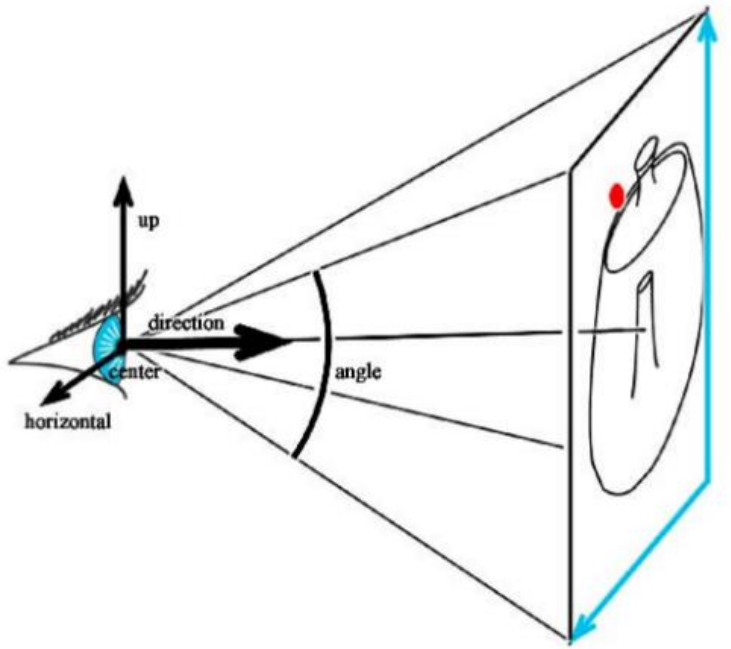
透视相机

- 相机外参:
  - 视点位置t
  - 视点朝向（指向画布中心点）direction
  - 画布的水平轴horizontal和数值轴up



透视相机

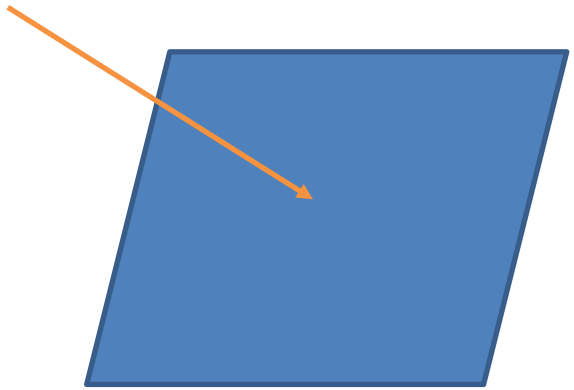
- 相机内参:
  - 图像大小(w, h)
  - 光心位置(c_x, c_y)
  - 视场角$angle$（水平、竖直）
- 给定图像坐标求射线?



几何求交

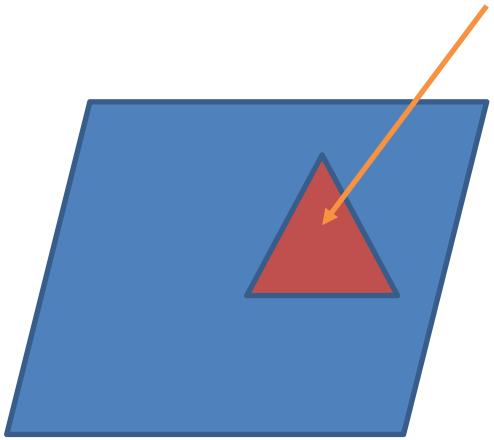
射线与平面的交点

- 射线参数方程$\vec{x} = \vec{o} + t \cdot \vec{r}$
- 平面方程$\vec{n} \cdot \vec{x} = d$
- 二者联立解出参数t



射线与三角形的交点

- 求出三角形所在平面
- 求出射线与平面交点
- 判断交点是否在三角形内

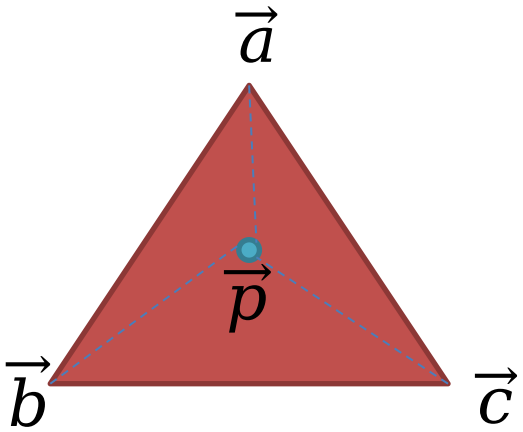


射线与三角形的交点

- 设三角形的三个顶点坐标为$\vec{a}, \vec{b}, \vec{c}$
- 一般约定顶点顺序为逆时针
- 法向量方向：$\vec{n} \parallel (\vec{b} - \vec{a}) \times (\vec{c} - \vec{a})$
- 所在平面方程$\vec{n} \cdot \vec{x} = \vec{n} \cdot \vec{a}$

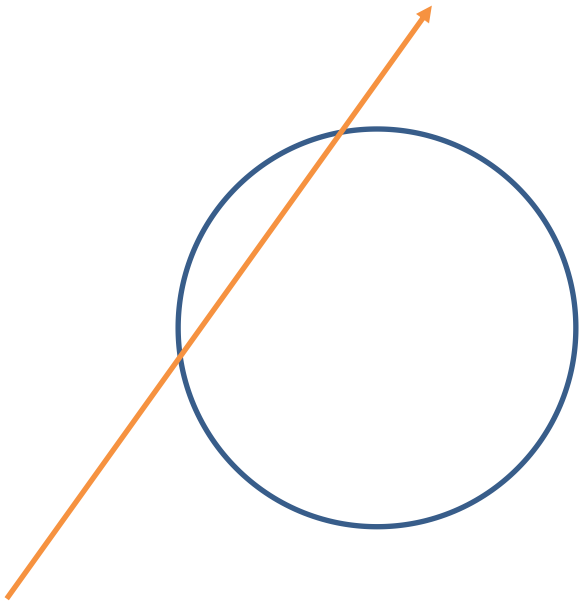
射线与三角形的交点

- 顶点坐标为$\vec{a},\vec{b},\vec{c}$，交点坐标为$\vec{p}$
- 若交点落在三角形内部，则
  - pab为逆时针顺序
  - pbc为逆时针顺序
  - pca为逆时针顺序
- $(\vec{b}-\vec{p})\times(\vec{c}-\vec{p})$与$\vec{n}$同向



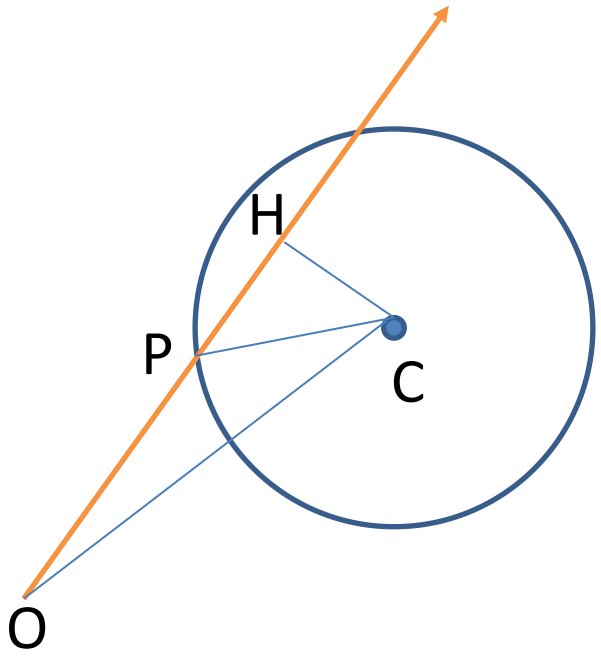
射线与球面的交点

- 射线参数方程$\vec{x} = \vec{o} + t \cdot \vec{r}$
- 球面方程$(\vec{x} - \vec{c}) \cdot (\vec{x} - \vec{c}) = R^2$



射线与球面的交点

- 计算OC在射线方向的投影长度OH
- 计算CH的长度
- 若CH>R说明不相交
- 否则计算PH的长度
- 参数t=OP=OH-PH



代码讲解

代码讲解

- vecmath库
- 主要用到的是Vector3f类
- 重载了+-*/和[]访问
- 一些常用的函数

```
float length() const;
float squaredLength() const;

void normalize();
Vector3f normalized() const;

Vector2f homogenized() const;

void negate();

// ---- Utility ----
operator const float* () const; // automatic type conversion for OpenGL
operator float* (); // automatic type conversion for OpenGL
void print() const;

Vector3f& operator += ( const Vector3f& v );
Vector3f& operator -= ( const Vector3f& v );
Vector3f& operator *= ( float f );

static float dot( const Vector3f& v0, const Vector3f& v1 );
static Vector3f cross( const Vector3f& v0, const Vector3f& v1 );
```

代码讲解

- Image类
- 数组data存储颜色值

```
const Vector3f &GetPixel(int x, int y) const {
    assert(x >= 0 && x < width);
    assert(y >= 0 && y < height);
    return data[y * width + x];
}

void SetAllPixels(const Vector3f &color) {
    for (int i = 0; i < width * height; ++i) {
        data[i] = color;
    }
}

void SetPixel(int x, int y, const Vector3f &color) {
    assert(x >= 0 && x < width);
    assert(y >= 0 && y < height);
    data[y * width + x] = color;
}
```

代码讲解

- Object3D类
- 所有类型物体的基类
- 需要在派生类中实现intersect求交函数

```
// Base class for all 3d entities.
class Object3D {
public:
    Object3D() : material(nullptr) {}

    virtual ~Object3D() = default;

    explicit Object3D(Material *material) {
        this->material = material;
    }

    // Intersect Ray with this object. If hit, store information in hit structure.
    virtual bool intersect(const Ray &r, Hit &h, float tmin) = 0;
protected:

    Material *material;
};
```

代码讲解

- Material类
- 定义了材质属性
- 需要根据Phong模型实现Shade函数

```
    Vector3f Shade(const Ray &ray, const Hit &hit,
                   const Vector3f &dirToLight, const Vector3f &lightColor) {
        Vector3f shaded = Vector3f::ZERO;
        //
        return shaded;
    }

protected:
    Vector3f diffuseColor;
    Vector3f specularColor;
    float shininess;
```

代码讲解

- SceneParser类，读取输入的场景文件和模型

代码讲解

- 场景文件（scene*.txt）和模型文件（*.obj）

```
Materials {
    numMaterials 1
    PhongMaterial {
        diffuseColor 0.79 0.66 0.44
    specularColor 1 1 1
    shininess 20

    }
}

Group {
    numObjects 1

    MaterialIndex 0
    TriangleMesh {
        obj_file mesh/bunny_200.obj
    }
}
```

```
v -1 -1 -1
v 1 -1 -1
v -1 1 -1
v 1 1 -1
v -1 -1 1
v 1 -1 1
v -1 1 1
v 1 1 1
f 1 3 4
f 1 4 2
f 5 6 8
f 5 8 7
f 1 2 6
```

环境配置

环境配置

- 为什么使用CMake?
  - 自动处理复杂的源码依赖关系
  - 能够生成跨平台的编译文件

- 配置示例：
  - Windows 10 + Ubuntu Subsystem
  - https://docs.microsoft.com/en-us/windows/wsl/install-win10

- IDE： CLion, VS Code, Visual Studio

环境配置

- 不论你采用何种方式编译和调试，请将补充完整的程序文件按原路径保存好，以确保能够按照introduction中的方式编译运行

Thank You！
Any Questions?